Designing multimodal composition activities to support integrated primary programming and storytelling

Robert Whyte, BEd (Hons.) MA

Thesis submitted to the University of Nottingham

for the degree of Doctor of Philosophy

July 2021

Table of Contents

Table of Contents i
Table of Figures vi
Table of Tables vii
Acknowledgements viii
Abstract ix
Research dissemination xi
CHAPTER 1 Introduction 1
1.1 Overview 1
1.2 Context 2
1.2.1 The computing curriculum 2
1.2.2 Curriculum implementation 3
1.2.3 Potential for integration 7
1.3 Thesis goals 9
1.4 Conclusion 10
1.5 Layout of thesis 11
CHAPTER 2 Literature review 14
2.1 Introduction 14
2.2 Computational thinking (CT) 14
2.2.1 Traditional approaches 14
2.2.2 Visual programming 15
2.2.3 Fostering CT 16
2.3 Alternative conceptions of CT 17
2.3.1 Computational action 18
2.3.2 Computational action across the curriculum 19
2.4 Literacy education 22
2.4.1 Literacy in the digital age 22
2.4.2 Literacy across the curriculum 24
2.4.3 Supporting narrative construction 26
2.4.4 Supporting literacy through programming activities 30
2.5 Multimodal composition 34
2.5.1 Researching multimodal composition 38
2.5.1.1 Artefact *39*
2.5.1.2 Rhetorical skills *40*
2.5.1.3 Substance *41*
2.5.1.4 Process management *43*
2.5.1.5 Habits of mind *44*
2.6 Research aims 46
2.7 Summary 48
CHAPTER 3 Methodology 50
3.1 Overview 50
3.2 Design-based research 50

3.2.1 Limitations of DBR 53
3.2.2 Employing DBR in the thesis 54
3.2.3 Phases of DBR 54
3.2.4 Outputs of DBR 55
3.3 Data generation 56
3.4 Data analysis 57
3.4.1 Task-oriented analyses 58
3.4.2 Analytical procedure 59
3.4.3 Proposed analytical framework 60
3.5 Reliability of findings 62
3.6 Researcher position 63
3.7 Research ethics 63
3.8 Conclusion 65
CHAPTER 4 Design 67
4.1 Overview 67
4.2 Pedagogical goal 67
4.3 Design principles 68
4.3.1 Build on learners' prior knowledge and skills (DP1) 69
4.3.1.1 Theory 69
4.3.1.2 Instantiation 71
4.3.2 Engage learners in constructive activities (DP2) 72
4.3.2.1 Theory 72
4.3.2.2 Instantiation 73
4.3.3 Scaffold student-generated construction (DP3) 74
4.3.3.1 Theory 74
4.3.3.2 Instantiation 77
4.3.4 Provide learners with authentic and purposeful activities (DP4) 81
4.3.4.1 Theory 81
4.3.4.2 Instantiation 82
4.3.4.2.1 Initialisation and sequencing 82
4.3.4.2.2 Concurrency 83
4.4 Proposed design 83
4.4.1.1 Structured activities 86
4.4.1.2 Independent project 87
4.5 Summary 88
CHAPTER 5 Study 1 89
5.1 Introduction 89
5.1.1 Study rationale 89
5.1.2 Research questions 90
5.2 Method 91
5.2.1 Setting and participants 91
5.2.2 Intervention procedure 92
5.3 Findings 94
5.3.1 Structured activities (A1-C5) 95
5.3.1.1 Programme dialogue between characters (A1-A3) 95
5.3.1.2 Debug a story (B4-B5) 97
5.3.1.3 Coordinate narrative elements (C6-C7) 99

5.3.1.4 Execute multiple narrative elements simultaneously (D8) *101*
5.3.2 Independent project (EF9) 102
5.3.3 Integration of learning strands 107
5.3.3.1 Structuring text(s) using cohesive device(s) *108*
5.3.3.2 Using multimodal effects to engage and hold a reader's attention *111*
5.4 Summary of findings 112
5.4.1 Characterising programming and storytelling practices (SQ1) 113
5.4.2 Relating programming and storytelling practices (SQ2) 116
5.5 Design implications 117
5.5.1 Embedding relevant programming concepts (DP2) 117
5.5.2 Scaffolding student-generated construction (DP3) 118
5.5.3 Providing more time for 'purposeful' activities (DP4) 121
CHAPTER 6 Study 2 122
6.1 Introduction 122
6.1.1 Study aims 122
6.2 Method 123
6.2.1 Setting and participants 123
6.2.2 Design iterations 125
6.2.2.1 Engage learners in constructive activities (DP2) *125*
6.2.2.2 Scaffolding student-generated construction (DP3) *126*
6.2.2.3 Engaging learners in authentic and purposeful activities (DP4) *128*
6.2.3 Intervention procedure 129
6.2.4 Data analysis 131
6.3 Findings 132
6.3.1 Structured activities (A1-C5) 132
6.3.1.1 Programming dialogue between characters (activities A1-A2) *132*
6.3.1.2 Debugging a story (activities B3-B4) *134*
6.3.1.3 Coordinating narrative elements (activities C5-C6) *136*
6.3.2 Independent project (DEF6) 138
6.3.3 Integration of learning strands 142
6.3.3.1 Structuring narrative episodes with cohesive devices *143*
6.3.3.2 Using multimodal text features for narrative effect *144*
6.4 Summary of findings 145
6.5 Research questions 147
6.6 Design implications 150
6.6.1 Design prompts for distributed scaffolding 150
6.6.2 Adapt intervention for formal curricular use 152
CHAPTER 7 Study 3 154
7.1 Introduction 154
7.1.1 Study rationale 154
7.1.2 Study aims 155
7.2 Method 156
7.2.1 Setting and participants 156
7.2.2 Design iterations 158
7.2.2.1 Build on learners' prior knowledge and skills (DP1) *158*
7.2.2.2 Engage learners in constructive activities (DP2) *158*
7.2.2.3 Scaffolding student-generated construction (DP3) *159*
7.2.3 Intervention procedure 162

7.2.3.1 Original research procedure ... *162*
7.2.3.2 Revised research procedure ... *163*
7.2.4 Data generation ... 167
7.2.4.1 Observation field notes ... *168*
7.2.4.2 Artefact-based interviews ... *169*
7.2.5 Data analysis ... 173
7.2.5.1 Interview coding ... *174*
7.2.5.2 Task-oriented analyses ... *175*
7.3 Findings ... 176
7.3.1 Teacher's experiences ... 176
7.3.1.1 Motivations ... *176*
7.3.1.2 Experiences of facilitating ... *178*
7.3.2 Intervention factors ... 179
7.3.2.1 Computing subject knowledge and skills ... *179*
7.3.2.2 Literacy and multimodal composition ... *181*
7.3.2.3 Time management ... *182*
7.3.2.3.1 Activities ... *182*
7.3.2.3.2 Timetable ... *184*
7.3.2.4 Differentiation ... *185*
7.3.2.5 Teaching strategies ... *186*
7.3.2.5.1 Open-ended activities ... *186*
7.3.2.5.2 Integrated activities ... *187*
7.3.2.6 Availability of resources ... *188*
7.3.2.6.1 Technology ... *188*
7.3.2.6.2 Stimuli ... *189*
7.3.3 Adaptation and future use ... 190
7.3.3.1 Annotated texts ... *191*
7.3.3.2 Integration of learning strands ... *191*
7.3.3.3 Time management ... *193*
7.3.3.4 Differentiation ... *194*
7.3.3.5 Modelling and demonstration ... *195*
7.3.3.6 Structured activities ... *195*
7.3.3.6.1 Giving feedback ... *195*
7.3.3.6.2 Structured planning frame ... *197*
7.4 Summary of findings ... 198
CHAPTER 8 Discussion ... 202
8.1 Introduction ... 202
8.2 Research aims and objectives ... 204
8.2.1 Study 1 ... 204
8.2.2 Study 2 ... 206
8.2.3 Study 3 ... 208
8.3 Implications ... 210
8.3.1 Design principles ... 210
8.3.1.1 Building on learners' prior knowledge and skills (DP1) ... *211*
8.3.1.2 Engaging learners in constructive activities (DP2) ... *213*
8.3.1.3 Scaffolding student-generated construction (DP3) ... *215*
8.3.1.4 Engaging learners in authentic and purposeful activities (DP4) ... *218*
8.3.2 Analytical approach ... 221
8.4 Current limitations and suggested future research ... 222
8.4.1 Recruiting multiple teachers ... 222

8.4.2 Broadening computing for multiple purposes 224
8.4.3 Conjecture-based analyses 226
8.4.4 Generalisation of findings 227
8.5 Conclusion 229
References 230
Appendix 1: Intervention lesson plans 253
Appendix 2: Intervention scaffolding materials 259
Appendix 3: Task-oriented analyses 261
Appendix 4: Ethics documents 263

Table of Figures

Figure 1: Scratch 2 programming interface 5
Figure 2: Hayes and Flowers' (1981) cognitive process theory of writing model 44
Figure 3: Classroom prompts 78
Figure 4: Proposed storyboard 79
Figure 5: The Scratch programme interface 80
Figure 6: Intervention site (Study 1) 93
Figure 7: Screenshot of debugging task (stage and code) (Study 1) 98
Figure 8: Coordinating narrative elements using show/hide blocks (Study 1) 100
Figure 9: Use of broadcast messages (left) vs wait blocks (right) (Study 1) 101
Figure 10: Sample project (Study 1) 103
Figure 11: Tadgh's use of glide block (Study 1) 103
Figure 12: Charlie's use of looping motion to simulate a meteor shower (Study 1) 104
Figure 13: Natasha's final project (Study 1) 107
Figure 14: Aoife's use of the ghost effect (Study 1) 109
Figure 15: Mark's use of lines to indicate motion (Study 1) 112
Figure 16: Intervention site (Study 2) 124
Figure 17: Screenshots of differentiated scaffolding prompts (Study 2) 128
Figure 18: Screenshot of Eric's loop animation (Study 2) 134
Figure 19: Screenshot of debugging activity (stage and code) (Study 2) 135
Figure 20: Screenshot of Eric's project using a repeated loop animation (Study 2) 140
Figure 21: Example of sprite manipulation (Study 2) 142
Figure 22: Intervention site (Study 3) 157
Figure 24: Revised storyboard (Study 3) 161
Figure 23: Scaffolding prompts (Study 3) 166
Figure 25: Student-modified resources (Study 3) 190
Figure 26: Student-generated resources (Study 3) 190
Figure 27: Screenshot of Scratch animated text features (beta) 226

Table of Tables

Table 1: Progression in multimodal text composition 37
Table 2: Domains of multimodal assessment 39
Table 3: Phases of DBR project (from Herrington et al., 2007) 55
Table 4: Decide on mode and content for specific purpose(s) (conjectures) 61
Table 5: Proposed activities (adapted from Balch et al., 2013) 86
Table 6: Intervention sessions and lesson activities (Study 1) 92
Table 7: Analyses of activities A1-A3 (Programme dialogue between characters) 96
Table 8: Analyses of activities B4-B5 (Debug a story) 98
Table 9: Analyses of activities C6-C7 (Coordinate narrative elements) 100
Table 10: Analyses of activity D8 (Executing narrative elements simultaneously) 101
Table 11: Participants' project goals (Study 1) 102
Table 12: Analyses of activity EF9 (Independent project) 105
Table 13: Structure text(s) using cohesive device(s) 108
Table 14: Structuring longer texts with cohesive devices 110
Table 15: Using multimodal text features for narrative effect 111
Table 16: Participant pseudonyms and levels of experience 125
Table 17: Original and revised activities 130
Table 18: Analyses of activities A1-A2 (Programming dialogue between characters) 133
Table 19: Analyses of activities B3-B4 (Debugging a story) 136
Table 20: Analyses of activity C5 (Coordinating narrative elements) 137
Table 21: Participants' project goals (Study 2) 139
Table 22: Analyses of activity DEF6 (Independent project) 141
Table 23: Structuring narrative episodes with cohesive devices (conjectures) 144
Table 24: Multimodal text features (conjectures) 145
Table 25: Original research procedure 163
Table 26: Intervention activities 165
Table 27: Revised research procedure 167
Table 28: Pre-intervention interview questions 170
Table 29: Intervention feedback questions 171
Table 30: Post-intervention artefact-based interviews 172
Table 31: End-of-study interview 173
Table 32: Coding scheme for artefact-based teacher interviews 175
Table 33: Analyses of activity DEF6 (Independent project) (subset) 181

Acknowledgements

I would like to thank the following people for helping with this research project:

First, my supervisor Professor Shaaron Ainsworth, for her unwavering support and guidance from the very beginning (including the advice not to conduct a historical review of ICT in schools). Thank you for everything, Shaaron.

Second, to Dr Jane Medwell, for her guidance and support throughout the PhD. I learned a lot about how to be a better researcher. Thank you, Jane.

To my colleagues and friends past and present in the LSRI: Dr Pryce Davis, Dr Freydis Vogel, Dr Hoa Ta, Dr Ahmet Durgungöz, Dr Haci Yapici, Caroline Anderson, and Emma Stringfellow. Thank you for your encouragement, guidance, and support.

Special thanks to Dr Jonathan Halls, Dr Jannah Mohd Jaafar, Dr İpek Saralar-Aras Kate Murphy, Arjmand Kokab, and Rachel Lehner-Mear for your support and friendship throughout the PhD.

To the many students and teachers from local schools who made this research possible. Thank you for your enthusiasm and patience.

To my family, especially Mam, Dad, Kim, Sarah, Paul and Roberto. I hope this acknowledgement expresses my profound and enduring gratitude to you all.

Finally, a special mention to Jake, for his patience, kindness, and support throughout. Cheers pal.

Abstract

A broad interest in embedding computer science (or computing) education in formal learning contexts has resulted in a proliferation of research on the teaching and learning of programming. Much of this research has focused on supporting learning in science, engineering, and mathematics (or STEM), as computing education is seen to promote problem solving (Wing, 2006), with much less work conducted in non-STEM (e.g., literacy) content areas. This thesis adds to this latter by investigating the potential for integrated computing and literacy education by proposing multimodal composition (MMC) as a framing device for supporting children's storytelling and programming practices.

Employing a design-based research methodology, this thesis begins by reviewing literature on computing education, approaches to multimodal storytelling and prior work in fostering computational thinking. It then describes the development of a theory-driven curriculum design to support integrated programming and storytelling goals. Three empirical studies are reported that test and refine this design.

The first two studies test the curriculum design as part of an afterschool primary computing club. Process data generated included audio-recorded interactions between participants and facilitator, and student-generated data, including planning materials, project work, and screen recordings. These data were tested against the intervention goals using conjecture-based analyses to generate insights into the learning process. The results of these studies were then used to iterate the curriculum design. Analyses demonstrated where composing multimodal texts in a visual programming environment supported novice learners' programming practices, such as algorithm design, program execution and loops, and visual storytelling practices, including representation choice, sequencing, and visual design. However, it also indicated areas where MMC was less successful in supporting learners' programming practices, including conditional logic and variables, and storytelling, including descriptive (i.e., monomodal) writing. The third empirical study consists of a teacher-led implementation of the curriculum unit during regular lessons. Findings demonstrate the

potential for productive integration between programming and storytelling, but also highlight the challenges for teachers in embedding activities, particularly those that sit across curriculum boundaries, into their regular practice.

Findings from all studies are integrated to uncover their theoretical and practical implications for future work in this area and their overall contribution to knowledge.

Research dissemination

Peer-reviewed conference proceeding papers

*__Whyte, R.__, Ainsworth, S.E & Medwell, J. (2020) Designing multimodal composition activities for integrated K-5 programming and storytelling. Presented at the International Conference of the Learning Sciences (ICLS 2020), June 19-23 2020, Vanderbilt University Nashville (TN), USA.

Whyte, R., Ainsworth, S.E & Medwell, J. (2019) Designing for integrated K-5 computing and literacy through storymaking activities. Presented at the International Computing Education Research Conference (ICER '19), August 12-14 2019, Toronto, ON, Canada.

Whyte, R. (2018) Integrating computational thinking and literacy instruction: A design-based study. Presented at the Learning Sciences Graduate Conference (LSGS '18), October 12-13 2018, Vanderbilt University, Nashville (TN), USA.

Conference presentations and posters

Whyte, R. (2019) Designing multimodal composition activities for integrated primary programming and storytelling. Presented at *Scratch Conference Europe*, August 23-25, University of Cambridge, UK.

Whyte, R. (2018) Iterating on computing lessons: Stories of integrated computing and literacy activities. Presented at *Pecha Kucha Research Showcase*, November 21 2018, University of Nottingham, Nottingham, UK.

Whyte, R. (2018) Multimodal composition and visual programming: Designing integrated lessons in computing and literacy. Presented *at School of Education Postgraduate Research Conference,* November 16 2018, University of Nottingham, Nottingham, UK.

* Shortlisted for Best Student Paper, ICLS 2020

Whyte, R. (2018) Employing multimodal composition to support literacy skills and computational thinking. Presented at *LSRI Learning Lunch Series* (seminar), November 8 2018, University of Nottingham, Nottingham, UK.

Whyte, R. (2018) Integrating computational thinking and primary literacy: A design-based study. Presented at *London Computing Education Research Symposium (LCERS '18)*, June 11 2018, King's College London, UK.

Whyte, R. (2017) Designing, implementing and evaluating an integrated approach to computational thinking. Presented *at School of Education Postgraduate Research Conference*, November 10 2017, University of Nottingham, Nottingham, UK.

Whyte, R. (2017) Computing across the curriculum. Presented at *LSRI Learning Lunch Series* (seminar), November 7 2017, University of Nottingham, Nottingham, UK.

CHAPTER 1
Introduction

1.1 Overview

In 2015, I moved to the UK from Ireland and started my first teaching post in a small primary school (or elementary school, ages 5-11) in Nottingham. At that time, the education system, including the curriculum (or 'programmes of study'), was undergoing significant change. This challenged me to familiarise myself with new curricula and policy as a newly qualified teacher, while also offered me and my experienced colleagues an opportunity to plan and facilitate new and exciting experiences for learners in my classroom.

One of the more novel additions to the curriculum was the introduction of computing education, replacing the existing Information and Communications Technology (or ICT) curriculum. Having taken an elective module in ICT-based learning during my undergraduate degree, I was familiar with some of the activities and tools recommended as part of the new curriculum. Finding opportunities to integrate computing—primarily programming—activities across the existing curriculum proved both exciting and challenging. Typically, this centred on designing games, animations, and digital stories that integrated learning in other subject areas.

Throughout the school year, I found myself taking an increasing role in organising and facilitating computing lessons and professional development (or PD) sessions with other teachers. The experience solidified my interest in the subject, while highlighting the challenges and opportunities it presented to me and my colleagues. A significant challenge was felt in adequately furnishing lessons as resources were stretched across classes and schools. In conversations with teachers in other schools across the city, similar accounts were reported of teachers feeling overwhelmed or underprepared for implementing the new programme of study.

These developments, coupled with a personal interest, led to my desire to undertake doctoral research in this area. The contents of this thesis constitute my doctoral research project which takes a participatory approach to design, test and refine an approach to integrate computing education into formal learning contexts. More specifically, it explores the potential of integrating computing education into literacy education and how this could have pedagogical benefits for both teachers and students. By looking at how curricula can be designed to support learners in both subjects and collaborating with primary school teachers currently implementing the new curriculum, I address how such an innovative pedagogy could be embedded in classroom practice and what challenges might arise from this.

1.2 Context

The advent of increased digital technologies and applications has had a significant impact on young children's learning experiences (Buckingham et al., 2005). Through varied experiences with tablets, PCs, smartphones and other digital devices, young children have frequent experiences playing and create games, explore virtual worlds, watch tutorials and YouTube channels, and search and use information online (Marsh, 2010). Concerns have been noted by some educators and parents over the need to position learners as creators, not merely consumers of digital and online media, through activities that require critical thinking, creativity, and problem-solving. Similarly, researchers in media education (e.g., Buckingham, 2008) have foregrounded the importance of critical media skills in enabling young people to participate in a digital world as both an informed producer and consumer of media. Burnett and Merchant (2018), for instance, argue for the understanding and promoting of '21st-century literacies'—or those increasingly commonplace in young people's everyday experiences—including programming, filmmaking, digital and online media, and design activities.

1.2.1 The computing curriculum

In the past few years, a similar trajectory has emerged in computing education. As researchers and industry figures argue that computer science is an essential literacy (e.g., Guzdial, 2019), calls for its integration in formal educational settings have grown. Lobbyists have partly driven

this resurgence, with advocates arguing for computer science as "a fundamental skill for everyone, not just for computer scientists" (Wing, 2006, p. 33).

More locally, this demand resonated with some researchers and policymakers, and prompted a broader scheme to embed computing within the outdated ICT curriculum in England (Dredge, 2014). In primary schools alone, a growing effort to embed computing education into formal curricula prompted a state-wide review in 2013. In that review, computing was proposed as a mandatory subject at state-maintained primary (ages 5-11) and secondary (ages 11-16) schools in England. Thus, in 2014 computing replaced the outgoing Information and Communication Technology curriculum (or ICT), a subject widely regarded as focusing too heavily on using software and technological tools. Programming activities were instead favoured as a means to promote a deeper understanding of computing by exploring the underlying properties and mechanics of systems and programmes (Brown et al., 2014).

Though the curriculum change was not consistently implemented across the four devolved nations (England, Scotland, Wales and Northern Ireland), a broad move towards compulsory computer science from ages five and upwards was mandated. The change primarily focused on integrating the fundamentals of computation—often termed 'computational thinking' (or CT) (Wing, 2006)—alongside the existing curriculum of digital literacy and e-safety. These 'fundamentals' commonly include authentic concepts within computer science, including, but not limited to, sequencing, loops, variables, and authentic practices, including designing, debugging, and iterating on algorithms and computer programmes. These are commonly developed through visual programming (or 'coding') tools, including Scratch (see Figure 1).

1.2.2 Curriculum implementation

With an increase of £84m in government funding over five years announced (Cellan-Jones, 2019) to train and upskill 8000 teachers, along with a commitment to establishing a National Centre for Computing, it appears significant progress in implementing the curriculum has been

made. However, concerns remain over the ‘quality’ of this implementation. Though evidence is limited, some notable findings have been reported.

Evidence of success has been mixed, with advocates praising the ambition of the programme of study in England (Brown et al., 2014), while critics have noted the additional logistical and professional challenges the change has brought (Sentance et al., 2012). These include the need to train new teachers and ‘upskill’ existing teachers with greater subject and pedagogical knowledge. More recently, Cellan-Jones (2017) found significant concerns about the ability of teachers to deliver these curricular changes, particularly within the primary sector. Larke (2019) also found a growing trend of teachers rejecting the mandated change. Similarly, evidence from other countries introducing similar curricular changes also points to the challenges in furnishing schools with adequate resources and training teachers (e.g., in the United States, Grover & Pea, 2013).

Several years after the curriculum was first integrated, an influential report on the state of computing education (The Royal Society, 2017) found the implementation of the new curriculum to be “patchy and fragile” (p. 6). In discussing uptake and engagement in computing with young people, it found that a lack of interest was cited as the main reason for not choosing to study computer science at secondary school level (47% of respondents). The Department for Education drew criticism for these findings, including their recruiting of newly qualified teachers and upskilling of existing teachers in the field.

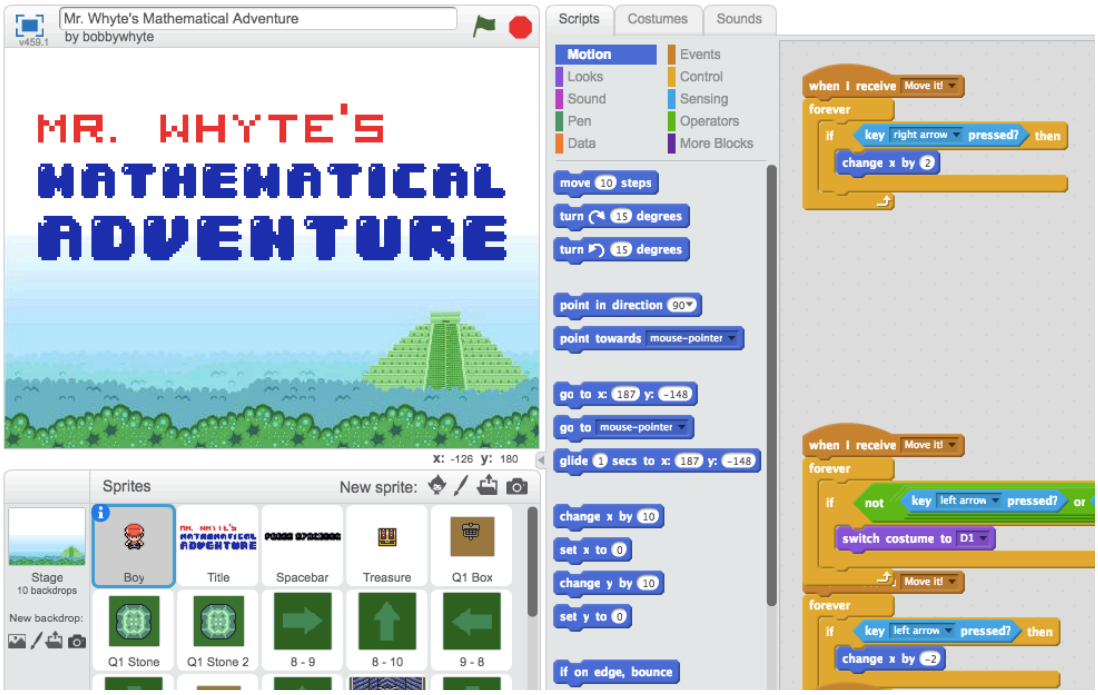


Figure 1: Scratch 2 programming interface

Research into teachers' experiences with implementing the new curriculum are also mixed. For instance, a survey of 400 UK primary school teachers' perceptions of computational skills (here, 'tech literacy') found that most of those surveyed regard computational thinking as having a positive benefit to numeracy (96%) and literacy (69%) in the classroom (BT & Ipsos MORI, 2016). However, more than 43% reported not understanding what computational thinking was while 31% reported difficulties in both teaching CT in computing lessons and embedding CT across subjects. The report highlighted that the challenges of implementing the curriculum were acutely felt by teachers integrating computational thinking into their existing practices.

In secondary schools—where the subject has an assessment component—the problem of teacher training and teacher retention has been acutely felt as schools struggle to 'upskill' ICT teachers to satisfy the new curricular demands (Sentance et al., 2012). Though this problem is mostly reported in secondary schools in England, a growing issue has also emerged in primary schools where almost half (45%) of surveyed teachers reported insufficient familiarity and low confidence in their ability to deliver the requirements of the

computing curriculum (The Royal Society, 2017). These teachers noted how insufficient "theoretical and technical knowledge of computing that included aspects of programming and coding" (p. 54) had impacted their ability to deliver these requirements. Some requirements were reported by teachers as less difficult such as those carried over from the previous ICT curriculum (e.g., e-safety and digital literacy). However, programming and computational thinking were the areas reported as needing the most CPD as many teachers felt ill-equipped and overwhelmed. Additionally, teachers reported a relatively low amount of time dedicated to computing CPD, with a third averaging 1-9 hours per year (34% of reported teachers), and another third (29%) reported no CPD or additional training to prepare for the curriculum change.

More recent research by Larke (2019), whose work investigated the experiences of primary school teachers, echoes the results of the reports detailed above. Using survey data, she found teachers reported the content of the curriculum as "developmentally inappropriate, too narrow and too difficult" (p. 1137) and highlighted how its function within the curriculum served as "a narrow range of activities which were not as broadly applicable as subjects like maths, English or science" (p. 1146). Further, teachers reported needing additional support and material resources, prerequisite knowledge and skills to understand and teach the subject and felt worried their students would overtake them.

Low engagement in computing is problematic but the problem might not only relate to student interest or teacher recruitment. Previous research in higher education internationally has similarly indicated that low engagement in computing education may stem from a common 'image problem' that emphasises abstraction over application (Forte & Guzdial, 2004). Recent media reports in England (e.g., Cellan-Jones, 2019) paint a similar picture as computing was perceived by students as "being harder than other subjects" and "off-putting". These reports reinforce the idea that while the old ICT curriculum provided learners with opportunities to practice and develop touch typing, word processing and search skills, it posed few to no opportunities for learning basic programming in a formal setting (Larke, 2019). Within an already "crowded curriculum" (The Royal Society, 2012, p. 30), the purpose and outcomes of

computing education are arguably too "specific and narrow" (Cellan-Jones, 2017) for teachers and learners to productively engage with and benefit from the new subject. Moreover, with little evidence of the learning benefits of engaging in computing activities available, there exists a strong need to consider how these issues could be mitigated.

1.2.3 Potential for integration

Research has suggested that programming presents several distinct learning opportunities for young children. Through visual programming languages (VPLs), physical programmable devices, and 'maker' tools, learners are introduced to computer programming activities, as well as programmable (or 'computational') media (Peppler & Kafai, 2007). The role of computing in facilitating these experiences is crucial as it attempts to minimise the "new digital divide" (Buckingham, 2007) not only by attending to the divide between the technology 'rich' and 'poor' but also to the gap between children's digital' experiences at home and in school.

Programming has been shown to offer distinct affordances when integrated with other content areas. Grover and Pea (2013), in their meta-analysis, suggest that research could further explore how CT approaches could be applied to other subjects and how teachers and pupils might profit from this. Similarly, though many research efforts have explored this integration within STEM disciplines (Guzdial, 2008; Sengupta et al., 2013), applications of CT approaches to non-STEM disciplines are less common and arguably under-explored (e.g., literacy and the English curriculum). In response, a recent trend in fostering CT through compositional activities and creative production has indicated its potential as a suitable and motivating context for computing development. Indeed, many have pointed out the parallels between programming (or 'coding') and traditional (i.e., textual) conceptions of literacy as analogous forms of construction (diSessa, 2000; Vee, 2013). Empirical studies that take this perspective typically view visual programming as an environment that encourages learners to develop introductory programming skills by designing games, animations, simulations, and digital stories (e.g., Kafai & Burke, 2015).

This approach to computer science is often termed programming ‘with a purpose' where computational activities are integrated with disciplinary content learning (e.g., science, mathematics or literacy skills development) rather than being taught in a standalone context. This application of programming has historically been explored for its motivational and affective benefits (Kelleher et al., 2007; Robertson & Good, 2003) yet its potential to represent and construct knowledge, involving the manipulation of words, images, animations and sound, to create games, animations, simulations, and digital stories, has been notably under-explored. Research into this form of integration has to date predominantly been undertaken in the United States (e.g., Burke, 2012; Grover et al., 2018) and most commonly at middle school level (ages 11-13).

Framing programming tools as composition tools presents an opportunity to explore new ‘literatures’ (diSessa, 2000) that result from children’s design work. It also ensures pupils become digitally literate, or “able to use, express themselves and develop their ideas through information and communication technology” (Department for Education, 2013, p. 178). The Cambridge Primary Review (Alexander, 2009), in outlining domains of recommendation for curriculum reform, focuses on the dominant role of ICT in young children’s out-of-school literacy experiences, and the need for “ICT and other non-print media” (p. 24) to play a central role in the new curriculum. In the report, they argue that ICT (the pre-2014 curriculum area) reaches across the curriculum while the literacy curriculum pays “attention to text but ignores txt” (p. 24). It similarly criticised the Rose Report (2009) for promoting ICT as a core component of the curriculum. Instead, it argued that grounding ICT within the literacy curriculum enables teachers to explore relationships between conventional and novel communication forms. This aligns with literacy theorists who argue that learners need the skills to interpret and construct their own representations to be both informed consumers and active producers of digital artefacts (e.g., Kress, 2003). It also aligns with computer science advocates who similarly maintain that learners should be encouraged to engage with

computational media and tools, but also to understand how they work (The Royal Society, 2012).

At primary school, this could present an opportunity to explore how integrated literacy and computing might encourage *all* participants to engage purposefully in the construction of personally meaningful computational artefacts. If many computing initiatives aim to encourage 'computing for all' (Wing, 2008), and literacy theorists stress the need to bridge the digital divide (Buckingham, 2007), then successfully implementing the computing curriculum requires a collaborative effort to improve our understanding of subject-specific pedagogies. This includes situating computing within other contexts (e.g., literacy through multimodal composition). It could also contribute to our understanding of computing and computational thinking, regarded as in its infancy compared to more established disciplines (Yadav et al., 2018). A third benefit is that it would offer insights into the role of digital technologies and computing in the literacy classroom. Though some progress has been made, significant questions remain on how to integrate computing and CT into existing curricula across many international contexts (Blikstein, 2018; The Royal Society, 2017).

1.3 Thesis goals

In this thesis, I want to argue that an integrated approach to primary computing and literacy education could provide distinct benefits for teachers and learners. By looking at how a curriculum unit can be designed to support learners through integrated activities, I will explore whether and how such an innovative approach could provide mutual learning benefits. The goal of this thesis is therefore to develop young children's programming and storytelling practices by designing, testing and refining a series of activities. The central argument of this thesis is that that multimodal composition—or the practice of composing digital texts using multiple representations—can support young children in context-specific programming and storytelling concepts and practices and provide an authentic learning context in which to practice these skills.

Engaging in multimodal composition (MMC) or, more specifically, multimodal *storytelling,* is argued as a framing device to provide an engaging and meaningful purpose for young children's learning in primary literacy (or English) and computing, providing opportunities for purposeful engagement in a novel context. This thesis also considers how focusing activities on certain concepts and practices of MMC could promote different and arguably more productive aspects of children's storytelling and programming. The following chapters report the design of these activities, including the integration of teaching strategies, learning supports, and structured goals, and their implementation and subsequent revision across a series of empirical interventions. The purpose of these interventions was to test and refine the design of the curriculum unit, and to consider how young children could better be supported in achieving programming and storytelling goals in an integrated context.

1.4 Conclusion

Since its introduction in formal educational settings, computing has been contested by both researchers and practitioners, partly due to the limited evidence base in primary settings. Further research into the educational benefits of computing education is arguably needed to further our theoretical and practical understanding of the discipline.

Computing education researchers including Grover and Pea (2013) and Lye and Koh (2014) point to the value of school-based interventionist studies as a means to improve our understanding of subject-specific pedagogies. More specifically, they argue for the design and exploration of learning environments that foster computational practices and perspectives and research that explores how programming can be framed as a medium for other subjects. According to Guzdial (2008), this is more prevalent in STEM learning, where computational ideas are more naturally integrated.

In this chapter, the current context of computing in the English primary curriculum was discussed, providing insights into the challenges of implementation in schools. Despite a substantial body of research to support this, there are few studies that explore the application of computational thinking to non-STEM disciplines (e.g., literacy). The chapter has

highlighted this gap within the existing research literature and offers a focus for the thesis. In the next section, a brief description of each chapter is detailed to provide an overview of the structure and content of the thesis.

1.5 Layout of thesis

This final section outlines the layout and content of the thesis. Chapter 2 details the background and context of computing and literacy in the English primary school curriculum. It begins by introducing the notion of ICT, the precursor to Computing, as an example of an integrated context for young children's computing- and literacy-based skills. It provides examples of where computer-based or digital learning tools have provided powerful and engaging environments in which to practice and develop storytelling and programming skills. The chapter then describes relevant literature that pertains to developing computational thinking, most commonly through project-based work in programming environments. It equally reviews research on the practices of multimodal composition relevant to storytelling and considers how relevant aspects of MMC can augment the storytelling process. Thirdly, it reviews the curricular demands within the English primary school context and considers how a curriculum unit could serve mutual learning goals within an integrated context. Finally, a theoretical framework is proposed that frames the values and aims sought in this thesis, while demonstrating the unique perspective adopted in empirical work. The chapter ends by outlining relevant research questions based on the gaps noted in the literature review and describes them in terms of what empirical work could be undertaken.

Chapter 3 (Methodology) details the chosen methodological approach, design-based research (DBR), in designing, testing and refining the curriculum unit through a series of empirical investigations. The chapter discusses the process and outputs of DBR, highlighting its argued contributions to knowledge, and potential limitations in deriving and communicating research findings. It then discusses the procedure taken across multiple empirical studies, including the role of design principles in framing and communicating design findings. The next section outlines the data generation methods adopted across multiple studies

and details the analytical strategies used to derive insights into the curriculum unit in practice. It then argues how this approach was used to determine the relative successes (and failures) of the enacted curriculum unit and highlight areas for iteration in successive studies.

Chapter 4 details the proposed design of the curriculum unit enacted in Study 1 (Chapter 5). It first details relevant design principles derived from relevant research and then outlines how these were instantiated in key design features within the curriculum unit. It then discusses how these principles informed the sequence of curriculum unit activities, scaffolding measures and supports, and instructional strategies proposed to achieve the overall pedagogical goals.

Chapter 5 describes the first iterative study to test the design. In seeking to answer the broad question of whether and how multimodal text construction can mutually support students' programming and storytelling practices, it focuses on two areas of inquiry: (i) identifying what forms of programming and storytelling practices are fostered through multimodal composition activities, and (ii) determining how the practices of programming and storytelling in the production of multimodal texts relate to one another. The chapter details the procedure of enacting the proposed design and applying an analytical procedure to consider its pedagogical value. It then reports the findings of the empirical work, highlighting areas where MMC supported learners in achieving programming and storytelling goals, but highlighted weaknesses in the design in terms of chosen activities and scaffolding measures afforded learners. The chapter ends by summarising these findings and offering in terms of the enablers and constraints afforded by the design and offers proposed design modifications enacted in the subsequent study.

Chapter 6 details the second empirical study, which focused on implementing the revised design based on findings and areas for development identified in the first study. It first introduces design modifications derived from Study 1 and argues how they might provide more productive opportunities for learners. This study also focused on the need to design for learners with a breadth of prior experience of computing and multimodal storytelling, and to use flexible and differentiated strategies to support and challenge them accordingly. The

design of the curriculum unit is then discussed in relation to the challenges and contextual demands brought on by the final empirical study (Chapter 7) and how these could be mitigated through design measures.

Chapter 7 details the third and final empirical study. This study investigated how the curriculum unit could be collaboratively adapted and then facilitated by a practicing teacher and then embedded in regular classroom instruction. It sought to achieve this aim by focusing on three important areas of inquiry: (i) investigating teacher/student experiences participating in the intervention during their normal classroom time, (ii) determining perceived factors that enhance or inhibit the pedagogical aims of the intervention, and (iii) considering how the curriculum unit could be further adapted to meet the needs of stakeholders. Empirical findings illuminate the perceived benefits the curriculum unit afforded the participant teacher, while identifying challenges during implementation. These included the difficulties of timetabling integrated activities and the relative availability of resources to furnish activities. Finally, the chapter discusses the future use of the curriculum unit by the participant teacher and offers recommendations for future practice.

Chapter 8 discusses the findings and implications of the research reported in the thesis. It begins by summarising the findings of each empirical chapter and synthesises these results against existing research. It then considers how findings compare to previous research, whether and how they support previous findings or offer new insights, and what contribution to knowledge has been made. It then considers whether and how aspects of the curriculum unit supported the intended learning goals, and how these were adapted across successive iterations. As the empirical work centred on research-practice partnerships and enacting theory-derived principles in real-world contexts, findings are discussed both in terms of their implications for future theoretical work and practical work. Likewise, it discusses the limitations of the empirical studies and analytical approach and offers areas for suggested improvements.

CHAPTER 2
Literature review

2.1 Introduction

The central goals of the thesis are to test and refine a theory-derived design that aims to support mutual programming and storytelling practices. The purpose of this chapter, therefore, is to provide background literature on the diverse fields that this thesis brings together, including computing and literacy education or, more specifically, research on programming and storytelling, multimodal composition, and relevant learning theories.

The first two sections in this chapter outline literature relevant to both computing and literacy education, respectively. The first explores the notion of computational thinking and the challenges in defining and integrating it into classroom practice. It then offers a conception of computational thinking and details how this informs the subsequent literature consulted and design enacted thereafter. Correspondingly, the second part of this chapter discusses the context of literacy education, the practices of storytelling, and the theoretical motivation for digital storytelling in this context. The third section discusses multimodal composition and argues for its use as an umbrella term to integrate the mutual goals of programming and storytelling in the intended curriculum design. Finally, the research aims of the thesis are discussed and a summary of the chapter is offered.

2.2 Computational thinking (CT)

2.2.1 Traditional approaches

A central requirement of primary school teachers is to integrate computational thinking into their computing lessons. From a curricular perspective, computational thinking (or CT) refers to the concepts and practices derived from the field of computer science, including, but not limited to: using sequences, loops, and variables to create, debug, iterate, and decompose computer programmes (Department for Education, 2013). Integrating these concepts and

practices into existing curricula has emerged in recent years as a central goal within computing education research (Grover & Pea, 2013).

Though a relatively recent phenomenon, the history of CT stretches back decades. In fact, Papert (1980) coined 'procedural thinking' as a set of critical thinking skills that emphasises the connections between computing and everyday life (e.g., giving motorists directions) through computational ideas and metaphors (e.g., naming, manipulating, and recognising patterns in complex problems). His work in the development of LOGO, an early programming environment, was an attempt to concretise these skills (Papert, 1980). In recent years, Wing (2006) has repopularised the term as a problem solving approach. She argues that the concepts and practices of CT are universally applicable within and outside the field of computer science.

2.2.2 Visual programming

Though implementing the computing curriculum has demonstrably proved difficult for educators, a proliferation of age-appropriate tools is available to support the design of activities to support learners in developing skills and understanding. Most notably, visual programming languages (VPLs) such as Scratch (see Figure 1) and Alice have gained prominence in recent years. These low-entry visual programming tools that can serve a valuable pedagogical role by reducing the barriers to introductory programming for novice programmers. Moreover, they aim to support programming skills by foregrounding project-based applications such as multimedia projects, digital stories, animations, simulations, and games.

Visual programming languages are valued widely for mitigating some of the challenges that novice programmers encounter. For instance, dragging blocks of 'code' (see Figure 1) and snapping them together in a LEGO-like fashion allows learners to quickly programme functional projects and not worry about syntax errors affecting their programmes. The use of drag-and-drop blocks of 'code' to write scripts and create basic programmes may account for the significant adoption in primary and secondary school settings. In fact, the Royal Society report (2017) found that that 91% of primary school teachers in England

reported using Scratch over all other available tools. Scratch was the most used programming language at the primary level (38% of responses) and the second most used at secondary level (19% of responses). While these results are limited to 174 primary school respondents, the secondary school figures (1,508 respondents) give a much clearer picture of the popularity of VPLs in education in England.

2.2.3 Fostering CT

The evidence from the literature detailed in the previous chapter highlights a growing problem in computing education: the failure of the subject to engage students' interests and connect meaningfully with their wider learning experiences. Though colourful, age-appropriate tools exist to supplement instruction and provide engaging learning experiences, the goals of the computing curriculum (or 'programme of study') in England arguably fail to serve all learners.

The problem of designing 'computing for all' is not new, however. In reporting on the problem of interest and uptake in higher education, Guzdial (2017) underscores the need to design a curriculum for all learners and not just those interested in pursuing computer science. In this sense, the curriculum should serve the needs of both students and teachers. As a starting point for the design and study of learning environments in computing, he suggests this curriculum should encompass a broader set of learning activities and diverse curricular goals. One attempt toward this end is demonstrated in his work on Media Computation (Forte & Guzdial, 2004), an undergraduate course at Georgia Tech. The designed curriculum aimed to address how computing could be designed with a broad set of learning goals, including the integrating of computation through the creation and manipulation of meaningful and engaging digital media. They argue that media in the context of learning computation provides a motivating framework for students and argued that broader literacy skills as an alternative goal within computing education could also be developed.

2.3 Alternative conceptions of CT

Since Wing's original definition of CT was given in 2006, research has challenged this conception and sought to reframe CT to encompass not only concepts and practices of CT, but learners' dispositions and perspectives, as well as the additional creative, collaborative and communicative dimensions of the discipline (Grover & Pea, 2018). Alternative views on what CT is, why it is important and what should be taught have existed for several years. In some ways, these conceptions are an attempt to respond to the problem of what Papert (1987) termed technocentrism. A portmanteau of techno- and egocentrism, it refers to the idea that computers and computational tools are privileged over the learning goals and experiences of those using them and the environment in the technology is being used.

More recent research has embraced the complexity of learning with computational tools as existing beyond the confines of the computer. For instance, Lye & Koh (2014) argue that CT can be understood not only as an individual cognitive tool for problem solving and reasoning, but as a means to integrate computation through project-based learning or the production of computational media. This perspective foregrounds designing personally meaningful projects and embeds relevant problem solving throughout.

This position is also noted by Kafai and Burke (2014) who argue that learning should be framed in terms of computational participation, emphasising the development of communities in which computational activities occur. Similarly, diSessa (2000) underscores the social and cultural dimensions of computation as a series of "technology-dependent communicative skills" (Vee, 2013, p. 45). As computer programming extends beyond the domain of computer science and connects with other areas of learning, some have argued, including Vee (2013), that learners should be encouraged to compose and interpret new types of text or new literatures. This delineates computing as a discipline, comprised of elements of computer science, data science and information technology, and computing as a medium which permits the exploration of other domains such as science or mathematics.

2.3.1 Computational action

For the purposes of this project, CT is defined as the concepts and practices derived from computer science through creating and exploring computational media and tools. As such, CT is posited as both a learning goal within computing education and as a means to connect the discipline of computing with other areas of learning through constructive tasks (e.g., project-based activities). This conception aligns strongly with the ideas of computational action (Tissenbaum et al., 2019). Instead of framing CT within cognitive parameters—where problem-solving and an understanding of computation are the ultimate learning goals, computational action (CA) aligns more with a situated and applied perspective to learning. This instead emphasises interest-driven and real-world activities, drawing on a range of learning theories and traditions, most notably constructionism (Harel & Papert, 1991). Computational action seeks to reframe computing education from its focus on teaching the fundamentals of programming—or computation—towards more meaningful project-based ends. Instead of learning the definitions of variables, loops, and conditionals, for example, they argue that learners should be provided opportunities to create with programming in ways that directly affect their lives and have personal and social relevance. From the design of the learning environment to setting desired curricula and pedagogical goals, they argue that computation should provide learners with more engaging opportunities to connect with computing through meaningful and relevant learning experiences.

Underlying this conception is a theory of learning that emphasises building meaningful connections between disciplines. Papert (1980), who grounded his work in the constructivist theories of Piaget, in particular the metaphor of knowledge construction, extended this to consider how knowledge is constructed and used. He theorised that building relationships between old and new knowledge and through interacting with others could be achieved through constructing “artifacts of social relevance” (Kafai, 2005, p. 35). He also claimed that this process would enable more desirable and positive outcomes in learning contexts.

Many examples of computing curricula exist that point toward providing learners with meaningfully embedded CT in authentic contexts. Naturally, these contexts are situated, in that learning is bound by the context in which it is taking place (Lave & Wenger, 1991). For example, Balch et al. (2014) devised the *Creative Computing Curriculum* to enable learners to develop personal connections with computing. They argue that learners' creativity, imagination, and interests should be involved in early computing experiences through meaningful artefact construction, including game design, storytelling, and animation. These activities privilege learner agency in computing by emphasising the need to empower learners to create "dynamic and interactive computational media that they enjoy in their daily lives" (p. 7). Similarly, Tissenbaum et al. (2019) describe research using MIT's App Inventor as a means to enable learners to tackle relevant problems in real-world contexts thereby demonstrating the potential of reframing CT around more meaningful projects and applications.

2.3.2 Computational action across the curriculum

Beyond the cognitive framing of CT as a domain of higher-order problem-solving and thinking skills, computing education researchers have suggested that CT incorporates new "material, cognitive, and social elements" that encourage new ways of thinking, learning and representing knowledge (Blikstein, 2018, p. 10). They further this claim by emphasising how disciplines could be transformed using computational tools (diSessa, 2000), including mathematics (e.g., Benton et al., 2018).

A fundamental characteristic of computational action is the commitment to embedding experiences of applying computation across a wide variety of authentic learning experiences. Within formal educational settings, existing research points toward many different forms of integration with much of this centring on the application of CT within STEM (science, technology, engineering, and mathematics). For instance, in the United States, Kelleher and Pausch (2005) help to clarify the role of programming in integrated learning environments. They point out that we can understand programming tools as being broadly

situated within two categories: (i) ‘teaching’ systems, where the goal is to learn programming, and (ii) ‘empowering’ systems, where programming is taught in pursuit of another goal (e.g., project-based learning). Though much historical work has centred on the former category, particularly the cognitive benefits of programming (e.g Pea & Kurland, 1984), the trend towards researching the latter category is growing (Brennan & Resnick, 2012; Resnick et al., 2009). A US report (National Research Council, 2011) extends this view of the role of computing in a modern curriculum; indeed, many of its authors maintain that, while difficult to achieve, integrating computational experiences across disciplines might also be more efficient than preparing stand-alone disciplinary courses.

Many international examples of successful computing integration exist. Kafai (1995), for instance, explored computer game construction and how it might benefit the teaching and learning of both programming and fractions in mathematics. Her work demonstrated how LOGO, an early example of a VPL, could be used to teach basic computational concepts such as loops and conditionals as participants created fraction-based multiple-choice problems for younger learners. Drawing on Papert (1980), she noted that design activities, in this case software design, could provide a ‘microworld’ wherein learners could explore fraction concepts and develop their ideas with fantasy contexts. In a follow-up study, Kafai (1998) found that guiding questions or challenges were necessary to guide and focus learners during the design process as results indicated their game designs and fraction content were often unrelated.

Researchers have also investigated the integration of CT in the science classroom. Sengupta et al. (2013) found that integrating CT into science lessons resulted in distinct learning benefits, such as increased modelling and simulation of complex scientific phenomena, visualisation of abstract phenomena, and synergies between scientific and computational practices (e.g., algorithm design and generating formal models). Similar examples of programming languages used as ‘reconstructible’ computational media (diSessa & Abelson, 1986) for other topics such as science and mathematics are widespread (Lye &

Koh, 2014). These studies point towards the potential benefits of visual programming to enhance learning in science and mathematics classrooms.

The integration of programming within these contexts takes different forms. In observing classroom teachers integrating computational thinking into the mathematics classroom, Israel and Lash (2020) note the multiple levels of integration that resulted in practice. They categorise these as: (i) no integration, or activities and lessons taught in isolation; (ii) partial integration, or learning strands used to reinforce content in a different learning strand, and; (iii) full integration, or learning strands taught together where the affordances of one or both learning strands are exploited to support learning in another, and vice versa. In practice, they noted how practicing teachers mostly facilitated lessons and activities in isolation, commenting on the "commitment to teaching the curriculum with fidelity", the "need to cover the mathematics content to prepare students for […] assessments, and a strongly held "belief that the mathematics curriculum was an effective way to teach mathematics concepts" (p. 373). The insights demonstrate the challenges of cross-curricular integration of learning strands within a strict curricular system. These are particularly notable where the perceived learning benefits of integration are less known or, understandably, perceived as being marginal. Nonetheless, the findings of previous research, particularly within the STEM tradition, and the growing body of literature focused on the integration of CT within non-STEM areas, present a strong case for investigating the pedagogical value, if any, of such an approach. Taken together, these studies indicate the potential for computational activities to provide tangible and fruitful benefits when integrated across the curriculum. Moreover, learning experiences have the potential to embed aspects of computational thinking and satisfy some of the curricular requirements experienced by teachers.

A lesser explored area for CT integration is literacy education. Despite a growing interest in STEM, little research has been conducted that focuses on the productive integration of programming and composition activities and how they might mutually support one another. This is surprising as arguably programming shares many traits with traditional compositional practices (e.g., deciding on a goal, organising different elements, and revising a text). For

instance, many programming curricula and activities task learners with constructing and manipulating multiple representational forms, including word, image and sound, to create projects, including games, animations, and digital stories. Proponents of project-based programming activities often draw comparisons with traditional writing, suggesting that learners should be encouraged to code new forms of digital media including stories, games and animations (Resnick, 2002). Literacy theorists similarly stress that as communication become increasingly multimodal—that is, involve more than a single representation (e.g., text, image, sound)—learners need the skills to interpret and construct their own representations to be both informed consumers and active producers of contemporary forms of literacy (Kress, 2003).

In the following section, these issues are discussed further as relevant literature is consulted to consider whether and how visual programming could meaningfully integrate within literacy education, how learners could be supported in this process, and what might be required. Through considering alternative and relevant pedagogies to embed CT in regular instruction, it is hoped to resolve some of the challenges noted earlier in this section, particularly those that relate to accessibility and interesting in computing (e.g., Larke, 2019).

2.4 Literacy education

2.4.1 Literacy in the digital age

In recent years, many researchers have argued that literacy education places an overreliance on the teaching of the written language 'basics' at the expense of privileging a broader set of literary goals and outcomes. For instance, Moje (2008) conducted a survey of adolescent literacy preferences and practices in the United States and found that teenagers reported few experiences of composing outside school. However, in observing the same participants, she noted rich and varied examples, including comics, magazines, websites and music lyrics, that were not immediately reported by participants. She also found that these experiences were compelling to learners though not immediately understood as forms of digital communication or literacy. She concluded that the presence of these informal literacy experiences—at least

from a curricular perspective—suggested that learners' implicit understanding of writing conventions was overly dependent on forms of composition (or genres) in school; conversely, composition forms outside of these genres (e.g., digital communication, texting, email) were not understood as meaningful forms of writing. Andrews and Smith (2011) suggest that multimodal forms of composition should be privileged as they supported increased ways to communicate and represent meaning.

In some ways, the pervasiveness of digital media in the personal lives of young children has challenged us to reconsider the notion of literacy and what it means to be literate. Traditional conceptions of writing tend to focus on its nature as a problem-solving endeavour, emphasising the hierarchal and recursive, as opposed to linear, structure of writing (Flower & Hayes, 1981). For instance, Bereiter and Scardamalia (1987) suggest that novice writers form ideas 'by association', writing one after another, whereas expert writers maintain communicative goals and seek to achieve these in writing. This might include defining a narrative goal, generating an initial outline and notes before beginning the composition process. However, these models have been met with criticism, with some citing their lack of consideration for the social aspects of writing (Olson & Torrance, 2010) as well as the spontaneous nature of idea generation or writing as a process of 'discovery' (Elbow, 1998). Learners need to be able to articulate and communicate ideas, as well as express these in a logical and coherent way for the benefit of the reader (or audience). In a curricular sense, this aligns quite closely with the particulars of composition (i.e., articulating ideas and structuring them in different forms) (Department for Education, 2013).

Despite the emphasis on multiple forms of representation and communication, dominant approaches to literacy teaching and learning still privilege written-lettered representation (Kress, 1997; Marsh, 2004). For instance, the revised English programme of study (Department for Education, 2013) is notably thin with recognition of alternative forms of composition beyond the "[articulation] and structuring" of ideas into speech and writing (p. 15). Contrastingly, the former National Curriculum for England (Department for Education

and Employment, 1999) presents a different, arguably broader view, of literacy and composition though still has received some criticism for its disregard for multimodal forms of communication (Andrews & Smith, 2011). Though it centres explicitly on "the ability to construct and convey meaning in written standard English" and "the use of compositional skills to develop ideas and communicate meaning to a reader", it makes some references to "non-fiction and non-literary texts... ICT-based reference and information materials... CD-ROMs, [and the] internet" (Department for Education and Employment, 1999, p. 55).

Advocating for less traditional forms of literacy such as digital storytelling is challenging as it falls outside firm curricular boundaries. Bearne and Reedy (2017) note that the literacy curriculum (in England) bears "no mention of any texts other than books and no suggestion that writing might include writing on screen" (p. 407). Further, compulsory examinations at the end of primary school focus solely on the examination of written text. The lack of an explicit mention of multimodal—or even screen-based—texts presents a tension for teachers who seek to contextualise their literacy lessons within a wider multimodal context by considering the both formal, in-school and informal, out-of-school communicative practices of learners. Despite the lack of formal recognition, Bearne notes that examples of innovation are increasingly common. Through detailed case study work, she argues that MMC can help students develop timely skills for contemporary literacy demands. Indeed, researchers have increasingly recognised and documented the impact of children's knowledge and experience of multimodal texts on their composing practices (Dowdall, 2009; Jewitt, 2008; Millard & Marsh, 2001). Andrews and Smith (2011) argue that a revised model of literacy should take into account the "digitization" of writing (p. 15) and recognise "the multimodal possibilities for composition" (p. 27).

2.4.2 Literacy across the curriculum

Alongside a rise of research looking at digital technologies, researchers have sought to determine their utility for acquiring new skills and knowledge across disciplines (Mills, 2010). From a curriculum perspective, both the Rose Report (Rose, 2009) and Cambridge Primary

Review (Alexander, 2009) necessitate the use of cross-curricular work for learners in applying their subject-based knowledge and skills in diverse areas of learning. Moreover, it is suggested that subject teacher (or learning) should be complemented with rich examples of cross-curricular studies to enable learners to see value and purpose within and across disciplines.

As writing is an important activity in daily life for representing and communicating ideas, the importance of new digital technologies and media on children's literacy development is widely debated. A review of the primary curriculum (Rose, 2009) stressed the important of ICT and digital media in children's lives as both something to learn and something with which to learn other things. This follows similar recommendations given for the development of children's literacy skills which suggests that the teaching and learning of writing skills should permeate all subject areas (Medwell et al., 2009). It recommended that ICT be embedded throughout the curriculum to develop deep cognitive skills and an informed understanding of the potential and limitations of technology. The importance of developing children's 'digital literacy' skills is also highlighted though these are not clearly defined in this context. Some of these 'digital literacy' skills might include the use of word processing tools (e.g., Microsoft Word) for composition tasks to facilitate planning and drafting (or redrafting) of text.

Concurrent with the proliferation of computing and computational tools in educational settings is their function and potential as digital writing tools (Tyner, 2014). Prior to the introduction of computing, the ICT curriculum had been widely integrated across the primary English curriculum in several different ways. Most notably, the use of word processing tools as a means to support revision in writing, and presentation tools to enhance meaning and consider the addition of different media in texts (Andrews & Smith, 2011). These communicative or integrative practices though have no official role within the wider primary curriculum (e.g., creating multimodal projects). Likewise, the new computing curriculum noticeably removed this cross-curricular 'link', instead stressing computing's "deep links" (Department for Education, 2013) with mathematics, science, and design and technology.

However, the importance of literacy's scope in computing is found elsewhere in the curriculum where it is suggested that language and literacy are "integral aspects of the teaching of every subject" (p. 10) and should be integrated where possible. This is further emphasised through a focus on writing skills such as adapting "language and style in and for a range of contexts, purposes and audiences" (p. 3). Though not made explicit, this presumably refers to the medium of production (e.g., word processing). A view of literacy as being inherently a cross-curricular enterprise has implications for the delivery and scope of the primary English curriculum. More specifically, it provides a rationale for teachers and schools to "range beyond the national curriculum specifications" (p. 6) and explore new contexts for literacy.

One view of this synergy is articulated by Walsh (2010) who defines the 'systems-based' (p. 24) digital literacy practices of learners as they create programmes, games, or other interactive media. This design-oriented view of storytelling and programming is useful as it offers a bridge between both disciplines and suggests this integration may not only be complementary, but also beneficial. If programming involves designing systems then a view of digital literacy practices as systems-based in nature might better connect the 'new' and 'old' practices within a single context. For example, in programming interactive games and animations, learners often construct 'paratexts' (Walsh, 2010), or texts which are embedded within media products such as computer programmes and games and emerge from traditional literacy practices; these might include instructions, narrative descriptions, quizzes, or fan fiction. By critically investigating how these distinct applications of literacy may be synergized, we can better understand how learners connect different learning experiences and develop new competencies in the digital age.

2.4.3 Supporting narrative construction

Supporting learners to construct multimodal stories first required us to understand how they construct stories. Bruner (1986, 1990) argues that making stories, or using representations to tell stories, is fundamental to our ability to interpret, explain and understand the world; in short, he suggests that we construct a narrative sense of reality. He noted that storytelling

illuminates how human beings organize their experience of the events they encounter in the world. Likewise, Schank (1995) suggests that our knowledge is made up of stories, and intelligence is the ability to find and use those stories. For this reason, among others, narrative structures (e.g., stories) are a privileged genre in the primary English curriculum (McCabe, 1997). In contrast to this belief, a recent report (Clark, 2016) indicated that digital environments for composition, such as texting and instant messaging, continue to dominate the writing of young children and adolescents (8-18) outside the classroom. It also highlighted the increased writing of 'code' via programming tools outside the classroom where upper primary pupils (ages 7-11) were more likely than secondary school pupils to write code and boys more likely than girls (16.0% vs 9.2%).

Constructing narratives not only enables learners to represent their understanding of events, experiences and knowledge of literary conventions, but also stipulates how they can develop that understanding through exploration within a specific medium (Bruner, 1991). Stein and Glenn (1979) in their work on 'story grammars', suggested that this understanding can develop learners' internal cognitive structures, or 'narrative schemas'. These schemas emerge out of learners' experience of creating personal or fictional narratives including fairy tales, oral recounts, and other literary forms. They are also understood as a fundamental structure of human meaning-making (Bruner, 1991) wherein storytelling activities help organise and represent our knowledge and experience of the world (Schank, 1995).

An extensive literature documents the developmental changes in children's narrative abilities from producing oral to written and other forms of narrative. Roth (2000) argued that narrative writing is complex and has its basis in children's oral narration skills. As children grow older, oral narratives change in structure and coherence (Stein & Trabasso, 1981) and narratives grow in length and syntactical complexity. Despite claims of ICT or computational tools supporting children's storytelling in different ways, many of the cognitive demands of writing persist. Kellogg (2008) argues that writing is a complex process as learners engage in processes of idea generation, deciding on spelling and grammar, ordering their thoughts, and

drawing on their knowledge of story structure. In planning for writing, learners must also draw ideas from memory, develop concepts, create an image of the reader, test ideas and text against that image, translating ideas into words, transcribing these ideas onto paper or screen (Flower & Hayes, 1981).

There is also evidence to suggest that computer-based writing tools can engage and support learners in this process. For instance, Graham et al. (2012) found in a meta-analysis of writing studies the quality of student writing was enhanced using word processing tools. One common reason for this quality shift is given by Owston et al. (2009) who maintain that learners consider the quality of writing more when they know that it will be read by others; indeed, this is a common view shared by others across the field (e.g., Graves, 1983).

Conversely, there are also additional concerns about the pedagogical challenges and additional demands for both teachers and students in writing (or composing) using computer-based tools as they necessitate the production of visual texts. As Kress (2003) notes, understanding visual texts requires new logics (or grammars) to be considered. He argues that if the logic of alphabetic text is sequential and temporal, then the logic of image and visual design is spatial and simultaneous (Kress, 2003). In science education Hubber et al. (2010) argue that the cognitive demands in exploring, generating, coordinating and refining representations constitute a significant challenge for students and teachers alike. In researching how teachers approached science concepts (e.g., force) through a representational approach, they further this by suggesting that student-generated non-standard or non-canonical representations present additional demands on teachers' conceptual understandings and classroom time.

McLuhan's thesis that "the medium is the message" (McLuhan, 1964, p. 8) is useful here in considering the role of the medium in constructing, representing and communicating narrative ideas to both writers and readers. Applying this idea to the context of digital storytelling, the problem of content (or what to say) and rhetoric (or how to say or represent it) arguably represents a demand on learners' attention and working memory resources. A recent survey (Clark, 2016) substantiates this by revealing a majority of the young children

surveyed in schools struggled over ‘what to write’, with this impacting on their enjoyment of the writing process. In narrative writing, these challenges extend to the conventions of the genre.

Drawing on Graves’ writing workshop model (1983), Graham and Harris (2012) argue that pre-writing, or the preparatory stages before a writing activity, can positively impact on these challenges when composing. They suggest question prompts as a tool to help students both recognise and engage with the key components of narrative in story writing. These include:

- Who is the main character; who else is in the story?
- When does the story take place?
- Where does the story take place?
- What does the main character do; what do the other characters want to do?
- What happens when the main character tries to do it?
- How does the story end?
- How does the main character feel; how do the others feel? (p. 28)

Gelmini-Hornsby et al. (2012) found that similar questions prompts fostered increased engagement in interactive discussions and resulted in learners producing richer and more coherent stories. Likewise, Saddler et al. (2018) found explicit reference to these narrative elements through targeted questioning and note-taking on “think sheets” (p. 10) prompted learners to produce more elaborate and better stories. By modelling the pre-writing process, and explicitly referring to various story parts, learners were also found to have spent far greater time planning in advance post-treatment with this resulting in better writing outcomes.

Research on the use of scaffolding measures in composition has provided some insights into the use of graphic organisers (e.g., writing frames) though their effectiveness in the writing classroom is contested. While Fu and Shelton (2007) found evidence that struggling writers benefited from illustrating or ‘mapping’ initial story ideas, Saddler et al. (2004) found that learners did not use organisers despite teachers emphasising their use during

writing instruction. As well as functioning as an organisational device to focus learners on whole-text structure, graphic organisers may arguably help learners to reflect on both the textual and non-textual elements of multimodal texts (e.g., considering the 'framing' of a narrative); both seen as important considerations during planning and writing digital texts (QCA/UKLA, 2004). Though not explicitly mentioned in their original framework, Flower and Hayes (1984), in their multiple representation thesis, note that writers often "represent their current meaning to themselves in a variety of symbolic ways" (p. 129). These might include "nonverbal or imagistic representations, abstract knowledge networks, text-based representations, and […] formal text representations" (p. 129). In designing a learning environment to support children's writing and multimodal considerations, adequate supports are essential to ensuring successful outcomes.

2.4.4 Supporting literacy through programming activities

Exploring computing or ICT as a context for literacy skills development requires that we consider both the benefits and limitations of such an approach. Previous research indicates where computer games have been used to support children's narrative writing (Buckingham & Burn, 2007; Pelletier et al., 2010; Walsh, 2010). Games, in particular, have been noted for their use of dialogue, multimodal dimensions, and narrative structure, providing young writers with opportunities to connect fictional game worlds with more formal literacy learning. Kelleher and Pausch (2007), for example, noted that storytelling using an adapted version of Alice promoted student interest in learning how to programme; in other words, students were motivated to programme in order to tell stories. While this finding is consistent across the literature (Burke & Kafai, 2012), less attention has been given to the role of MMC, more specifically digital storytelling, in the contemporary literacy classroom.

A notable study by Kafai (1995) found that through programming activities, children began to incorporate narrative elements into their LOGO game design projects and focus less on the ludic elements of their designs. She argued that narratives provide an overarching structure for learners to connect scenes and give focus to their projects. Buckingham and Burn

(2007) similarly argue that exploring the narrative elements of analogous media, such as character types, narrative events (e.g., conflicts and resolutions) and story settings, are not only central to the design of digital texts (in this content, games) but fundamental to their appeal to learners.

Some evidence suggests that this might be the case. For instance, Robertson (as Howells & Robertson, 2012; Robertson, 2005) point to similar insights in their work on Adventure Author, a computing game authoring tool. By repurposing an existing commercial role-playing game, Neverwinter Nights (NWN), their aim was to develop specific literacy (i.e., narrative) skills in a programming environment. Unique features of the tool included a visual representation of the story structure being provided, mapping out scenes or individual episodes within a plot structure and a branching mechanism to create non-linear (or interactive) dialogue. An initial evaluation suggested that learners were motivated using digital tools and that their engagement with the tool enabled them to practice narrative skills including dialogue writing, creating multiple plot lines and multi-linear narratives, and reflecting on character and situation within a story. Learners were also found to have engaged in metacognitive behaviours, including generating and selecting ideas, planning, testing and revising when encouraged to reflect and revise their narratives.

One point stressed is the additional demands placed on learners as they consider the means of delivery (i.e., the representational forms chosen) in telling a story, as well as anticipating the readers' actions and responses. As both writers and designers, learners had to make decisions about the textual, visual and spatial elements of their projects, and how they form a unified story. The authors also introduce the notions of story 'locations' to convey how learners are encouraged to approach the task of situating a narrative visually. Indeed, they recognise the importance of visual design skills in creating a more complex and rich multimodal narrative text. The study suggested that learners benefited from 'new media' storytelling and visual design but that they needed additional guidance to develop their sense of audience (Robertson & Howells, 2008).

Alternatively, it may be the case that visual programming as a context for literacy, and vice versa, could also hinder learning in both. In contrast with studies mentioned previously, Kelleher et al. (2007) suggest that integrated programming and storytelling poses several limitations. Their work explored the use of Storytelling Alice, a programming tool adapted for story authoring, exploring whether creating stories and practicing the basic concepts of programming can be understood as essentially "the same activity" (p. 63). Though integrated object-oriented programming and storytelling activities was achieved, the authors found that the level of programme 'complexity' encouraged by story-like structures was an issue. These results are supported by more recent work by Franklin et al. (2013) on the integration and assessment of Scratch programming projects within an project-based programme that incorporated storytelling activities. Though students demonstrated competencies in areas such as event-based programming, initialisation, and message passing (or 'broadcasting' in Scratch), they found that few student pairs were successful in using more complex programming features, including loop functions, in their final project work.

Additional literature suggests that the 'linearity' of stories in the creation of complex computational media does not encourage complex computational ideas (Adams & Webster, 2012; Burke, 2012). Despite this, the authors found that while learners' projects initially followed a linear sequence, over time they transitioned to including more complex programming constructs and practices as their applications became clearer. They also point out that certain elements of digital stories encouraged certain programming constructs over others (e.g., high-level animations encouraged learners to explore methods, loops and parameters). The use of an existing bank of characters, scenery and visual resources also supported the generation of story ideas during the drafting phase. They concluded that learners who used Storytelling Alice were not only motivated to programme, but also encouraged to construct longer, more complex narrative structures.

Results from these studies are not without their caveats. It is important to note that, with the exception of Burke & Kafai (2012), these studies employed programming

environments that were purpose-built or adapted for narrative goals. Unlike many of these tools, Scratch is not designed for any specific purposes and, as such, can arguably be repurposed for different outcomes. Furthermore, this research has largely focused on designing and evaluating tools for learning with narrative and visual design skills positioned as secondary outcomes while programming serves as the main outcome. By contrast, this PhD seeks to use Scratch, an existing and relatively common tool, to consider how these practices can be mutually beneficial and to explore ways in which they can feasibly be used in practice.

These studies suggest that by grounding programming tasks within the framework of learners' existing story knowledge and narrative skills, the mutual benefit of supporting programming at earlier ages can be achieved. They detail how tasking learners with selecting, manipulating and coordinating different representational forms, and drawing attention toward salient features of narrative construction such as the role of the audience, dialogue, setting, character development, and how these relate to producing a coherent narrative. Though these studies have demonstrated how storytelling through multimodal composition might motivate learners into programming, less is known about the process of multimodal storytelling in programming environments, how it can be supported, and what pedagogical value it brings to both subject areas, respectively. Understanding programming as more than a means to learn computation but also as a compositional tool, requires that we can explore the productive outputs, or new literatures (diSessa, 2000) as children learn to express themselves in different forms. As mentioned earlier, computing entails an entirely new medium of composition, 'code' (Vee, 2013) which may impact learners in how they approach composition tasks. As diSessa (2000) notes, it might allow learners to see new paths and connect existing ideas (i.e., of narrative and story structure) with the features and possibilities of a new medium. Moreover, developing our understanding of this approach requires that we move beyond researching the motivational or affective benefits of multimodal and media composition (e.g., Forte & Guzdial, 2004) and seek to understand what and how students learn through integrated approaches to CT.

2.5 Multimodal composition

Integrating the distinct learning strands of computing education through programming activities and literacy education through storytelling activities required a framing device to align the distinct and complex practices of both learning strands. Toward this end, the notion of multimodal composition is invoked as umbrella term to frame the process and product of generating multi-representational narrative texts in a visual programming environment. Multimodal composition (MMC) is broadly defined as the use of multiple representations (e.g., sound, text, images, gestures) to produce a composition for a communicative purpose (Bearne & Wolstencroft, 2007). MMC as a literacy practice is therefore the process of learners selecting, coordinating and manipulating representations to produce 'texts' largely in a digital or computer-based context (New London Group, 1996; Smith, 2017). Recent years has seen its popular grow as an increasingly common form of composition (see Pantaleo, 2017; QCA/UKLA, 2004) in which learners construct multimodal texts, including comic books, graphic novels, computer games and animations, and even traditional picture-books (B. E. Smith, 2017). Millard (2006) suggests that MMC offers a means to 'fuse' home and school literacy practices by encouraging the 'hybridisation' of popular culture (e.g., games and comics) and classic texts (e.g., children's literature). The process of creating multimodal texts can be undertaken digitally though this is not exclusive to any particular format or medium.

Over the past two decades, a growing interest in embedding formal experiences of multimodal composition has resulted in efforts to study their role in the writing classroom. For instance, research undertaken by the United Kingdom Literacy Association (QCA/UKLA, 2004, 2005) into children's multimodal work yielded many examples of multimodal texts and their merits against then current writing standards. However, the assessment tools used to evaluate students' work have been contested, with many noting its focus on a limited definition of multimedia and reliance on adapting existing writing assessment criteria (McGrail & Behizadeh, 2017; Vincent, 2006)

Existing literature has also suggested the benefit of MMC practices in applying learners' storytelling skills in new contexts. Recent scholarship has argued that composing digitally promotes new 'literatures' to explore (diSessa, 2000), including adventure-based hypertexts and interactive fiction (Proctor & Blikstein, 2017). Myhill and Cremin (2013) argue that as the conventions and audiences of writing change, so too do the task goals and written content they need to create and revise in light of a changing brief.

Unlike traditional writing practices, multimodal composition requires learners to consider additional aspects of their texts. If we take this view that certain contexts for writing include more than words, sentences, and paragraphs, then our definition of literacy must broaden to encompass digital, multi-linear, multimodal (e.g., multimedia presentations), and 'hybridised' (i.e., combining multiple genres) contexts for literacy development (Merchant, 2007). In the case of composing a multimodal text digitally, children can use a range of fonts, pictures, sounds, and other means to represent and construct meaning.

The shift away from 'monomodal' views of literacy (Andrews & Smith, 2011), while important, has yielded little impact in embedding multimodal composition practices in primary classrooms in the UK (Nixon, 2012). Nonetheless, the widespread use of visual programming languages in computing arguably constitutes a rich multimodal experience for young learners. For instance, in visual programming tasks, learners are tasked with coordinating and integrating multiple representational forms for different purposes and goals, one being the construction of multimodal 'texts. The combination of multimodal composition in a VPL requires learners not only to select and manipulate these resources, but to program how these resources are sequenced, in what order, and how this might be interpreted by a 'reader'.

Like many researchers interested in MMC, the thesis of this research project has been that curricula and classroom practices should take account of the increased multimodal and digital nature of students' literacy experiences and develop appropriate frameworks and approaches for supporting students' engagement with a range of 'new' media (Dyson, 2008).

The Essex Writing Project, the UKLA/National Strategy *Raising Boys' Achievement in Writing* and *More than Words* projects were designed to satisfy this aim by focusing on the development of a set of progressive statements detailing children's composition practices of multimodal texts (or MMC) (see Table 1). Over a period of four years, the research project involved working with forty-eight teachers in three local authorities in England on an action research project designed to raise boys' achievements in writing through drama and visual texts. The result of this co-design process informed the development of curriculum descriptors for multimodal text composition in line with comparable descriptors for progress in writing. The need for these descriptors was born out of a desire to characterise and understand the unique nature of multimodal composition and the contribution it can make to children's literacy experiences and meaning. Though explicit attention had been paid to the multimodal dimensions of children's writing in earlier work, the Scale of Progression in Multimodal Text Composition (Bearne & Wolstencroft, 2007) specifically examined both text making on screen and on paper. In developing screen-based multimodal—as opposed to monomodal—texts, Bearne argues that learners need to take increasing responsibility in multiple defined areas of multimodal composition. Table 1 (Bearne & Reedy, 2017, p. 406) details this progression.

The development of the progression in multimodal composition (see Table 1) emerged to provide tools to assess both learners' screen-based and paper-based multimodal text construction. Recognising the monomodal nature of existing frameworks and reference assessment tools, the progression details additional considerations not commonly found in formative or summative assessments of children's writing. For instance, some of the practices detailed including deciding which modes will best communicate meaning for specific purposes or choosing font/typography effects to enhance and shape meaning. In short, Bearne's tool serves as an approach to characterise and understand the nuances of constructing multimodal texts for different purposes:

Table 1: Progression in multimodal text composition

MMC activity	Practices
Decide on mode and content for specific purpose(s) and audience(s)	Choose which mode(s) will best communicate meaning for specific purposes – for example, deciding on words rather than images or gesture/music rather than words
	Use perspective, colour, sound and language to engage and hold a reader's/viewer's attention
	Select appropriate content to express personal intentions, ideas and opinions
	Adapt, synthesise and shape content to suit personal intentions in communication
Structure texts	Pay conscious attention to design and layout of texts, use structural devices (pages, sections, frames, paragraphs, blocks of text, screens, sound sequences) to organise texts
	Structure longer texts with visual, verbal and sound cohesive devices
	Integrate and balance modes for design purposes
	Use background detail to create mood and setting
Use technical features for effect	Handle technical aspects and conventions of different kinds of multimodal texts, including line, colour, perspective, sound, camera angles, movement, gesture, facial expression and language
	Choose language, punctuation, font, typography and presentational techniques to create effects and clarify meaning
	Choose and use a variety of sentence structures for specific purposes
	Handle technical aspects and conventions of different kinds of multimodal texts, including line, colour, perspective, sound, camera angles, movement, gesture, facial expression and language
Reflect	Explain choices of modes(s) and expressive devices including words
	Improve own composition or performance, reshaping, redesigning and redrafting for purpose and readers'/viewers' needs
	Comment on the success of a composition in fulfilling the design aims
	Comment on the relative merits of teamwork and individual contribution for a specific project

This PhD proposes that the respective goals of programming and storytelling can be framed within the domain of multimodal composition. In doing so, it argues the need to consider whether and how these learning strands complement one another and the impact of MMC on this process. As noted by Vincent (2006), assessing learners' digital texts or composing practices require that relevant criteria in outlining, constructing and evaluating texts are proposed. For instance, the lack of concern for the multimodal aspects of texts in writing assessment literature has resulted in a notable lack of multimodal-specific assessment criteria (McGrail & Behizadeh, 2017). Characterising and assessing learners' multimodal compositions becomes more challenging when seeking to embed specific programming and storytelling goals. One reason for this is the complex nature of the activities and the wide range

of criteria that can be used. For this thesis, a literature search was conducted to source research related to characterising and assessing children's multimodal work. Adopting a framework to such a particular context needed to be approached flexibly and be adapted to suit the tools and approaches used. Andrews and Smith (2011), for instance, argue that such a framework should provide flexible yet robust learning goals that have implications for curriculum design and assessment. In the following section, multiple perspectives on this are reviewed and a framework is proposed.

2.5.1 Researching multimodal composition

Different models (or rubrics) for assessing multimodal texts are difficult to reconcile owing to their use of multiple theoretical frameworks and different language to describe the process and product of MMC (e.g., McGrail & Behizadeh, 2017). Work commissioned by the United Kingdom Literacy Association and the Qualification and Curriculum Authority (see UKLA, 2004, 2005) has sought to provide assessment tools to evaluate children's multimodal work though these have since been criticised (see Vincent, 2006) for their predominance of non-digital texts (e.g., picture books and graphic novels) and reliance on existing writing standards and rubrics used to assess print-based compositions. Internationally, nationwide assessment projects in the US by the National Writing Project sought to resolve this issue by developing a theory-inspired framework for assessing multimodal work. The committee of the Multimodal Assessment Project (MAP) proposed five domains to establish a shared language to discuss and evaluate valued aspects of multimodal composition (Wahleithner, 2014; Eidman-Aadahl et al., 2013). These are provided in Table 2 with a short description to illustrate each one. Further, each of these domains is discussed in greater detail in the following sections.

Table 2: Domains of multimodal assessment

Domain	Description
Artefact	Technical outputs of multimodal composition activities
Rhetorical skills	Considerations for the purpose, audience, and composing environment (or medium) in which the composition is taking place
Substance	Criteria related to the credibility, accuracy, and significance of compositions in relation to a given task
Process management	Skills related to planning, developing and evaluating multimodal work, as well as the ability to collaborate with others,
Habits of mind	Patterns of behaviour or attitudes that develop over time and exist independent of specific instances of composition work

Though the MAP framework (McGrail and Behizadeh, 2017) reflects a 'consensus' definition of multimodal composition with no single domain fully accounting for its processes and products, it remains reflexive to each instance in which it is instantiated. Similarly, it is noted for its purpose to assess a single instance (or production of a 'text') or a writer's development over time. As such, specific criteria for assessing multimodal composition are not prescribed but suggested. In the following five sections, each domain of the MAP framework is discussed respectively. They offer distinct and valuable perspectives in which to characterise and assess learners' multimodal texts. Each domain is accompanied with select examples of the framework being applied are given. Each domain is explored within a different case study undertaken by a member of the assessment group. Applying this framework intends to build on this ambition by considering how multimodal composition can be assessed across each of these domains.

2.5.1.1 Artefact

A holistic assessment of the value of MMC as a learning activity begins with the product of participants' engagement, the artefact. Eidman-Aadahl et al. (2013) define this as the "final consumable (readable/viewable) product" that offers a coherent message through the use of "structure, medium, technique, [and] message" (p. 4). Wyatt-Smith and Kimber (2005) define three criteria for considering 'what counts' in multimodal text production: (i) cohesion, or the organisation of a composition; (ii) content, or the individual multimodal elements, form, and originality of a composition, and; (iii) design, or the way multimodal elements interact with

one another, as well as the grammars (or logic) of visual, verbal, and computational 'communication' used. In their appraisal of student-generated online multimodal texts (PowerPoint presentations, concept maps, and Macromedia Flash applications), these criteria are explored for their utility in evaluating student work.

Assessing the *content* and *design* of a multimodal text necessitates a consideration of how students use and manipulate representational resources (e.g., images, sounds, or text). It also concerns the way in which resources are manipulated and grouped, both within and across 'episodes' (Wyatt-Smith & Kimber, 2005). Unlike the print medium, visual and supporting materials serve not only as a supporting role, but as a series of "inter-related elements" (p. 28) that work together to communicate more effectively. These elements observe grammars (or logics) of visual, verbal, and computational 'communication' such as verbal (traditional grammar), visual (positioning, symbolic representation), and electronic text (specific logics relevant to a software environment). For print-based elements, text is often organised sequentially, temporally or some other logical order (Kress, 2003). However, for multimodal (including computational) elements, different logics could be employed to yield a 'good' multimodal narrative text; that is, to programme a story effectively.

2.5.1.2 Rhetorical skills

Rhetorical skills encourage us to ask about the environments surrounding the creation of the artefact and how the artefact enters into that world (Eidman-Aadahl et al., 2013). In the case study detailed, participants in a ninth grade American Studies classroom used Twitter to write a microblog as a collection of short stories related to American soldiers in the Vietnam War was read aloud. Assuming the character of the soldiers,, participants were asked to tweet thoughts felt as the characters, responding to other soldiers in assigned platoons, and develop poems using the resultant Twitter feeds. One group of students then composed a piece, "A Poem of the Vietnam War", that synthesised the account given on Twitter. The researchers noted how deeply students considered the environment and medium used in the production of multimodal artefacts, as well as the purpose of the task and intended audience. The purpose of

the task was to tweet how a soldier might feel in battle and consequently develop empathy and resonance with characters. By each assuming a character role on Twitter, participants and their peers and teachers served as a defined audience.

Other research into MMC pays similar attention to the role of audience as a contextual factor of the process of creating a text. In Penrod (2005), for example, the production of contemporary "e-texts" is compared to traditional "papertexts" (p. 30) where specific attention is paid to the increased role of audience in a writer's rhetorical skills. For instance, she argues that while monomodal texts require careful consideration of words and phrases to provoke a certain response, writing multimodally includes an additional level of complexity in compositing multiple modes to anticipate an audience reaction.

Context, in another sense, refers to affordances of the platform in which composition is taking place and the specific design considerations for that context. In the case study noted above, using Twiter presented participants with the challenge of keeping their posts to a 140 character limit. This meant that language needed to be consise and meaningful. One affordance of composing in a Twitter feed is the creation of a timeline of posts which are then used to create the outcome (written poems). In this sense, multimodal work functions not as the artefact for assessment, but as a process used to create the artefact. Applying this understanding to a programming environment might involve an understanding of the conventions of programming languages and how they operate. In the context of this thesis, knowledge of the affordances of visual programming, as well as the constraints of such a medium, would fall under this category.

2.5.1.3 Substance

Aside from the features and processes required to create a multimodal text, the *substance* of a text refers to the "quality of ideas, credibility, accuracy, and significance" in relation to purpose and audience (Eidman-Aadahl et al., 2013, p. 10). It is described as an attempt to re-join form and content, two elements artificially separated in the Artefact domain. For example,

a technically sophisticated presentation (e.g., a PowerPoint with meaningless transitions) might detract from the overall message or impact of the text.

While criteria such as credibility, accuracy or significance might be important for expository multimodal texts, their function in examining narrative texts is contestable. Where the accuracy and relevance of information, the selection of sources, and documentation function to create a quality expository text, evaluating narrative texts requires a different means of evaluation. For instance, narrative texts typically rely on other criteria, such as story structure (McCabe & Peterson, 1991). In *What Makes A Good Story?*, McCabe & Peterson (1984) discuss *structure* as a key factor in determining the substance of children's narratives. To structure a cohesive multimodal narrative text, multiple representational resources such as images, sounds, and blocks of code (the 'content' of a text) must be assembled. A detailed example given by Eidman-Aadahl et al. (2013) discusses a student-generated video project, *Cheese*, a three-minute action movie centred on two protagonists who infiltrate a high-security fort to obtain "the cheese". The structure, content and use of specific design and medium affordances are assessed. Their analysis suggests that due to a lack of explanation and background information, the film is found to leave the audience with many questions and confusion. The researchers note how these omissions give the project an 'end of a movie' structure rather than a complete one. For stories such as *Cheese* to embody a more organised and coherent structure, a suitable knowledge of story structures and conventions is required (McCabe & Peterson, 1984).

As noted earlier, the roots of storytelling lie firmly in oral narrative. Story schemata, or a person's knowledge of the hierarchical components of a story and the logical relationships among these components, have been theorised under many forms. Assessing this knowledge and how it is understood and used in practice can take many forms such as high point analyses (Labov, 1972), dependency analyses on linguistic complexity (Deese, 1983), and episodic or story grammar analyses (Rumelhart, 1980; Stein & Glenn, 1979). Surveying these different approaches, Peterson & McCabe (1983) note that no one approach provided a superior

representation of the quality of a narrative structure with each one demonstrating different merits. However, an analysis of children's stories using each method demonstrated a clear sensitivity to several criteria common to all. Completeness in 'episodic structure', or the inclusion of at least three categories of events, namely (i) initiating events, (ii) goals, (iii) attempts to accomplish these goals, and (iv) consequences of actions, are seen as common measures of story cohesion or 'completeness'.

2.5.1.4 Process management

Process management refers to the "skills, capacities, and processes involved in planning, creating, and circulating multimodal artefacts" (Eidman-Aadahl et al., 2013). An example given by the authors describes an after-school computer camp for middle-school girls who collaboratively developed their attitudes and skills by using computers to design a website aimed at future computer camp participants. Though the authors discuss some of the complex processes and the technical skills required to effectively compose in a multimodal authoring environment, including coordinating different rhetorical tasks, sharing knowledge, and taking collective responsibility, specific criteria or sub-themes are not given. To explore the themes proposed, more established literature is needed for explicit criteria. For this, existing literature on processes in composition is consulted (e.g., Flower and Hayes, 1981).

The original cognitive process theory of writing (or print composition) proposed by Flower & Hayes (1981) suggested that writing functions not as a linear process, but a complex, recursive one. Accounting for this theoretical shift produced three iterative writing processes writer monitor when composing: (i) planning, (ii) translating, and (iii) reviewing. Elsewhere in the model, they additionally propose the Task Environment, including the Rhetorical Problem, and writer's knowledge of Topic, Audience, and Writing Plans, as important aspects of the writing process though these are arguably satisfied elsewhere in the framework (e.g., Task Environment as Environment/Medium) (Flower & Hayes, 1981, p. 370):

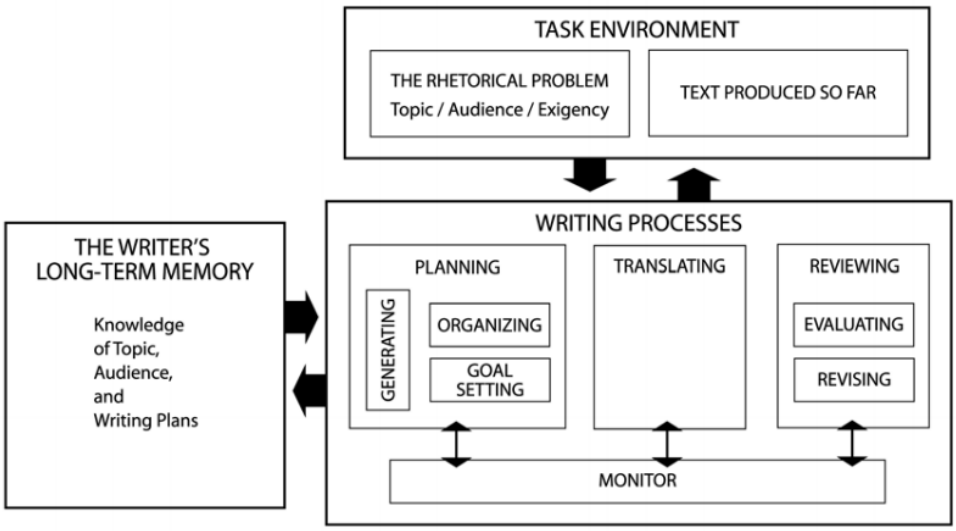


Figure 2: Hayes and Flowers' (1981) cognitive process theory of writing model

More recently, this model has been adapted for novel forms of composition. Bruce (2016), for instance, employs an adapted cognitive process model for examining video composition as an analogous process of visual conceptualisation, production and evaluation. Like written composition, he argues that video composition follows many of the sub-processes in the model, including generating ideas, revising, and evaluating (or editing and trimming), and employing knowledge of writing plans (or imitation or mimicking of existing images and sequences). Taking these processes as a basic model for composition work, assessing children's multimodal composition work could be framed as both a product (a composition) and a series of processes needed to produce it (the process of composing).

2.5.1.5 Habits of mind

Habits of mind are characterised as "the patterns of behaviours or attitudes that reach beyond the artefact being created at the moment" (Eidman-Aadahl et al., 2013). In one case study, these habits are explored through an analysis of projects created in VoiceThread, a multimedia authoring tool. A group of fourth grade (9-10 year olds) students collated multimodal resources to explore the issue of Japanese-Americans being placed into WWII internment camps. Participants added to these multimedia 'mash-ups' though a microphone, webcam, keyboard, and telephone. Their appraisal of the project suggests that most writers who participated in the

project demonstrated how their current multimodal work connected with historical episodes, including questioning societal values as both readers and writers of historical multimodal compositions. This, in turn, requires learners to connect 'in-school' learning with the real world and reflect on the lasting impact of their multimodal work (McGrail & Behizadeh, 2017).

In the case of a programming environment, Brennan & Resnick (2012) similarly refer to these as children's growing understanding of "themselves, their relationships to others, and the technological world around them" (p. 10). These are summarised as (i) expressing, (ii) questioning and (iii) connecting; collectively, these relate to children's perceptions of the value of computational media and tools, their explicit questioning of the role or value of these tools, and appraisal of their limitations. Though they point out that these shifts in perspective were observed when working with children using Scratch independently, as well as promoted through teacher-led activities, it is arguable that this dimension of computational thinking is not observable in 'the moment'. Rather, they argue this dimension is better identified over an extended period of time (e.g., across multiple instances or projects). Though these described outcomes and responses seem complex and difficult to comprehend—and perhaps more difficult to identify—they can be characterised as the values, attitudes and patterns of behaviour which ought to be cultivated as opposed to observed in learning contexts (Penrod, 2005). Indeed, Bearne (2017) cites habits of mind as important criteria for literacy educators to value students' knowledge, experiences, and lives outside the classroom.

Taken together, these frameworks point to multiple perspectives in which to characterise and assess multimodal composition. As no theoretical framework that specifically aimed to address the aims of this thesis existed, these perspectives are used in this thesis to inform a proposed framework. This framework aims to assess learners' production of multimodal texts in the pursuit of storytelling and programming goals. This process began by aligning existing work with this research project and adapting relevant criteria and adopting them according to the needs of the project. These perspectives are revisited in Chapter 4.

Further, relevant literature is reviewed that connects the desired learning goals with theory- and practice-oriented literature to inform what activities can support learners' progression towards those goals.

2.6 Research aims

Though evidence that multimodal composition can be employed to support learners' pursuit of programming and storytelling goals respectively, researching how these could be enacted within a singular context has not been clearly defined. The pedagogical value of an integrated context has also not been adequately established. As such, little is known about what constitutes best practice in computing education when compared with more established STEM disciplines (e.g., science and mathematics) (Guzdial, 2017). Moreover, with issues of uptake and interest in computing education, there also exists a need to explore alternative pathways to CT and computing education. Additionally, empirical evidence of primary school interventions is lacking.

This thesis proposes that MMC can function as a constructive process in which to exploit learners' programming and storytelling practices. While most conceptions of computing position it as a problem-solving endeavour, this research project argues that it can also be understood as a compositional tool for the production and dissemination of multimodal texts. Thus, a warrant exists to explore how MMC could be framed as a series of overarching learning activities in which to mutually support learning strands of programming and storytelling. This research therefore seeks to design, test and refine a designed curriculum unit to establish the reciprocal benefits, if any, of such an integration. In doing so, the proposed research would seek to contribute to the following areas of inquiry:

- Theory on how MMC can be productively employed in the pursuit of programming and storytelling goals
- An enacted design in the form of a curriculum unit in which multimodal composition activities are used to support programming and storytelling goals

- Knowledge about the learning processes as promoted by the design

The main research question (**MRQ**) derived from the literature reviewed above is therefore articulated as follows:

> *How can multimodal composition activities enable primary students' programming and storytelling practices to be meaningfully integrated?*

This overarching question functions as an advisory research goal (Plomp & Nieveen, 2013), intended to provide inform what a theory-driven instructional design (elaborated in Chapter 3, *Design*) based on relevant research could look like. To answer this question, it was important to first consider activities and approaches that have been used in prior research and to explored how these could be used for dual outcomes. In Study 1, these issues are considered by answering the following research question and corresponding sub-questions:

- Research question 1 (**RQ1**): Which multimodal composition activities promote storytelling and programming goals?
    - Sub-question 1 (**SQ1**): What forms of programming and storytelling practices are fostered through multimodal composition?
    - Sub-question 2 (**SQ2**): How do the practices of programming and storytelling in the production of multimodal texts relate to one another?

In Study 2, a second design cycle iterated the intervention design to address design weaknesses in the first cycle (Study 1). Again, these issues are considered by answering the following questions:

- Research question 2 (**RQ2**): How can learners with varying experiences of programming and storytelling be better supported during multimodal text construction?
    - Sub-question 3 (**SQ3**): What forms of programming and storytelling practices are fostered through multimodal composition?

- Sub-question 4 (**SQ4**): How do the practices of programming and storytelling in the production of multimodal texts relate to one another?

In Study 3, the focus of the inquiry shifts to a teacher-led implementation of the learning activities. As such, the research questions are given as follows:

- Research question 3 (**RQ3**): How can multimodal text construction that aims to support students' programming and storytelling practices be embedded within formal classroom instruction?
    - Sub-question 5 (**SQ5**): What are the experiences of participants (teacher and students) taking part in the intervention during normal classroom time?
    - Sub-question 6 (**SQ6**): What perceived factors enhance or inhibit the aims of the intervention?
    - Sub-question 7 (**SQ7**): How can the intervention be adapted to the needs of stakeholders (teachers and students)?

2.7 Summary

The literature review has reviewed relevant literature relating to computing and literacy education respectively. It first explored the context of computing in the primary school curriculum, highlighting the multiple theoretical perspectives on computational thinking (CT) and how they are instantiated in classrooms. It then discussed how CT has been embedded in existing curricular practices (e.g., mathematics and science).

The literature review then detailed research on contemporary approaches to literacy, including digital storytelling, and discussed the extent to which multimodal composition could be used to support children's storytelling practices. It also considered to what extent contemporary accounts might extend our understanding of the notion oftext and what it means to be literate. It then highlighted the multiple perspectives that can be taken to understand, characterise and ultimately assess multimodal composition for different purposes. The literature discussed in this section has suggested that MMC can be beneficial in developing

learners' narrative skills by providing opportunities for learners to construct new forms of composition and exercise their existing storytelling skills in new contexts. However, it cautioned that research in this area was limited, and that further research was not only desirable, but warranted. Finally, the research aims and objectives of this thesis were detailed.

CHAPTER 3
Methodology

3.1 Overview

This chapter details the methodological approach adopted in this thesis. The chapter is divided into four sections: the first part describes the chosen methodological approach, design-based research (DBR), and provides a rationale for its use in this PhD. It then discusses the contributions to knowledge that can be made, as well as potential limitations in deriving findings. The second part details the procedure undertaken as part of this thesis and how this was shaped by the methodological approach. The role of design principles in framing, and communicating design findings, is also discussed. The third section outlines the data generation methods adopted across each study and details the analytical strategies used to derive findings. The fourth section discusses issues relating to positionality, my roles in the research process, and research ethics.

3.2 Design-based research

This thesis employs design-based research (or DBR), an emerging paradigm of research inquiry (The Design-Based Research Collective, 2003). DBR typically involves an empirical investigation of a theory-driven learning environment to generate contextualised theories of learning (Bakker, 2018). DBR thus aims to bridge theory and practice by focusing on both the design of learning activities, environments, materials and approaches, as well as theory on how learning emerges from these. In this sense, DBR is an important methodology for understanding how, when, and why educational innovations work in practice. DBR typically denotes a family of similar approaches, including design studies, design experiments, and formative experiments; these terms are often used interchangeably (Collins et al., 2004).

The warrant for DBR emerged from the notion that educational research had little impact on or implications for practice (Brown, 1992). Collins (1992) argued that this methodological approach was needed to research learning interventions in naturalistic

contexts, as opposed to laboratory settings. Findings from a DBR project centre on design innovations that embody specific theoretical claims about teaching and learning, and help us understand the relationship between educational theory, designed artefacts based on that theory, and the impact of these artefacts on practice. In this sense, DBR researchers acknowledge that the contexts of learning matter. This both recognises of the "messy… multivariate world of learning, teaching and education" (Cohen & Manion, 2018, p. 331) and rejects artificial or 'laboratory' contexts for educational research (Barab, 2014). This is a point where DBR differs from experimental approaches (e.g., randomised controlled trials) in that design studies of this nature emphasise process rather than two points of measurement (pre- and post-test). In this sense, important contextual information is arguably missed if measures are only taken before and after an instance of learning.

Unlike experimental approaches such as randomised controlled trials (RCTs) that seek to determine the extent to which something works (i.e., the efficacy of X) in a controlled laboratory setting, DBR is concerned with understanding and documenting *how* and *why* a designed intervention or innovation works in practice. Reinking and Bradley (2004) argues this by framing a typical experimental question as "Which intervention is better on average, X or Y?" (or which is better?) while a DBR study seeks to answer "Given that intervention X (or pedagogical theory Y) shows promise to bring about a valued pedagogical goal, can it be implemented to accomplish that goal, and, if so, how? (What could be?)" (p. 153). The difference between experimental approaches to research (e.g., RCTs) and DBR is a commitment to the complexity of learning contexts by focusing on the mechanisms of how something works, under what conditions, and how these insights can inform practice in real-world contexts.

DBR equally differs from naturalistic accounts of learning (e.g., ethnographies) by actively manipulating the learning environment and examining what impact this has on the desired pedagogical goal. A similar trend in literacy research is found in the practice of formative experiments (Reinking & Watkins, 2000) that seek to design and evaluate learning environments.

Design is central in efforts to foster learning and advance theories of learning and teaching in complex settings. In this sense, DBR can be characterised as interventionist or adopting an 'engineering' metaphor to designing learning environments (Cobb et al., 2003). As well as being set in naturalistic contexts, it involves the refinement of conjectures—or predictions of how learning will occur over a series of iterations (Barab & Squire, 2004). These theoretical conjectures are typically embodied in a learning artefact, "a sequence of activities that together or in combination intervene in existing knowledge, beliefs, dispositions, or routines in order to prompt new learning that leads to new practices" (Mintrop, 2020, p. 133). They also culminate in the production of humble theories about the nature and process of learning in context which can be refined through iterative or 'informing' cycles (The Design-Based Research Collective, 2003).

Results of DBR inquiries can thus generate 'storied truths', or usable theories of learning which have direct implications for practice. These are characterised by their ability to "guide, inform and improve both practice and research" (Anderson & Shattuck, 2012, p. 16) in educational settings. Barab (2014) maintains that to produce and understand the positive impact of an educational innovation and thus have implications for practice, innovations must be located in learning environments which are "designed and systematically changed by the researcher" (p. 151). Moreover, inquiries must be grounded in previous research. If a cumulative design knowledge is to be advanced, this typically involves the use of multiple cycles of iterative refinement.

DBR ultimately serves as an approach to study learners and learning within naturalistic contexts, and by manipulating this environment in different ways, evidence-based claims through theoretical accounts of teaching and learning in situ can be made (The Design-Based Research Collective, 2003). In short, DBR can be defined by the following characteristics given by Reeves (2006, p. 58):

- [DBR] addresses complex problems in real contexts in collaboration with practitioners

- [DBR] integrates known and hypothetical design principles with technological affordances to render plausible solutions to these complex problems
- [DBR] involves conducting rigorous and reflective inquiry to test and refine innovative learning environments as well as to define new design principles

3.2.1 Limitations of DBR

Deciding upon an appropriate research methodology requires the affordances and limitations of different approaches to be addressed. For instance, concerns have been raised over the perceived 'anything goes' approach of DBR methodology and the need to improve consistency and rigour (Easterday et al., 2018). Euler (2017) defines this loose-structured design or research focus as follows: "The DBR research process begins with the following question: How can an aspired, initially vaguely formulated goal be reached by a yet to be developed design?" (p. 7). Researchers acting in the dual role of 'advocate' and 'critic' raise questions about the reliability of findings (The Design-Based Research Collective, 2003).

One solution argued for addressing the issue of rigour in DBR is given by Cohen et al. (2018). Drawing on criteria outlined by Lincoln & Guba (1986), they argue that an audit trial can improve the dependability or confirmability of findings. An audit trail consists of a detailed record of the processes of data generation, analysis and findings derived at all stages (i.e., not just during empirical work). Another solution often proposed is to maintain collaborative partnerships between key stakeholders in the research inquiry. By involving multiple parties, including designers, researchers, and teachers (or practitioners), in the design, enactment and evaluation of interventions, the risk of researcher bias or subjective influences can be minimised. As collaborators perform member checking—that is, checking how findings are being derived, interpreted, or presented—this can arguably improve consistency and transparency in the research process. Such a relationship can span over many years, requiring firm commitments and a sustained relationship if progress is to be made. Details of the member checking process in this thesis are discussed under reliability of findings (see 3.7).

3.2.2 Employing DBR in the thesis

The aim of this thesis is to design and evaluate activities to achieve the pedagogical goal defined in the previous chapter. More specifically, the goal was to consider how learners' programming and storytelling practices could be supported through multimodal composition activities. Many researchers in computing education advocates argue the need for domain-specific theories of learning which are necessary to both understand how to integrate computing better (Guzdial, 2017) and to provide practitioners with theory-driven designs and applications (Lye & Koh, 2014). Therefore, adopting a DBR approach to the research project would allow these activities to be iteratively tested and refined over time.

The proposed learning design detailed in Chapter 4 aims to achieve this goal by being grounded in existing theory and relevant literature. The design aims to provide opportunities for learners to exploit their existing programming and storytelling skills and apply these in a novel context. In the following section, the procedure of conducting this DBR project is detailed.

3.2.3 Phases of DBR

This section provides a description of the phases of DBR undertaken for this thesis. The structure of empirical work broadly follows the process detailed by Herrington et al. (2007, pp. 4091–4092). They outline the four phases of DBR based on Reeves's (2006) earlier work. These are given in Table 3. As the thesis generally follows the structure proposed, a third column maps each thesis chapter onto the phases and procedures of a typical DBR project. Moreover, the structure of each chapter generally follows the list of elements given in the table.

Table 3: Phases of DBR project (from Herrington et al., 2007)

Phase	Element	Thesis chapters
Phase 1: Analysis of practical problems by researchers and practitioners in collaboration	Statement of problem	Chapter 1 – Introduction
	Consultation with researchers and practitioners	
	Research questions	Chapter 2 – Literature review
	Literature review	
Phase 2: Development of Theoretical framework solutions informed by existing design principles and technological innovations	Theoretical framework	
	Development of draft principles to guide the design of the intervention	Chapter 4 – Design
	Description of proposed intervention	
Phase 3: Iterative cycles of testing and refinement of solutions in practice	Implementation of intervention (First iteration)	Chapter 4 — Study 1
	Participants	
	Data generation	
	Data analysis	
	(Second and further iterations)	Chapters 4-5 – Studies 2-3
	Participants	
	Data generation	
	Data analysis	
Phase 4: Reflection to produce 'design principles' and enhance solution implementation	Design principles	Chapter 8 – Discussion
	Designed artefact(s)	
	Professional development	

3.2.4 Outputs of DBR

Design researchers are interested in defining, understanding and, ultimately, solving problems across multiple disciplines. In education research, this often explores the use of technology-mediated environments, innovative pedagogies, or novel instructional contexts. As well as detailing the process of DBR, Herrington et al. (2007) describe the form that these distinct contributions from DBR can take. These are noted as follows:

- **Scientific outputs**: Design principles: "The knowledge claim of design-based research [...] takes the form of design principles [...] that can inform future development and implementation decisions". Therefore, design principles embody the procedures, results and context so that future research can derive insights relevant to their own specific settings. As such while generalisation of DBR is rather limited, the contribution of design principles to knowledge functions as a form of analytical generalisation to guide future work.

- **Practical outputs**: Designed artefact(s): "[...] the product of design is viewed as a major output. Design artefacts in this field may range from software packages to professional development programmes. Practical outputs (e.g., learning materials, tools, approaches) are stable insofar as that they are valued by stakeholders as having inherent value outside of the research project itself. These outputs are commonly designed by the researcher or research team in collaboration with stakeholders though this is not necessary.
- **Societal outputs**: Professional development of participants: Participants reflect positively on their experiences within the research (p. 4096).

These three outputs serve as goals in the pursuit of success in DBR. The work undertaken to arrive at these outputs is discussed in detail in Chapter 8 (Discussion).

3.3 Data generation

Adopting a suitable data generation strategy required the suitability of methods to be considered. As DBR is characterised "a family of related research approaches with internal variations in aims of characteristics" (Van den Akker, 2006, p. 3), it doesn't prescribe any fixed approach to research—neither in data generation nor analysis. This necessitated the need to carefully consider the focus of the chosen research project and what data would allow relevant research claims to be made. The nature of multiple data generation processes was first considered to determine the appropriateness of methods to the chosen research questions (Robson & McCartan, 2016).

The literature reviewed in the previous chapter, particularly in computing education, characterises learning as "building relationships between old and new knowledge" through the generation of 'artifacts of social relevance' (Kafai, 2005). As such, the decision to analyse the process and products of students' composition work during the learning intervention seemed to align with existing research approaches. Given that the thesis centres on the design of a learning environment which supports learners' programming and storytelling practices, observing and interpreting students' constructed artefacts was necessary. Studies would

typically then involve observing learners during constructive activities with appropriate data generation methods to reflect this goal. Three sources of data were chosen to achieve this goal:

- Participant-generated materials, including participants' planning materials, written reflections, and weekly 'snapshots' of participants' project work.
- Audio recordings of the interactions between teacher and participants. Discussions with participants throughout the intervention acted as 'artefact-based interviews' to focus and guide participants in the design process. A small dictaphone generated audio recordings of these interactions for transcription and analysis.
- Field notes and screen-data recorded during the intervention. Learners' storytelling and programming were then assessed through a coded analysis (discussed below) of screen-capture data to explore the concepts and practices employed in their engagement in activities. This data was triangulated with the resultant projects to determine common themes.

Data generation varied across studies though these methods were common across all studies. In addition, data generated in specific studies are discussed in each chapter respectively (e.g., teacher interviews in Study 3).

3.4 Data analysis

Data analysis consisted of multiple analytical strategies. As some of these are common across each of the three studies (Chapters 5-7), they are discussed here. However, one analysis is specific to Study 3 and is discussed in greater detail in that chapter. The analysis employed in each study focuses on the storytelling and programming processes that were supported through the curriculum design (detailed in Chapter 4) (i.e., through students' responses to regular prompts when viewing, constructing and discussing multimodal texts, and their resulting projects). Often, these products are designed by the researcher and developed by programmers and resource specialists. Though it is not necessary for the researcher to be the developer, in

this case, I served as both the designer in (Studies 1-2) and co-designer (in Study 3) as well as the facilitator (in Studies 1-2).

3.4.1 Task-oriented analyses

A common set of approach to data analysis in DBR involves conjecture mapping (Sandoval, 2014) or the "systematic test of particular conjectures about learning and instruction in specific contexts" (p. 19). Conjecture maps function as heuristic accounts of the proposed curriculum design and how it relates to learning outcomes. As the designed environment is studied in detail, the exogenous (or 'created') elements and endogenous (or 'natural') elements not initially conceived as part of the conjecture are studied and used to refine the model.

In many cases of DBR, visual representations such as conjecture maps serve to highlight the relative strengths and weaknesses of the designed artefact or practice against its implementation. Similar approaches to DBR (e.g., Simon & Tzur, 2004) refer to these as hypothetical learning trajectories (or 'HLTs'). Another analytical approach taken by Dyson (2008) defines a practice as a single analytical unit, consisting of the expectations, values, actions and interactions of participants in a given context. By contrasting official and unofficial events within the classroom, she makes explicit the structural expectations—those governed by teacher or school expectations—peer or social expectations, and, most commonly, hybrid events that focus on a convergence of the two. Each event was categorised for their focus (what was being taught), their thematic content, and symbolic tools being used (e.g., writing, talk, or drawing).

Similarly, Dierdorp et al. (2011) proposes the use of task-oriented analyses in DBR contexts. In the examples given, this strategy sought to assess how a teaching and learning sequence could support students in making statistical inferences. The authors gathered evidence and counter evidence to review conjectures that predicted how learning would occur. They then analysed these using the approach of assessing predicted against actual learning. Actual learning, in this case, is defined through observation and appropriate data generation strategies. Where a conjecture—or prediction based on anticipated learning—is confirmed for

at least two-thirds of all students, a plus sign (+) is recorded. Conversely, where a conjecture is confirmed for less than one-third of participants, a minus sign (-) is recorded. By generating conjectures—or theories about how learning is anticipated to occur—and then comparing these with the data generated during the intervention, apparent weaknesses in either the design or implementation of a curriculum design are highlighted. They argue that the analysis provides a 'shape' to a learning sequence and provides a warrant for follow-up work to improve the tested design.

The task-oriented analyses employed in this thesis differ from the use case detailed by Dierdorp et al. (2011) in two ways: (i) as the intervention is concerned with multiple learning conjectures—those of both storytelling and programming respectively—two conjecture maps are generated, and; (ii) these are combined to visualise whether multiple tasks align with one another. In other words, a combined task-oriented analysis should illuminate whether programming goals (e.g., create an algorithm to move a sprite) and storytelling goals (e.g., establish the narrative environment) mutually support one another or not. In this thesis, examples of participants' successes and failures are both illustrated to demonstrate the value of the curriculum unit in practice.

3.4.2 Analytical procedure

In this thesis, data generated were analysed using the task-oriented analysis strategy detailed above. This enabled the proposed design (see Chapter 4) to be analysed at the task level and to enable general trends in learners' progressions to be observed. The process of preparing for data analysis started by deriving pre-determined categories based on extant literature related to MMC, programming and storytelling. These analytical categories served to identify learners' progression in both learning strands during activities. As the primary purpose of the literature review is to inform the design of a learning intervention, the literatures consulted and discussed in that chapter were used to design a solution to accommodate many of salient aspects of multimodal composition, storytelling and programming as they relate to one another.

The defined conjectures embody the argument proposed in the literature review that composing multimodal texts in a visual programming environment can support the development of learners' programming and storytelling practices. It also suggests that by employing a visual programming tool (Scratch), we can observe the resulting processes (or practices) and products (or artefacts) and make theoretical conjectures about learning in context.

3.4.3 Proposed analytical framework

The proposed analytical framework focuses on types of MMC activities that could meaningfully support learners in areas of programming and storytelling. These included concepts and practices associated with CT approaches to programming (e.g., Brennan & Resnick, 2012) as well as more applied approaches (e.g., Franklin et al., 2013). Though certain conjectures relating to long-term learning goals (e.g., computational perspectives, Brennan & Resnick, 2012) are equally desirable in this context, it was assumed that these would be difficuklt to observe as their development is anticipated over a longer period than fieldwork would permit. Table 4 details illustrated one section of the framework as initially formulated. It outlines how progression in multimodal composition can be achieved through specific activities (Bearne & Reedy, 2017). The full conjecture-based framework is available in Appendix 3. The analytical framework functions as a starting point for articulating the relationship between the multiple learning strands and diverse theories of action. As it instantiates the proposed understanding of how learning is anticipated to occur in the enacted curriculum, its function is refined in future iterations to improve its function and value.

Table 4: Decide on mode and content for specific purpose(s) (conjectures)

MMC task	Storytelling conjectures	Programming conjectures
Decide on mode and content for specific purpose(s) and audience(s)	Chose which mode will best communicate meaning for narrative purpose(s)	Add character(s) or object(s) using sprite(s)
	Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., say blocks)	Create sequence using *say* block(s)
	Maintain readability using multiple *say* block(s)	
	Establish setting(s) using backdrop(s)	Adds backdrop(s)
	Use varied text block(s) (e.g., *say*, *think*) to represent dialogue/thoughts	Use looks block(s) for multiple purposes (e.g., say, think)
	Use *show/hide* block(s) as cohesive device(s)	Add *show/hide* block(s) to sequence(s)
	Illustrates multiple story episodes through setting change	Use *motion* blocks to illustrate movement (e.g., glide/go to X: Y:)
	Illustrates multiple story episodes through text blocks	Use loops to create animation or repeating motion (e.g., looping costume changes)
	Illustrate characterisation (behaviour, appearance) through costume(s)	Uses *change costume* block(s) to change and reset costume(s)
	Add or manipulate costume(s) to illustrate action(s) (e.g., looking around)	Use switch costume block(s)
	Adapt graphic or text elements (e.g., dialogue, character or object) for narrative effect	Edit sprite(s) or backdrop(s)
		Use change effect block(s)
	Define a narrative goal	

In both learning strands, conjectures were developed to embody some of the perspectives on researching multimodal composition noted earlier (see section 2.5.1). Conjectures therefore took into consideration the artefacts to be produced during each intervention, the desired practices central to their construction, and the criteria necessary to evaluate their substance. In Table 4, columns 1-2 detail the areas of MMC derived from Bearne & Reedy (2017) and delineates these to consider how conjectures around storytelling and programming could be met. Column 3 focuses on conjectures that assess participants' storytelling practices in a programming environment. These are adopted from literature relating to multimodal composition and visual approaches to writing in the primary classroom (Bearne, 2017; Burnett & Merchant, 2018). To assess programming, conjectures derived from relevant programming literature (Meerbaum-Salant et al., 2013; Weintrop et al., 2018) are employed, relating to the

programming concepts and skills that comprise an applied perspective on computational thinking.

Once the draft framework was generated, it was tested in Study 1. Students' project work was viewed, including screen recordings of the process of creation, and coded according to the learning goals and sub-goals associated with each activity. This framework was then used to assess the processes and artefacts of multimodal composition as it relates to both programming and storytelling. As new concepts and practices emerged that were deemed pedagogically desirable, these were used to refine the framework for subsequent studies. In this sense, the analyses were neither fully inductive nor deductive but rather iterative. Tracy (2013) argues that iterative analyses take a "pragmatic iterative approach" (p. 183) to data analysis with cycles of data analysis being ground in both the emergent data and informed by existing theories and models. In the framework in Appendix 3, some of these emergent structures are highlighted.

3.5 Reliability of findings

As noted earlier, the reliability of findings in (or outputs of) DBR has caused some concern. The proposed strategy for checking the reliability of findings—that is, the tested conjectures against the dataset—was to invoke member checking and invite external partners to code a sample of the dataset. This was done to test the reliability of the research instruments used. One researcher unaffiliated with the research project independently coded three participants' complete project data. These were taken from multiple studies using the researcher-developed analytical framework detailed above. They were asked to apply the framework across participants' work and submit these to the lead researcher for analysis. In the first pass, both coders reached 86% agreement with the proposed conjecture-based analyses indicating very good agreement. The remaining conjectures were then discussed and consensus was reached on the disagreements.

3.6 Researcher position

In this section, my role in the research process as well as my relationship with participants is outlined. In detailing this relationship, I hope to maintain transparency within the DBR process as argued by Cohen et al. (2018). In the following chapters, I provide a detailed audit trail of the process and outcomes of conducting this DBR project in the hope of providing future researchers guidance and relevant considerations for similar research.

From its outset of this thesis, I worked closely with classroom teachers, school computing coordinators to develop and enact the aims of the DBR project. In Studies 1-2 (Chapters 5-6), I served as facilitator for both interventions. This involved working closely with students and experiencing the research process first-hand; the nature of this role and the procedure undertaken during each study are detailed in their respective chapters. In Study 3 (Chapter 7), I served as participant observer while a participant teacher facilitated the intervention.

This relationship between myself and participant schools, teachers and students emerged organically through a pre-existing relationship with them. This grew out of a series of collaborative projects and activities I organised before pursuing doctoral research. Over the course of each fieldwork period, these activities included organising research projects, managing project data, engaging in pilot work to test ideas, and redesigning the curriculum unit and research design considering emerging findings. Between cycles of design, this also included organising CPD sessions to support primary school teachers in delivering parts of the new computing curriculum. One intention of this commitment was to use these CPD courses as an authentic site to recruit teachers for subsequent studies. Part of this was to encourage teachers to consider how the designed materials and lesson plans could be adapted to fit their teaching needs. Further details of these activities are discussed in Study 3.

3.7 Research ethics

Ethical approval for each study in this thesis was sought and approved by the School of Education's Ethics Committee. The British Educational Research Association (BERA) Ethical

Guidelines for Educational Research and the University of Nottingham's Code of Research Conduct and Research Ethics were consulted during each application process. Once approved by the School of Education Ethics Committee, outlier permission was sought from the Headteacher of each participating school. I was asked to give a short presentation to all participants where a number of points were clarified, including the focus of each study, the nature of data being generated, how it would be handled carefully and its eventual use, and participants' right to withdraw. Further, any lingering questions that participants had could be answered verbally. All potential participants who then decided to take part in the study were asked to consult with their parents/guardians and to sign a consent form. In seeking to inform participants of their role in the study in an appropriate way, consent forms and information sheets were adapted for different stakeholders; for instance, participants in each study (7-11 year olds) were given age-appropriate consent forms and information sheets with simple language and accompanying illustrations (see Appendix 4). Once consent was given, participants were invited to participant in research activities. In the case of Studies 1-2, they attend an after-school computing club (approx. 15:15-16:15). In Study 3, this took place during the regular school day (13:00-15:00).

All data generated, including participant-generated materials, audio recordings and screen data, were anonymised from the outset. Screen-capture data and audio recordings of the intervention were be transcribed and stored securely on OneDrive. Additionally, original copies of participant-generated materials were placed in a securely locked drawer at my University office. All data were analysed at the University and no primary data will be used or taken outside of this secure context. To maintain confidentiality, all data was be anonymised before being transferred to this location. The data were treated with confidentiality and only available to me and my supervisors. Finally, all primary and secondary is stored in this secure location for a period of seven years after any research is published before being destroyed.

To ensure anonymity of all participants, pseudonyms were used to protect the identity of individual children and all consent forms were kept in a locked drawer within my

office. This was made explicit in all documentation given to participants. Furthermore, a clear understanding of the nature of the research project was achieved by ensuring that all participants' queries are answered and that the consent form and information sheet were carefully explained to them.

Following the impact of COVID-19 on the original research plan, an amendment to the original ethics application (ref: 2018/8) was submitted to the School of Education Ethics Committee and approved by the Chair. The amendment concerned the impact on in-person data generation methods and instead proposed moving to online methods. In line with the university's concerns around Zoom's security vulnerabilities, and to ensure compliance with GDPR requirements, interviews were conducted through Microsoft Teams which offers end-to-end encryption. Furthermore, interviews with the lead participant were recorded using a separate dictaphone. From there, its contents were handled in the same way as existing data generated in a face-to-face setting and complied with GDPR regulations in respect of the management and storage of the data (i.e., anonymised before being transferred to OneDrive, treated with confidentiality and only available to me and my supervisors).

3.8 Conclusion

This chapter has argued that design-based research can provide strong theoretical and practical insights. It first discussed the nature and process of DBR and justified its use in this thesis. Research methods were outlined and argued for their ability to answer the key research questions. In addition, reliability of findings, particularly those that relate to researcher-developed instruments, was clarified and the process of achieving this outlined. As noted earlier, methods specific to each study are discussed in their respective studies (Chapters 5-7).

To instantiate the design considerations noted in the literature review (Chapter 2), the following chapter outlines the theory that informs empirical work and how this correspondingly applies to the design of the curriculum unit. By testing these theory-derived products in real-world contexts, the pursuit of practical and theoretical insights can arguably be achieved. The aim of this is to ensure that key stakeholders in the research partnership

(researchers, teachers, and students) can productively benefit from the approach: for the lead researcher, insights into how theory is applied in practice, and how this affects learning, can be derived; for teachers, the experience of conducting research-based teaching can yield tangible benefits in professional development and engaging in curriculum design with others.

CHAPTER 4
Design

4.1 Overview

The literature review (Chapter 2) explored the context of computing in the primary school curriculum and relevant research relating to the integration of computing in existing curricular practices, primarily literacy. Given the paucity of literature relating to integrated storytelling and programming, prior research on integrated programming within STEM contexts (e.g., integrated CS and mathematics) was also consulted. Accordingly, this chapter focuses on theory and literature across relevant fields of interest that relate to the design of the intended curriculum unit.

As noted in Methodology (Chapter 3), design-based research takes an interventionist approach by typically articulating a learning design in three components: (i) the pedagogical goal that frames the desired learning in context; (ii) the design of the learning environment (e.g., activities or approaches, lesson plans, supporting materials) to achieve this goal, and (iii) the hypothetical learning trajectories that offer a prediction of how student learning will occur in the designed environment (Simon, 1995). In the following sections, the pedagogical goal and associated design principles are outlined. Subsequently, this learning design is tested by analysing data generated in the first iteration of the study. These insights are then used to iterate the learning design in subsequent studies. Results and interpretation are also be discussed in terms of these principles.

4.2 Pedagogical goal

The design of the curriculum unit was guided by a simple pedagogical goal: to support students to explore, discuss and create short multimodal stories in a visual programming environment. It aimed to do so by engaging them in a series of multimodal composition (MMC) activities, requiring them to draw on experience and skills in both learning strands in the pursuit of narrative goals. It also sought to exploit previous findings reported on the benefits of

integrating literacy and computing education (Burke & Kafai, 2010) and challenge previous findings on the limitations of this approach (e.g., Adams & Webster, 2012). Enacting this design was also done to observe the unique role that multimodal—or multiple representation—construction might play during programming and storytelling activities.

4.3 Design principles

In DBR studies, the intervention is defined by its essential components. Although these components can be implemented and adapted in many ways, they are the constants that identify an intervention as a distinct pedagogical entity (Reinking & Bradley, 2007). Fidelity in a formative experiment is not an inflexible implementation of instruction but instead the expectation that all of an intervention's essential components are integral to its success. If DBR is concerned with identifying key characteristics of an ideal intervention (the 'what'), as well as understanding how particular learning goals can be achieved (the 'how'), then an intervention design should seek to serve both of these research goals effectively (Bakker & van Eerde, 2015).

This thesis aligns with Howell et al. (2017) who argue that an intervention is an instantiation of design principles relevant to an explicit pedagogical goal. Accordingly, four design principles are proposed to guide the development, implementation, and evaluation of the curriculum unit. Drawing on relevant theories of learning, along with domain-specific research into the teaching and learning of programming and storytelling, the design principles outlined below serve as an justification for each component of the design that aim to achieve the pedagogical goal detailed above. Further, they provide a series of breadcrumbs that detail the decisions taken in developing the curriculum unit using theory from existing literature. Each consideration is reified in one or several aspects of the design and functions as conjectures about how students' programming and storytelling practices could be supported when composing multimodal texts. Each section below describes first the relevant theory and then details how this is instantiated in the design.

4.3.1 Build on learners' prior knowledge and skills (DP1)

4.3.1.1 Theory

It is generally understood that learners' prior knowledge plays a central role in the development of new knowledge (Piaget, 1926; Vygotsky, 1978). Constructivism, a dominant theory of learning, posits learners as active participants in the learning process who construct knowledge based on their prior knowledge and their interaction and engagement with new experiences. Moreover, from the perspective of constructionism (Harel & Papert, 1991), these learning experiences are mediated by the externalised construction of "learning artefacts" (Kafai, 2005) and the discussion of those artefacts between, most commonly, students and teachers.

The first principle emphasises the existing knowledge that learners bring to the classroom. For this, the role of prior knowledge and skills in storytelling, multimodal composition, programming, both in and out of school, cannot be underestimated. For instance, in constructing multimodal texts, learners are asked to draw upon existing knowledge and skills relating to storytelling, narrative conventions and their understanding of the functions of representations (e.g., textual or graphical) in generating a coherent text (e.g., a story).

Researchers have long argued that learners already possess a wealth of ideas about how to create, critique and adapt representations for their purposes and goals (Bearne & Wolstencroft, 2007; diSessa, 2004). Indeed, diSessa and Cobb (2004) argue that learners should be encouraged to draw upon their own existing representational knowledge and engage in self-directed representational construction as opposed to reproducing given or 'sanctioned' representations. However, in looking at dominant perspectives on multimodality in the literature, a different perspective is taken. Socio-semiotics (e.g., Jewitt, 2008), for instance, emphasise a highly prescriptive 'induction' into representational conventions and traditions. As such, there are criticisms that this arguably diminishes the role of students' constructing their own representations. For instance, Prain and Tytler argue that socio-semiotic accounts of multimodal construction generally place little value in open-ended student-generated

representational construction. Instead, they argue that "highly directive teaching practices" are favoured for providing students with an induction into "semiotic conventions and traditions" (2012, p. 2769) when viewing and constructing representations. If we accept the view that learners possess a rich understanding of representational conventions for constructing and representing their current knowledge, then encouraging learners to draw on this 'bank' may prove productive during design-based activities.

As argued in the literature review, this approach, though grounded in science learning, could meaningfully be applied to literacy as discipline-sanctioned conventions are not common. However, in multimodal storytelling, rather than prescribe certain representations or conventions, demonstrating the considerations that need to be taken into account might encourage students to reflect more deeply on their multimodal constructions (Hull & Nelson, 2005). In developing screen-based multimodal—as opposed to monomodal—texts, Bearne argues that learners need to take increasing responsibility in deciding on mode and content for specific purposes and audiences (Bearne & Reedy, 2017). Their work on supporting young boys' literacy learning through the construction of multimodal texts resulted in the development of descriptors for planning and assessing multimodal text construction in line with descriptors for progress in writing. These descriptors reflect the arguments of others (e.g., Hull & Nelson, 2005) by acknowledging learners' existing literacy experiences and consciously privileging these in the classroom. These might include learners' implicit understandings of multimodal affordances, the information conveyed by words, pictures, animation or a combination of them, the organisation and appropriateness of representations (in terms of economy, redundancy), the basic principles of narrative, such as the main character, the quest to overcome obstacles or complete missions, and, in the case of interactive texts, user-driven features designed to encourage interaction with the text.

Drawing on these 'funds of knowledge' within formal educational contexts arguably makes an explicit commitment to the informal cultural experiences and expertise that learners can employ in the classroom. An attempt is made to bridge learners' home and school—formal

and informal—learning experiences within a 'third space' (Moje et al., 2004) and build upon existing knowledge and experience.

4.3.1.2 Instantiation

The first design principle is therefore reflected in activities which encourage learners to draw on their existing knowledge of visual programming tools and computational thinking, their understanding of narrative and multimodal conventions, and to build on existing ideas and construct personally meaningful learning artefacts. For instance, the proposed activities ask learners to reflect on their favourite multimodal texts (e.g., comic strips or picture books) and prompt them to reflect on different aspects of the narrative of these texts. They are also encouraged to bring examples into the classroom when constructing their own texts. As no 'sanctioned' representations are explicitly mandated, student-generated representations are analysed against their narrative goals to determine how successful their multimodal texts are at communicating intended meaning.

Despite the seemingly common-sense nature of this design principle, potential limitations or design 'tradeoffs' are noted. For instance, providing opportunities for open-ended representation construction assumes that learners can decide on appropriate representations when not prescribed any and are able to seek solutions when encountering problems. Moreover, generating 'new' representations (e.g., drawing a character sprite) in a visual design tool would be unwieldly within a limited timeframe (i.e., classroom lessons).As such, a 'bank' of existing symbolic resources (e.g., characters, costumes, backdrops, and objects) in the Scratch programming tool are employed for practical reasons.

Though diSessa and Cobb (2004) argue that learners should be provided with ample opportunities to draw on their own rich set of representational resources, providing existing template resources and providing learners with opportunities to manipulate and repurpose existing representational resources was deemed more suitable for the context. This fits with what Tytler et al. (2013) suggest as a key element to a representation construction approach, namely the "evaluation, negotiation, and refinement" of representations in class and individual

discussion as well as "constructing their own representations" and "explicit discussion of the role of representation in learning and knowing" (p. 51). These activities are central to multimodal composition as they encourage an understanding of the role and 'appropriateness' of representations in narrative texts. Subsequent iterations of the intervention will determine whether explicit instruction on multimodal construction is needed.

4.3.2 Engage learners in constructive activities (DP2)

4.3.2.1 Theory

The design was also guided by the principles and practices of constructionism (Harel & Papert, 1991). In short, constructionism proposed that learning should involve constructive activities that afford learners the ability to 'concretise' knowledge by externalising ideas as "objects-to-think-with" and reflecting on these (Papert, 1980, p. 11). To form and transform ideas, learners need to engage in a conversation with their own representations (Ackermann, 2004). This view of learning through constructing 'evocative objects' (Turkle, 2007) stresses 'making thinking visible' by encouraging learner-generated representations of knowledge (diSessa & Sherin, 2000). Collins et al. (1991) argues that in generating constructions, learners can represent their skill level or conceptual understanding. In turn, this can facilitate debugging processes or enable problems students may be facing to be diagnosed, allowing for additional scaffolding or support measures to be justified. Experiential, or reflective, artefacts are a preferred means of achieving this goal (Papert, 1980).

Constructive activities also involve a certain degree of iteration, or the act of 'mashing' or 'remixing' existing resources. This practice is common within studies of introductory programming to lower the entry barrier for novice programmers (Resnick et al., 2009). Enacting this process involves the use of a pre-existing 'bank' of representational resources while simultaneously encouraging learners to reflect on the 'appropriateness' of representations for their narrative goals; indeed, Millard (2006) argues this is a privileged activity within contemporary literacy approaches, such as multimodal composition as they

encourage hybridised forms of literacies where learners connected and combine ideas and experiences across multiple domains of learning.

4.3.2.2 Instantiation

This principle guides the framing of open-ended activities that encourage students to constructively engage and apply their own ideas in the construction of multimodal stories. Drawing on ideas developed by Eve Bearne (as Bearne & Reedy, 2017) as part of the Essex Writing Project, this encourages students to apply out-of-school literacy resources (e.g., comics, games, animations, movies or books) to the pursuit of in-school literacy projects. This principle is also reflected in how students are encouraged to draw on their personal preferences and apply these in the pursuit of narrative goals. Moreover, this aligns with one of the tenets of constructionist design activities (Harel & Papert, 1991) where the need for expressive tools for students to construct personally-meaning artefacts is central. Constructing multimodal stories should encourage students to reflect on how their ideas are encoded and interpreted by a visual programming agent while providing opportunities to diagnose or self-diagnose issues with projects.

Further, employing the visual programming tool Scratch offers a means for students to construct and manipulate multimodal resources (e.g., images, pictures, sounds) to create on-screen multimedia projects. As the programming interface can be employed to represent student thinking, making visible their current understanding of programming or storytelling concepts, it should allow teachers or facilitators the opportunity to diagnose or students to self-diagnose issues relating to their project work. This principle is intended to encourage sharing and discussing of ideas by taking Papert's 'objects-to-think-with' metaphor (1980) as a useful framework to structure interactions. These interactions can be understood as 'design discussions' between students and teachers, encouraging both to reflect on ideas and concepts found in projects and to use these reflections to guide discussions and feedback.

4.3.3 Scaffold student-generated construction (DP3)

4.3.3.1 Theory

In constructionist learning environments, a challenge arises in balancing discovery-based activities where resources are made available when they are needed, and direct instruction where resources are provided in advance by anticipating and navigating learners' needs. Though constructionists often argue for 'bottom-up' approaches to artefact and knowledge construction, others suggest that minimal guidance or 'pure' constructivist (or constructionist) approaches are ineffective at promoting certain learning outcomes (Kirschner et al., 2006). As Cobb (1994) notes, pedagogies deriving from constructivist theories of learning commonly emphasise a diminished role in the instructor or teacher. However, this is a common misconception of constructivism—a theory of learning as opposed to a theory of instruction or teaching. Instead, Papert argues that a constructionist-based approach can be characterised as "supporting children as they build their own intellectual structures with materials drawn from the surrounding culture" (1980, p. 31) as opposed to an instructionist approach.

In thinking of means in which to support children through the construction of intellectual structures, seminal work by Wood et al. (1976) offers some valuable insights on the practices of scaffolding. In their work on the instructional strategies needed to assist young children solving problems, they recognised that learning could be support through interactions with more knowledgeable others (or 'experts'). They argue that learning is often "a kind of 'scaffolding process' that enables a child or novice to solve a problem, carry out a task or achieve a goal which would be beyond his unassisted efforts" (p. 90). They propose six scaffolding strategies to aid learners' skill acquisition and problem-solving, including: (i) "recruitment", or maintaining interest in the learning task, (ii) "reduction of degrees of freedom", or simplifying the task to make it more manageable, (iii) "direct maintenance", or keeping learners focused on the objective(s) of the task, (iv) "marking critical features", or highlighting the relevant features to help learners accomplish a given task, (v) "frustration control", minimising risk without resulting in teacher dependency, and (vi) "demonstration",

or modelling the procedure required to complete a task (p. 98). These elements point to level of instructional support provided in a learning environment.

In designing for learning in a programming environment, these measures are necessary as the challenges students face when programming are well documented in the literature. For instance, Qian and Lehman (2017) found that conditionals, loops, and programming execution presented significant challenges for novice programmers and stressed the need for instructional strategies to help address these misconceptions. Whereas exploratory tasks require less scaffolding, 'directed' tasks adopt several scaffolding functions. Though Wood et al. (1976) proposed six elements in scaffolding the learning process, two of those most relevant to this project are employed in the current design model and detailed below.

Research literature relevant to this thesis offers some insights into the scaffolding measures needed to support students. A common thread of research into intervention approaches to learning has centred on the value of direct instruction (DI) for improving student understanding in programming tasks. Rather than allowing learners to experiment on their own, DI proponents (e.g., Kirschner et al., 2006) argue that unsupported approaches are unhelpful and, in some cases, can inhibit learning. However, some research suggests that where discovery-based learning approaches are employed, they result in increased capacity to retain learning beyond the context in which it was initially learned (i.e., increased transfer, H. S. Lee & Anderson, 2013). Papert (1980) refers to these learning environments as 'microworlds' and noted that in order to exercise powerful ideas or skills, it was necessary for a "constructed reality" where "structures match that of a given cognitive mechanism so as to provide an environment where… particular powerful ideas or intellectual skills" can be exercised (p. 204). Bers (2020) echoes to this continuum as a "playground vs. playpen" metaphor (p. 33) where a playground promotes experimentation, autonomy for exploration, creative opportunities and taking risks; conversely, a playpen lacks these features or promotes them to a lesser extent.

Within the computer science education literature, Brennan (2014, 2015) notes a tension between learning that is encouraged through exploration and learning that is structured or scaffolded; Noss and Hoyles (1996) refer to this as the 'play paradox'. On the one hand, Grover et al. (2018) found that while open-ended programming projects created by middle school children promoted the use of basic programming concepts (e.g., loops, sequences, and conditional statements), more advanced constructs were not used by learners; this evidence is supported by many studies in similar contexts (Field et al., 2014; Mayer, 2004). Lye and Koh (2014), borrowing from Wood et al. (1976), argue that teachers or facilitators should demonstrate more complex computing concepts and practices and articulate out loud the thinking process of an 'expert' programmer. Addressing this issue, Benton et al. (2016), in their work on the ScratchMaths project, propose a continuum of activities from the 'explored' to the 'directed', beginning with open-ended activities which introduce specific tools to encourage exploration of the medium before planning structured activities to ensure learners encounter 'powerful ideas' embedded within the design and relating these to their own ideas.

A secondary challenge in scaffolding novice students when working in a visual programming environment is the lack of prior experience in programming (Guzdial, 2017; Kafai & Burke, 2014; Kelleher & Pausch, 2005) and the need to provide structured experiences to guide students in unfamiliar contexts. In the computer science literature, the *use-modify-create* cycle (I. Lee et al., 2011) offers a means of structuring these activities by providing learners opportunities for ownership and agency within their own learning. In the first activity ('use'), students are led through worked examples and asked to reflect on the programme construction with discussion-based questions (e.g., *what block of code caused that change?)*, design choices made during construction, and potential errors that need addressing. The second type of activity ('modify') provides students with opportunities to modify existing work (e.g., sample projects) or require debugging of common problems through 'remixing'. Finally, the third activity ('create') provides students opportunities to create original work based on their experiences of project work during the previous two activity types.

Lee (2011) argue that the use-modify-create cycle encourages ownership and agency in learning by providing learners with opportunities that are flexible enough to encourage multiple learning pathways and support different learner preferences while still structured enough so that learners can be supported in achieving their intended goals. Though the use of sample projects could arguably be seen as too structured or inflexible, the use of a third 'create' phase aims to encouraging autonomous learning by proposing that learners should have responsibility and agency in their own learning. Further, creating projects from scratch should enable design discussions to explore student thinking through cycles of diagnosis (with a teacher or facilitator) or self-diagnosis (individually). It is hoped that encouraging student to create and evaluate their own project work and regularly review this on screen, instead of exploiting the facilitator to resolve issues, should encourage student self-reflection and, in turn, fewer issues in creating multimodal stories.

4.3.3.2 Instantiation

The third principle relates to the levels of support or 'scaffolding' learners are provided during the intervention. This principle is operationalised using both designed and adopted scaffolding measures. Designed scaffolding measures focus on the challenges of composing multimodally in a programming environment. For instance, the process presents learners with additional task demands to be managed, such as the representations to be employed, programming using 'code' blocks, the demands of the target audience (or 'reader'), their intended narrative goals, and navigating the user interface. The result of this complex activity is that some learners might overlook specific aspects of their designs. Wood et al. (1976) point out how scaffolding through various strategies can mark 'critical features' that are vital to the success of a task.

Drawing on the narrative features noted by Bearne (as Bearne & Reedy, 2017), classroom prompts (see Figure 3) were designed to focus students' attention on relevant storytelling and programming concepts. For instance, the function of relevant coding blocks in producing narrative features is emphasised in one classroom prompt and used throughout constructive activities where students are encouraged to adopt these freely.

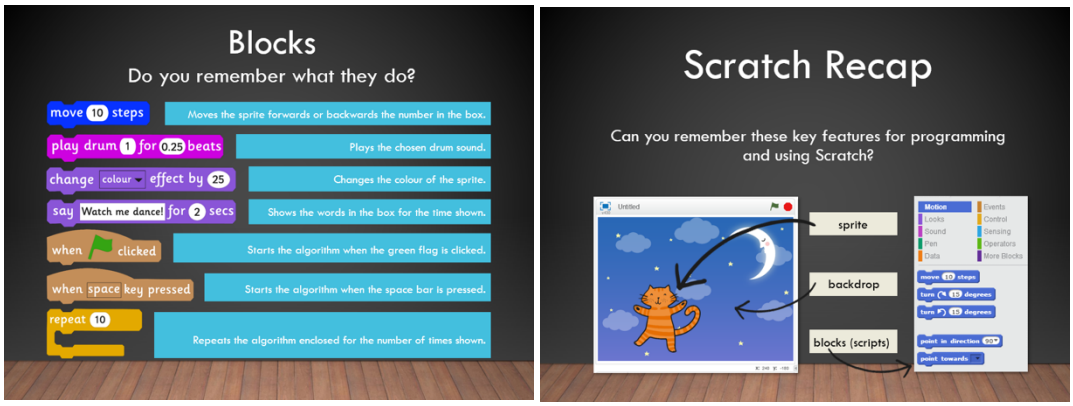


Figure 3: Classroom prompts

In addition to instructional prompts, a storyboard was designed to students to consider the critical features of their story projects (see Figure 4). The storyboard functions as a visual storyboard to guide students to reflect on how their story will be structured episodically and how these will proceed in sequence. Similar approaches to employing storyboards or concept maps (e.g., Liu et al., 2011) note the importance of providing such a scaffold to enhancing the "sequential quality" of the story being told. As multimodal stories in this context are primarily interpreted through the 'stage' (see Figure 5, right image), the use of a visual storyboard is more aligned with the planning demands of the task as students must represent their narrative goals in a similar form to the final product. Equally, the storyboard is designed to encourage students to reflect on the necessary blocks required to orchestrate their stories. These design prompts are embedded into the instructional component of each activity and modelled for students using the overhead projector; it is hoped that by suggesting design considerations for students to reflect on, they can be guided toward the necessary the features of their texts. It is also hoped that this contributes to a learner's 'inner speech' (Reiser & Tabak, 2014) and encourages them to engage in more challenging aspects of the work (e.g., considering alternative representations in their multimodal texts).

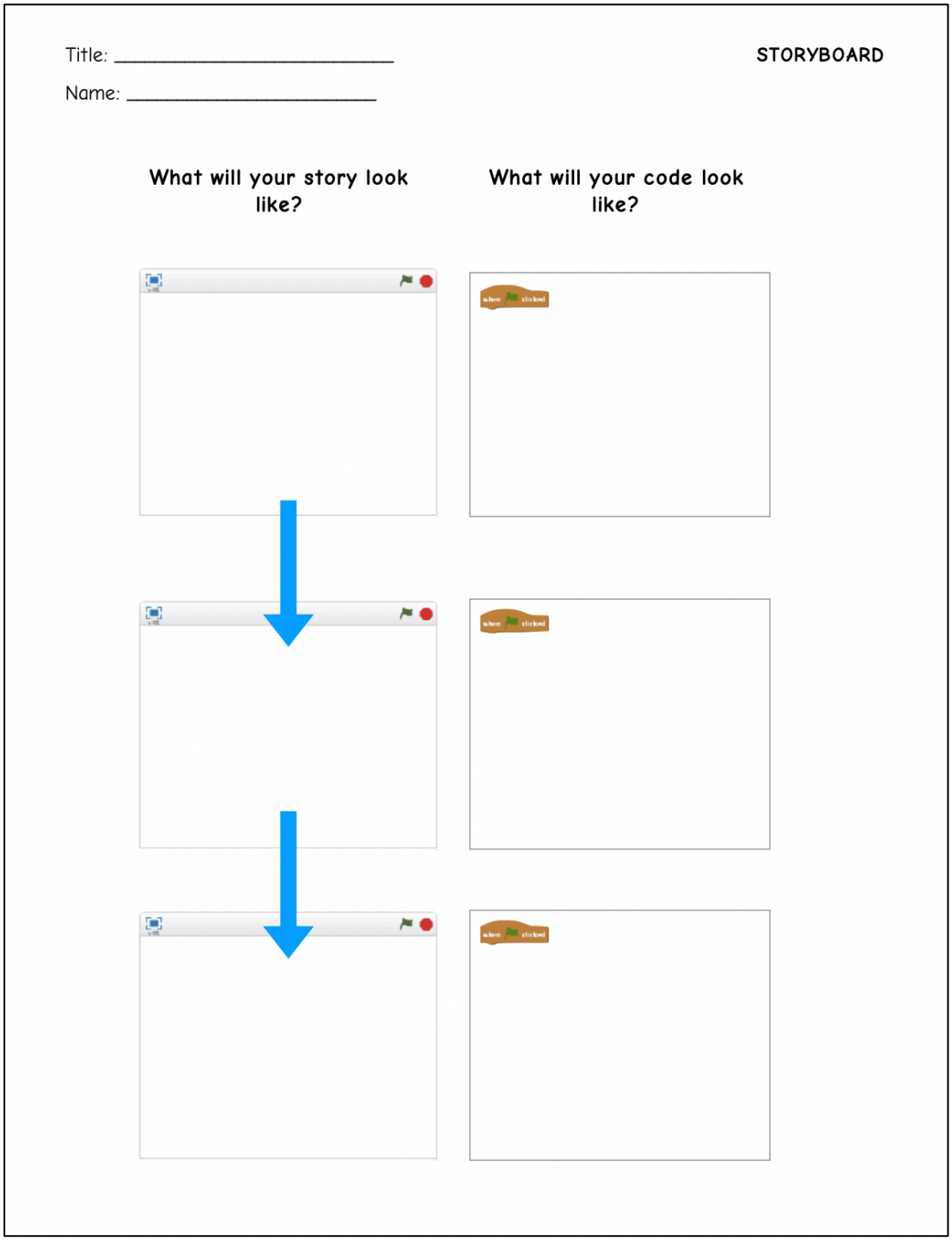


Figure 4: Proposed storyboard

Adopted scaffolding measures refer to the *off-the-shelf* resources that serve to scaffold the learning process for students in programming. In this case, employing Scratch (Maloney et al., 2008), a free visual programming tool, offers inherent scaffolding measures to support novice programmers when creating code (or block-based) projects. For instance, the interface is colourful and intuitive, allowing learners to snap together drag-and-drop blocks of code—similar to Lego—to create a broad range of multimedia projects (see Figure 5).

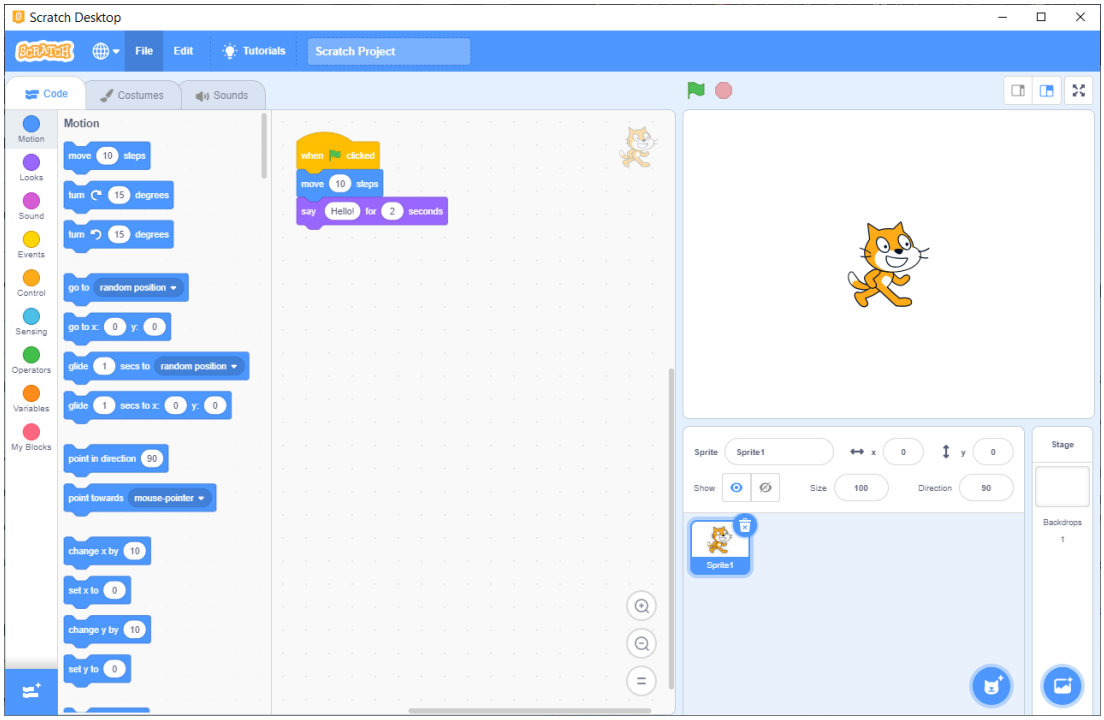


Figure 5: The Scratch programme interface

As code is employed through pre-written blocks, the interface support students' programming practices by eliminating the problems of syntax errors common to other text-based programming languages. Further, the 'bank' of code blocks enables learners to explore the nature and function of code blocks and receive live feedback when code is executed via a 'stage' window. These features aim to highlight the relevant features of create projects in a block-based programming language while offering a 'low floor' for novice programmers.

4.3.4 Provide learners with authentic and purposeful activities (DP4)

4.3.4.1 Theory

Previous research into the teaching and learning of computational thinking has emphasised the synergistic use of computational ideas in other curricular areas. For instance, the application of computing across STEM subject areas has been well explored both recently (e.g., Sengupta et al., 2013) and historically (e.g., Harel & Papert, 1991). These desired 'deep links' (Department for Education, 2013) between computational activities and science, technology, engineering, and mathematics (STEM) are widely accepted with much research to support this claim. One such effort previously discussed saw Kafai (1995) integrate fraction learning into computational game design. The study explored the use of a game design authoring tool to teach basic computational concepts such as loops and conditionals by creating multiple-choice problems for younger students. Similar examples of computational activities as vehicles for other topics such as science are widespread (Sengupta et al., 2012; Weintrop et al., 2016)

Recently, researchers have considered the need for programming instruction to be contextualised in authentic practices (Forte & Guzdial, 2004; Tissenbaum et al., 2019). While this denotes a focus on applications of programming to real-world examples, it does not exclude students' understanding of how and when to apply certain CS concepts when programming. While this trend of research is relatively common, prior consideration of research integrating computational thinking with existing non-CS or non-STEM instruction is less common. For instance, research exploring the integration of storytelling and programming is notably less common than research in STEM contexts such as science or mathematics (Weintrop et al., 2016) arguably owing to the integrated and symbiotic nature of STEM learning in programming contexts. Further, the wider applications of programming across subject boundaries (e.g., supporting storytelling practices) are not entirely novel within the frame of the primary curriculum. For instance, the revised 2014 National Curriculum for England argues that language and literacy are 'integral aspects of the teaching of every subject' (Department for Education, 2013, p. 10) and suggest that profound links between computing

and other areas of the curriculum should be explored. Prior to the introduction of the Computing curriculum, the previous curriculum has placed significantly more emphasis on the links between ICT and the wider curriculum. As programming commonly focuses on sequencing, events, and conditionals (Brennan & Resnick, 2012) it potentially offers an authentic context in which to embed storytelling practices to consider how these are affected by the demands of the context.

4.3.4.2 Instantiation

This design principle is instantiated in the use of programming (e.g., event handling) and storytelling (e.g., characterisation) concepts and activities that relate to multimodal composition. These activities are principally framed as multimodal composition activities, providing a frame to explore links between the visual elements of block-based visual programming—such as sprites (characters), backdrops (settings), speech blocks (dialogue), among others—and the process of digital storytelling. These are discussed in greater detail below.

4.3.4.2.1 Initialisation and sequencing

One benefit of constructing multimodal stories in a visual programming language is their fixed nature. As stories conventionally have a set of characters, a fixed outcome, and a sequence of set decisions and events, it presents the possibility to explore how to initialise sprites and programmes by setting certain characteristics (e.g., character position, state, or expression, setting) and ensuring the programme is executed in line with students' narrative goals. This approach should motivate students to engage in processes of debugging and iterating on their projects as they arrive at closer approximations to their narrative goals. This principle is operationalised throughout as the activities in the proposed curriculum concern the design and testing of programmes that tell stories when executed; crucially, seeing the same output is important as the story may suffer from sequencing or coherency issues otherwise.

4.3.4.2.2 Concurrency

Composing multimodal stories requires the employment of multiple representations to communicate meaning, often simultaneously (e.g., text and images concurrently) (Bearne & Reedy, 2017). Consequently, it offers a strong opportunity to exploit aspects of concurrency within programmes—or synchronising multiple processes (Meerbaum-Salant et al., 2013)—to create more complex projects. Though the challenges for novice students in understanding the complexities of concurrency both within and across sprites/objects are well documented in high school and university-level interventions (e.g., Ben-David Kolikant, 2004), the value of concurrency as a tool to support multiple and synchronous forms of representations for storytelling in a primary/elementary school context is less known. In the first instance, these ideas are embedded in structured activities to explore how broadcast scripts can be used to coordinate students' story elements. Further, it presents an opportunity to explore how programme execution (Meerbaum-Salant et al., 2013) can be managed and how event handling can be coordinated through sending and receiving messages.

4.4 Proposed design

The proposed design of the curriculum unit required appropriate activities and tasks to be reviewed. Given the novelty of the context, few existing materials serve this exact purpose. As such, existing materials were reviewed and evaluated for their potential in this context. This entailed consulting research that adopted story-based approaches (e.g., Kelleher & Pausch, 2007) and broader studies in subject integration (e.g., Benton et al., 2016) to determine appropriate curricular material, approaches, and teaching strategies.

The intervention was designed to support students' programming and storytelling practices through appropriate MMC activities. Research on similar interventions integrating discreet learning strands existed though, in this case, no known intervention could be consulted. The fourth design principle (**DP4**), *engaging learners in authentic and purposeful activities,* indicated how, in the case of an integrated intervention, new concepts and practices should be contextualised (or situated) within specific and meaningful contexts. In this case, to

consider how multimodal composition could support the storytelling and programming process, it was first necessary to consult appropriate literature on the process of creating multimodal texts (e.g., Bearne & Wolstencroft, 2007). From there, practices relevant to the context could be targeted. The practices of composing a story (or storytelling) were first considered, and how this might be augmented through a multimodal context (i.e., visual programming). For instance, as well as deciding on a narrative goal, learners needed to decide on which representations to use, how to structure their text, and to consider which technical features to use for effect. The corresponding storytelling and programming practices (Bearne & Reedy, 2017; Meerbaum-Salant et al., 2013; Weintrop et al., 2018) that would be most salient for MMC were then aligned with the proposed activities discussed below.

The intended curriculum unit primarily adapted one module of the Creative Computing Curriculum (Balch et al., 2014), *Stories*, to focus on aspects of multimodal composition that relate to curriculum-relevant programming and storytelling practices. Unlike other curriculum models that seek to integrate computing and literacy through mutually relevant tasks (e.g., Wolz et al., 2011), the module explicitly focuses on how the programming features within Scratch can be leveraged to support storytelling goals. For instance, activities focus on key aspects of storytelling as interpreted through visual programming features: characters, conversations, scenes, and debugging scripts (or revising texts). These tasks were favoured as they aligned well with the *progression in multimodal text composition* descriptors detailed above (see Chapter 2, Table 1). However, the relative paucity of programming concepts and practices—relative to other approaches (e.g., making games, generating simulations)—was deemed limiting to the success of the curriculum unit. For instance, creating characters (sprites) and conversations (text-based), while important for telling a story, arguably provide little opportunities for learning more advanced programming concepts or skills beyond sequencing and execution. In adapting the activities from this module (see Table 5), additional programming concepts (e.g., event handling) were embedded to provide suitable challenge and conceptual depth to learning activities.

The nine proposed activities comprise of eight structured activities and one independent activity intended to support students' emerging programming and storytelling practices by providing opportunities to work on meaningful projects within an applied context. Table 5 details the content of each intended activity: the first eight activities are short problem-based exercises with learning goals related to specific programming features (or 'blocks') or narrative features. These activities focus on introducing the Scratch interface, including the snapping together of coloured blocks (or commands) to construct algorithms, combining these to build programmes, and testing and debugging these when unexpected results occur. Finally, the last activity invites learners to create an original multimodal text, using the concepts explored to create, execute and revise their developing texts. The two sections that follow provide a walkthrough of the curriculum unit activities, including the pedagogical moves made by both facilitator and students.

The use of structured followed by unstructured activities follows the trajectory set by Lee et al. (2011) in designing activities to support learners approaching novel ideas while providing opportunities for self-directed learning and agency. Consistent with a constructionist approach, and broader theories of engaged or deep learning (Blumenfeld et al., 2005; Scardamalia & Bereiter, 2014), this approach should arguably provide students with opportunities to spontaneously apply concepts or practices from previous learning in other occasions by encouraging them to forge connections across learning experiences. Likewise, the third design principle, *Scaffold the process of student-generated construction* (**DP3**), argues that learners should be supported through multiple means during complex tasks (e.g., programming). For this reason, one way to support students who were unfamiliar with the strands of programming and storytelling, particularly in a multimodal environment, was to provide structure to those with limited prior experiences.

Table 5: Proposed activities (adapted from Balch et al., 2013)

Activity	Aim	Description
1	Programme dialogue between characters	Participants are guided to initialise a sprite and use say blocks to illustrate speech
2		Participants are guided to add additional say blocks and maintain readability
3		Participants are guided to initialise a second sprites and simulate a conversation through managed programme execution
4	Debugging a story	Participants are guided to debug an existing narrative by successfully managing programme execution
5		Participants manipulate sprite attributed (e.g., show/hide blocks) to introduce a set of changes to their character's environment (e.g., the introduction of a natural disaster)
6	Coordinating narrative elements	Participants are guided to debug an existing narrative project using wait blocks, defining and manipulating sprite attributes (e.g., visibility)
7		Participants are guided to use change backdrop blocks to illustrate successive narrative events
8	Execute multiple narrative elements	Participants use broadcast scripts to coordinate narrative elements simultaneously (e.g., change backdrop and introduce a new sprite)
9	Create an original multimodal narrative text	Participants apply a variety of programming concepts to create a 'complete' multimodal story—that is, one with all (or most of) the elements of a basic narrative structure along with selected multimodal effects (relevant to their goals)

The proposed activities and practices in which students are expected to achieve during the process are described as well as the desired pedagogical moves that are regarded as necessary to support learners to exploit their multimodal composing practices toward narrative and programming goals. Each of the nine activities planned are trialled in empirical studies and then assessed using the analytical framework detailed in the Methodology chapter to consider learners' success in concurrent learning goals.

4.4.1.1 Structured activities

The curriculum unit is intended to guide learners through each activity, programming basic sprites to read out text, adding characterisation through dialogue and actions, and introducing multiple narrative episodes. After viewing a demonstration of each new programming feature, learners are asked to recreate each programme or narrative effect and encouraged to add their own ideas. Multiple possibilities and solutions for each task and possible modifications to the project are discussed (e.g., replacing characters without affecting their 'code'). Throughout these tasks, students are guided through each step and given time to give feedback and discuss solutions to tasks as well as issues they were facing. Additionally, initial tasks focusing on

navigating the Scratch user interface and initialising and synchronising multiple sprites where necessary.

The intervention begins by discussing the types of on-screen texts student had previously created (e.g., *PowerPoint* presentations, *Textease* documents, and/or *Scratch* games and animations). Students are shown a sample Scratch project intended highlighting the affordances of visual programmes—interactive features, images, video, animation, sound, and colour. The facilitator and students then discuss the difference between 'on-screen' and 'off-screen' texts. Students are then tasked (activities 1-8) with using existing Scratch resources to programme short story-based projects to explore relevant concepts and ideas to achieve certain programming and storytelling goals. Each activity builds cumulatively on concepts and skills that were employed in previous tasks before the intervention culminates in a final activity (activity 9) that asks students to apply all of the concepts and skills employed in previous tasks and exploit these in the production of an independent (i.e., unstructured) composition.

4.4.1.2 Independent project

Students then complete an independent activity (9) targeting the concepts and practices explored in the guided portion of the intervention. This final task was decidedly more complex as students were asked to independently create a complete story in the programming environment with minimal input from the facilitator. As such, the task took place over multiple sessions (Sessions E and F) to give students enough time to complete their respective texts.

The final independent activity draws upon these computational ideas and tasks learners with creating an original multimodal story by employing some of the features and concepts explored during the guided tasks. As noted in the *Design* chapter, the second design principle, *engaging learners in constructive activities (***DP2***),* emphasised opportunities for personally meaningful and constructive learning. It was anticipated that students would engage in many of the programming and storytelling practices to accomplish their individual narrative goals through engaging in the open-ended composition activity. As such, it was hoped that a 'drop off' effect would not occur during this activity.

4.5 Summary

The design detailed above serves to support students designing and editing on multimodal stories and to consequently support their storytelling and programming practices. Key features of the intervention, including the sequence, selection and scaffolding of the activities, have been guided by relevant literature in the field. Moreover, the intervention serves as a proposed theory-informed design to serve primary school teachers integrating computing across the curriculum or embedding multimodal composition within their existing literacy practices.

In the empirical chapters (Chapters 5-7) to follow, this design is tested, iterated, and refined to better serve students and their teachers and yield empirical findings from the three empirical studies on the intervention in practice. These serve to iterate on the intervention design proposed in this chapter and offer both theoretical- and practice-led insights into the value and experience of the intervention for students and teachers.

CHAPTER 5

Study 1

5.1 Introduction

The literature review (Chapter 2) sought to highlight the areas in which multimodal composition have been embedded within educational settings and suggest learning strands where this integration could further be exploited. Though there is much to celebrate in terms of its potential in diverse learning strands, this area of inquiry is in its infancy. Some introductory curricula have proposed multimodal composition activities, primarily in the form of digital storytelling (Burke & Kafai, 2010; Kelleher & Pausch, 2007) as a context for programming. Despite previous research (Burke & Kafai, 2010; Franklin et al., 2013), the argument for greater attention to the role of MMC for programming and storytelling purposes was made.

The literature review also highlighted the areas in which multimodal composition have been embedded within educational settings and suggest learning strands where this integration could further be exploited. Though there is potential in terms of the application in diverse learning strands, this area of inquiry is in its infancy. Moreover, such applications in programming and storytelling operate at a very general level (i.e., storytelling as a context, not a learning goal), and present a design problem—how to meaningfully integrate storytelling and programming activities within a learning environment.

5.1.1 Study rationale

Exploiting multimodal composition as a context for programming and storytelling could solve a number of issues relevant to both learning areas: for programming, current theorists have argued that most computing curricula run the risk of making computing appear irrelevant to students (Tissenbaum et al., 2019) Moreover, it has been argued that, within the current 'crowded' curricular context, the goals and values of computing education could be considered too narrowly defined with scant evidence of the use of computing or programming in applied

contexts available (Larke, 2019). For storytelling, theorists have long stressed the need to develop new theories on composition and a broadened definition of literacy to take into account the advent of new technologies and digital media though this largely exists in a theoretical state (Bearne & Bazalgette, 2010; QCA/UKLA, 2004).

In this chapter, the first iteration of a DBR study focuses on whether and how storytelling and programming practices could mutually support one another though multimodal composition has, to the author's knowledge, not been undertaken. Moreover, the possibilities and challenges associated with such an approach are not widely known or understood. Despite the large body of work investigating the nature and promise of MMC in the classroom, research has often tended to be theoretical with less empirical work conducted in school settings. Moreover, most of the empirical studies that exist have been largely conducted at secondary and university level. It is hoped to contribute to this growing area of research by first investigating multimodal composition as a context for literacy and computing—more specifically, programming and storytelling.

5.1.2 Research questions

As noted in the literature review, the aims of this study are to consider the nature and of the potential for the mutual practices of programming and storytelling to be integrated, within an authentic, meaningful context. In terms of the DBR process, this cycle serves to instantiate the curriculum unit detailed in Chapter 4 (Design) as a solution to the pedagogical problem identified in Chapter 2 (Literature Review) and to consider which design principles might underpin such an approach. These include: (i) what areas of programming and storytelling are mutually beneficial; (ii) what role does multimodal composition play in integrating these discreet set of disciplinary practices, and; (iii) what are the challenges or hindrances in integrating these learning strands and how might these be mitigated through explicit design choices. As such, the main research question was given as follows:

- Research question 1 (**RQ1**): How can multimodal text construction mutually support students' programming and storytelling practices?

To answer this question, the curriculum unit was trialled and analysed to consider the pedagogical value, if any, of integrating primary computing and literacy through multimodal composition activities. Moreover, this process was equally undertaken to consider whether and how these learning goals could be achieved through an integrated context. The necessary outcomes of this study therefore were to identify enablers, constraints, and outcomes in implementing the designed approach. The sub questions are therefore given as follows:

- Sub-question 1 (**SQ1**): What forms of programming and storytelling practices are fostered through multimodal composition?
- Sub-question 2 (**SQ2**): How do the practices of programming and storytelling in the production of multimodal texts relate to one another?

5.2 Method

In this section, I detail the context of this study. I begin by detailing the study location and participants, and then describe the procedure for the intervention. This includes how data were generated, and how this was then analysed.

5.2.1 Setting and participants

The study took place during an after-school computing club in an inner-city primary school in Nottingham. As facilitator, I enacted intervention with a group of 12 Year 5 (9-10 year olds) students comprising of six girls and six boys. The classroom teacher and group were surveyed and were found to have limited formal experience in multimodal text production. Students in this school typically had dedicated computing lessons once per fortnight. Though the 2014 curriculum had been in place for several years, the school, like many, was playing 'catch up' to curriculum expectations (The Royal Society, 2017); by the time the intervention was trialled, most of the Year 5 students (n=10) had limited to no experience of using Scratch or any other visual programming tool.

5.2.2 Intervention procedure

The intervention consisted of six sessions—a total of six one-hour lessons—taking place weekly. The first four sessions (sessions A-D) comprised of the short, structured activities relating to specific programming (e.g., initialising a sprite) and storytelling (e.g., creating characters) goals. These involved participants being scaffolded through facilitator intervention and support materials. The remaining two sessions (sessions E-F) focused on students creating a multimodal text independently (i.e., with minimal guidance). The breakdown of each session and the activities they comprise are detailed in Table 6.

Table 6: Intervention sessions and lesson activities (Study 1)

Session	Task	Aim	Description
A	A1	Programme dialogue between characters	Participants are guided to initialise a sprite and use say blocks to illustrate speech
	A2		Participants are guided to add additional say blocks and maintain readability
	A3		Participants are guided to initialise a second sprites and simulate a conversation through managed programme execution
B	B4	Debug a story	Participants are guided to debug an existing narrative by successfully managing programme execution
	B5	Extend a story	Participants manipulate sprite attributed (e.g., show/hide blocks) to introduce a set of changes to their character's environment (e.g., the introduction of a natural disaster)
C	C6	Coordinate narrative elements correctly	Participants are guided to debug an existing narrative project using wait blocks, defining and manipulating sprite attributes (e.g., visibility)
	C7		Participants are guided to use change backdrop blocks to illustrate successive narrative events
D	D8	Execute multiple narrative elements simultaneously	Participants use broadcast scripts to coordinate narrative elements simultaneously (e.g., change backdrop and introduce a new sprite)
E	EF9	Create an original multimodal narrative text	Participants apply a variety of programming concepts to create a 'complete' multimodal story—that is, one with all (or most of) the elements of a basic narrative structure along with selected multimodal effects (relevant to their goals)
F			



Figure 6: Intervention site (Study 1)

The intervention began in the classroom. Participants were briefed on the procedure for moving to the computer room and the rules in place while there. The conditions of their participation were explained again (e.g., their right to remove themselves from the study if desired) though no participants left the study at any point. In the computer room, I facilitated participants through the structured activities using the teacher's computer and projector (see Figure 6). This meant leading discussions on participants' previous experience in creating on-screen and off-screen texts. Aside from copying handwritten texts into PowerPoint presentations and Word documents, they had limited experiences in multimodal composition. Two participants, Mark and Charlie, recalled previously creating Scratch games and animations in an after-school coding club. This became evident in the analyses of participants' performance and is discussed further below.

Participants were then guided through each activity, programming basic sprites to read out text, add characterisation through dialogue and actions, and introduce multiple narrative events. After viewing a demonstration of each new programming feature, they were asked to recreate each programme or narrative effect and encouraged to add their own

personalisations. Multiple possibilities and solutions for each task and possible modifications to the project were discussed (e.g., replacing characters without affecting their 'code'). Throughout these tasks, students are guided through each step and given time to give feedback and discuss solutions to tasks as well as issues they were facing. As this was the first visual programming experience for most students, initial tasks focusing on navigating the Scratch user interface and initialising and synchronising multiple sprites, were necessary.

In the final two sessions, participants completed the independent composition activity (EF9). As this activity was unstructured, minimal guidance was provided. This encouraged participants to complete the task independently though it was understandably more complex. However, the aim was to encourage participants to apply some of the features and concepts explored during the structured activities in their final project. Though the activity was allocated more time for participants to complete their texts, in most cases, this wasn't enough. Again, this is discussed in greater detail in the Findings section below. The final session ended with a debriefing discussion. Participants were encouraged to share their experiences of the project and discuss what activities their enjoyed most. They were also given the opportunity to continue a computing club though outside of the boundaries of the research project.

5.3 Findings

The findings are divided into two parts. The first section details the task-oriented analyses conducted on participants' performance in the designed activities. The results are discussed in terms of the broad patterns across participants' work with illustrative cases provided to give a representative picture—or 'thick description'—of participants' storytelling and programming practices and how these were fostered through multimodal composition. These illustrative cases were chosen to provide evidence of patterns of performance (i.e., both successes and failures) across participants' work and resulting projects. In this sense, broad trends across the participant group can be observed. In the second part (see section 5.3.3), these analyses are

compared alongside one another to determine the strength of content integration and consider whether programming and storytelling tasks mutually supported one another or not.

5.3.1 Structured activities (A1-C5)

5.3.1.1 Programme dialogue between characters (A1-A3)

In the first activity (see Table 7, activity A1), participants were guided to programme a basic sprite (the Scratch 'cat') to speak. Table 2 illustrates the conjectures for the first three activities (A1-A3). As most participants were unfamiliar with Scratch or its features—despite evidence of previous formal experiences of Scratch—several opted to use *sound* blocks instead of programming text. As noted in the table, only four participants successfully completed this task which required a *hat* block to initialise their sprite (when it is programmed to speak); most (n=10) did manage to include a sprite (who is speaking), and a *say* block (what is being said).

In terms of storytelling (Table 7, columns 4-5), all participants manged to initialise a sprite as their character though few managed to use *say* blocks to develop characterisation or illustrate character's actions or intentions (n=2). In most cases, this was generally due to their inexperience with the software; indeed, many participants found difficulty in navigating the interface, including dragging and dropping programme and visual elements. For many participants unfamiliar with Scratch, their first attempt was to use *sounds* in order to programme the cat to speak; after a demonstration of the *say* function, they changed their code with most achieving dialogue (n=10).

In the second activity (see Table 7, activity A2), participants were asked to add a series of additional *say* blocks to create additional dialogue and characterisation in their scenes (Figure 7). As performance in the first task was unexpected, participants were asked to correct any bugs preventing their programmes from being executed; in almost all cases, participants hadn't connected the hat block to their *say* blocks; in the end, almost all (n=11) corrected the bugs in their code from the previous task though only five managed to add additional text blocks to their sequences by 'snapping' them together. This indicated how participants found difficulty in navigating the Scratch interface. Though the software is designed to scaffold

learners to avoid syntax errors, participants' limited understand meant that blocks were not immediately snapped together.

Table 7: Analyses of activities A1-A3 (Programme dialogue between characters)

Activity	Conjecture (storytelling)	#/12	Conjecture (programming)	#/12
A1	Add character(s) or object(s) using sprite(s)	12	Add sprite(s)	12
	Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., say blocks)	2	Create sequence using *say* block(s)	10
	Use structural device(s) to ensure cohesion (e.g., yellow 'events' blocks)	4	Initialise sprite using *hat* block (e.g., when green flag is pressed)	4
A2	Maintain readability using multiple *say* block(s)	1	Create sequence using *say* block(s)	5
	Maintain chronology using *hat* block(s)	10	Initialise sprite using *hat* block (e.g., when green flag is pressed)	11
A3	Illustrate opening (e.g., situation) using *say* block(s)	6	Create sequence using *say* block(s)	10
	Illustrate characterisation (roles, behaviour, actions) using *say* block(s)	5		9
	Maintain readability using multiple *say* block(s)	6	Execute two or more independent sprites concurrently	11
			Initialise sprite(s) using hat block (e.g., when green flag is pressed)	11
	Establish setting (surroundings, and habitual states) using backdrop(s)	6	Add backdrop(s)	6
	Use *wait* block(s) as cohesive device(s)	6	Use wait block(s) to synchronise events correctly	6

Another challenge for storytelling goals was maintaining the readability of their programme for their intended audience. This required that *say* blocks be kept short—usually a sentence or two—and timed accordingly. Most participants did not achieve this (n=1) finding challenge in the timing of each element for readability when running the programme. For example, Natasha exclusively used *say* blocks without a time specification, so the text was unreadable having not specified how long to display each block of text.

Finally, in the third activity (Table 7, activity A3) participants were asked to create a dialogue between two characters. Most participants (n=11) creates two sequences of code blocks in parallel and started (or initialised) these using hat blocks, such as *when green flag is pressed*, though fewer (n=6) used *wait* blocks to sequence their dialogue correctly; this half of participants (n=6) generated a project similar to Figure 3, combining multiple lines of dialogue

into a coherent conversation Though not explicitly specified as a learning goal, participants provided explicit context for the dialogue (e.g., friends lost in the wood). They also used dialogue to add characterisation to their scenes, offering character's feelings, emotions, and intentions.

Owing to the difficulties encountered by most in the first two activities, the focus of the demonstration for this task was modified to focus on navigating the Scratch interface, including adding and removing blocks, snapping blocks together, and testing and revising projects. Though the process inevitably became more complex as participants sought to ensure each element was sequenced correctly, more (*n*=6) added appropriate times to each block of text, ensuring the readability of their story despite increasing complexity.

5.3.1.2 Debug a story (B4-B5)

In Session B, participants were given a closed debugging activity (see Figure 7) with bugs—two sprites speaking over one another—and were tasked (Table 8, activity B4) to debug each sprite before testing their debugged programme; the correct (and most common) solution used multiple *wait* blocks interspersed between each *say* block to sequence the narrative correctly. Many participants (*n*=7) successful completed this task. As the success of the programme depended on the *cohesiveness* of the story, conjectures were closely aligned with these seven participants satisfying both conjectures (see Table 8, activity B4). The result indicated that the coordination of multiple characters was challenging for many; the remaining participants had difficulty in completing the task, ranging from blocks not being snapped together, or more than one *wait* blocks being required for the task. Participants also had difficulty in correctly sequence by timing each *wait* block to coordinate the text blocks correctly.



Figure 7: Screenshot of debugging task (stage and code) (Study 1)

In the second and final activity of the session (Table 8, B5), participants were asked to adapt the sample project to include additional story episodes (e.g., an opening complicating or concluding event). They first discussed what 'makes' a story, with some indicating the need for backdrops to show "where the story is set", and a simple plot as a "thing going to happen… all to build up to the main reason of the story." As the project was open-ended and participants could tell any story, all participants indicated a preference to use their existing project from Session A instead of using the sample project provided.

Table 8: Analyses of activities B4-B5 (Debug a story)

Activity	Conjecture (storytelling)	#/12	Conjecture (programming)	#/12
B4	Use *wait* block(s) as cohesive device(s)	7	Use *wait* block(s) to synchronise sprite(s)	7
B5	Illustrates first story episode through text blocks	11	Add *say* block(s) to sequence(s)	11
	Illustrates subsequent story episode(s) through text blocks	6		
	Coordinate narrative element(s) to execute correctly	10	Execute two or more independent sprites concurrently	10
	Establish setting (surroundings, and habitual states) using backdrop(s)	11	Add backdrop(s)	11
	Illustrates story episodes through setting change	3	Add *change backdrop* block(s) to sequence(s)	3
	Use structural device(s) to ensure cohesion (e.g., when blocks)	12	Initialise sprite(s) using hat block (e.g., *when green flag is pressed*)	12
	Illustrate characterisation (roles, behaviour, actions) using *say* block(s)	7	Add *say* block(s) to sequence(s)	7
	Use *wait* block(s) as cohesive device(s)	8	Use *wait* block(s) to synchronise sprite(s)	8

In terms of the storytelling conjectures, almost all participants established their characters and setting through dialogue or backdrops (*n*=11) while fewer (*n*=6) included an additional narrative event (e.g., complication). Participants notably began to use different programming features to indicate narrative events, relying less on dialogue alone. Though not conjectured to occur, the two participants with prior programming experience, Mark and Charlie, indicated a character's intentions through motion blocks while three participants used *cha*nge *backdrop/costume* blocks to illustrate this; they did, however, use dialogue alone to illustrate characterisation in their sprites (*n*=7). For instance, one participant, Mimi, used a forest backdrop in her story, commenting that her characters were there "trying to find like a hidden key and a book [and] they have to look everywhere for it".

In terms of programming, almost all participants executed two or more independent sprites concurrently (*n*=10) meaning that two characters were running simultaneously. They mostly used *say* blocks (*n*=7) and *change costume/backdrop* blocks (*n*=3) to tell their stories and mostly (*n=8*) used *wait* blocks correctly to sequence their dialogue between two sprites. The remaining participants used the correct number of *wait* blocks but inputted the wrong number of seconds, creating overlapping and incomprehensible dialogue.

Despite not being a feature of the instruction or sample projects, participants became increasingly keen to move and animate their sprites. Where some participants (*n=2*) succeeded in adding motion blocks to indicate movement in their stories, others were less successful and resorted to removing the code and taking a different approach. For instance, one participant, Alex, commented "I don't know how to get my other person back because I put [move] 1000 steps by accident" indicating that he could programme a sprite to move, but couldn't reset to its original coordinates.

5.3.1.3 Coordinate narrative elements (C6-C7)

In Session C, participants were asked to debug a project from a previous activity (see Figure 7, *The Boy and the Wizard*). In the first activity (Table 9, activity C6), participants were shown how to use *show/hide* blocks to correctly sequence narrative elements depending on the

context (see Figure 8). In total, almost all participants (*n*=11) were successful in adding show blocks to add and remove narrative elements at appropriate times in their stories with one participant, Charlie, failing to add a corresponding hide block to remove the element when no longer required.

Table 9: Analyses of activities C6-C7 (Coordinate narrative elements)

Activity	Conjecture (storytelling)	#/12	Conjecture (programming)	#/12
C6	Use *show/hide* block(s) as cohesive device(s)	11	Add *show/hide* block(s) to sequence(s)	11
	Use *wait* block(s) as cohesive device(s)	8	Use *wait* block(s) to synchronise events correctly	8
C7	Illustrate setting change(s) using *change backdrop* block(s)	5	Add *change backdrop* block(s) to sequence(s)	10

In the next activity (Table 9, activity C7), participants were asked to use *change backdrop* blocks to indicate a setting change. Using their existing projects from Session B, most participants added a backdrop as setting with most programming multiple backdrops to change at the appropriate moment (e.g., a chase sequence, scavenger mission, or time travel) (*n*=10). However, fewer (*n*=5) added an additional 'change backdrop' block at the beginning of their text to define the initial programme state; this meant that their respective programmes would not produce the same output each time. Similarly, though it was anticipated that participants could use these backdrops as organisational features within their texts, segmenting each part of their story into narrative *episodes*. few (*n*=5) chose to use this programme feature to indicate the next event in their story.

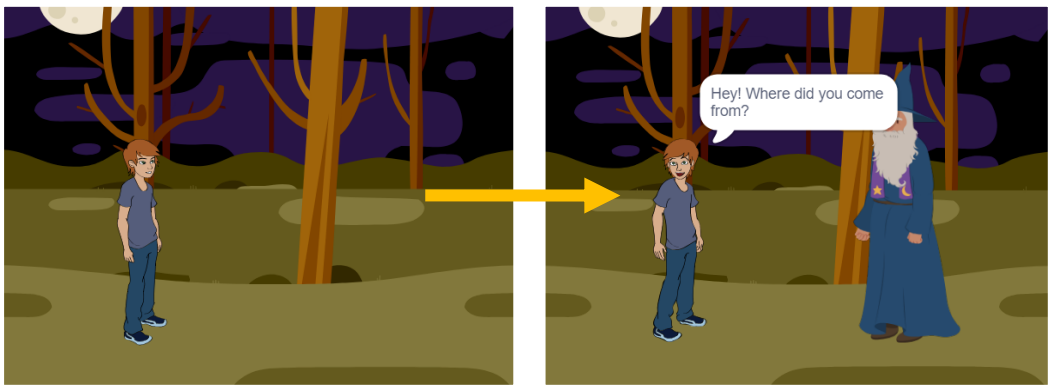


Figure 8: Coordinating narrative elements using show/hide blocks (Study 1)

5.3.1.4 Execute multiple narrative elements simultaneously (D8)

In Session D, participants were shown another sample project and presented with a problem, executing multiple narrative elements simultaneously. Broadcast messages were then demonstrated to enable multiple programme elements to be coordinated simultaneously (e.g., a backdrop changes and a sprite reacts to this).

Table 10: Analyses of activity D8 (Executing narrative elements simultaneously)

Activity	Conjecture (storytelling)	#/12	Conjecture (programming)	#/12
D8	Use *broadcast message* block(s) as cohesive device(s)	8	Use *broadcast message(s)* to coordinate multiple processes	5
			Use *wait* block(s) to synchronise sprite(s)	4

In terms of storytelling (see Table 10, activity D8), it was anticipated that participants would use these programme features to better sequence their stories, both when constructing them and presenting them; it was also hoped that participants would use the more advanced programming construct and then, in turn, this feature would 'spill over' in constructing their own stories. However, participants neither expressed a preference for either. Most (n=8) used *wait* blocks to 'delay' story elements in their story sequence (see Figure 9, right example). Aside from one participant, Charlie, participants had not previously used broadcast messages; a minority (n=5) elected to use these features with most (n=7) confusing a broadcast with a *say* block. Incidentally, others employed *wait* blocks to 'delay' these elements in their story sequences (n=4).

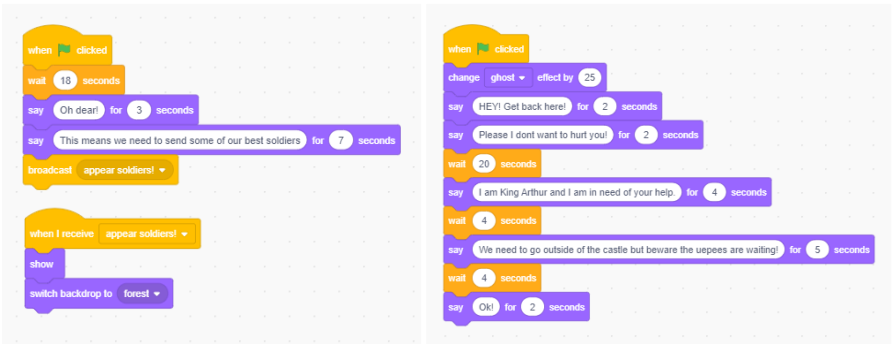


Figure 9: Use of broadcast messages (left) vs wait blocks (right) (Study 1)

5.3.2 Independent project (EF9)

In the final two sessions, participants independently created an original multimodal text (see Table 6, activity EF9). Their intended project goals are detailed in Table 11. These detail the key plot points of their story and offered a way to examine their intended against enacted projects. The first of the two sessions began with a demonstration of a sample text (see Figure 10). This featured a boy who lives on the moon and offered suggestions on how the story could proceed and what a suitable ending might look like. Participants then spent the remaining time in Sessions E and F adding sprites, backdrops and text blocks to their projects and programming each element to occur in the correct sequence.

Table 11: Participants' project goals (Study 1)

Participant	Narrative goal
Aisling	"My story is about a princess who gets cursed and has to find a way to be uncursed. The witch wants to get the castle but the princess said she'll give it to her only if she's nice."
Alex	"Two cars are going too fast and they crash and flip.. Superman comes and makes sure that they are ok."
Aoife	"Mr and Mrs Jackson get kidnapped by a ghost. Then a dino comes to save them."
Charlie	"My story is about where you meet a knight and help him get stronger to defeat a ghoul. You travel to a planet but you find problems along the way."
Jeremy	"My story is about a ghost who is a king."
Joseph	"Two knight teams are at war. Ghouls and wizards fight against one another. One army kills the other."
Marian	"Two people are captured by a witch who wants to hurt them. They find a way to call the police."
Mark	"Nano and Terra (Scratch characters) set off on a magical journey to find a mysterious potion. Nano drinks it and turns into Werenano."
Mimi	"A princess is looking for the key the sorcerer hid and asks Emily to help. She'll get her prince if found but she can't find it. She asks a girl to help her. It ends with a cliffhanger."
Miranda	"Three teens find a shiny rockon the side of the river and take it home. They find it is a cursed diamond gem. Their parents disappear and they have to get them back from an evil witch."
Natasha	"Two gobos are in love but they leave their best friend out and they forget about him. Then they say sorry."
Tadgh	"In this story, ghouls run low on diamonds and gems and they go to a powerful dragon to get resources. The dragon crashes into a wall and the ghouls get a lot of diamonds/gems."

In terms of programming (see Table 12), participants each programmed their multimodal texts using multiple sprites, backdrops, and blocks of code. Most programmes tended to use a combination of text- and image-based resources (e.g., text blocks, sprites, backdrops) to illustrate different elements of their stories (n=11); backdrops (n=11) were employed to

provide background information or illustrate narrative events, costume changes (*n*=5) were employed to illustrate character states or response to events, and text (*say* or *think)* blocks were added to illustrate character dialogue and internal thoughts.

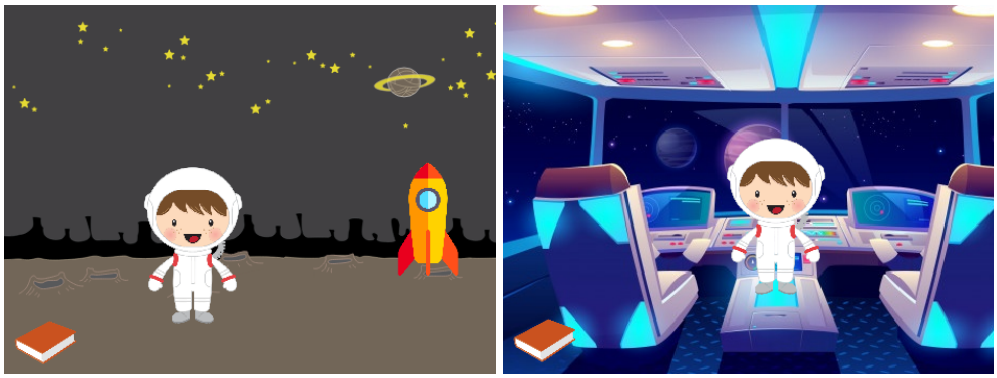


Figure 10: Sample project (Study 1)

Some participants noted the difficulty in correctly resetting their projects though this was dependent on the programming features they used. For instance, Tadgh used a glide block to illustrate a dragon swooping over his victims (see Figure 11) produced a terrifying effect for his main characters. Unfortunately, this feeling was short-lived as he struggled to repeat the effect on subsequent programme executions; as the dragon sprite had not been programmed to reset to its starting position, it didn't move on any subsequent attempt unless manually moved.

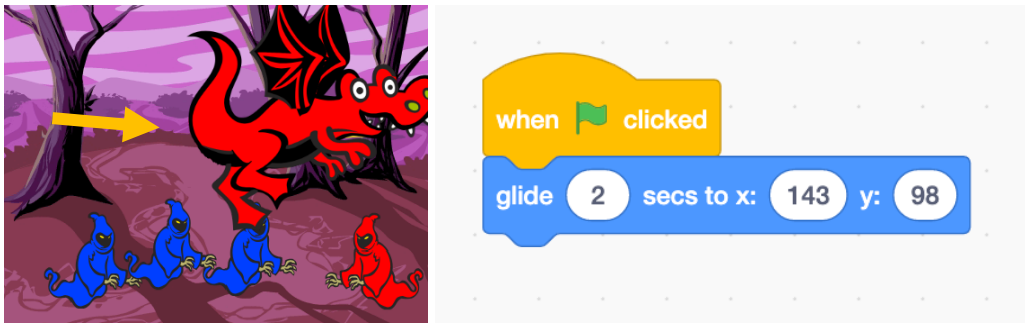


Figure 11: Tadgh's use of glide block (Study 1)

Elsewhere, where most participants modified sprite size, position and graphic effects for narrative purposes (*n*=9), they ensured sprite and programme states were similarly reset; this resulted in their programmes outputting the same story project each time. This was also evident in their correct sequencing of *show/hide* (*n*=8) and *wait* blocks (*n*=10) in their scripts. In the

final session, participants were asked to test their projects to check for programme errors; it was during these run-throughs that most (n=10) corrected bugs within their projects, preventing their programmes from running according to their story plans.

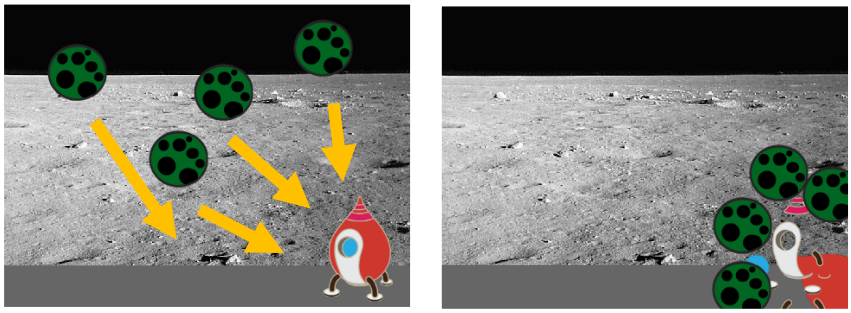


Figure 12: Charlie's use of looping motion to simulate a meteor shower (Study 1)

As this was an introductory module for programming for most participants (n=10), this was unsurprising. While this meant most stories proceeded chronologically and cohesively, it also meant that most projects were simple sequential programmes without conditional logic, loops, or broadcasting messages. However, the analysis provided some evidence of more complex programming concepts and practices in participants' project work. For instance, some (n=3) opted to use motion-based (e.g., *move, glide* or *go to X/Y)* blocks to illustrate actions in their stories. For this, it required them to define initial sprite and programme states so that repeat executions of the programme yielded identical results. For instance, one participant, Charlie, who had previously attended a code club in the school, opted to use a simple looping motion and broadcast a message to programme a meteor shower (see Figure 12, orange arrows indicate motion). These programme features drove his narrative forward by presenting a complication in his story; the meteors destroy the rocket (indicated with a costume change), forcing the main character to seek an alternative route home. Though multimodal text effects like this can promote more sophisticated programme structures, these were not widely common within the dataset.

Table 12: Analyses of activity EF9 (Independent project)

Conjecture (storytelling)	**#/12**	**Conjecture (programming)**	**#/12**
Define a narrative goal	12		
Add character(s) or object(s) using sprite(s)	12	Add sprite(s)	12
Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., say blocks)	12	Create sequence using *say* block(s)	12
		Uses *change costume* block(s) to change and reset costume(s)	5
Establish setting (surroundings, and habitual states) using backdrop(s)	12	Add backdrop(s)	12
		Edit or manipulate backdrop(s)	2
Use *show/hide* block(s) as cohesive device(s)	8	Add *show/hide* block(s) to sequence(s)	8
Use varied text block(s) (e.g., *say*, *think*) to represent dialogue/thoughts	3	Use looks block(s) for multiple purposes (e.g., say, think)	3
Adapt graphic or text elements (e.g., dialogue, character or object) for narrative effect	11	Edit sprite(s) or backdrop(s)	9
		Use change effect block(s)	4
		Use text block(s) (e.g., say, think)	11
Add or manipulate costume(s) to illustrate action(s) (e.g., looking around)	4	Use *change costume* block(s) to change and reset costume(s)	4
Illustrate setting change(s) using *change backdrop* block(s)	9	Use change costume block(s)	9
Consider audience effect when viewing project	12	Use initialising block(s) to execute programme (e.g., green flag)	12
		Set initial program state(s) (e.g., sprite size, position, graphic effect)	7
		Ensure all scripts have hat block(s) (i.e.no 'dead code')	12
Create title cards to orient reader(s)	1	Create text-based sprites (e.g., title cards)	1
Illustrates opening story episode through text blocks	12	Use looks block(s) for multiple purposes (e.g., say, think)	12
Illustrates multiple story episodes through text blocks	6		6
Illustrate setting change(s) using *change backdrop* block(s)	6	Use change costume/backdrop block(s)	6
Organise narrative event(s) using backdrop(s), hat block(s) or broadcast message(s)	9	Uses *change backdrop* block(s) to change and reset backdrop(s)	9
Use structural device(s) to ensure cohesion (e.g., *when* blocks)	10	Use *wait* block(s) to synchronise events correctly	10
Use *broadcast message* block(s) as cohesive device(s)	3	Use *broadcast message(s)* to coordinate multiple processes	3
Check project against plans	10	Run programme to check for errors	10
Revise project in light of plans	4	Debug errors	4

In terms of storytelling (see Table 12), participants each produced a multimodal text based on their emerging storyboards and ideas elaborated in previous activities. Analysis shows that all participants produced a text with at least one narrative episode (*n*=12). A secondary analysis

of narrative structure participants' final projects (Labov, 1972; Ukrainetz et al., 2005): some (*n*=6) produced a chronological story, containing multiple narrative events with temporal but not causal connections, and no conclusion: others (*n*=5) produced an *ending-at-a-high-point* story, with multiple successive events but no resolution, and; one, Natasha, produced a classic (or complete) story, with several narrative events leading to a high point that is then successfully resolved (see Figure 13). The ending-at-a-high-point structure is commonly used for narrative effect though, in this case, participants described not having enough time to programme all of their story ideas into a functional programme. In the case of those who produced a chronological story, these were well-structured with successive events, but the lack of a typical 'high point' meant these were not scored as a 'complete' story or classic narrative. In discussing their projects and the challenges they faced, participants described the difficulties in thinking of successive narrative events ("what to say next") with most (*n=11)* producing a story without a clear resolution.

Exploiting the multimodal features of visual programming allowed then to achieve multiple narrative effects. Most narrative events were represented through altering backdrops and costumes (*n=9)* to indicate a change of setting or character state and adding text/speech blocks (*n=12)*. While some instances of multimodal use were intended to drive their narratives forward, most participants chose to use these to illustrate background detail or provide characterisation (*n*=8).

Some participants' project work provided evidence of the potential of MMC in Scratch on storytelling practices. For instance, Natasha (see Figure 13) used multiple Scratch features for different narrative effects. In her opening episode, she used a *change costume* block to provide characterisation, illustrating the anger her main character felt being left out at a party. She also manipulated one of the other character's costumes to face away from the main character indicating isolation and distance from the other characters. In the next episode, the use of capital letters and exclamation marks heightens the tension. Each episode of her story is demarcated using backdrops, giving a visual indication of a moving narrative. Upon

reconciling their differences, and opting to attend a pool party, she employed a *change costume* block again to indicate the character's reaction to the preceding events; in this example, Natasha used the basic elements of Scratch to illustrate setting, provide a richer characterisation than could be provided through dialogue alone. These practices were reflected across most participants' scores and suggested their goal of providing further information about the characters economically and without lengthy description.



Figure 13: Natasha's final project (Study 1)

5.3.3 Integration of learning strands

Based on the preceding analyses, the targeted conjectures and, by extension, learning activities are considered against their intended pedagogical goals. The intention is to consider how the designed multimodal composition activities relate to the observed programming and storytelling practices identified in the analyses above. Further, these analyses were intended to demonstrate which MMC activities were productive for learners in the pursuit of their respective programming and storytelling goals and, conversely, which activities were less successful; that is, while tasks and activities might pedagogically promote certain integrated practices, these might not have necessarily been employed by participants or promoted by the facilitator. In each section below, exemplar conjectures are extracted from the datasets above to highlight where these activities were repeatedly successful in supporting participants' goals.

The results of the conjecture-based analyses point to clear successes in the pursuit of mutual programming and storytelling goals. The areas outlined below were found to have

strong integration across both programming and storytelling conjectures where engaging in the MMC activities generally resulted in higher conjecture success rates and, consequently, more complex and pedagogically desirable project work.

In this study's dataset, two areas were found to have benefit participants in the pursuit of mutual learning goals: (i) structuring narrative episodes with cohesive devices and (ii) using multimodal text features for narrative effect. These are discussed in the following two sections with illustrative examples from participant project work given.

5.3.3.1 Structuring text(s) using cohesive device(s)

Participants structuring their texts using cohesive devices were found to provide productive opportunities to succeed in both programming and storytelling across multiple activities (see Table 13). This was mostly evident from their use of structural devices common in the Scratch programming interface. Structuring texts was challenging as it required participants to defining individual sprites' states (e.g., position, size, visual effect) to ensure cohesion across repeat execution of their programmes. It also required using *wait* blocks to synchronise all elements of their story, or using more complex solutions (e.g., broadcast messages).

Table 13: Structure text(s) using cohesive device(s)

Activity	MMC task	Conjecture (storytelling)	#/12	Conjecture (programming)	#/12
C6	Structure text(s) using cohesive device(s)	Use *wait* block(s) as cohesive device(s)	8	Use *wait* block(s) to synchronise events correctly	8
D8			10		10
EF9		Consider audience effect when viewing project	7	Set initial program state(s) (e.g., sprite size, position, graphic effect)	7

Though participants had some difficulties in achieving these effects when programming, it offered a clear alignment between the demands of programming and storytelling. For a story to run coherently, it needed to be programmed correctly. If it wasn't programmed correctly, and the programme had bugs, it would reflect in the execution where the story would not finish or would be sequences out of order. This was most noticeable for participants when defining and altering the backdrop (i.e., in which setting to begin their story). Where most opted to use backdrops to indicate changes in setting or narrative events (n=9), these were reflected in their

correctly defining initial sprite and backdrop states. As the effect of a backdrop changing or not changing at the appropriate moment was more noticeable than the cumulative effect of a variable change (e.g., increased transparency), this was not surprising and served as an encouraging result.

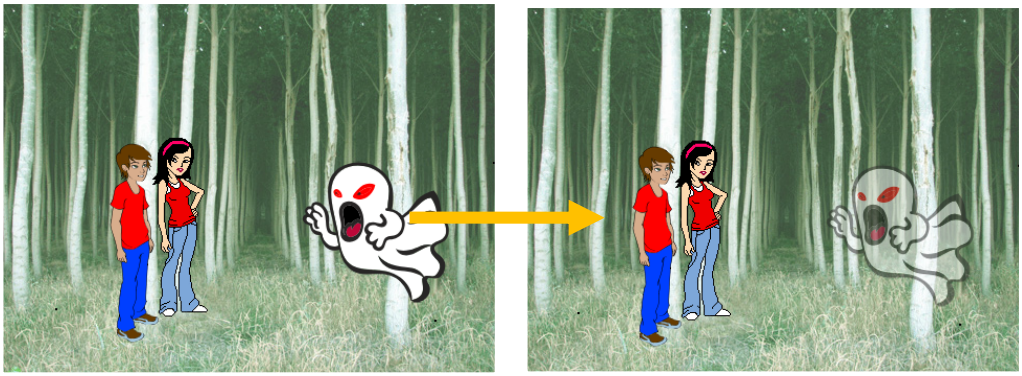


Figure 14: Aoife's use of the ghost effect (Study 1)

Creating multimodal texts in a visual programming environment was demanding as it required participants to handle multiple events, sequence sprites and backdrops correctly, and exploit the features of Scratch for narrative effect. Moreover, the challenge of programming characters to "run at the same time" or thinking about "how many seconds... you have to wait" before each sprite is executed was regularly called out. Nonetheless, participants' scores on each activity generally increased from the structured activities (A1-D8) through to the independent project (EF9).

The most promising result observed during the independent activity was participants' sequencing of their programmes. In most structured activities, participants were tasked with resolving issues relating to the sequence of story programmes. Given that the sequence in which instructions are carried out can affect the outcome, and stories generally have a defined sequence, the conjecture that participants would ensure correct sequencing to maintaining coherence was supported by the data. This increased as the intervention progressed: in the first structured activity (A1), one participant programmed a simple sequence; in the fifth activity (B5), more did so (n=8) and by the independent activity, most

correctly sequenced their narratives (*n*=10). This programming conjecture was largely achieved through their use of *wait* blocks (*n*=10) with fewer using broadcast messages for inter-sprite synchronisation (*n*=4).

Though less common across the dataset, structuring longer texts with cohesive devices additionally provided opportunities for participants to achieve mutual goals (see Table 14). Event handling in participants' projects occurred through multiple strategies, most frequently through using *when green flag is clicked* block (*n*=10) to handle events. However, they did use several unanticipated strategies for sequential execution. When defining the initial curriculum activities and associated conjectures, it was conjectured that participants would use broadcast messages to coordinate multiple sprites, backdrops, and scripts. Given that this was an explicit goal of a structured activity (D8), it was anticipated that broadcast messages would enable participants to execute multiple events (e.g., backdrop changes, dialogue, movement) though this was not supported by the data (D8, *n*=5; independent task, *n*=4) (see Table 14).

Instead of using broadcast messages, participants more commonly (*n*=10) used wait blocks to manage sprite execution in coordinating sprites and backdrops. For most, the use of broadcast messages to create short, more elegant programs was superfluous to their needs. In both structured and independent activities, they opted instead to use wait blocks and coordinate each program element to sequence relative in line with their narrative goals. This meant that most took longer to complete their projects as synchronising multiple program elements required repeated execution which took a considerable amount of time.

Table 14: Structuring longer texts with cohesive devices

Activity	MMC task	Conjecture (storytelling)	#/12	Conjecture (programming)	#/12
D8	Structure text(s) using cohesive device(s)	Use structural device(s) to organise longer compositions	5	Use broadcast message(s) to coordinate multiple processes	5
EF9			4		4

Participants used this strategy in two ways: (i) intra-sprite delays and (ii) inter-sprite synchronisation. In the former case, participants less commonly used the *wait* block in their final projects to manage the execution of individual sprite attributes (e.g., visual effect,

position or size) (*n*=4). In the latter case, they used the wait block to control program execution between multiple sprites (*n*=11); in other words, sprites and backdrops were successfully coordinated to appear in the correct sequence at the appropriate time. While the use of broadcast messages might be programmatically more complex and desirable in programming instruction, within this intervention and given the relative experience and demands of participants, a more productive approach could be taken within this period; this is discussed further in the design implications detailed below.

5.3.3.2 Using multimodal effects to engage and hold a reader's attention

A second area where MMC was found to promote mutual potential for programming and storytelling goals was participants' use of multimodal effects in their projects. Table 15 illustrates some of these conjectures and shows how participants used a variety of structures across multiple activities to tell their stories. In these activities, some participants moved beyond text to use multiple backdrops (*n*=6) and costumes (n=*6*) to indicate events and actions in their respective stories (*n=6)*. In fewer cases, they used motion-based blocks (e.g., *glide to/go to*) (*n*=5) to illustrate narrative events or character's actions in their projects though fewer again (*n*=3) correctly included a corresponding block to 'reset' each sprite's position at the beginning of each project; this was not surprising as motion had not been explicitly taught during structured activities.

Table 15: Using multimodal text features for narrative effect

Activity	MMC task	Conjecture (storytelling)	#/12	Conjecture (programming)	#/12
C7	Choose and use a variety of structures for specific purposes	Illustrate event(s) resulting from actions (e.g., setting change)	5	Add *change costume/backdrop* block(s) to sequence(s)	10
EF9		Illustrates multiple story episodes through setting change	6		6
		Illustrates multiple story episodes through text blocks	6	Use text block(s) (e.g., say, think)	6
				Use motion-based blocks (e.g., *glide*)	3

Though the conjectures were observed across fewer participants, unanticipated multimodal text effects provided evidence of strong integration between programming and storytelling.

For example, many expressed a preference for using animation in their projects. However, given their inexperience and lack of instruction, most sought alternative strategies to achieve similar effects. For instance, Mark decided to graphically edit one of his sprites to depict 'motion lines' upon realizing that his 'floating' animation had not worked (see Figure 15). Subsequent review of his screen recording reveals several attempts to create a loop. While this manipulation of his sprite was novel, and ultimately delivered a similar narrative effect, the appeal of using animation within multimodal projects was exciting as it suggested that more complex programming features could be embedded within the intervention. This would encourage more multimodally rich work and, naturally, a more complex and rich narrative. Moreover, more mutual benefit to both programming and storytelling could be achieved.

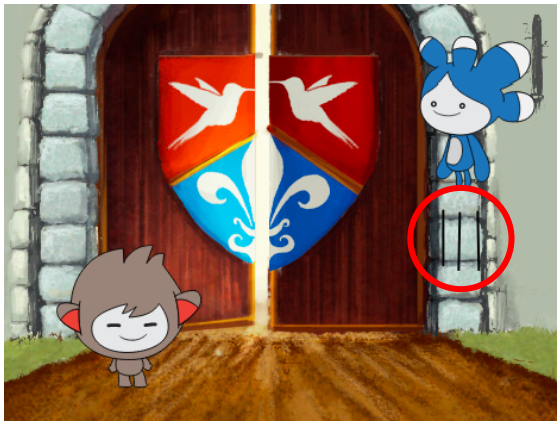


Figure 15: Mark's use of lines to indicate motion (Study 1)

5.4 Summary of findings

This study addresses the question of how an intervention could be designed to mutually support participants' programming and storytelling practices through multimodal composition activities. In this section, the intervention activities are discussed, along with the challenges and opportunities these presented in practice; these are then contextualised against previous findings. From the analysis of this data, it became clear that subsequent redesign of the intervention would be needed. As is customary in DBR, redesign for future use would need to leverage these findings by conceiving design solutions to weaknesses present in either the design, its implementation, or both.

5.4.1 Characterising programming and storytelling practices (SQ1)

The first aim of this study (**SQ1**) was to consider what forms of programming and storytelling practices are fostered through multimodal composition. This was achieved by observing the enactment of the designed learning environment and considering performance across programming and storytelling conjectures relative to each MMC activity. Findings from the conjecture-based analyses were encouraging and suggested that the intervention aims were reasonably met. Moreover, participants viewed the intervention as exciting, expressing contentment at the goal of using Scratch to create multimodal texts though some found certain programming structures difficult to achieve. The activities were equally purposeful as they encouraged participants to draw on their own experience of narrative and visual texts, combining text- and image-based resources, to construct interest-driven multimodal texts. Participants' project goals in Table 11 also demonstrated their desire to draw upon resources in Scratch, creating imagined worlds and stories using the bank of image-based resources (e.g., sprites).

Composing multimodal texts provided opportunities for participants to encounter many foundational programming concepts and engage in relevant programming practices. Employing a storytelling-based project promoted participants to consider the sequencing of their stories and how this could be programmed. It also encouraged them to consider how story elements (e.g., sprites and backdrops) were initialised and, in the case of multiple sprites, how these could be executed in parallel. In line with previous research (Adams & Webster, 2012), participants' projects tended to follow a linear structure where loops, conditional logic and broadcasting messages were less evident. However, structuring a programme for readability meant that the temporal aspects of programmes (e.g., using *wait* blocks to synchronise sprites) was more common. This echoes work by Weintrop et al. (2018) who reported how event-based programming encouraged novice programmers to manually control the speed of the execution for different outcomes; they do caution, however, the need to stress *how* events occur over

when events occur as temporal-oriented programming is primarily a feature of visual block-based programming languages and not more advanced, text-based languages.

Another challenging area where participants found success was initialising certain sprite attributes. Again, in line with previous findings (Franklin et al., 2017), participants struggled to set starting values for different sprite attributes; these include sprite position, size, orientation. For example, in the independent activity (EF9), participants programmed multiple narrative events (e.g., sprites and appearing and disappearing, backdrops changing) and were successful in resetting these values. In earlier activities, they were observed manually move sprites as opposed to seeking the solution to be automated. These earlier challenges provided ample opportunities to highlight relevant issues with participants during instruction (e.g., resetting sprites).

Composing texts in Scratch also provided opportunities for participants to tell stories multimodally in a visual programming environment. This process required participants to choose and manipulate environment-specific representations (e.g., sprites, backdrops, stage), structure and organise their texts using programming strategies, and exploit multimodal features for narrative effect. Participants largely employed text- and image-based resources to construct their multimodal texts. Though other representations were available (e.g., sound resources), these were not explicitly used due to temperamental computer room audio equipment.

At the beginning of the intervention, participants were encouraged to consider other multimodal texts they had encountered (e.g., digital texts, comic books, animation) and to to draw on these experience when constructing their own texts. Unsurprisingly, participants demonstrated oftentimes an implicit understanding of the multimodal resources to be used and their effect on meaning; indeed, it is widely expressed in the literature base that students bring a wide cultural knowledge of visual texts and their conventions to the literacy classroom (Bazalgette & Buckingham, 2013) and these need to be validated (Burn et al., 2010). However, given participants' relative inexperience of programming, their projects never fully succeeded in living up to their narrative goals. For instance, many participants struggled to integrate

motion in their final projects as programming attributes (e.g., initial sprite state, final sprite position), instead opting to express narrative events through dialogue alone. The desire to animated sprites for different purposes, though narratively rich, generated some disappoint among participants.

Composing multimodal texts also highlighted the challenges participants with limited prior programming experiences faced during activities. For instance, many participants struggled to find the required blocks to programme a sprite to speak in Session A; in these cases, they used sound blocks to programme the sprite to play a sound file. Further, in activity D8, participants found difficulty in applying broadcast messages to their projects. Participants were observed referring to broadcast messages as analogous to sprite 'speaking'. This lines up with previous findings (Pea et al., 1987; Sorva, 2013) that suggest the everyday semantics of natural language (e.g., say, broadcast) can produce misconceptions where meaning is derived from the definition of a word and not its function.

Additional challenges were noted when participants when planning and executing multiple narrative goals. Where participants were generally successful in completing the initial structured activities, they were less successful in applying these to an independent project. Performance in the independent project activity (EF9) was not entirely anticipated in the design of the task-oriented analysis in that most participants were unable to complete their story projects to a satisfactory level. Initially, it was unclear whether they were failing to embed suitable programming concepts into their projects or finding the interface difficult to use given the lack of prior programming experiences. However, during the independent project activity, participants commented on the challenge of thinking of successive events for their stories ("what to say next") with multiple participants restarting their projects when they couldn't think of a suitable ending. These discussions revealed that most found the process of generating story ideas difficult and required further practice in this area. Those who successfully completed the structured activities were still unable to complete the independent project. These responses might also indicate that participants could have benefited from more explicit instruction on story generation when compared with instruction on programming

processes or features. Though participants had more experience in reading and telling stories, they found both areas challenging; addressing the balance of instruction in both programming and storytelling is therefore proposed as an area for subsequent redesign.

5.4.2 Relating programming and storytelling practices (SQ2)

The second aim of the study (**SQ2**) was to consider whether and how the practices of programming and storytelling in the production of multimodal texts relate to one another. As the curriculum unite was designed to situate storytelling and programming alongside one another, task conjectures were compared alongside one another and scrutinised to determine their pedagogical value in a single integrated context.

Integration takes many different forms. As Israel and Lash (2020) note in their work on the integration of programming within primary (elementary) mathematics instruction, integration can occur on multiple levels. They define these levels as: (a) no integration (i.e., practices occur in isolation); (b) partial integration (i.e., areas in which one learning strand is used to reinforce the other), and; (c) full integration (i.e., both learning strands are mutually dependent and support one another) (pp. 374-376). As the goal was to consider how integration could meaningfully be achieved—in other words, that programming and storytelling would exist alongside one another in an authentic context—achieving this goal would create a warrant for the productive integration of both learning strands.

However, it was not always possible nor desirable to do so. By comparing the analyses of participants' performances against the target conjectures, areas of the intervention that offered the strongest case for integration across learning strands can be identified. As noted earlier, some activities were found to promote participants to apply multiple storytelling and programming ideas such as the use of cohesive devices to structure texts. As the conjectures associated with these activities yielded strong results, their use in the subsequent iteration is guaranteed.

Participants also expressed wide interest in using multimodal effects for narrative effect. Taking advantage of these features, such as using motion-based animations, might

encourage stronger integration though their success would likely require more explicit instruction. Participants would arguably need more practice in understanding and applying more complex programming features in telling stories. As the intervention ran for only six hours, the potential for curricular content was limited in scope. Nonetheless, these insights were useful as they illustrated how multimodal composition could be productively employed for programming and storytelling goals though this entails the future sequence and selection of activities to be carefully considered for the intended learners.

5.5 Design implications

The analysis detailed above suggest that several modifications might further improve the pedagogical aims of the study. As highlighted by Lewis et al. (2020), one challenge of effective DBR is deciding upon the core elements of iteration. They further this by suggesting that "defining which aspect… is most likely to make the project fail to make the desired impact because it fails to support stakeholder goals" (p. 1151) should serve as a guiding principle in this process. As such, one way of thinking about iteration in the context of these pedagogical goals is to reframe the initial research question of this study (**RQ1**): How can composing multimodal texts *better* support students' programming and storytelling practices? Toward that end, three specific amendments to the intervention are outlined so that it might better support integrated practices in this context. These are given below as: (i) embedding relevant programming concepts; (ii) scaffolding the storytelling process, and; (iii) providing more authentic debugging tasks. These are described in greater detail below, highlighting how empirical evidence indicates what design modifications are needed, and what theoretical supports there are to support such a claim.

5.5.1 Embedding relevant programming concepts (DP2)

During post-session discussions, participants were asked about the activities in which they found difficulty and were then invited to discuss what alternative strategies they sought. Employing more complex multimodal text features was stressed by many during the intervention as crucial to realising many elements of their stories and was not included in the

chosen activities. For most (*n*=9), they indicated a preference for animation/motion to further embellish their projects. One participant was observed graphically edit one of his sprites to depict 'motion lines' upon realising that his 'floating' animation had not worked.

The inclusion of animation for narrative effects like this might promote strong content integration between storytelling and programming by providing participants with additional tools for programming multimodal effects for specific narrative effects. Amending the intervention design to embed more relevant activities might presented an opportunity for more productive programming and storytelling conjectures. Adams and Webster (2012), for instance, analysed animation-based blocks in student projects—in their case music videos—and found that they promoted more complex computing concepts, including loops and nested loops, the use of repeated program execution, defining initial sprite state (i.e., their starting position) and motion blocks (i.e., their ending position).

A redesign of the curriculum unit could also allow more time for using broadcast messages might result in more participants using the concept, in this case, they adopted alternative strategies to achieve their narrative goals. Instead, an additional module—focusing on the use of motion-based blocks to create animation and control the positioning of sprites and objects—will be introduced.

5.5.2 Scaffolding student-generated construction (DP3)

Participants indicated difficulty in generating story ideas—in particular, story endings—with some suggesting that they "did not know what to say" so spent considerable time developing different story ideas before starting their programming work. The results of the independent project suggested that participants were mostly unsuccessful in completing their multimodal texts—that is, they struggled to consider what 'high points' or tensions could be resolved through successive narrative event. Results indicated that they struggled to write a complete story, with some (*n*=5) participants producing an 'end-at-a-high-point' narrative, others (*n*=6) producing a chronological narrative, and one producing a complete 'classic' narrative. Though it was initially conjectured that participants would have difficulty in programming tasks—and

subsequently structured activities (A1-D8) were designed to scaffold and support programming practices—one unanticipated challenge was the need for additional instructional scaffolding in the storytelling strand. Given the frequency of this finding across participants' conjectures, a second modification would seek to address this issue.

As scaffolding was already a design principle in the development of the intervention, it became necessary to revisit the underlying theory that led to the decision to include certain scaffolding materials, support structures, model texts, and demonstration activities. Though the scaffolding measures in Study 1 were not adequate enough—according to the dataset—to support storytelling practices, removing them would be a senseless option. Instead, referring back to theory (Langer & Applebee, 1987), as well as theory-inspired practice (Howell, 2018), to operationalise the design principle in a different, more productive way, was deemed a more worthwhile approach.

Reflecting on the nature of the structured activities, alongside the instructional supports, most of these centred on participants exploring a concept or tool within Scratch to achieve a narrative goal. What is notably absent from these tasks, and their accompanying instructional supports (e.g., demonstrating), is a focus on narratives structure or coherence. Looking at the limited range of models and examples of multimodal texts used, either from their own work or facilitator-generated examples, it became clear that a wider range needed to be introduced and explored. In doing so, it was hoped that participants would have a richer set of narrative- and programme-based examples to draw from and, in turn, would have greater success in programming multimodal stories more aligned with their goals and intentions.

Introducing the elements of a classic narrative (or 'complete' story) as defined by Peterson and McCabe (McCabe & Peterson, 1984) prior to the independent task might arguably resolve some of these issues. Existing activities in the activity sequence rightly introduced several programming concepts necessary to create a multimodal text within the programming environment. However, neither the tasks nor any of the instructions provided opportunities to reflect on the structure and sequence of their developing texts. It could be

argued that some of their shortcomings in producing a complete narrative stem from a lack of consideration of narrative during instruction. With most participants achieving an incomplete episodic structure, it suggests that instructional supports focusing on narrative might result in more complete narrative structures and, subsequently, better stories. It was anticipated that familiarity with domain-specific examples could further be exploited for the purposes of the intervention.

In addition, providing participants with opportunities to connect their in- and out-of-school literacy experiences is another area for further exploitation. In the current cycle, participants were limited to the in-built resources (e.g., sprites, costumes, backdrops) in order to construct their texts. With most expressing interest in using characters and property from popular media, and their associated stories, it seemed helpful to furnish participants with additional resources. By allowing participants to locate and source images from the internet related to their personal interests in future intervention cycles, it was anticipated that this would motivate participants to seek and apply their personal interests within more formal curriculum activities.

Elsewhere, the analyses provided useful information on novice performance in such tasks and suggested areas where the intervention design could be improved. Alongside the final stories produced—or not produced—participants regularly directed their attention to the multimodal features of their stories. Where they generated rich multimodal texts, combining appropriate modes to achieve their narrative goals, cases were *incidental* or emerged organically over the course of the intervention and were a minority of cases. As such, claims about the role of intervention tasks or guided support in supporting participants' storytelling or programming practices are difficult to make. In order to better support participants to analyse, create and evaluate multimodal texts, explicit teaching of the affordances of multimodal effects, and their relationship to narrative texts, is a worthy goal.

5.5.3 Providing more time for ‘purposeful’ activities (DP4)

Participants also indicated a strong preference for using their own narrative texts during the intervention. Looking closely at performance during these activities, many of the conjectures were confirmed where participants opted to use their own texts, as opposed to extending provided examples. This suggests that redesigning the set of activities to accommodate more flexible outcomes would be advantageous. That is, they were less likely to achieve the anticipated outcome where the intended learning goal was fixed, not flexible.

A second concern identified during the intervention was the need for additional programming time to accommodate the independent project. To ensure all participants have increased time for multimodal construction, it is proposed providing an additional session to enable them to achieve their narrative goals. The proposed structured activities in the previous iteration would be concentrated into three sessions. As the data showed that almost all participants favoured their own stories over provided examples, the revised set of activities would privilege learner agency by providing increased time (3 full sessions) for participants to work on their own stories.

CHAPTER 6

Study 2

6.1 Introduction

The results of Study 1 (Chapter 5) suggested that mutually supporting participants' programming and storytelling goals could reasonably be achieved through multimodal composition activities. First, conjecture-based findings based on student performance highlighted areas where integration was at its strongest and areas where tasks were neither useful for integration, nor purposeful for participants' needs. Second, analyses highlighted where participants found many tasks difficult due to their limited experiences of programming and multimodal storytelling.

A central goal of DBR is to leverage the findings of previous design cycles and the theory that guided them to identify areas for refinement or redesign. As the findings from Study 1 were used to locate areas for improvement and articulate these through specific design implications, the aim of Study 2 then is to observe these modifications in practice. Consequently, a second aim of the study, and one central to DBR, is to identify areas for further development to strengthen the robustness of the design. Likewise, as the curriculum unit is trialled with a different group of participants with different contextual demands, there also exists a need to consider how it can be suitably adapted for local needs.

6.1.1 Study aims

The second study of this PhD therefore concerns the iteration of the first design detailed in Study 1. It also concerns the implementation of the intervention with a different group of participants. In this case, participants were found to have had varying prior experiences of computing and multimodal storytelling. To accommodate this group of participants and consider how they could be supported, and to successfully implement the proposed design modifications, the main research question of the study is proposed:

- Research question 2 (**RQ2**): How can learners with varying experiences of programming and storytelling be equally supported during multimodal text construction?

To answer this question, the curriculum unit was redesigned to address the perceived limitations and recommendations noted in the previous chapter, namely: (i) embedding relevant programming concepts to encourage multimodal text effects; (ii) additional scaffolding mechanisms to support participants' text construction, and; (iii) more authentic contexts for debugging tasks. The success of these modifications is evaluated through the following sub-questions:

- Sub-question 3 (**SQ3**): What appropriate scaffolding measures can better support learners with diverse prior experiences of computing and multimodal storytelling?
- Sub-question 4 (**SQ4**): How can more relevant programming and storytelling practices be encouraged?

6.2 Method

6.2.1 Setting and participants

Like Study 1, this study took place during an after-school project-based computing club in the computer room of a primary school in the city of Nottingham (see Figure 16). The researcher again facilitated the intervention for a group of 10 upper primary school students (8-to-11 year olds; 3 girls, 7 boys).



Figure 16: Intervention site (Study 2)

Participants in this school typically had dedicated computing lessons once per fortnight with the school's computing coordinator facilitating each session. As in the case of Study 1, the 2014 curriculum had been in place for several years though the computing coordinator noted how the school was playing 'catch up' to curricular requirements (The Royal Society, 2017).

Participants were comprised of older students (n=5) (10-11 year olds) with some prior experiences of computing and younger students (n=5) (8-9 year olds) with little to no prior experiences of computing. Participants with prior computing experiences had previously taken part in the school's *digital leaders* programme, an extra-curricular club that invited students to participate in after-school clubs, technology fairs, and support younger students in other classes. The younger participants (n=5) had limited to no experience of using Scratch, or any comparative visual programming tool. As such, the curriculum design detailed below offers a differentiated approach to scaffolding learning to accommodate the diversity of experience in the group. Table 16 details participants' pseudonyms used in the study, as well as their prior experiences in computing, storytelling and multimodal composition.

Table 16: Participant pseudonyms and levels of experience

Year	Name	Digital leaders	Experiences of storytelling	Experiences of MMC
Y3	Derek	No	Yes	No
	Lily	No	Yes	No
Y4	Luke	No	Yes	No
	Oliver	No	Yes	No
	Marc	No	Yes	No
Y5	Theo	Yes	Yes	No
Y6	Elijah	Yes	Yes	No
	Eric	Yes	Yes	No
	Leah	Yes	Yes	No
	Mary-Lynn	Yes	Yes	No

6.2.2 Design iterations

Based on the findings of the previous study (Study 1), design iterations are framed against the original design principles proposed in Chapter 4. The proposed modifications are discussed in terms of these principles and how they could be instantiated differently so that the intervention design can provide greater success for learners. Prior to the study, this revised curriculum unit was presented to the computing coordinator to adapt to the appropriate context. These discussions centred on the design principles enacted in the curriculum unit design (e.g., the level of support provided, the instructional sequence) as well as school conventions. In the following sections, these are discussed in terms of the amendments made to the curriculum design.

6.2.2.1 Engage learners in constructive activities (DP2)

Another principle finding of Study 1 was the degree to which learners opting to not use some of the programming features explored in structured activities in their final independent project. While some learners chose to use other features for their narrative goals, many had expressed a preference for multimodally-rich features to enrich and enhance their stories. To accommodate this in Study 2, this iteration provided learners with opportunities to explore and apply a greater number of multimodal text features in their texts. These are demonstrated in Table 17 and include the applications noted in the previous chapter, namely embedding

animated sequences using repeated loops and character motion using motion blocks. Encouraging participants to understand and apply these multimodal texts also meant providing successful model texts as examples during instruction. These texts were specifically designed to embed complex and desirable programming features by grounding them in practical examples (e.g., animated sequences through repeat loops). It was hoped that, by demonstrating the impact of these multimodal features on meaning, participants would approach their independent work with a richer set of examples and structures in which to construct their project.

A secondary concern identified in the analyses was the need for additional programming time to accommodate the independent project; this modification is intended to ensure all participants had increased time for multimodal construction and to enable them to achieve their narrative goals. Table 17 outlines the previously proposed sequence of activities (column 2) and the revised sequence of activities (column 3) undertaken by participants in Study 2; changes are highlighted in bold.

Finally, the proposed structured activities in the previous iteration had largely focused on closed tasks—that is, those that result in the same outcome for all participants—with independent activities reserved for the final two sessions. The revised set of activities privileges learner agency by providing increased opportunities for participants to generate narrative events of their choosing; it is conjectured that this distinction will engage participants by providing them with agency to create richer, more complex stories.

6.2.2.2 Scaffolding student-generated construction (DP3)

Operationalising *scaffolding* required the local context and participants' prior experience in computing and in constructing multimodal stories to be considered. Though the dangers of inaccurate assessment of learners' ability for independent performance is commonly highlighted as an issue when designing of appropriate scaffolding measures (Reiser & Tabak, 2014), detailed conversations with the classroom teacher and computing coordinator prior to the intervention provided a better sense of the support that participants might need.

Research into scaffolding programming environments for novice learners has a long history of advocating reflective practices by encouraging learners to be mindful of design choices and consequences when creating programmes (Lye & Koh, 2014). As outlined by Lehrer et al. (1999, p. 248) common reflective teaching practices include (a) prompting conversations about principles to anchor programme designs; (b) a provision of programming templates (to embody prompt discussions around key principles of programming design; (c) summarisation of the action of programming code to encourage learners to 'chunk' blocks/scripts when executing, and; (d) contrasting examples of code to encourage learners to reflect on the relationship between chunks of code and their programme outputs.

In Study 2, these were adapted to suit learners with varying experiences in programming and multimodal storytelling in the group. Discussions with the classroom teacher revealed that half of participants had previously participated at an in-school *digital leaders* programme and consequently had more experience in designing and evaluating *Scratch* programmes. Conversely, half of the group had limited classroom experience of computing though their out-of-school experiences differed strongly. Incidentally, all participants had limited experience of formal (i.e., in-school) multimodal text construction. As MMC does not feature explicitly in the 2014 English curriculum, instead existing implicitly within the Computing curriculum (within the Digital Literacy strand), it was not widely taught nor featured regularly in classroom practice. This necessitated a differentiated approach to scaffolding to ensure all learners needs were being adequately met. In practice, learners with more experience programming in *Scratch* were challenged with more demanding tasks, including the use of hierarchically more complex programming features and, accordingly, more rich multimodal text features to use. These are reflected in Table 1 below (additional tasks are highlighted in green).

Learners with little to no experience of Scratch required similar accommodation during programming sessions. Whereas software-realised scaffolding in Scratch (e.g., blocks snapping together to eliminate syntax errors) provided some support, facilitator guidance was needed to encourage learners to apply regulatory strategies in debugging programme errors

and revising texts. This was primarily achieved through modelling and demonstration of each new programming feature and its corresponding function in a multimodal story; as facilitator, these think aloud process demonstrations were designed to encourage learners to regularly reflect on their evolving narrative goals and, consequently, promote learners' debugging processes to revise and improve their texts. Some of these prompts included asking learners to refer to their plans and consider how certain programming features could be used productively for their narrative goals. A measure of the success of these strategies then was the degree to which these programming features emerged organically in learners' independent work.

Additional prompts were distributed across different means of support; in other words, if one or both facilitators are working with a group, additional means of support were available. Unlike Study 1, paper-based prompts were designed for different groups of learners (see Figure 17). For instance, the group of *digital leaders* were tasked open-ended, free-choice activities to both promote learner agency and resist stifling their creativity in both programming and storytelling. However, some of the less experienced learners in the group required a synergistic approach to scaffolding, employing both teacher guidance and regular scaffolding prompts, to support their learning. This group were equally provided open-ended, free-choice activities with additional modelling and demonstrations.

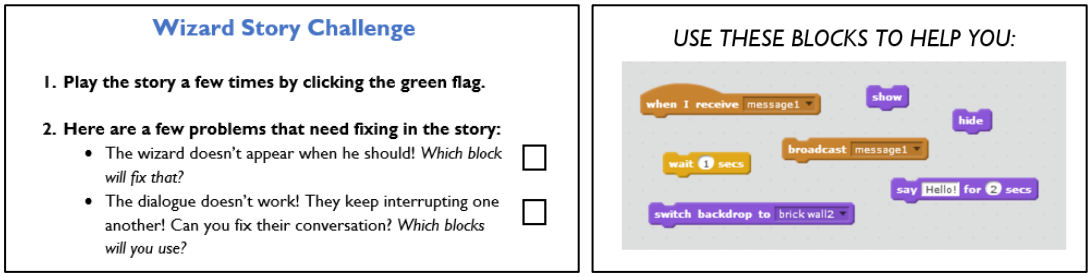


Figure 17: Screenshots of differentiated scaffolding prompts (Study 2)

6.2.2.3 Engaging learners in authentic and purposeful activities (DP4)

In Study 2, to promote greater learner agency while still encouraging more complex programming constructs, activities including demonstrations and discussions, were modified to focus on the outputs of certain programming choice. For instance, in activity C6 adding two

distinct *change costume* blocks within a repeat loop generates simple animated sequences. As noted in Study 1, these preferences emerged organically through discussions with participants as they sought to accomplish their narrative goals with a limited understanding of programming features. The revised sequence of activities aims to compensates for this disparity, providing ample opportunities to create narrative projects while focusing on appropriate programming constructs for rich multimodal text effects.

6.2.3 Intervention procedure

A pre-intervention meeting was scheduled with the classroom teacher and school computing coordinator to discuss the context of the study, including participants' prior experiences in computing and in constructing multimodal stories. The location of the research site, access to equipment, and timetabling was also discussed.

Each of the six hourly sessions took place weekly after timetabled classes in the school's computing room. Participants sat in a circle of tables with a computer connected to an overhead projector for instructional purposes. In the first three sessions, each activity was introduced through a live demonstration by the facilitator. As the space was limited participants moved between their computers and a row of chairs for discussion-based activities (see Figure 16). To encourage more active engagement, participants were asked to contribute to demonstration activities and asked to predict what the programme output would be. Participants then returned to their computers and attempted the tasks in each activity. Scaffolding prompts (see Figure 17) were distributed to different learner groups and displayed on the overhead projector. I acted as facilitator with the computing coordinator supporting participants where needed; that is, where design prompts were insufficient in supporting

learners' needs, additional scaffolding in that both facilitators provided guidance, offered suggestions and, in some cases, modelled the appropriate steps to accomplish certain tasks.

Table 17: Original and revised activities

Session	Activity	Original proposed activities (Study 1)	Revised activities (Study 2)
A	A1	Participants are guided to initialise a sprite and use *say* blocks to illustrate speech	Participants are guided to initialise a sprite and use *say* blocks to illustrate speech
	A2	Participants are guided to add additional *say* blocks and maintain readability	Participants are guided to initialise a second sprite and create an authentic narrative context. They are asked to maintain readability by managed programme execution
	A3	Participants are guided to initialise a second sprites and simulate a conversation through managed programme execution	
B	B4	Participants are guided to debug an existing narrative by successfully managing programme execution	Participants are guided to debug an existing narrative project by successfully managing programme execution using *wait* blocks or *broadcast message* blocks
	B5	Participants manipulate sprite attributed (e.g., *show/hide* blocks) to introduce a set of changes to their character's environment (e.g., the introduction of a natural disaster)	Participants are guided to add additional narrative event(s) of their choosing to the sample text, employing text blocks, change backdrops, and manipulating sprite attributes (e.g., size, position) to illustrate these events.
C	C6	Participants are guided to debug an existing narrative project using *wait* blocks, defining and manipulating sprite attributes (e.g., visibility).	Participants are guided to debug an existing narrative project using *wait* blocks or *broadcast message* blocks, defining and manipulating sprite attributes (e.g., position, size, visibility) and creating animated sequences using repeated loops. They are also asked to reset all of these attributes so that the programme outputs the same result automatically. Participants end by collectively drafting ideas for their independent projects and sourcing appropriate images to use in their projects.
	C7	Participants are guided to use *change backdrop* blocks to illustrate successive narrative events	
D	D8	Participants use broadcast scripts to coordinate narrative elements simultaneously (e.g., change backdrop and introduce a new sprite)	Participants apply a variety of programming concepts to create a 'complete' multimodal story—that is, one with all (or most of) the elements of a basic narrative structure along with selected multimodal effects (relevant to their goals)
EF	EF9	Participants apply a variety of programming concepts to create a 'complete' multimodal story—that is, one with all (or most of) the elements of a basic narrative structure along with selected multimodal effects (relevant to their goals)	

In the final three sessions, participants completed their independent narrative project. To address the limitations of the activities enacted in Study 1, the storytelling process was modelled to participants. This involved generating a model text (seen in Figure 19) that embodies the desired outcomes in its design. Outcomes included demonstrating multiple narrative episodes and how to communicate these (e.g., through backdrop changes), and how to use multimodal text features (e.g., motion blocks) to embellish their stories and contribute to meaning.

The demonstration materials, scaffolding prompts (e.g., see Figure 17), and sample projects were made available to them for use though all participants started their projects from scratch. As these sessions are largely independent, the classroom teacher and I identified common misconceptions or challenges participants were facing and organised whole-group discussions during each session to discuss problems and potential solutions; for instance, in one session, participants were show multiple examples of code based on common misconceptions (e.g., blocks not executing in sequence as they aren't connected) and asked to predict their corresponding outputs.

6.2.4 Data analysis

As in Study 1, the data generated was analysed using the task-oriented analyses framework (see Appendix 3). This framework was again applied across activities to assess participants' storytelling and programming practices and how these were fostered in a visual programming environment through MMC activities. However, unlike Study 1, additional conjectures were needed to accommodate the addition of different activities (e.g., using loops to create animation or repeating motion).

These additional conjectures also reflected the fact that participants had varying prior experiences in programming. As such, conjectures are reported separately in the next section where "**DL #/5"** is used to denote the five learners who took part in the *digital leaders* programme and therefore were anticipated to achieve more complex conjecture-related goals. Conversely, the five younger learners with fewer programming experiences are recorded

alongside (as **"#/5"**) (e.g., see Table 18). As younger learners were anticipated to satisfy fewer of the more complex programming and storytelling conjectures, there are 'greyed out' in the analyses; the first of these differentiated analyses is available in Table 19.

6.3 Findings

In this section, findings are discussed both in terms of performance in the structured and independent activities. As in Study 1, these results are described in three sections. The first two sections detail the results of the analyses of participants' performance against their differentiated conjectures. In the third section, conjectures are compared across the intervention to determine whether tasks in both learning strands complement one another or exist in isolation Illustrative cases from the dataset are provided to provide context to participants' programming and storytelling goals.

6.3.1 Structured activities (A1-C5)

6.3.1.1 Programming dialogue between characters (activities A1-A2)

In Session A, the first activity (Table 18, activity A1) tasked participants with adding and initialising a single sprite to speak at the beginning of a story. Participants chose one of the provided sprites as their character and used *say* blocks to provide additional information about their characters. Table 18 illustrates that all participants successfully added a sequence of say blocks (n=10) with a majority including characterisation details (e.g., personal details, character preferences, responses to events) (n=9). In programming the sequence of utterances, some participants (n=3) initially had some difficulty in distinguishing between *say* and *say for x seconds* blocks which, when not timed in a sequence, is not readable. This distinction led to a discussion around readability for the end user/reader and prompted participants to consider how the text was encountered by others. In total, all participants resolved this issue by the second activity (Table 18, activity A2).

Table 18: Analyses of activities A1-A2 (Programming dialogue between characters)

Act.	Conjecture (storytelling)	#/5	#/5 DL	Conjecture (programming)	#/5	#/5 DL
A1	Add character(s) or object(s) using sprite(s)	5	5	Add sprite(s)	5	5
	Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., say blocks)	4	5	Create sequence using *say* block(s)	5	5
	Maintain readability using multiple *say* block(s)	3	4			
	Use structural device(s) to ensure cohesion (e.g., yellow 'events' blocks)	5	5	Initialise sprite using *hat* block (e.g., when green flag is pressed)	5	5
A2	Illustrates first story episode through text blocks	4	5	Create sequence using *say* block(s)	4	5
	Illustrate characterisation (roles, habitual actions, personality, appearance) through dialogue (i.e.., *say* blocks)	3	5		3	5
	Maintain readability using multiple *say* block(s)	5	5		5	5
	Coordinate narrative element(s) to execute correctly	3	5	Execute two or more independent sprites concurrently	3	5
	Use structural device(s) to ensure cohesion (e.g., yellow 'events' blocks)	5	5	Initialise sprite using *hat* block (e.g., when green flag is pressed)	5	5
	Establish setting(s) using backdrop(s)	4	4	Adds backdrop(s)	4	4
	Use *wait* block(s) as cohesive device(s)	3	5	Use *wait* block(s) to synchronise events correctly	3	5

The second activity (Table 18, A2) tasked participants with initialising a second sprite and programme a conversation between their chosen sprites. They were also asked to consider an authentic narrative context, or why their characters would be talking (e.g., one character asking another for help). In most cases (*n*=8), participants used the nature of the character (e.g., a ghoul) as the basis for a meaningful interaction between two characters (e.g., scaring a friend). Where some completed the activity within the allotted time (*n*=4), they opted to add additional details to their story sequence (e.g., backdrops, additional costumes) (*n*=8) and, in the case of older participants, some opted to use broadcast scripts to coordinate narrative elements (*n*=3). In one case, an older participant, Eric, added loop animations and motion sequences to his project, to simulate a flying ghoul and a dragon breathing fire and engage and hold his reader's

attention (see Figure 18). As noted above in design iterations, these results prompted a reconsideration of conjectures for future sessions.



Figure 18: Screenshot of Eric's loop animation (Study 2)

The second activity also proved challenging for two younger participants, Derek and Lily, who struggled to coordinate dialogue using *wait* blocks (n=8). Up until this point, scaffolding measures had been applied evenly across the group regardless of age or prior experiences in programming/multimodal storytelling; however, this result indicated that these measures needed to be reconsidered for the second session.

6.3.1.2 Debugging a story (activities B3-B4)

Session B started with a debugging activity. Participants were given a sample project and tasked with fixing a series of issues relating to a story sequence (Table 19, activity B3). This required participants to execute the project and check if the output is readable. Owing to the group's diversity of experience, additional conjectures (highlighted in bold) relating to *broadcast message* blocks and initialising sprites were included; these challenges were equally outlined to participants who had more experience using Scratch.

In the second activity of the session (see Table 19, B4), participants were asked to complete the original story prompt using an ending of their choice. It was anticipated that at this point participants would generally use *say* blocks to illustrate key narrative events in their stories with some using additional features (e.g., backdrops) to illustrate successive events. Participants' stories commonly centred on journeys to multiple locations, including volcanoes,

music concerts, space, and enchanted forests, or a character requesting help from another in missions and quests. As conjectured, they widely used *say* blocks to illustrate key events (n=10) with fewer (n=6) using backdrops to represent narrative events (e.g., characters being transporting to space).

In both cases, participants applied different strategies to coordinate multiple narrative elements. In most cases (n=8), they used *wait* blocks to determine how and when different elements interacted with one another. More experienced participants used more complex programming features (e.g., *broadcast message* blocks, n=2) and defined initial sprite states (e.g., character position, size and graphic effect) (n=3); examples of these include the sprite size attribute being used to indicate characters walking into the distance, or a costume change illustrating a character being transformed into a frog.

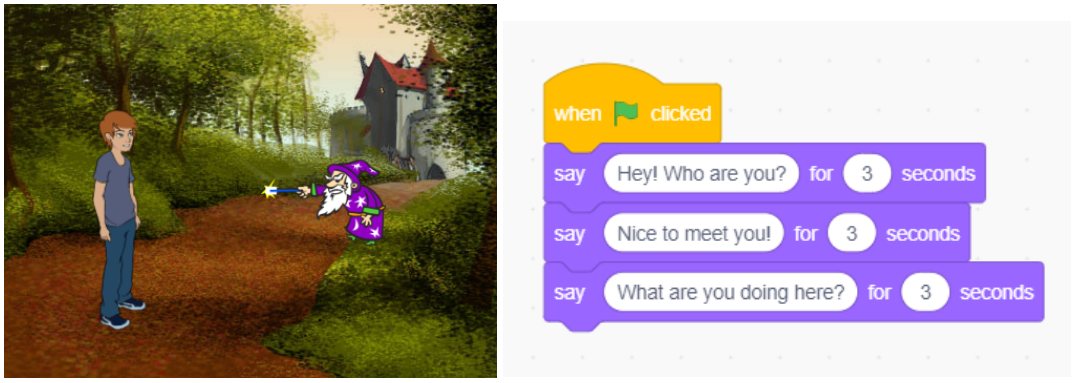


Figure 19: Screenshot of debugging activity (stage and code) (Study 2)

Table 19: Analyses of activities B3-B4 (Debugging a story)

Act.	Conjecture (storytelling)	#/5	DL #/5	Conjecture (programming)	#/5	DL #/5
B3	Illustrates opening episode through text blocks	5	5	Create sequence using *say* block(s)	5	5
	Use *show/hide* block(s) as cohesive device(s)	4	3	Add *show/hide* block(s) to sequence(s)	4	3
	Illustrates story episodes through setting change	3	3	Uses change backdrop block(s) to change and reset backdrop(s)	3	3
	Use *wait* block(s) as cohesive device(s)	5	5	Use *wait* block(s) to synchronise sprite(s)	5	5
	Use *broadcast message* block(s) as cohesive device(s)		2	Use *broadcast message(s)* to coordinate multiple processes		2
	Coordinate narrative element(s) to execute correctly	5	5	Use initialising block(s) (e.g., *when* blocks) on all scripts	5	5
		5	5	Execute two or more independent sprites concurrently	5	5
B4	Illustrates opening episode through text blocks	5	5	Create sequence using *say* block(s)	5	5
	Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., say blocks)	3	5		3	5
	Consider audience effect when viewing project		4	Use *change effect* block(s)		4
			4	Define initial program state(s) (e.g., size, position, graphic effect)		4
	Illustrates multiple story episodes through setting change	3	3	Uses *change backdrop* block(s) to change and reset backdrop(s)	3	3
	Use structural device(s) to ensure cohesion (e.g., yellow 'events' blocks)	4	5	Initialise sprite using *hat* block (e.g., when green flag is pressed)	4	5
	Use *wait* block(s) as cohesive device(s)	3	5	Use *wait* block(s) to synchronise sprite(s)	3	5
	Use *broadcast message* block(s) as cohesive device(s)		2	Use *broadcast message(s)* to coordinate multiple processes		2
	Coordinate narrative element(s) to execute correctly	3	4	Execute two or more independent sprites concurrently	3	4
	Establish setting(s) using backdrop(s)	3	3	Add backdrops(s)	3	3

6.3.1.3 Coordinating narrative elements (activities C5-C6)

In the third session, participants were first tasked with completing a debugging activity (see Figure 7). To execute the story correctly, participants needed to add *wait* blocks or *broadcast message* blocks at the appropriate point in the narrative. They then needed to programme a third sprite, a bat, to appear inside the castle and the boy to react to its appearance. This task also required participants to initialise all sprites and backdrops correctly as they correspond to later events. For instance, by programming the backdrop to switch to the castle, they must

correspondingly programme the backdrop to switch back to the forest at the beginning of the programme. The session concluded with participants drafting ideas for their independent projects and sourcing appropriate images to use in their projects. Differentiated scaffolding and instruction meant that the *digital leaders* group were additionally tasked with adding animations using repeated loops (conjecture highlighted in bold).

Table 20: Analyses of activity C5 (Coordinating narrative elements)

Activity	Conjecture (storytelling)	#/5	DL #/5	Conjecture (programming)	#/5	DL #/5
C5	Illustrates story episodes through setting change	5	5	Create sequence using *say* Uses *change backdrop* block(s) to change and reset backdrop(s)	5	5
	Use cohesive device(s) (e.g., hat blocks, *show/hide* blocks, *wait* blocks)	5	5	Use *wait* block(s) to synchronise sprite(s)	5	5
	Use *show/hide* block(s) as cohesive device(s)	5	5	Add *show/hide* block(s) to sequence(s)	5	5
	Consider audience effect when viewing project	5	5	Set initial program state(s) (e.g., sprite size, position, graphic effect)	5	5
	Illustrate event(s) resulting from actions (e.g., setting change)	5	5	Add *change backdrop* block(s) to sequence(s)	5	5
	Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., say blocks)	5	5	Create sequence using *say* block(s)	5	5
			5	Use switch costume block(s)		5
	Establish setting (surroundings and habitual states)	5	1	Edit or manipulate backdrop(s)	5	1
	Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., say blocks)		5	Use *motion* blocks (e.g., glide. go X/Y)		5
				Use loop constructs to create animation		5
	Use cohesive device(s) (e.g., hat blocks, *show/hide* blocks)	5	5	Use *wait* block(s) to synchronise sprite(s)	5	5

In the main activity (Table 20, C5), participants all successfully managed to synchronise sprite timings and achieve parallelization in both tasks (n=10). Additionally, all participants managed to programme the bat sprite to appear at the appropriate time, and their character to react accordingly using *say* blocks.

6.3.2 Independent project (DEF6)

In the final three sessions, participants completed their independent project. During this time, participants designed and evaluated their projects against their original plans. These plans are briefly summarised in Table 21. Separately, Table 22 outlines the conjecture-based analyses that feature additional conjectures (bolded) for the *digital leaders* group.

Participants were generally more successful in creating their independent project in satisfying both the conjectured learning and their own project goal. Providing additional time to work on the independent project—which constitutes most of the conjecture-based analyses—provided breathing room for participants to fully realise their intended outcomes. It also meant that participants had more time to explore different forms of multimodal text features to represent their emerging storyboards and ideas elaborated in previous tasks.

Analyses showed that most participants produced a complete text. As in Study 1, an analysis of narrative structure of each participants' project (Labov, 1972; Ukrainetz, 2006) showed that most (n=9) students produced a classic story, with several events leading to a high point that is then successfully resolved (i.e., a narrative resolution); the remaining student, Theo, produced an ending-at-a-high-point story, with narrative orientation, a series of successive events though no resolution of the 'high point' event (in this case, a basketball player who helps a witch recover an object to save the city). In this case, the ending-at-a-high-point structure emerged as an intentional feature of Theo's story as opposed to an unintended effect of running out of time. The other participants produced stories that were well-structured with successive events, generally featuring a 'high point' event and a resolution.

Table 21: Participants' project goals (Study 2)

Participant	Narrative goal
Derek	"DJ Marshmallow is captured by the Fire King. He raises him like a father. When Marshmallow is older he has his first concert but he is very scared he doesn't know anyone.. He finds his own kind at the concert."
Elijah	"My story is about Greg who wants to be rich and famous but has a horrible family to deal with. He goes to middle school and gets into a fight with Bryce but he tells the truth and doesn't have to be punished."
Eric	"My story is about Skelly and Munchkin who are friends and they are celebrating Munchkin's birthday but a dragon comes and steals his present. They go on a quest to rescue the present from the dragon's castle. It turned out it's a magical snow globe."
Leah	"My story is about a girl who gets kidnapped by an invisible man who you can only see his shoes. She escapes from the rubbish bin she's locked in but then she falls off a cliff by accident."
Lily	"My story is about DJ Marshmallow and goes to the seaside with his friends and they have a lovely day."
Luke	"My story is a battle in Fortnite. Ninja wants to win but he has no loot and Banana hits Blackheart so he gets a point. He gets a legendary chest and finally reaches the Arctic and then he wins the battle."
Marc	"Harry, Ron and Hermione are on their way to get the Philosopher's Stone but the basilisk is in the way. They cast a spell to kill the snake and get the stone.
Mary-Lynn	"Zara is a kidnapper who is trying to kidnap an invisible person. Suddenly she sees him and tries to capture him but he recognises her from the news and fights her. She gives up and he is safe."
Oliver	"Ninja wats to win the battle with my scar and get their loot. He hits Skull and wins a legendary chest in the Arctic. He wins."
Theo	"My story is about a gymnast in a school and he doesn't like his sister. A wizard asks for their help to save Nutropolis. They must find the Light of Time to save the city but it's too late…"

The conjecture-based analyses also highlighted a wider range of multimodal text features, such as the use of *motion* blocks to simulate movement and character actions, and a corresponding use of more complex programming features. As proposed in the design modifications (see 6.2.2), demonstrating the purpose and practice of using loops (e.g., forever, repeat X times) to create animation and *motion* blocks to simulate movement provided participants with a repertoire of tools and resources in which to construct their stories. In some cases, participants combined these functions for additional visual effects. For instance, one participant, Eric, used a combination of a repeated loop and *change size* block to simulate characters moving far into the distance (see Figure 20).



Figure 20: Screenshot of Eric's project using a repeated loop animation (Study 2)

Participants additionally exploited many multimodal text features to enhance aspects of their stories. For instance, some participants (*n*=4) opted to use looks blocks (e.g., say, think) to indicate different character responses (e.g., speaking to another character, having an internal monologue). They also adapted existing image-based materials to suit their individual narrative goals. Though arguably cumbersome, participants commonly indicated a preference for manipulating sprites and backdrops to change setting details, assign objects to characters, and remix existing narrative resources for new purposes. In the case of the *digital leaders* group, these modifications to sprite costumes could then be looped to create simple animated sequences; all *digital leaders* (*n*=5/5) embedded these sequences in their texts. For instance, one participant, Elijah, modified an existing sprite (a popular children's book character) to include additional gestures, one indicating a walking motion and another illustrating a grabbing gesture (see Figure 21). In his text, he then used a repeating loop of *change costume* blocks when Greg moves setting (e.g., from his house to school) and the grabbing gesture costume when Greg steals a handbag.

Table 22: Analyses of activity DEF6 (Independent project)

Conjecture (storytelling)	#/5	DL #/5	Conjecture (programming)	#/5	DL #/5
Define a narrative goal	5	5			
Add character(s) or object(s) using sprite(s)	5	5	Add sprite(s)	5	5
Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., *say* blocks)	5	4	Create sequence using *say* block(s)	5	4
Illustrate characterisation (behaviour, appearance) through costume(s)	3	2	Use *change costume* block(s) to change and reset costume(s)	3	2
		5	Use loops to create animation or repeating motion (e.g., looping costume changes)		5
Establish setting(s) using backdrop(s)	5	5	Add backdrop(s)	5	5
			Edit backdrop(s)	4	3
Illustrate setting change(s) using *change backdrop* block(s)			Use *change backdrop* block(s) to change and reset setting(s)	5	4
Use *show/hide* block(s) as cohesive device(s)	5	5	Add *show/hide* block(s) to sequence(s)	5	5
Use varied text block(s) (e.g., *say*, *think*) to represent dialogue/thoughts	2	2	Use looks block(s) for multiple purposes (e.g., say, think)	2	2
Adapt graphic or text elements (e.g., dialogue, character or object) for narrative effect	5	5	Edit sprite(s) or backdrop(s)	5	5
			Use *change effect* block(s)	1	3
Add or manipulate costume(s) to illustrate action(s) (e.g., looking around)	5	4	Edit costume(s)	4	3
			Use change costume block(s)	5	4
Use *broadcast message* block(s) as cohesive device(s)		5	Use *broadcast message(s)* to coordinate multiple processes		5
Coordinate narrative element(s) to execute correctly	5	5	Use initialising block(s) (e.g., *when* blocks) on all scripts	5	5
			Set initial program state(s) (e.g., sprite size, position, graphic effect)	5	5
			Ensure all scripts have hat block(s) (i.e.no 'dead code')	5	4
Illustrates story episode(s) through text blocks	5	5	Use text block(s) (e.g., *say, think*)	5	5
Illustrates story episode(s) through character action (e.g., motion)	5	5	Use change costume block(s)	5	4
			Use *motion* blocks (e.g., glide. go X/Y)		5
Create title cards to orient reader(s)	2	1	Create text-based sprites (e.g., title cards)	2	1
Use structural device(s) to ensure cohesion (e.g., yellow 'events' blocks)	5	5	Use change backdrop block(s)	5	4
Use *wait* block(s) as cohesive device(s)	5	5	Use wait block(s) to sequence narrative correctly	5	5
Use *broadcast message* block(s) as cohesive device(s)		5	Use broadcast messages(s) to coordinate multiple processes		5
Check project against plans	5	5	Run programme to check for errors	5	5
Revise project in light of plans	4	5	Debug errors	4	5

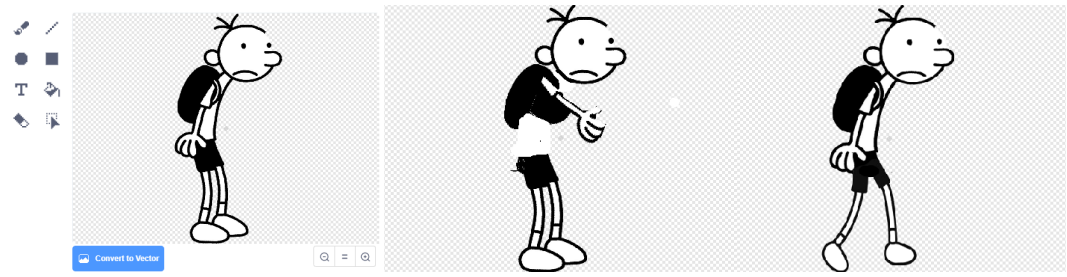


Figure 21: Example of sprite manipulation (Study 2)

6.3.3 Integration of learning strands

In this section, learning activities and their associated conjectures are considered against their intended pedagogical goals. As in Study 1, applying conjecture-based analyses across multiple activities served two purposes: (i) they provide evidence where multimodal composition activities and their associated practices supported learners' programming and storytelling goals, or (ii) where activities were less successful or productive in supporting these goals. In each section below, exemplar conjectures are extracted from the full dataset to highlight where these activities were successful.

The results of the conjecture-based analyses in this study point to greater successes in learners' pursuit of programming and storytelling goals. The area outlined below were found to have the strong integration across both programming and storytelling conjectures where engaging in such MMC tasks generally resulted in longer, more multimodally rich stories and, consequently, more sophisticated programmes. Where some activities and corresponding conjectures were deemed less successful or productive for participants in Study 1, the enacted modifications to the instruction sequence were intended provide greater opportunities for integrated learning. The results of these changes are discussed under design implications (see section 6.6).

As in Study 1, the following areas were considered fruitful in the pursuit of mutual learning goals: (i) structuring narrative episodes with cohesive devices and; (ii) using multimodal text features for narrative effect. However, additional conjectures are noted in this study and are detailed below.

6.3.3.1 Structuring narrative episodes with cohesive devices

Participants structuring their stories using cohesive devices (e.g., *hat* blocks) was found to be associated with the highest number of conjectures. In the first activity (A1), participants' use of multiple *hat* blocks (e.g., *when backdrop switches to X*) ensured their stories were cohesive and followed a pre-defined sequence. For instance, *when backdrop switches to X* blocks were commonly used to anchor specific events to setting changes (*n*=8) (e.g., characters moving inside a castle/a bat appearing). These blocks improve both the readability of the programme while making it easier to review each narrative event separately. Moreover, it prompted participants to abstract their storytelling goals to manageable sections when programming (i.e., one scene per script). Similar results were noted during the independent project (Table 13, activity DEF6) as participants again used *wait* blocks (*n*=10) and, in the case of the *digital leaders*, multiple *broadcast message* blocks (*n*=5). Managing programme execution through these strategies generated a lot of bugs when executing participants' stories; one of the final tasks involved checking participants' projects against their plans by running through their project multiple times. This enabled participants to identify and correct any deviations from original plans, bugs affecting story sequencing and cohesion, and corrected; in total, all participants correctly identified one or more errors when revising their projects with most (*n*=9) making the necessary changes to fix these.

Table 23: Structuring narrative episodes with cohesive devices (conjectures)

Act.	MMC task	Conj. (story.)	#/5	DL #/5	Conj. (prog.)	#/5	DL #/5
B4	Structure longer texts with cohesive device(s)	Maintain story cohesion using structural device(s) (e.g., when blocks)	4	5	Initialise sprite using hat block (e.g., when green flag is pressed)	4	5
DEF6			5	5	Use wait block(s) to sequence narrative correctly	5	5
		Use structural device(s) to organise longer compositions (e.g., broadcast message blocks)		5	Use broadcast message blocks(s) to coordinate multiple processes		5
	Improve own composition, redrafting for purpose and readers' needs	Check project against original plans	5	5	Run programme to check for errors	5	5
		Revise project in light of plans	4	5	Debug errors	4	5

6.3.3.2 Using multimodal text features for narrative effect

A series of successful conjectures related to learners choosing which representations to best communicate meaning. In total, 136 instances of storytelling conjectures and 112 of programming conjectures were identified in the process data; these not only suggest some meaningful integration between storytelling and programming when deciding on modes to communicate meaning, they also highlight the dominance of this practice in creating multimodal stories where participants continually adapt and refine their projects for differing narrative goals.

Table 24 provides a sample of where participants choosing modes for different purposes resulting in a high number of conjectures related to programming and storytelling. As the more advanced programming features were difficult to accomplish by the younger participants (e.g., repeat loops), more widely used multimodal text features included multiple costume blocks, backdrops and manipulated image-based resources. For instance, the use of multiple costumes to add characterisation (habitual actions, personality, or appearance) or illustrate character actions requires costume attributes to be reset for multiple programme

executions. Additionally, the diversity of experience across the group warranted multiple trajectories for participants whose capacities in Scratch vary widely.

Table 24: Multimodal text features (conjectures)

Act.	MMC task	Conjecture (story.)	#/5	DL #/5	Conjecture (prog.)	#/5	DL #/5
C5	Choose which mode(s) will best communicate meaning for specific purposes	Illustrate characterisation (roles, behaviour, appearance) through dialogue	5	5	Use repeat loops to create animated sequences	5	5
		Provide character detail(s) (e.g., habitual actions, personality, appearance)		5	Use switch costume block(s)		5
		Add character(s) using sprite(s)	5	5	Initialise sprite(s)	5	5
		Establish setting(s) using backdrop(s)	5	5	Add backdrop(s)	5	5

6.4 Summary of findings

Participants' performance was again assessed by analysing screen recordings, final project work to conduct the conjecture-based analyses. In this study, findings from these were used to observe performance in programming and storytelling and assess the impact of the design iterations.

Participants were found to have employed many of the basic programming and storytelling constructs introduced in the structured activities during their independent project work. Results from the conjecture-based analyses were encouraging and suggest that participants were more successful in applying constructs from the structured activities during their independent project work; unlike the previous design cycle, most participants successfully completed most or all of the structured tasks. Moreover, most participants (n=9) created a full multimodal text relative to their original narrative goals. Providing additional time to work on the independent project—which constitutes most of the conjecture-based analyses—provided breathing room for participants to fully realise their intended outcomes. Participants reported feeling satisfied with their performance and motivated to continue creating narrative-based projects through online tools.

Some results in the conjecture-based analyses were unanticipated though welcome. First, the temporal sequencing of narratives, coupled with participants' clear understanding of their individual story sequencing, provided an opportunity to explore programme execution in storytelling. It was initially conjectured that participants—through iterative testing—would ensure their programs could be executed multiple times with the same outcome; in other words, that learners would ensure each sprite would reset to its original position/size/attributes. This emerged from a discussion on the desire to use *motion* blocks to set each narrative element at the beginning of the programme. As participants were encouraged to routinely test their stories for coherence, they used *motion* blocks (e.g., *go to X Y*) and looks blocks (e.g., *set size to 100%)* to reset their projects for repeat execution. This was necessary as, through multiple programme executions, participants were able to see how a reader would be unable to know the nature of the original programme state (i.e., where sprites should begin on screen). One unanticipated observation concerned some participants executing scripts using specific key inputs. An unintended consequence for a reader meant their ability to execute the program correctly was limited to their understanding of the running order of the text.

Second, comparing conjectures across both learning strands highlighted the centrality of event coordination and structure through cohesive and structural devices. Given the temporal nature of participants' stories, where certain multimodal features are required at a certain time, programme coordination was crucial to telling their stories. To achieve this, participants adopted multiple event coordination strategies to structure and coordinate multimodal features. These took two different forms: (i) all participants used *wait* blocks (e.g., in-program events) to coordinate their narrative structure and, in the *digital leaders* group; (ii) broadcast features (e.g., in-program messages) were used (n=5) to coordinate parallel scripts. Moreover, all participants made use of intra- and inter-script delays (*wait* blocks) to ensure all blocks were executed at the appropriate time. While most learners' programs could be executed automatically, 2/10 featured scripts bound to specific key inputs (e.g., 'when x is pressed') requiring the eventual reader to know the exact key sequence to run the story.

Though this became a general point of discussion for the group, it was not corrected in the final projects.

Third, the multimodal dimensions of participants' projects framed how and what learners programmed. During the independent composition activity, participants were asked to consider these dimensions particularly as they relate to the programming features they adopted. When translating their initial story plans from paper to screen, participants were encouraged to employ more multimodal text features as, in turn, this contributed to more sophisticated programming structures. In one example, it was noted that many participants (n=6) used loops to illustrate motion in their stories. This was usually coupled with costume change blocks to further simulate motion or change within a character. For instance, one participant, Eric, used loops to automate characters *shrinking* into the distance (see Figure 20). In another example, this effect was used by two participants to simulate a character growing up before being shrunk down by a wizard. In both cases, this led to a discussion on the benefits of showing a certain story element over telling it in the same instance. It was also recognised that some participants (n=4) chose not to use such an effect (or were unable to do so) and instead opted for text-based resources.

6.5 Research questions

The main goal of this study has been to consider how learners with varying experiences of programming and storytelling goals can be better supported during multimodal text construction activities (**RQ2**). This involved a redesign and reimplementation of the proposed curriculum design.

Based on findings from Study 1, three design iterations to the intervention were made, including (i) differentiated scaffolding measures for learners with varying experiences of computing and multimodal storytelling; (ii) open-ended storytelling tasks and; (iii) contextualised programming features added to the instructional sequence. To do this, the additional of these scaffolding measures was tested to see whether and how it could better support learners with varying goals and experiences (**SQ3**). Observation fieldnotes indicated

participants made regular use of visual prompts related to each activity; moreover, when participants encountered difficulties during structured tasks, these prompts provided an unobtrusive reminder of a particular goal and the necessary components to achieve that goal. These prompts were also useful in helping participants decompose larger narrative-based tasks (e.g., programme a narrative event) into smaller manageable tasks (e.g., programme a character to move in multiple directions, then a backdrop change, and finally add character reactions through text blocks or other means). Moreover, where additional support was required—or the activity prompts failed to accommodate participants' design work during open-ended tasks—participants were given sample blocks of code and encouraged to manipulate code and test its output until the desired outcome is reached relative to their individual narrative goals. This form of support helped participants apply regular debugging by iteratively testing their evolving projects and making and testing adjustments to see if an error could be resolved independently. Screen recordings further evidence participants' frequent execution to test the output of their programmes. However, in cases where errors could not be resolved—largely in working with less experienced learners—facilitator input was necessary to overcome design challenges.

There are some caveats to these findings. While the integrated analyses point to areas where programming and storytelling were meaningfully integrated, some limitations are noted. Existing research on novice programmers, posits that younger learners tend to begin by writing "spaghetti code" (Lehrer et al., 1999), or long sequences of scripts that eventually accomplish a programming goal, but promote an understanding of programming structure as flat rather than hierarchical. Owing to the linear nature of participants' stories, in most cases, projects used a flat structure though some hierarchy was promoted through managed programme execution and in-programme messaging.

A second design modification intended to accommodate longer, more productive engagement with sample projects was adapting the instructional sequence to accommodate a greater diversity of learners; in particular, participants were encouraged to spend longer in sessions adding additional narrative events of their choosing in the hope that richer

opportunities for achieving programming and storytelling goals could be fostered through extended engagement with fewer texts. The results of the conjectured-based analyses indicate that participants were generally more successful in completing these activities. In particular, the addition of experience-based conjectures (e.g., using repeat loops to create animated sequences) provided more experienced participants with more complex and appropriate challenges.

A secondary goal was to consider how the modification of learning activities to include meaningful and relevant programming constructs could result in a more productive relationship between storytelling and programming goals (**SQ4**). The findings of the conjecture-based analyses in Study 1 highlighted multiple areas where potential for integration was high though this was not executed by most participants. In Study 2, participants' preferences for multimodal text features (e.g., animations, movement) were embedded in the instructional sequence to enrich and enhance their stories and provide a greater set of multimodal resources in which to create stories. These examples were embedded in model texts and routinely demonstrated through facilitator-led reflective discussions, focusing on the impact of certain multimodal design decisions on participants' projects. In this sense, it is conjectured that through context-led examples, participants will approach their independent work with a rich set of examples on which to construct their project. One notable finding in the conjecture-based analyses is the prevalence of conjectures relating to these additional modifications; for instance, *motion* blocks to simulate movement (n=9) were commonly found among participants, with repeat loops to create animated sequences (n=5), and *broadcast message* blocks to coordinate narrative elements (n=5) in the *digital leaders* group. This result contrasts with previous findings (Lehrer et al., 1999) though notably the proposed intervention is more structured than those detailed in the cited literature.

Like previous theorists (Tissenbaum et al., 2019), this thesis argues that the goals of computing education should focus on the application of computational ideas in the development of authentic products. In this study, participants were encouraged to consider how loops, parallelism and event-based scripts could be used to create multimodal effects and

apply this to tell more engaging visual stories. Likewise, visual programming provided learners with a rich set of representational resources to consider both what story to tell and how to do so. Integrating programming and storytelling in this way also demonstrated where complementary learning goals support one another at different points. For instance, using event-based (e.g., initialising) blocks functioned as a means to structure their stories episodically. That is, students were able to create and revise each section of their story separately. Moreover, multimodal effects (e.g., *motion* blocks) required students to consider the necessary programming features and affordances of such an effect to tell more complex, engaging texts. However, the

6.6 Design implications

In this section, proposals for refinements to iterate the curriculum design based on the analysis presented above are discussed. In looking toward Study 3, while further design modifications to improve on the pedagogical goals of the study are needed, the aims are structurally different and require a different approach. For instance, Study 3 implements the curriculum unit within regular classroom time and is facilitated by the classroom teacher. As noted previously, Lewis et al. (2020) argue that "defining which aspect… is most likely to make the project fail to make the desired impact because it fails to support stakeholder goals" (p. 1151) should serve as a guiding principle in this process. As such, the design modifications required are those that arguably support the stakeholders (teachers) in the given context.

6.6.1 Design prompts for distributed scaffolding

The third design principle, *scaffold student-generated construction* (**DP3**), embodies the argument that the curriculum unit should support learners to achieve learning goals through various scaffolding strategies. For instance, discussions with the classroom teacher revealed that half of participants had previously participated at an in-school *digital leaders* programme and consequently had more experience in designing and evaluating *Scratch* programmes. Conversely, half of the group had limited classroom experience of computing though their out-of-school experiences differed strongly. In Study 2, some of these scaffolding strategies were

adapted to suit learners with varying experiences in programming and multimodal storytelling in the group.

A discussion-based activity was added in Study 2 to discuss story ideas, including different genre conventions and story endings. The activity involved a demonstration of the storytelling process, including generating story ideas, and possible endings, structuring narrative episodes, and considering multimodal text effects. Embedding structured storytelling experiences provided participants with appropriate supports to prepare and discuss their intended narrative goals with the facilitator and teacher in the room. By discussing the 'critical features' (Wood et al., 1976) of the storytelling process, participants were found to have written detailed storyboards and were generally more successful in achieving their narrative goals. While they also had additional time to complete the activity, few participants commented on the challenge of generating story ideas or resolving stories.

Composing a text multimodally in Scratch also required participants to negotiate multiple task demands, such as the multimodal resources to be employed, programming a narrative using code blocks, the demands of the target audience (or 'reader'), and navigating the user interface. The ability to support participants in negotiating these multiple and provide them with individualised feedback was partly enabled by working with a small group (n=10). The classroom teacher, who observed the lessons, noted this point and expressed concern in working with larger classroom groups (c. 27-32 students). One consequence of working with larger groups is that learners might overlook concepts and practices that are central to activities.

In thinking ahead to Study 3, where the facilitator is supporting an entire classroom of approx. 30 students, providing targeted support becomes a challenge. In Study 2, this was not the case. As scaffolding depends both on the facilitator's knowledge of the domain and their knowledge of the learner during interactions, supporting whole class groups in complex tasks requires further scaffolding measures. For instance, guiding learners toward critical features within constructive activities becomes more difficult as the demand of supporting

larger groups is felt. In one sense, the multiple scaffolding measures detailed in this study, such as teacher facilitation, discussion-based activities and paper-based materials arguably provide "support that is greater than the sum of the constituents" (Tabak, 2004, p. 308). The potential challenges associated with enacting these measures across the curriculum unit in Study 3—that is, in a formal classroom setting—might be mitigated through these means though a discussion with participant teachers is needed.

6.6.2 Adapt intervention for formal curricular use

The two iterations of the curriculum unit conducted so far have taken place in an after-school informal learning environment. Moreover, whereas the first two cycles of research explored the design of learning activities to support integrating programming and storytelling at primary (ages 7-11) level, the focus shifts to the implementation of the curriculum unit by in-service teachers and how their perceptions on pedagogy, programming as a curriculum activity, and the role of multimodality in the literacy classroom could influence the co-design and implementation of these activities. To satisfy the original aims identified in the literature review (see Chapter 2), the intervention needed to be trialled in a more formal capacity. It is hoped this direction will stimulate further research concerning the value of multimodal composition activities in developing storytelling for the literacy classroom and programming to the computing classroom.

The next iteration in this research project involves working with teachers to adapt the curriculum unit to fit the needs of local contexts. This means collaborating with teachers on facilitating the intervention within the existing curriculum. It also meant working with teachers to adapt the intervention to accommodate the local context and its individual demands; for instance, the current intervention has been facilitated in a small-group teaching capacity. In formal settings during timetabled lessons, the class size increases to c. 28-30 students per class. This presents not only has practical and logistical implications, in terms of being able to facilitate activities at this size, but also pedagogical and theoretical implications, in terms of the viability and appropriateness of activities given the changing context.

One of the key design principles when designing the learning environment for the intervention is to consider how learners can be scaffolded through the process (**DP3**). Operationalising this principle for a formal school setting required the way scaffolding strategies are embedded across the learning environment to be carefully considered. To date, this has been distributed in multiple forms, from the structure and selection of activities, differentiated scaffolding prompts for different learners, and generally through individual interactions. With a larger group and in the hands of a different facilitator, the recipient of scaffolding measures changes as support is required for both students and teachers whose experience of computing is limited. In this sense, preparations for the intervention could include support sessions for teachers to better support teachers in facilitating these activities.

CHAPTER 7

Study 3

7.1 Introduction

Two designed cycles have been conducted to date (Studies 1-2). These sought to design tasks in multimodal composition and consider how students' engagement in these can foster specific programming and storytelling practices. Previous design cycles have sought to consider the design of the intervention for informal educational settings (after-school computing clubs). In the literature review, however, the timetabling of Computing was noted to be an issue with much of this research taking place in afterschool clubs and extra-curricular activities (Larke, 2019). For this, the role of teachers in enacting such curricula is central. The focus of Study 3, therefore, was to co-design the intervention with collaborative partners (primary school teachers) for formal educational settings (regular classroom practice) and consider this could be valuable for all stakeholders.

7.1.1 Study rationale

Within the DBR tradition, research-practice partnerships are favoured for their commitment to improving practice (O'Neill, 2016). One such partnership is a co-design model with teachers. Our collaborative partnership involved a co-design effort between two participant teachers in two schools and the lead researcher. Co-designing the intervention involved the participation of practitioners as active curriculum co-designers in the context of a curriculum design and testing (Penuel et al., 2007). To date, Study 1 and Study 2 have been conducted with practitioners as collaborators and gatekeepers; however, these have taken place in after-school settings with little impact on classroom practices. Indeed, for research to address an educational need or realise a new teaching or learning innovation, collaborative partnerships are regarded for their greater potential to address implementation issues (Penuel et al., 2018).

By employing teachers as co-designers—and not just curriculum 'executors'—the aim is to ensure activities are effectively and sustainably implemented. Cviko (2014) define

three roles taken by teachers during curriculum design and implementation: (i) executor-only, or the implementation of ready-made activities; (ii) re-designer, or the collaborative redesign of existing activities, and; (iii) co-designer, or the collaborative design of new activities. Within the DBR tradition, co-design is favoured for its commitment to improving practice, involving practitioners for their expertise in adapting designs for their local context and ensuring greater feasibility (Severance et al., 2016). In the current context, due to several factors (teacher commitment, curriculum constraints) it was decided to commit teachers to the project as re-designers adapting the existing intervention design for their own purposes and contexts. Bakker (2018) argues that this process can yield a set of negotiable targets are established as one of the outcomes of design-based research.

7.1.2 Study aims

The aim of the third study, then, was to adapt the previous curriculum unit enacted in the second cycle (Study 2). This iteration considers how the current intervention design could be adapted by in-service primary school teachers to complement their existing curricula. Reinking and Bradley (2007, pp. 73–78), in their organisational framework on design (or formative) experiments in language and literacy search, outline several areas of inquiry to explicitly conceptualise, conduct and report design-based research:

1. What is the pedagogical goal to be investigated, why is that goal valued and important, and what theory and previous empirical work speak to accomplishing that goal instructionally?
2. What intervention, consistent with a guiding theory, has the potential to achieve the pedagogical goal and why?
3. What factors enhance or inhibit the effectiveness, efficiency, and appeal of the intervention in regard to achieving the set pedagogical goal?
4. How can the intervention be modified to achieve the pedagogical goal more effectively and efficiently and in a way that is appealing and engaging to all stakeholders?

5. What unanticipated positive and negative effects does the intervention produce?
6. Has the instructional environment changed as a result of the intervention?

Several of these research aims (questions 1-3) have been explored in previous studies, albeit in an informal learning environment. For the intervention to have value and appeal to practitioners, one of the principal goals of DBR, it needs to be considered against the backdrop of formal classroom use. To that end, the overarching research question in this study is:

- Research question 3 (**RQ3**): How can multimodal text construction that support students' programming and storytelling goals be embedded within regular classroom instruction?

Given the broad nature of this question, it seemed necessary to divide this into more explicit research questions:

- Sub-question 5 (**SQ5**): What are the experiences of participant students and teachers taking part in the intervention during their normal classroom time?
- Sub-question 6 (**SQ6**): What perceived factors enhance or inhibit the aims of the intervention?
- Sub-question 7 (**SQ7**): How can the curriculum unit be adapted to the needs of stakeholders (teachers and students)?

These sub-questions, adapted from Reinking and Bradley (2007), serve to evaluate how the design as implemented so far can be suitably trialled to test its merit in an authentic context. They also serve to generate exploring processes, products and artefacts that yield insights that are "usable, actionable, and adoptable" (Bannan-Ritland, 2003, p. 24) for stakeholders.

7.2 Method

7.2.1 Setting and participants

The study took place in a primary school in the Nottingham city area. The school is mixed gender with 245 3-11-year-old students, respectively. One participant teacher facilitated the

six-hour intervention with Year 5 students (*n*=28) (aged 9-10) in the school computer room (see Figure 22).

The participant teacher, Mr Thomson (pseudonym), served a dual role as the school's computing coordinator and special educational needs (SEN) teaching assistant. He assumed the role of computing (previously ICT) coordinator since the introduction of the revised Computing curriculum in 2014. The role primarily involves facilitating professional development in Computing with the core teaching staff, developing curriculum materials for each class group, and supporting teachers to implement the new computing requirements in bi-weekly lessons. As part of these responsibilities, Mr Thomson also frequently supported teachers through lesson observations, co-facilitating lessons, and organising evening CPD workshops.



Figure 22: Intervention site (Study 3)

Mr Thomson's additional duties include leading the school's digital leaders programme, a scheme designed to encourage older (i.e., Year 4-6, 8-11 year old) students to support teachers and younger students' computing skills through after-school clubs, assisting in lessons, and technology showcases. As facilitator, Mr Thomson had worked with the group (i.e., the Year 5 class) for several years, both leading computing sessions and supporting the classroom teacher to implement the school's computing curriculum.

7.2.2 Design iterations

Based on the findings of the previous two studies (1-2), modifications are discussed in terms of the design principles (see Chapter 3). The proposed modifications for this study are discussed in terms of these how they instantiate each principle differently as the unit was adapted to fit the local context and stakeholder goals. Prior to the study, the proposed activities were presented to the computing coordinator to adapt to the appropriate context. These discussions centred on the design principles enacted in the curriculum unit design (e.g., the level of support provided, the instructional sequence). The design meetings yielded some necessary design modifications. These are discussed in terms of how they were operationalised through activities, scaffolding measures, and pedagogical goals (**DP1-DP3**); the iterative work of **DP4** remains unchanged. As the process was co-design where additional stakeholder (i.e., teacher) goals were taken into account, these were also modified to suit these needs. Further, findings from Study 1 and 2 were used to refine the design of the curriculum unit for Study 3.

7.2.2.1 Build on learners' prior knowledge and skills (DP1)

The first design principle (**DP1**) centres on facilitate students through a cumulative sequence of learning activities, beginning with introducing multimodal storytelling through sample texts, building on their existing experiences of programming and (multimodal) storytelling (where possible), and culminating in a final independent project. In previous cycles, this principle was instantiated by engaging learners in short, structured activities—with largely fixed outcomes—to a longer independent activity—with flexible outcomes. This principle resonated with the participant teacher during early design meetings as a necessary design component in supporting students' learning and therefore its embodied features remained unchanged.

7.2.2.2 Engage learners in constructive activities (DP2)

The second design principle (**DP2**) centres on providing learners with ample opportunities to construct personally meaningful texts. In previous cycles, students were encouraged to decide

on their own genre and inspiration for their independent text. Results from Study 2—where costume loops were added—indicated that providing learners with open-ended tasks and opportunities to organically apply this concept for different purposes (e.g., demonstrating a man walking). Moreover, removing the closed nature of tasks—those that seek the same outcome for all participants—was found to encourage learners to produce longer, richer and more programmatically complex stories.

In Study 3, however, a concern for the variation in student projects during open-ended tasks was raised in an early design meeting. The participant teacher reported feeling overwhelmed at the idea of providing feedback and support to students with varying project and individual needs. In practice, the participant teacher compensated for this by nominating multiple genres to demonstrate using sample text (see Appendix 3). This design decision was chosen to address the teacher's perceived ability to provide spontaneous feedback to multiple projects. It was also an attempt to satisfy the desire to allow students to draw on existing narrative and representational resources (e.g., images, text and sounds) to support the authoring of 'original' texts. Though this principle worked well in previous cycles, this was partly achievable due to the small-scale nature of each intervention group (10-12 participants). An attempt was made to maintain learner agency by providing multiple structured opportunities for participants to generate their own stories within a defined set of text examples. The results of this decision are discussed further in the Findings section (3).

7.2.2.3 Scaffolding student-generated construction (DP3)

The third design principle (**DP3**) focuses on scaffolding learning through multiple means as the learning strands of programming and multimodal storytelling were largely unfamiliar to participant students in previous cycles. In Studies 1-2, some of the scaffolding measures employed did not serve participants well. For instance, participants reported difficulties in generating story ideas and were not observed using the provided storyboard to plan their texts. It was conjectured that few critical features (e.g., multimodal text effects, narrative episodes) were prompted using these materials and that a redesign would be needed. Based on these

findings, one suggestion noted by the classroom teacher was the need for explicit criteria to guide students in the process of constructing multimodal stories. Those suggested including creating setting, atmosphere and tension using multimodal effects, showing character's point of view using dialogue or text effects, or justifying how meaning is conveyed through images and words. These were highlighted by the participant teacher as necessary not only to scaffold students' learning but to support teaching as evaluative criteria to be used during observations, giving feedback and instructions. Thus, although the design principle remains as before, its implementation is adjusted based upon the teacher's understanding of whole class teaching.

Modifying the design prompts provided could guide learners by asking specific questions about the features of their texts and suggesting design considerations to reflect on. These prompts would be explicitly tied to the corresponding conjectures, centring both on the process *and* product of composing multimodally.

A second storyboard (see Figure 23) was co-designed with the participant teacher to problematise specific aspects of learners' work that might otherwise be take for granted. In terms of storytelling, prompts included questions asking participants to think about the beginning of their story, what happens in successive episode(s), and how their stories end. Additional prompts were provided for learners to consider what multimodal text effect(s) would be employed in their story and what effect(s) this would have on meaning.

Employing a structured storyboard also required that the process of planning a text be modelled by the facilitator. Using a sample text (discussed below), the facilitator would model creating a story in three parts and model the process of deciding on explicit multimodal text effects to contribute to meaning and enhance his story for the reader. It was conjectured that this solution would support participants to employ similar narrative structures and texts effects in their projects.

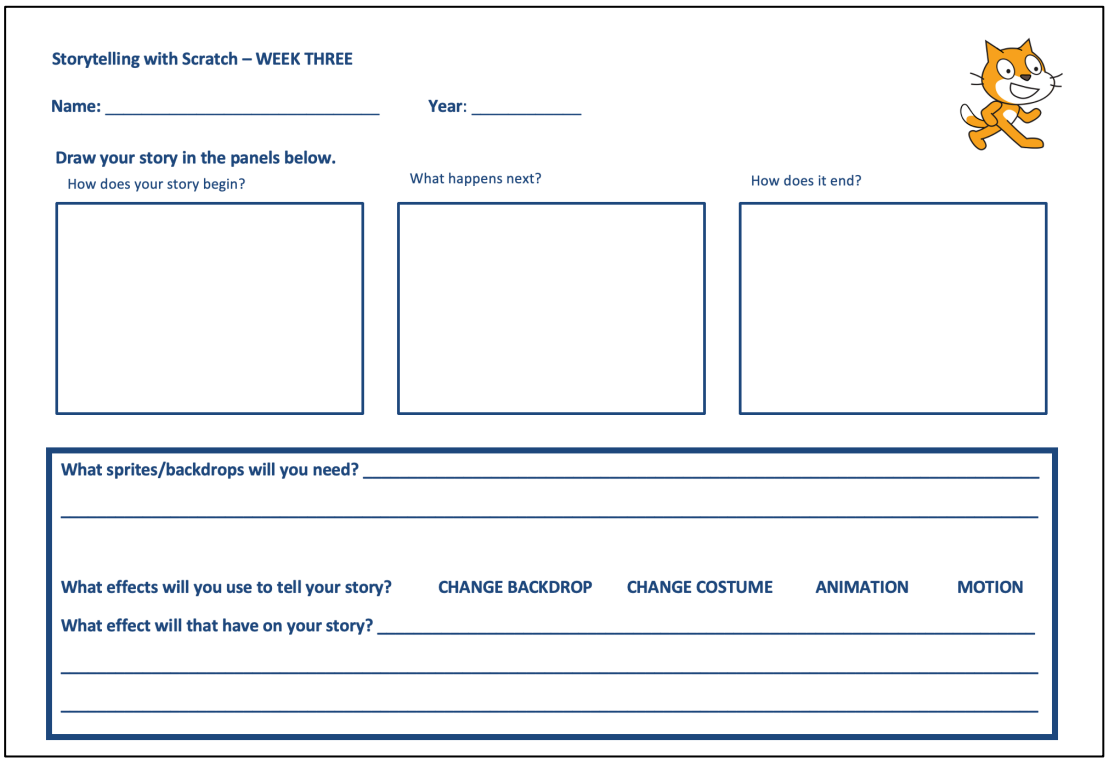


Figure 23: Revised storyboard (Study 3)

The advent of facilitating multimodal composition activities provided the participant teacher with some concerns. As MMC does not feature explicitly in the 2014 English curriculum, instead existing as an implicit feature of the computing curriculum (within the digital literacy strand), it was unsurprising that the teacher was both unfamiliar and uncomfortable in leading the session without support materials. Therefore, some of the scaffolding materials were designed to support the facilitator during sessions. These materials were co-designed over the course of multiple design meetings and are detailed in Appendix 3. The first set of materials is a detailed lesson plan for each teaching session. These outline the step-by-step processes as well as the instructional moves in leading each session. The lesson plans also included specific language and criteria mutually decided upon to facilitate discussion on the multimodal aspects of students' projects and to encourage students to reflect on and justify their design choices. As multimodal composition is concerned with encouraging students to look at texts holistically, a detailed sample multimodal text (see Appendix 3) was co-designed to provide the facilitator with a model in which to both identify multimodal features and use these in practice.

Providing multiple and distributed scaffolding measures through teacher facilitation, paper-based materials, and technological supports might prove more supportive of participants' individual learning goals by producing "support that is greater than the sum of the constituents" (Tabak, 2004, p. 308). In addition, these measures were designed to address potential logistical challenges associated with implementing the curriculum unit in a formal classroom setting.

7.2.3 Intervention procedure

7.2.3.1 Original research procedure

Before the impact of the COVID-19 pandemic, Study 3 included four stages: (i) recruitment; (ii) preparation; (iii) data generation, and; (iv) data analysis. The intention was to recruit primary school teachers to participate as facilitators in trialling the intervention. The process would have entailed co-designing the intervention by discussing the aims and outcomes of the intervention, noting previous successes and failures, and how the intervention could be adapted to fit their needs. An initial design meeting was proposed where the lead researcher and teachers discussed the learning activities and considered how they could be adapted for individual teaching goals—a contextualised literacy project based on previous work. The adapted intervention would then have been trialled by each teacher and subsequently redesigned and reconfigured as each lesson progressed.

Given the challenges of supporting teachers to embed and promote digital technologies in the classroom (Hall et al., 2013), it was assumed that participant teachers would require an understanding of the curriculum requirements and a familiarity with the Scratch programming interface so that the intervention could run successfully. To accommodate this, one of the activities of the partnership between the lead researcher and school-based partners involved a CPD session focused on areas of the Computing curriculum—specifically the programming component—and how this can be realised through visual programming. This procedure is detailed in Table 25.

Table 25: Original research procedure

Phase	Procedure	Products	Timeframe
Recruitment	Recruit 2-3 local primary school teachers with an interest in developing their computing expertise Organise CPD session on the computing curriculum (esp. programming) Discuss aims and outcomes of the study with participant teachers Receive informed participant consent via consent forms	2-3 participant teachers Informed participant consent	1-2 months
Preparation	Organise 2 meetings (~1hr each) with each participant teacher to discuss the intervention and conduct a needs assessment Provide samples of previous student work and lesson plans and discuss how lessons could be adapted to suit local needs and context Researcher and teachers co-design a series of lessons focused on integrating computing and literacy goals relevant to each class context Provide support to participant teachers where needed to deliver the intervention	Intervention resources (i.e., lesson plans) Audio recordings of preparatory design meetings	2 weeks
Data generation	Organise design meetings with participant teachers after each observation (~45min each) Conduct 6 observations in regular classroom setting (computer room) using an observation protocol (~1hr each) Collect student-generated work and discuss during design meetings	Recordings of 6 meetings Observation data of 6 lessons Student-generated work	6-7 weeks
Data analysis	Apply deductive codes relevant to research aims based on previous literature (e.g., feasibility of intervention implementation, barriers to implementation, and role of teacher) Analyse dataset and write up findings	Theoretical and practical products related to int. implementation	4-6 weeks

7.2.3.2 Revised research procedure

With the advent of COVID-19 forcing universities and schools to close, the original research plans needed to be amended to mitigate the impact of national lockdown its associated challenges in conducting research. The original plan involving recruiting multiple teachers to trial the intervention and interviewing them on a weekly basis was untenable as the threat of forced school closures presented several challenges: (i) the time needed to recruit multiple teachers, collaborate on a series of lessons, and conduct interviews over a six-week period; (ii) the risk of empirical work being halted for an extended period, and; (iii) the lack of availability of teachers to commit to collaborating on a study remotely with significant disruption to their personal and professional lives. At the beginning of lockdown, these issues with discussed in

remote online meetings with the lead researcher, his supervisors, and research participants. The revised research plan considered these challenges and presented an alternative procedure that would allow for classroom observations and participant reflections in a reduced timeframe; this revised procedure is outlined in Table 27.

The revised procedure began by contracting the six-week period of classroom observations into a two-week period. These lessons took place in the afternoon and were facilitated by the lead participant teacher and observed by the classroom teacher and me. Before the first session, a series of meetings were scheduled with the participants teacher, Mr Thomas, to discuss the project and our respective goals, the location of the research site, access to equipment, and timetabling.

Each of the six hourly sessions took place over the course of a two-week period. These were scheduled in the afternoon timetable slot (c. 1-3pm) and were based in the classroom and the school computing room (see Figure 22). Participants sat in a circle around the room, each with a computer/laptop, and a computer with an overhead projector for the teacher facilitating. The classroom teacher, Mr Thomson, introduced the project in the classroom before bringing participants into the computer room.

Table 26: Intervention activities

Session	Activity	Proposed activities
A	A1	Participants are guided to initialise a sprite and use *say* blocks to illustrate speech
	A2	Participants are guided to initialise a second sprite and create an authentic narrative context. They are asked to maintain readability by managed programme execution
B	B3	Participants are guided to debug an existing narrative project by successfully managing programme execution using *wait* blocks ~~or *broadcast message* blocks~~
	B4	Participants are guided to add additional narrative event(s) of their choosing to the sample text, employing text blocks, change backdrops, and manipulating sprite attributes (e.g., ~~size, graphic event,~~ position, visibility) to illustrate these events
C	C5	Participants are guided to debug an existing narrative project using *wait* blocks ~~or *broadcast message* blocks~~, defining and manipulating sprite attributes (e.g., position, ~~size,~~ visibility) and creating animated sequences using repeated loops. They are also asked to reset all of these attributes so that the programme outputs the same result automatically. Participants end by collectively drafting ideas for their independent projects and sourcing appropriate images to use in their projects.
DEF	DEF6	Participants apply a variety of programming concepts to create a 'complete' multimodal story—that is, one with all (or most of) the elements of a basic narrative structure along with select multimodal effects (see lesson plans)

Unlike previous studies, the group consisted of 28 students. As space was limited, participants sat at their computers and were asked to turn off their monitors during discussions or live demonstrations. Participants were generally encouraged to offer suggestions and share their progress during discussions though the size of the room meant that this wasn't entirely possible. As such, participant was mediated through voting. For instance, during live coding sessions, participants were provided with multiple prediction options and asked to raise their hand to guess the programme output would be. Participants were then asked to run a sample piece of code on their computers to test their prediction. The teacher facilitated each activity (see Table 26) and displayed scaffolding prompts on the overhead projector (see Figure 24) to support students. As students were younger and had less experience that those in previous studies, certain programming features were omitted (Table 26, see crossed-out sections). In the final three sessions, as participants worked on their independent project, Mr Thomas sought to identify common misconceptions or challenges participants faced and scheduled

small discussions throughout each session. These focused on common problems shared by participants and allowed them to discuss problems and share potential solutions.

As participant observer, I interacted with students though played no role in contributing to the outcome of the sessions. However, I worked with the school's *digital leader* students to prepare the equipment when needed. In my interacting with students, it was decided to not influence or provide overt guidance or suggestions as to do so would minimise the ecological validity of the study; that is, with an explicit role in facilitating, I would have exercised undue influence on the 'success' of the intervention compromising any ability to yield insights into a teacher's experiences of trialling the curriculum unit largely alone.

As the frequency of lessons were intensified, the weekly design meetings were shortened to accommodate the teacher's limited availability. The purpose of the revised meetings was therefore to address any issues the participant teacher faced when facilitating the lessons. The original purpose for these meetings—to reflect on the successes and challenges of facilitating the intervention—was revisited through a series of remote, online post-intervention reflective interviews and are detailed further under Data generation (see 7.2.4).

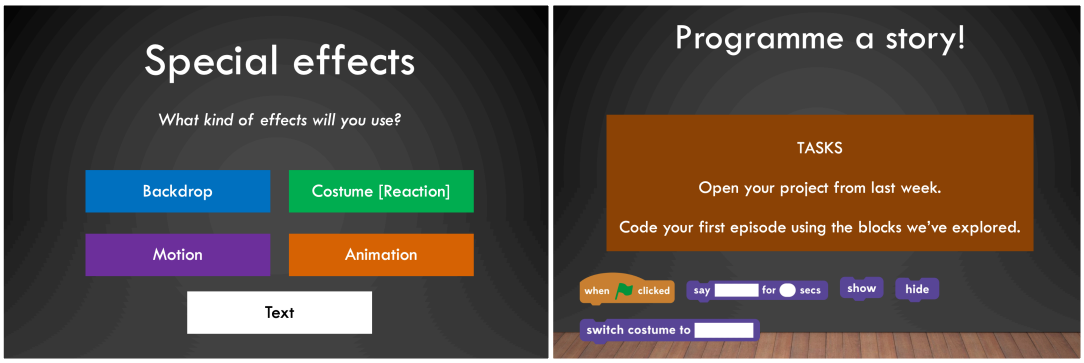


Figure 24: Scaffolding prompts (Study 3)

The need for additional programming time to accommodate the independent project was similarly felt in this context.; As this was modified after Study 1 to ensure all participants achieved their respective narrative goals over three sessions (as opposed to two), it was carried

over to this study. However, challenges to timekeeping were noted during implementation; these are discussed in greater detail in the Findings section (7.4).

Table 27: Revised research procedure

Phase	Procedure	Products	Timeframe
Recruitment	Recruit local primary school teacher with an interest in professional development Discuss aims and outcomes of the study with participant teachers Receive informed participant consent via consent forms	Study participants Informed participant consent	2 weeks
Preparation	Organise 2 meetings (~1hr each) with each participant teacher to discuss the intervention and conduct a needs assessment Provide samples of previous student work and lesson plans and discuss how lessons could be adapted to suit local needs and context Co-design a series of lessons focused on integrating computing and literacy goals relevant to each class context Provide support to participant teachers where needed to deliver the intervention	Intervention resources (i.e., lesson plans) Audio recordings of preparatory design meetings	1 week
Data generation	Organise design meetings with participant teachers after each observation (~45min each) Conduct 6 observations in regular classroom setting/computer room (~1hr each) Collect student-generated work and discuss during design meetings Conduct 6 post-intervention remote interviews	Recordings of 6 meetings Observation data of 6 lessons Student-generated work Audio recordings of remote interviews	2 weeks (observe.), 6 weeks (remote interviews)
Data analysis	Apply deductive codes relevant to research aims based on previous literature (e.g., feasibility of intervention implementation, barriers to implementation, and role of teacher) Analyse dataset and write up findings	Theoretical and practical products related to int. implementation	4-6 weeks

7.2.4 Data generation

Data generated during the study consisted of a total of 6 hour-long observations of 1 teacher during regular classroom time followed by post-intervention interviews. The participant teacher taught a redesigned unit of 6 lessons (a total of 6 hours) during afternoon computing lessons. The researcher acted as a participant-observer, recording fieldnotes and observing student participants and their interactions with the classroom teacher. Data generated during these observations included field notes, and student work. These data sources were then used

to guide the analysis of the intervention, how teachers came to adapt and enact the curriculum to fit their needs, and to what extent the intervention supported their teaching goals.

The current study used some of the data generation methods employed in Studies 1-2. For instance, data generated during the three-week period included: (i) participant-generated materials, including student plans, and project work; (ii) audio recordings of the interactions between the classroom teacher and participants and; (iii) observational field notes generated during and after the intervention. As in previous studies, data on participants' previous experiences in computing and multimodal composition was gathered through interviews with the classroom teacher and computing coordinator.

7.2.4.1 Observation field notes

Observations were undertaken to record the participant teacher's experience in facilitating lessons, and to ground design discussions in observations made by the researcher. The group was observed and followed between the classroom and computing room where most of the activities took place. These observations also informed post-session design meetings and grounded these discussions in matters that arose during the lessons. These notes detailed a "thick description" (Geertz, 1973) of aspects of the learning environment (e.g., seating plan, location of devices, teacher's use of technology). Observations of each lesson allowed students and their teacher to be observed to consider how this adaptation is realised to suit the local context and demands. Observation data were also used as a triangulation method to support or refute observations made during the reflective interviews. As both the lead research and participant teacher were present during each lesson, triangulating participant observations with researcher-generated fieldnotes functioned as a form of 'member' check (Lincoln & Guba, 1986), eliciting responses (i.e., interviews) from a key participant and corroborating these observations with related empirical data (classroom observation fieldnotes). Finally, these observations were used to inform discussions in weekly design meetings and post-intervention reflective discussions. These notes were reviewed immediately following each session and

used as artefacts to frame meetings between the participant and researcher in relevant observations and points of discussion.

7.2.4.2 Artefact-based interviews

Artefact-based interviews with the participant teacher were conducted after each teaching session served to both structure design meetings and discuss the current lesson and plan for subsequent lessons. These 'design discussions' focused on experiences of trialling the intervention, challenges, and whether further adaptations or solutions were needed. Analysed student-generated artefacts (e.g., project work) were used as a stimulus to guide discussions and provide focus to discussions. These were also used to inform any modifications to the intervention design and resources (e.g., scaffolding materials, lesson plans). Audio recordings of these interviews/meetings were also made for transcription and analysis. As these interviews were conducted remotely, artefacts (student project work, lesson plans and resources) were shared digitally and referred to throughout the interviews; additionally, screen sharing was employed to reference artefacts where necessary.

Interviews with the participant teacher were conducted before, during and after they facilitated the intervention. Due to the disruption of COVID-19, intervention feedback sessions were conducted partly face-to-face and then, following advice from the UK government and School of Education Research Ethics Committee, moved to a remote format; these follow-up interviews took place using Microsoft Teams. The first pre-intervention interview questions explored the participant teacher's existing practices and perceptions as they relate to the intervention (see Table 28). As the teacher was the school computing coordinator and classroom assistant, questions were framed to suit his professional context and teaching demands. As such, interviews sought to elicit his professional judgement on the potential and value of the intervention.

Weekly feedback meetings between the participant teacher and researcher were then held after each lesson. These were scheduled immediately after each lesson to discuss issues that arose during that time and to plan for future sessions. As these sessions took place

immediately before the school closures caused by COVID-19, where the intervention was scheduled to be completed across a two-week period, the meetings were shorter than originally planned and focused on planning for subsequent lessons based on observations made during the teaching session.

Table 28: Pre-intervention interview questions

Topic	Question
Teaching background (SQ5)	• Tell me about your teaching responsibilities. **Prompt**: What do they entail? • How often do you teach computing/literacy? • What motivated you to take part in this study? **Prompt**: Have you used project-based curricula before?
Existing practices (SQ5)	• What challenges, if any, have you faced in facilitating the computing curriculum to date? **Prompt**: Are there any extrinsic (e.g., resource/school/timetable issues) challenges? **Prompt**: How confident are you in teaching the 'computer science' component in computing? **Prompt**: What tools/strategies/resources have you used to achieve these aims? • To what extent have you used ICT (e.g., iPads/apps/PCs) across the curriculum (e.g., literacy)? **Prompt**: If so, what kinds of texts do students explore? **Prompt**: Have you explored multimodal (i.e., more than words) storytelling with students before?
Perceptions (SQ6)	• How best should programming for computing be taught? **Prompt**: What approach would you normally take? • How best should storytelling for literacy be taught? **Prompt**: What approach would you normally take? • Do you see value in exploring an integrated approach to these areas? **Prompt**: If so/not, could you elaborate?
Intervention (SQ7)	• [Artefact: lesson plans] • How would you normally teach the lesson aims in this project? Prompt: What activities would you plan? Prompt: What teaching strategies would you use? • Is there anything that you would change about the lesson aims? **Prompt**: If so, how could they better align with your teaching needs? / If not, why?

Table 29: Intervention feedback questions

Topic	Question
Previous lesson	• How did you find that lesson? • Did you note any challenges in facilitating the lesson as prepared? **Prompt**: If so, how did you adapt them, if at all? Why? **Prompt**: If not, how would you address this in subsequent lessons? • Is there anything that you would do differently now, and why? **Prompt**: How would you adapt subsequent lessons/lesson plans?

As the timetable for the intervention sessions and subsequent feedback sessions was rescheduled to minimise disruption caused by COVID-19, two sets of retrospective interviews were organised to elicit (i) responses based on the participant teacher's experiences in facilitating the intervention (ii) his reflections on the factors that enhanced or inhibited the pedagogical goals. The first set of post-intervention interviews centred on using a sample of student work collected during each teaching session as prompts for discussion (see Table 30). Sample projects were chosen based on the results when coded using the existing task-oriented analysis (see Study 1 and 2). Sample projects with differing programming and storytelling task completion rates were chosen as representative of the classroom (i.e., projects with high programming and storytelling, high programming and low storytelling, low programming and high storytelling and low programming and low storytelling conjecture rates). This allowed the participant teacher and researcher to reflect more deeply on their experience facilitating each session.

Table 30: Post-intervention artefact-based interviews

Topic	Question
General	• [Artefact: lesson plans/observation notes] • How satisfied were you with the lesson? **Prompt**: What worked? / What didn't work? • Did anything unexpected happen during the lesson? **Prompt**: How did you respond?
Programming	• [Artefact: sample of student projects] • To what extent do you feel the programming aims were met? **Prompt**: If so, how? / If no, why not? **Prompt**: How might you they have been better achieved? • What issues arose, if any, that affected students during these tasks? **Prompt**: If so, how did you address this?
Storytelling	• [Artefact: sample of student projects] • To what extent do you feel the storytelling aims were met? **Prompt**: If so, how? / If no, why not? **Prompt**: How might you they have been better achieved? • What issues arose, if any, that affected students during these tasks? **Prompt**: If so, how did you address this?
Planning	• [Artefact: lesson plans] • Were you happy with the changes made to the lesson plans? **Prompt**: Why? / Why not? • Is there anything you would have done differently? **Prompt**: Why? / Why not?

A final end-of-study summative interview focused on the teacher's experiences in planning and facilitating the intervention and his professional recommendations for future efforts (see Table 4). As the intention for the original study was to collaborate with the computing coordinator and classroom teachers to iterate on the intervention design, the summative interview was designed to focus on the changes the participant teacher had made to the intervention design and recommendations based on his experiencing of facilitating.

Table 31: End-of-study interview

Topic	Question
General **(SQ5)**	• Why did you take part in this study? • What were your expectations of the experience? • What are your thoughts on the intervention (e.g., lesson plans, aims, materials, scope)? **Prompt**: Was it enjoyable/challenging/useful? How?
Usability **(SQ6)**	• How did you find your overall experience of planning and facilitating the intervention? **Prompt:** What were some of the successes/challenges you noted throughout this experience? • How do you think students experienced the intervention? **Prompt:** What successes/challenges did you find they had experienced? **Prompt:** What changes would you make to better support them, if any? • How confident would you feel running the intervention independently? • How do you think other teachers in your school would find the intervention? **Prompt:** What challenges might they face, if any? **Prompt:** Did you face any of these challenges yourself? **Prompt:** How might you adapt the intervention to better support them, if at all?
Usefulness **(SQ7)**	• How might the intervention support your teaching in the future, if at all? **Prompt**: To what extent might it support others teachers in your school delivering the aims of the CS/literacy curriculum? **Prompt**: Do you see any benefits in taking an integrated approach to programming and storytelling? **Prompt**: Conversely, do you see any trade-offs in taking such an approach? • Would you consider using the intervention in your future teaching, if at all? **Prompt**: If so/not, why do you think that?
Final reflections **(SQ5)**	• Is there anything else you wanted to talk about in relation to the intervention or your experience?

7.2.5 Data analysis

Data generated was analysed in multiple ways. Data including observation notes and student work were used to inform design discussions on a weekly basis. In this section, both analytical strategies are briefly discussed.

7.2.5.1 Interview coding

Interviews with the participant teacher, conducted before, during and after the intervention were recorded, transcribed, coded, and analysed using an iterative coding scheme, adopting features of both a deductive and inductive approach. The analysis was shaped by the study's particular context, the impact of COVID-19 on data generation, and relevant literature on design implementations and evaluations. Drawing on salient themes from studies focused on teacher implementation of designed interventions (e.g., Reinking & Bradley, 2007) and those that relate to computing (Sentance et al., 2019; Sentance & Csizmadia, 2017), the coding scheme was iteratively developed using a combination of high-level categories grounded in the research goals (experiences, implementation factors, adaptations and future use) and additional themes yielded from the dataset. The coding scheme focused on the experiences of the participant teacher, the affordances (or enhancing) and limitations (or inhibiting) factors of the intervention as perceived by the participant teacher and lead researcher, and what adaptations of the intervention could inform the design and related theory (see Table 32).

However, as the nature of the intervention and its implementation were unique—at least in terms of applying a pre-determined coding scheme—an inductive coding process constituted a second pass of the dataset. The rationale for this decision was to consider how extant literature could inform the analysis in this study while remaining flexible to emergent themes specific to this study. For instance, a prominent theme in the teacher interviews was intervention status where the participant teacher discussed the challenges of placing the intervention within the existing curriculum scheme and how cross-curricular learning activities—though highly regarded—were difficult to justify to senior management as either 'pure' computing or literacy. Thus, the analytical process yielded both inductive themes that emerged in the dataset as well as high-level categories dictated by the research goals. Tracy (2013) argues that iterative analytical strategies are increasingly favoured for their rigour, yet flexibility, enabling researchers to connect their individual research contexts with broader research frameworks and tools. Analysing teacher interviews also enabled the intervention to

be evaluated by a third party, the intended audience for the intervention (i.e., practicing teachers). Further, corroborating these accounts with field notes generated during classroom observations provided additional triangulation of perceptions during lessons.

Table 32: Coding scheme for artefact-based teacher interviews

Research aims	Codes	Description	Example
Experiences of participant students and teachers (**SQ5**)	Motivations	Motivations to participate in the project/facilitate lessons	"I was interested in knowing how to apply… computing"
	Experiences of facilitating	Experiences in facilitating the sessions	" I enjoyed thinking of… things that have affected my practice"
Perceived enhancing or inhibiting factors or strategies (**SQ6**)	Subject knowledge and skills	Subject knowledge and skills in computing, literacy, MMC, and their impact on teacher confidence	" I don't see myself as being a "coder", I see myself as a learner"
	Time management	Managing time within lessons/across the curriculum	" There was quite a lot to get through in one lesson"
	Differentiation	Differentiating lessons for different students	"Computing is about… differentiating what you're learning [and also for how much access they've had on the computers."
	Teaching strategies	Comments related to strategies planned/enacted during lessons	"I think it's a challenge… in itself, how we get from, "Do this, do this, do this, do this" to… "here's this, go and have a play around a bit".
	Availability of resources	Material/technical resources available to teachers	" I wouldn't be able to […] create all the resources […] to deliver at that time"
	Curriculum timetable	Perceived value of integrated activities within the existing curriculum	"I think there's value in cross-curriculum learning […] and the application of using their computing skills and their literacy skills."
Intervention adaptation and future use **(SQ7)**	In-class adaptations	Examples of 'micro-cycle' iterations to intervention design	"[We added the annotated text because] I found it quite difficult to describe the difference between the effects that each one of those gives."
	Proposed adaptations	Examples of prospective change to intervention design	"[I want] the lessons more separate […] that I want to teach one computer lesson and then a literacy lesson […] ".
	Future use of intervention	Descriptions of future use of intervention in classroom practice	" I would give them more time on [their projects] as they were maybe cut short."

7.2.5.2 Task-oriented analyses

As the interviews conducted were based on artefacts generated during lessons, these needed to be analysed as in Studies 1-2 to provide a context in which to discuss how the lessons were proceeding. However, as the potential data to be generated and analysed had substantially

increased, a sample of the entire dataset was used to guide the artefact-based interviews. This sample was purposefully chosen to give a representation of varying experiences across the group, such as learners with previous programming experiences. The chosen participants' data were again coded using the task-oriented analytical framework (Dierdorp et al., 2011) from previous studies.

Unlike previous studies, the focus on the implementation included the teachers' experiences and how their teaching strategies affected the success of the intervention. In this sense, the analysis enables the success of the teaching and learning sequence to considered in authentic contexts and provide guidelines for future research and practice. Unanticipated events that emerged throughout are also discussed. Then, a retrospective analysis addressing what overall progress was made toward accomplishing the pedagogical goal and what modifications to the intervention the results suggest for future iterations.

7.3 Findings

This section is divided into three main areas, aligned with the research sub-questions. Analysis of teacher accounts of the intervention are reported here in terms of the guiding research questions. Respectively, these are concerned with issues of teachers' experiences, factors influencing the success of the intervention, and adaptation and future use. The overall research question is then discussed in the following section (4). Excerpts from interviews are provided to illustrate the identified themes.

7.3.1 Teacher's experiences

7.3.1.1 Motivations

The process of debriefing after the participant teacher's experiences of facilitating demonstrated his commitment to developing and extending his professional practice as a computing coordinator, teacher, and workshop facilitator. As both an SEN teacher and computing coordinator, Mr Thomson primarily facilitated interventions with multiple class groups and had an interest in seeing how computing and literacy could meaningfully be

integrated. Moreover, he emphasised how planning and facilitating *purposeful* activities was important to him:

> *MT: I was interested in knowing how that was going to be applied and taught alongside computing […] I expected the children to engage more with computing because… there was a purpose for it.*

Likewise, his commitment to designing and facilitating project-based activities was highlighted during initial design meetings where he discussed the use of projects as a means to apply concepts and practices in the pursuit of project-based outcomes. Previous 'schemes' of work facilitated in the school had differed greatly in that shorter schemes of work were more commonly used. This was partly due to the commitment required to facilitate anything more substantial:

> *MT: We do one-off lessons… or maybe take a couple of lessons on a project, but we don't really do a full-length project and actually it was quite nice to have like six weeks looking at building something up and making it something bigger.*

It further stemmed from the position of computing education in the school as an 'outlier' subject in the curriculum. Due to the subject not having an assessment component, it has often been side-lined in the lead-up to final exams (or SATs) for multiple class groups. As computing coordinator, Mr Thomson commented on how prior opportunities to implement cross-curricular or project-based activities were limited, particularly in computing:

> *MT: It tends to be pushed. The first thing that gets pushed out in the… school curriculum […] we're in a Catholic school [so] RE is always ahead of computing […] as well as a lot of the other subjects.*

As computing was observed by many in the school as an extra-curricular subject with little relevance to the core curricular areas (mathematics, science and literacy), little work had been previously undertaken to demonstrate its value or relevance across the wider curriculum. The desire to undertake an extended project-based curriculum was therefore positioned as a means

to justify how and why computing activities can feature in the wider curriculum. Previously, Mr Thomson described his use of pre-written project-based teaching resources (e.g., Code Club, Twinkl) though noted how adapting these materials was necessary for his own teaching needs. He concluded that co-designing this curriculum unit served to derive valuable resources and develop novel approaches to his future practice.

7.3.1.2 Experiences of facilitating

Mr Thomson reflected on his experience positively. Reflecting on the goals of collaborating on a scheme of work, developing resources, and observing his approach to teaching, he noted his satisfaction with the project. He noted how the experience provided opportunities to trial an alternative approach and consider how it might embed in his future practice after the intervention period.

A second point of reflect was on students performance as determined through their project work. In post-intervention interviews, student project work was reviewed and compared against expectations and intended learning goals, particularly final student projects. Where time was limited, the lead researcher analysed student projects and presented these during remote interviews. In looking at example student project work, Mr Thomson noted how most students were successful in completing basic expectations:

> *MT: ...being able to code multiple […] characters with backdrops and most of them were using the correct timings.*

When reflecting on student performance against the agreed pedagogical goals, the analyses of student work was reviewed (see Table 33). In this discussion, Mr Thomson commented on the need to balance student autonomy with teacher expectations:

> *MT: As it stands, I'm quite happy with what they've done […], most of them have had a good go at it. [In future] I would have prescribed it an awful lot more, but then that would have squashed a lot of creativity so that's the big question.*

Given the limitations of the context, it was decided that participants' performance in the analyses was moderately successful. Most of those from the sampled dataset (*n*=5/6) were found to have created a multimodal story with multiple narrative episodes and a resolution. In addition, all of them (*n*=6) used costume/backdrop changes to denote different narrative events or characterisation and most (*n*=4) had set initial sprite states correctly to ensure projects ran identically each time they were executed.

7.3.2 Intervention factors

7.3.2.1 Computing subject knowledge and skills

A primary factor Mr Thomson identified that impacted the success of the intervention was the level of subject knowledge and skills required to teach the subject effectively and his perceived abilities. Given his level of experience in teaching programming, as well as the infancy of the subject in schools, he highlighted the time commitment in 'tinkering' with project work before feeling confident to teach:

> *MT: I don't see myself as being a "coder", I see myself as a learner […] I have to have a fiddle around with it, play with it myself, to know what it's all about, and to learn it myself before I teach it.*

This point emerged in a further discussion on his ability to provide appropriate feedback for students. He suggested feeling anxious at the idea of working with students with better programming skills and the challenges in facilitating them:

> *MT: If I had a complex coder you know, more competent even than myself, I probably wouldn't be able to decipher [what they coded] in that sense.*

More broadly, he indicated that this challenge would be felt by most teachers in the school. This emerged from his experience as computing coordinator as his primary role involved working with other teachers to design and facilitate computing lessons. Discussing this issues with other classroom teachers, he found that certain processes yielded the most anxiety, such as debugging and troubleshooting:

MT: The general fear from most of my staff is the fact that, well, "what happens when it doesn't work? What happens when the computer crashes? What happens when Scratch doesn't do what it's supposed to do?"

Addressing this factor was challenging as it required additional support, particularly, when troubleshooting, and this was costly. During the interviews, he reflected on the upcoming school year and the challenges rolling out the curriculum unit with multiple teachers could provide:

MT: If you've got lesson like this then […] you'll need not one but even two [teaching assistants] to keep running around after people, debugging stuff or walking through that micro-teaching sort of style.

One solution he proposed to address these concerns was taking a dual-role during co-teaching with other teachers. This would involve teachers facilitating the curriculum unit while he would co-facilitate and offer support in troubleshooting hardware and software bugs. Though permission had been given to trial this approach previously, it was concluded as unsustainable for the long-term and therefore required alternative support to be provided (e.g., detailed teaching materials, additional scaffolding). Some of these provisions are discussed further under Adaptation and future use (see 7.3.3).

Table 33: Analyses of activity DEF6 (Independent project) (subset)

Conjecture (storytelling)	**#/6**	**Conjecture (programming)**	**#/6**
Define a narrative goal	6		
Add character(s) or object(s) using sprite(s)	6	Add sprite(s)	6
Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., *say* blocks)	6	Create sequence using *say* block(s)	6
Illustrate characterisation (behaviour, appearance) through costume(s)	4	Use change costume block(s)	4
Establish setting(s) using backdrop(s)	6	Add backdrop(s)	6
		Edit backdrop(s)	6
Use *show/hide* block(s) as cohesive device(s)	4	Add *show/hide* block(s) to sequence(s)	4
Use varied text block(s) (e.g., *say*, *think*) to represent dialogue/thoughts	3	Use *looks* block(s) for multiple purposes (e.g., say, think)	3
Adapt graphic or text elements (e.g., dialogue, character or object) for narrative effect	6	Edit sprite(s) or backdrop(s)	6
Add or manipulate costume(s) to illustrate action(s) (e.g., looking around)	6	Use change costume block(s)	5
Illustrate setting change(s) using *change backdrop* block(s)	6	Use change backdrop block(s)	6
Coordinate narrative element(s) to execute correctly	6	Use initialising block(s) (e.g., *when* blocks) on all scripts	6
		Set initial program state(s) (e.g., sprite size, position, graphic effect)	4
		Ensure all scripts have hat block(s) (i.e.no 'dead code')	4
Illustrates story episode(s) through text blocks	5	Use text block(s) (e.g., *say, think*)	6
Illustrates story episode(s) through character action (e.g., motion)	6	Use change costume block(s)	6
		Use *motion* blocks (e.g., glide. go X/Y)	3
Create title cards to orient reader(s)	1	Create text-based sprites (e.g., title cards)	1
Use structural device(s) to ensure cohesion (e.g., change backdrop blocks)	6	Use change backdrop block(s)	6
Use *wait* block(s) as cohesive device(s)	6	Use wait block(s) to sequence narrative correctly	6
Use *broadcast message* block(s) as cohesive device(s)		Use broadcast messages(s) to coordinate multiple processes	
Check project against plans	6	Run programme to check for errors	6
Revise project in light of plans	4	Debug errors	4

7.3.2.2 Literacy and multimodal composition

Previous work integrating computing and literacy through project-based activities organised as was discussed. As multimodal composition—or indeed multimodal storytelling—is neither

a requirement nor a feature of the 2014 English curriculum, most existing work undertaken by teachers centred on repurposing students' existing literacy work as opposed to creating or revising new work (e.g., multimodal texts). These included 'typing work up' for presentation or repurposing existing work for different purposes (e.g., PowerPoint presentations). Likewise, this novelty of approach meant that Mr Thomson was unfamiliar with creating multimodal texts. Several modifications made during sessions (e.g., the addition of lesson plans and sample materials, see Appendices 1-2) focused on providing support to teachers unfamiliar with this approach. In post- intervention interventions, the value of these materials as well as the importance of subject knowledge and professional experience in facilitating computing lessons was highlighted:

> *MT: So because I've experienced and been through it once I would feel a lot more confident […] I'm happy to do it [again] but I've got the knowledge and skills.*

Through support materials and reflecting carefully on the challenges faced by teachers with limited teaching experience, he concluded that the use similar pedagogical approaches in the future would be possible provided that adequate support was maintained through his work as computing coordinator.

7.3.2.3 Time management

7.3.2.3.1 Activities

A central theme of the post-intervention discussions was time management, both within sessions and within the wider timetable, and how this was managed in practice. For instance, the first session focused on exploring the concepts of sequencing (i.e., creating sequences) and execution (i.e., wait blocks) to programme sprites in terms of programming goals. In terms of storytelling goals, it also focused on using sprites and blocks to establish character and setting. Reflect on his experiences of facilitating this session, Mr Thomson added how time was overestimated to complete each activity though emphasised modifying instructions to encourage individual solutions:

MT: I thought it was going to be too simple [but] I think it spread its time out quite quickly […] we made it open-ended for them so that there if there was time […] they could take it further.

As all activities had an open-ended component where students could continue to develop a project, this issue was not immediately a concern. However, other activities presented some issues due to their pacing within sessions. For instance, during a planning meeting, Mr Thomson commented on the number of activities in Session 2—debugging stories, manipulating backdrops and costumes, and resetting scripts:

MT: I sort of wonder whether there's a way of using what they're making and building it with them rather than deal with some things separately and then get them do it on their own […] but it's hard to find a balance between [them].

To compensate for this issue, he demonstrated one activity (debugging an existing story) and encouraged students to look out for the same problem (e.g., timing issues) in their projects. However, after facilitating the session, the need to keep up with the pacing of activities to achieve the lesson goals was reemphasised:

MT: I think the lesson ran away with itself […] I think I was having to keep very prompt with my teaching and to keep it moving. I think there's quite a lot to get in [and] a lot to get through in one lesson.

This concern was partly due to the additional time needed to work with a larger classroom group. However, it also stemmed from a contextual issue. As part of the session involved moving between rooms, going from planning their independent projects in the classroom, generating story ideas, to working in the computer room, the time allocated for programming activities was more limited. This was due to the limited space in the school's designated computer room—which doubled as a school canteen and rehearsal space—meaning that planning activities (e.g., storyboarding) could only take place in the classroom and that lessons were divided between both spaces with 'walking time' factored into the lesson plans. This

point was reiterated in multiple sessions as moving between rooms caused concern over the time allocated to different activities:

> *MT: I think the overall arching thing was… time is critical. Other teachers might not be as time precious… 'cause otherwise […] it won't be usable.*

Though the timings were suggested, he reiterated the point multiple times as commitments on either side of the intervention sessions—photography course and afternoon lessons—meant that achieving the lesson goals within the allocated time was crucial.

7.3.2.3.2 Timetable

Committing to the project also required the participant teacher to commit to a series of co-design meetings taking place before and after each session. As the implementation period progressed and—given the circumstances surrounding the impending lockdown period—intensified, we reflected on the process of co-designing the intervention and how this was implemented in practice; in particular; Mr Thomson noted the demand of multiple workplace commitments and the challenge to facilitate the session alone:

> *MT: We spoke a bit about the planning process and what was going to be happening each week [but] I needed you to turn up [and] all the resources would be there. The planning would be there, and I'll be able to just deliver it quite honest.*

The commitment to the research study—including all requirements of co-design meetings, interviews, debriefing sessions—was understandably more *involved* than merely adapting and facilitating an existing intervention without a research objective though the process, including preparing the classroom, computing room, resources, and seating arrangements, was notably labour-intensive. To accommodate this, Mr Thomson sought assistance from students in the *digital leaders* group to prepare devices, learning materials, and classroom organisation.

7.3.2.4 Differentiation

A central purpose of the teacher interviews was to receive feedback from the participant teacher on the intervention content, and where effective changes could be made. Reflecting on this point, Mr Thomson noted the lack of feedback in managing a larger group of students:

> *MT: I think that regular feedback as well […] we'd not really looked at their projects and […] had a couple of marking sessions where we just look at all the work […] and therefore see some more commonalities from the class[…], if they're all missing the reset and we'll do a restart lesson or whatever.*

While the impact of COVID-19 precluded these weekly marking sessions, student work was reviewed between sessions by the lead researcher and feedback given during design meetings. For instance, sample projects were demonstrated to highlight common issues or features across student work. In one session, this was used to highlight issues relating to project timings—the time allocated to sequencing sprites and backdrops. In subsequent sessions, these patterns were addressed through targeted intervention work with individual students or small groups. In some cases, particularly in later sessions, whole group discussions were necessary to discuss recurring issues in student projects (e.g., resetting scripts).

Another point on differentiation and supporting and assessing students' performance was noted during the interviews. Mr Thomson reflected on his prior SEN experience in working with children with diverse learning needs and prior experiences and the challenges this poses in a computer-based environment:

> *MT: My skillset is in differentiating… to meet the needs of […] my children. [It's] horrific when you were doing computing… it's not just differentiating in […] what you're learning about […] It's also differentiating [for] how much access they've had on the computers. […]*

In relation to the intervention expectations, these considerations were highlighted as central to the success as basic computer skills were limited in the group. For instance, some students

found difficulty in using left and right click on a mouse, using some of the commands on Windows computers, or saving and retrieving project work. Where this issue manifested itself during the lessons, Mr Thomson noted how his ability to facilitate the lesson and support students was hindered as some of these more technical issues occupied a large part of his time:

> *MT: Another point… is that there's more troubleshooting going on than there is teaching which is why [our] teaching of computing has been limited […] so that we can ensure that the whole class is getting moved in the right direction.*

Observations notes corroborated this account as Mr Thomson was routinely asked to resolve technical issues (e.g., saving files, reconnecting to the internet) during lessons. Though this observation underscored the need for basic digital literacy skills in facilitating computer-based interventions, it further underscored the importance of—or lack of—individual feedback in providing adequate support for students to realise their project goals.

7.3.2.5 Teaching strategies

7.3.2.5.1 Open-ended activities

Enacting the curriculum unit presented Mr Thomson with multiple opportunities to practice new teaching strategies. For instance, many of the activities set were open-ended without fixed outcomes or processes (e.g., creating a narrative episode). Reflecting on his desired teaching strategies and approach, he noted his desire to include more open-ended activities in his regular practice. He noted how existing curricular schemes were limited in allowing students to decide the outcome of their projects or providing means to adapt to their needs. As these issues affected most schemes of work he had used, he recognised the challenge in designing activities and materials to facilitate open-ended, project-based activities:

> *MT: I think it's a challenge… in itself, how we get from (quoting) "Do this, do this, do this, do this" to (quoting) "here's this, go and have a play around a bit".*

It also presented additional challenges in facilitating a larger group of students and their multiple project needs and demands. This concern manifested through some amendments to

the curriculum design. For instance, in one session, a brainstorming activity was modified to include sample text types (fantasy/adventure, scary stories, and mysteries) based on stock resources in the software tool, to provide more structure for prospective teachers and students, and manage expectations. However, it was also noted that while resources were provided and encouraged, the outcomes of student project work was not mandated.

7.3.2.5.2 Integrated activities

The practice of integrating both learning strands within a series of lessons was a factor strongly identified as contributing to the success of the intervention. Commenting on the benefits and trade-offs of applying an integrated approach, Mr Thomson noted how the literacy standards and expectations in the school, and the balance between computing and literacy goals, affected the value he placed on applying the project in the current context:

> *MT: Is it beneficial for them to let's say (quoting) "waste" a literacy lesson on developing [an] application in their computing? […] I would feel more confident saying yes to that knowing they had […] a good knowledge of all of the other computing aspects […] so that they will be able to create something a bit better.*

This point was reiterated during further interviews where he noted the limitations of applying storytelling in a novel context where students' basic computer literacy skills limited their ability to engage in the intervention activities:

> *MT: …and the reason they've not managed to come up with something that's brilliant or flashy […] it's because they've just not got the computing side […] that they can't apply it because their computing understanding is not developed enough…*

He described how integrating both learning strands was challenging in practice as he felt that student projects were limited by their programming knowledge and skills and that the time allocated to computing ("once a week…if at all") paled in comparison to literacy ("five times per week"). Given that computing had only been introduced several years earlier, this was unsurprising though it meant that introducing cross-curricular activities was more difficult.

Discussing how this issue could be resolved in future practice, he reiterated his desire to plan separate storytelling and programming activities and then plan an integrated project at the end of this. When asked about the balance of storytelling and programming activities, he noted that the balance was adequate for the context, but emphasised his preference for teaching separate lessons to yield greater success than attempting to integrate throughout:

> *MT: It's not necessarily more balanced, it's more separate […] I want to teach one computer lesson and then a literacy lesson […] Once I've done that, I'm going to bring them together and we're going to go into the computer [and] join them together because that's what's coming together is for you project to be that sort of project.*

Due to students' relative lack of experience in both learning strands, particularly multimodal storytelling, he concluded that facilitating separate lessons in both learning strands would yield better results in an integrated context. Moreover, he argued, it would enable the curriculum unit to be better understood and adopted by teachers as the application of both strands would be more better understand.

7.3.2.6 Availability of resources

7.3.2.6.1 Technology

One of the primary challenges identified in the lesson observations was the limited number of computers (14) and the variety of devices, platforms and tools used to facilitate lessons in regular practice. This emerged when teaching strategies (e.g., setting pair programming tasks) were discussed in relation to lessons. This challenge was highlighted before the intervention and addressed it by adding additional desks, chairs, and portable computers (e.g., laptops) to the room to allow one device per child:

> *MT: I think… that's a failure as a…lesson. [because] now one device per child […] there's a minimum amount of computers in that room and if one fails so does that lesson in a sense.*

Resolving this issue required additional laptops to furnish the lesson which added considerable preparation time given to each lesson. As existing activities in the school were largely conducted in pairs (i.e., pair programming), he suggested that using a single device platform (iPads using the browser-based Scratch) would yield greater success. In practice, he aimed to trial this approach with smaller groups in remote teaching sessions during lockdown.

Working across multiple devices with multiple operating systems meant that managing student work became an additional challenge. Using the web-based Scratch Editor, logging all devices into the same account, and ensuring students used unique filenames (e.g., first name, session #) to prevent overwritten files, resolved this issue; however, it presented an logistical challenge in multiple sessions. Similarly, challenges arose in multiple lessons in syncing students' project work to their cloud account, allowing them to save and resume progress in subsequent sessions:

> *MT: I know we had a few teething problems with saving things and loading and stuff […] there were some technical issues with Scratch.*

The scarcity of teaching materials to adapt for lessons was also highlighted as a factor that affected the intervention. During one interview, he discussed using existing resources (e.g., Code Club materials) and how these were limited in their scope. Collaborating on the design of project-based activities then proved useful in discussing his context and needs and what support might be needed:

> *MT: It… enabled me to try something as a project that I wouldn't be able to do on my own. Like, I wouldn't be able to… plan and prepare and… create all the resources […] I think it's enabled me to have a project that we can do as a school.*

7.3.2.6.2 Stimuli

The availability of rich stimuli for storytelling was also referenced during interviews. Mr Thomson noted how using the web-based resources for student projects presented additional challenges for teachers as opposed to the use of existing in-built resources in Scratch (e.g.,

sprites, backdrops). He reflected on how student expectations could be managed by encouraging students to remix and edit existing resources:

> *MT: I think, I think there's a limit […] to where you can go with it [..] I just said, you got the stuff that's in the bank. […] If you want a robber, get a person and paint in black and white […] I think that a lot of their creativity can come through the database […] a wizard can be whatever you want it to be if you just create it differently.*

Though most students opted to use existing resources, this decision resulted in many students adapting these resources to suit narrative goals. For instance, students modified costumes to reflect events in their stories (e.g., expressing emotions, a spell being cast, magic dust to indicate movement; see Figure 25). Less commonly, students created resources from scratch to express narrative features. (e.g., portals to transport characters, simple backdrops to highlight important objects; see Figure 26) where existing resources were more commonly used to represent character actions, reactions, and attitudes.

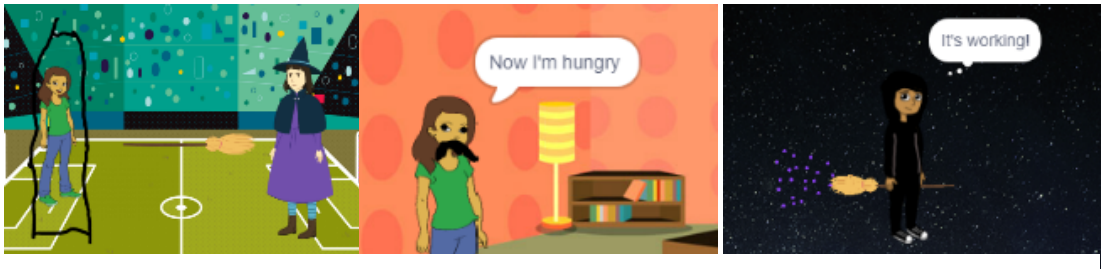


Figure 25: Student-modified resources (Study 3)



Figure 26: Student-generated resources (Study 3)

7.3.3 Adaptation and future use

This theme refers to adaptations to the intervention design made before or during the implementation period.

7.3.3.1 Annotated texts

In Session 4, activities focused on planning the multimodal elements of students' projects and revisiting programming features to date. One of the key adaptations made to the intervention design was the design of annotated texts (see Appendix 2) used during instruction. These texts were modelled in class to draw comparisons between monomodal (i.e., just text) and multimodal (e.g., text and image) texts. They were also used to encourage students to see *how* and *where* multimodal features affect meaning in a text. Reflecting on the need for and use of this tool, Mr Thomson commented on his unfamiliarity with the learning content, remarking:

> *MT: I found it quite difficult to describe the difference between the effects that each one of those gives.*

However, subsequent discussions based on the addition of the annotated texts highlighted their importance as both a modelling tool and a teaching support. In practice, Mr Thomson was observed using the tool to demonstrate the similarities and differences between the two text forms, while emphasising what multimodal text effects could be achieved through multiple programming strategies, and inviting students to indicate their preference (i.e., mono- or multi-modal texts).

7.3.3.2 Integration of learning strands

Another one of the adaptations made during lessons was the decision to separate literacy and computing activities into separate times during the afternoon sessions. Mr Thomson indicated that he preferred to facilitating activities separately, where programming instruction and planning for storytelling are taught separately and then subsequently integrated at a later stage. His justification for doing so partly due to the limited spaces for project work. As the computing room had no physical desk space, it meant that planning activities or writing reflections were limited to the classroom and programming activities in the computer room. This necessitated the timetabling of each learning strand into separate times in the afternoon

as opposed to integrated throughout. It also stemmed from his perception that the school's 'progress', particularly in computing, against National Curriculum standards was behind:

> *MT: I would rather do one lesson in the computer room learning the basics of all the computing issues […] and then maybe having a lesson that's more literacy based [on] how do we build stories […] what's the difference between a story with pictures and one without pictures and images […] and do more of that multimodal story teaching.*

A second justification he argued was that the dominance of formal exam preparations and the secondary nature of applied project-based activities, particularly in Literacy, meant that curriculum unit did not fit neatly within the school's existing curriculum 'boundaries' He argued that while Year 6 (10-11 year old) students would benefit most from the curriculum unit, particularly where activities are more complex, their upcoming final exams (or SATs) exempted them from less formal curricular activities:

> *MT: It's a nice idea to do in Year 6 as an end of year project after they finished their SATs and got a bit more time on their hands […] we probably put less weight on sort of 'other' learning and Computing at the beginning and the end of the year when they tried to do SATs prep and stuff.*

A third justification for organising activities into separate sessions was the need to accommodate students with limited experiences of programming and equally multimodal storytelling. As the experience was new to him and his students, he was concerned that it was too early to employ applied project-based activities and integrate learning strands:

> *MT: Maybe we're not quite ready for it yet […] I think as students and as staff, our knowledge of computing, might not quite be enough for us to be creative with it yet, what we're looking at doing here is creating a project and although I taught bits of it, it's hard to apply that straight away.*

This preference stemmed from both his work in designing curricula for each year group and working with classroom teachers in facilitating these lessons. He noted that the level of

experience and subject knowledge required to derive successful outcomes from the curriculum unit was great. This meant that separate lessons in both introductory programming and multimodal storytelling would be preferable before the curriculum unit could provide meaningful integration in both learning strands through project-based activities.

7.3.3.3 Time management

This theme refers to prospective or proposed changes to the intervention design discussed either following the implementation period or during the post-intervention interviews. Given that the intervention was intended to be further adapted through an additional iterative cycle with other classroom teachers, Mr Thomson was asked to comment on adaptations he would prospectively make to the intervention design to better support students based on his experience of facilitating. On multiple occasions, he noted the shortcomings of allocating sufficient programming time to the independent activity and his desire to frame initial structured tasks around students' own work:

> *MT: I would make it all about their one […] project which would give them more time on it as well, because I reckon […] they were maybe cut short […] or just [give them] extra time in the last two sessions to complete that whole story they wanted to create.*

Despite this point, he underscored the importance of these activities in structuring learning experiences. As initial structured activities were designed to ensure students encountered in specific aspects of programming or storytelling, these were predominantly based on sample texts and not student projects as such aspects might not be present or such issues might not present themselves (e.g., debugging synchronisation issues):

> *MT: It's nice to have that [structure]… but I, I felt earlier on that they felt like they were doing one-off activities that weren't necessarily purposeful […] so it's just maybe thinking how that could be started earlier so it's all their project.*

The discussion concluded by presenting the use of template structures (discussed in 7.3.3.6) as a means to work on students' projects earlier while still encouraging students to encounter

important programming and storytelling features. Moreover, providing additional time to working on students' final projects across multiple activity sessions would provide additional breathing room for students to complete their projects on time.

7.3.3.4 Differentiation

As Mr Thomson regularly co-taught lessons with teachers across all age groups, a primary concern for him was the differentiation of lessons and progression of concepts and practices across year groups. In discussing future use of the intervention, he underscored the importance of differing expectations and how the intervention could be differentiated to support different learning goals and expectations.

> *MT: …doing a similar sort of project but with different expectations for different year groups […] Year 2s, we are going to do a similar sort of project. We're just not going to have the costumes, motion and animation. We're just going to have the text and the backdrops. […] It's all dependent upon where they're at […] to have that progression.*

This preference aligned strongly with the use of differentiated learning conjectures outlined in Study 2. In conversation with Mr Thomson during this study, he further highlighted the role of *digital leaders* in each classroom and how this necessitated the use of differentiated learning goals and expectations. As digital leaders served to assist the computing coordinator, teachers and other students to use devices, prepare lesson materials, and set up equipment, defining learning goals for classrooms need to take into account the breadth of experience in each classroom group. A post-intervention interview focused on his use of the curriculum unit in future teaching sessions, including differentiating both learning activities and goals for younger class groups (e.g., Year 2-3, 6-8 year olds) and the use of more age-appropriate tools (i.e., Scratch Jr).

7.3.3.5 Modelling and demonstration

A third point noted was the need to model the multimodal storytelling process through modelling and verbal demonstrations. Mr Thomson underscored his explicit preference for modelling all project work through 'live coding' (Rubin, 2013) demonstrations:

> *MT: To show it to the demo […] like a sort of sample project […] I think it helps the children see you working it out […] just modelling it as you speak […] that sort of style suits me because […] the children can learn as I show them and walk them through my thought processes and then we obviously have questions from that.*

Modelling the multimodal storytelling process also served to provide him with experience of creating multimodal stories. As noted earlier, Mr Thomson highlighted his lack of familiarity with some of the learning goals, citing their lack of prevalence in daily literacy activities. Instead, he argued that the use of live coding and modelling strategies could aid his understanding of the curriculum materials and intended learning goals.

7.3.3.6 Structured activities

7.3.3.6.1 Giving feedback

The use of open-ended (e.g., narrative activities) also highlighted a challenge in working with large groups of students. More specifically, Mr Thomson identified the challenge of motivating students to take multiple pathways to achieve their individual goals. In post-intervention interviews, he noted the challenge in seeking flexible outcomes:

> *MT: The problem comes when the children become creative because then we've got to start looking and debugging […] we'd struggle when the students do something, and we can't see how they're going or where they've done it and how it's wrong.*

In discussing implementing the intervention with classroom teachers, he commented that planning debugging activities would require additional technical assistance or heavily prescriptive learning material (e.g., sample projects and resources):

MT: For them to deal with [debugging] 30 children, it's just not going to happen […] so you would just need to be either having the technical support there or having it more prescriptive or more prescribed is the two options I can set.

As Mr Thomson was planning for the forthcoming term, the adaptations he was making to the curriculum unit for future lessons were discussed. Specifically, how the level of structure through prescriptive elements to support teachers could be used while still providing students enough opportunity for individual expression. In this discussion, he proposed that prescribing the output of activities (e.g., through sample projects and resources) would better support teachers in achieving the project goals. By showing both students and teachers the features and structure of a desirable project, he concluded that novice teachers and programmers would be adequately supported in achieving the set goals of the curriculum unit. In one post-intervention, the idea of template structures was discussed in greater detail. He described in detail a form of story 'template' to structure project-based activities and manage variation across projects:

MT: It can be just the same model […]cause basically whatever the code is going to look the same again, [it will] have a green flag, they're going to switch costumes, switch backdrops to that scene, switch over to that one […] two characters in each one […] for the basis and then they can put in text and then timings […].

In proposing his 'balanced model', he further suggested that the text structure could serve as an instructional support for teachers while offering sufficient flexibility for students to embed their own ideas and individual creativity into their projects. However, this model presented a tension between his preference to scaffold students through structured activities and support his own teaching through modelled examples, and his desire to provide opportunities for students to creatively employ programming and storytelling for their own means. Reflecting on this, he noted the delicate balance between structure and agency as an additional concern. This stemmed from the idea that altering the structured tasks would strongly alter the project goals and processes:

MT: The problem is you take the creativity away from my kids but that is the only way to do it is to make it more prescriptive, you know?

This discussion ended by proposing that the level of prescription should focus on a series of inclusion criteria (e.g., programme should feature a loop) to allow students to creatively embed features and ideas into their own projects. It also meant that teachers with limited experiences in teaching programming would have adequate support structures to identify and diagnose code as samples could be provided. Further, he noted how comfortable he was with this approach, enabling more creativity within a structured set of activities.

7.3.3.6.2 Structured planning frame

A second structural support was proposed as a future adaptation in the use of structured questioning in the story planning frame. This idea stemmed one planning activity where participants had some difficulty 'filling in' each box (e.g., *What's your story about? What blocks will you use to* programme *it?)* and sought support from Mr Thomson. He noted that the loose structure of the planning frame questions didn't adequately draw students' attention to the critical features of both their projects as stories and programmes:

MT: I was trying to put the ownership on them, but maybe it's like, almost like a boxing […] Draw your character. What's your character's name? Where does your character live? […] Is it a quest? […] What is the problem?

These question prompts are notably more targeted and declarative though he argued that the need to provide this level of scaffolding prompts was needed in younger classroom groups. He also articulated how the prompts could prompt specific story structures (e.g., "Is your story an adventure/quest?") and allow students to adapt from rich model texts. As these prompts could be decided in advance, it would also enable teachers to direct students to follow specific structures and aid debugging and providing feedback.

7.4 Summary of findings

The main goal of this study (**RQ3**) was to consider how the intervention could be implemented by classroom teachers and productively incorporated into their classroom practice. This is considered by exploring the experiences of participants (teachers and students), the factors affecting implementation—both enhancing and inhibiting—and the adaptations and productive modifications made during lessons and recommended.

The overall experience (**SQ5**) of co-designing an intervention with a participant teacher, Mr Thomson, was a positive one. Though the impact of COVID-19 on data generation was not insignificant, classroom observations of the intervention being implemented by a practicing teacher followed by a series of reflective interviews were nonetheless possible within the limited period. These data were crucial to answering the main research question—how multimodal text construction that supports students' programming and storytelling goals can be embedded within regular classroom instruction—and provide insights into the challenges and opportunities this posed to the participant teacher.

The experience also provided an opportunity for Mr Thomson to reflect on his own practice and consider alternative pedagogies and approaches. Though this seemed sensible in promoting teacher agency when trialling the unit, it revealed another tension between the intended as opposed to enacted curriculum. van den Akker (in Nieveen et al., 2006) offers three distinctions between the intended (the idealised or formal written curriculum), the implemented (as perceived by teachers or enacted in the classroom) and the experiential (as experienced by students and the resulting outcomes) curriculum. In this case, Mr Thomson perceived the intended curriculum—students pursuing narrative goals through visual programming—as pedagogically desirable. However, integrating activities from two learning strands created between his beliefs about cross-curricular (or integrated) activities ("[It helps] because it's jam-packed [and] the fact that they can use skills in different way is really good") and the schools' expectations and his professional responsibilities. This was due to the fact that recognising the integrated project as either a 'pure' literacy or computing lesson was

difficult. In qualifying the lesson as either literacy or computing, he felt that the headteacher would raise issues with the use of integrated learning goals within a strictly-defined set of subject areas.

Nonetheless, the participant teacher's positive experiences were encouraging as they indicated his openness to trial new approaches, consider how computing might integrate with other subject areas, and willingness to overlook technological and pedagogical challenges in the pursuit of novel teaching practices. This contrasted with recent work by Larke (2019) who argues that a growing number of teachers have rejected the mandated curriculum change. Moreover, he neither viewed the new curriculum as "developmentally inappropriate, too narrow and too difficult" (p. 1137) and instead sought to find novel and authentic contexts in which to apply computing in multiple subject areas, including mathematics, science, and literacy education. These experiences were supported by the additional material resources and preparatory design meetings though it indicated that the right approach could enable teachers to provide authentic and engaging learning experiences for their students. While challenges still remain in equipping teachers and schools with adequate resources or training an adequate number of teachers, the results of Mr Thomson's experiences were encouraging.

The second research question in this study concerned factors that affected movement toward the pedagogical goal (**SQ6).** These varied from internal (subject knowledge and skills, preferred teaching strategies) to external (time management, availability of resources, intervention status) during the implementation period. One of the perceived central factors affecting future uptake was the timetabling of the curriculum unit as both a project designed to satisfy goals in both Computing and Literacy. According to the participant teacher, where the project would more comfortably lie is as separate learning strands within their respective disciplines before integrating these strands in a single project-based activity. This response highlighted the teacher's need to articulate the value of the intervention though it posed a challenge in considering how computing could be embedded across the wider curriculum, if at all.

Integrating existing curricular strands also meant that Mr Thomson also needed to justify the curriculum unit as existing firmly within one subject or another. In design meetings, he noted that achieving full integration, or where learning strands are taught together, was difficult to achieve in practice. This point is reflect in recent work by Israel and Lash (2020). In their study on the integration of CT into the mathematics classroom, they found that classroom teachers indicated a preference for enacting lessons in isolation. Mr Thomson was similarly committed to "teaching the curriculum with fidelity" (p. 373) particularly where subject areas had assessment components (e.g., end of primary school tests). However, there were grounds for optimism as he also indicated the pedagogical value of the curriculum unit in facilitating applications of curriculum content. In other words, while he felt that integrated lessons were less beneficial for introducing areas of computer science or multimodal storytelling, he concluded that providing meaningful and authentic contexts in which to exploit concepts and practices was his preferred means of instruction.

The process of co-designing—from initial design meetings to implementation—presented several opportunities to adapt the intervention **(SQ7)** both formally—through design meetings—and informally—through on-the-spot modifications—to suit the local context and the participant teacher's needs. As a computing coordinator with co-teaching and co-design responsibilities with the entire teaching population of the school, proposing cross-curricular projects that connect computing with the wider curriculum, namely literacy in this case, was naturally a strong fit. The participant teacher responded positively to the intervention activities, highlighting their applicability and value in his own future teaching. However, based on classroom experience, he anticipated greater structure being required to accommodate students and teachers with varying experiences in computing and multimodal storytelling. These points contradicted earlier points made about wanting to provide students with more opportunities to be creative through less prescriptive approaches. They also indicated the tension between Mr Thomson's desired as opposed to enacted teaching.

The results of the final study highlighted areas where the design could be improved. Tatar (2007) argues that design exists as a tension between "what is" and "what ought to be" (p. 415). The process of facilitating the intervention presented tensions in "what was" and "what ought to have been" in arriving at design compromises and adapting to suit the participant teacher's needs while equally serving the researcher's needs. These mostly manifested through the teacher's need to adapt the intervention design to suit challenges that presented during lessons. As arriving at a workable solution is one of the theoretical imperatives of DBR, it is important to address these tensions and how they affect practicalities and value. Some of the key tensions arose in the contrast between the participant teacher's preference for proposed activities (e.g., open-ended narrative activities) and his inability to achieve many of them. For instance, he indicated his preference for single-session projects as the inability to anticipate timetabling issues had, up until that point, prohibited multi-session projects.

The process of co-design also highlighted adaptations made during the implementation and proposed adaptations based on the teacher's experiences of facilitating the sessions. These proposed modifications to the intervention design are discussed further in the Discussion chapter (8) and contextualised against relevant literature to arrive at a theoretically grounded redesign. For instance, one key area for adaptation during interviews was the use of multiple differentiation and scaffolding strategies to support both the participating students and prospective teachers with varying experiences of programming and multimodal storytelling. In facilitating lessons, the teacher noted how the provided sample projects were not adequate for facilitating larger groups. The proposed modification, adding structured sample projects to use for demonstrating and as a basis for student-generated projects, could arguably support teachers in facilitating more anticipated (or fixed) outcomes while still providing ample opportunities for students to pursue their own ideas flexibly and creatively.

CHAPTER 8
Discussion

8.1 Introduction

The research undertaken for this PhD was motivated by two emerging strands of research inquiry. First, significant concerns have been raised by teachers, researchers and policymakers concerning how the computing curriculum has been implemented, particular at primary level, and how better support could be provided (Larke, 2019; Sentance et al., 2018). Secondly, scholars in literacy and computing education research (Burke et al., 2016; S. Smith & Burrow, 2016) have suggested that engaging students in multimodal composition, primarily in the form of digital storytelling, can motivate and support learners' programming and storytelling practices though evidence of this in an integrated content is limited.

The overarching aim of this thesis was thus (i) to explore whether and how employing multimodal composition activities in a visual programming environment might mutually support children' storytelling and programming practices, (ii) to consider the learning potential of an integrated approach, and (iii) to articulate how this form of integration could be enacted in a formal setting. The resulting curriculum unit was designed to engage children in a series of integrated activities that drew on their existing experience and skills in both learning strands. The primary empirical work tested the unit through multiple successive intervention cycles by developing and refining context-specific analytical techniques.

The empirical work undertaken to achieve these aims and objectives has been outlined and discussed in the previous three chapters. Three empirical cycles of testing the curriculum design were conducted. The first empirical cycle, Study 1 (Chapter 5) first considered how multimodal text construction could mutually support students' programming and storytelling practices by (i) characterising what forms of programming and storytelling practices are fostered through multimodal composition activities and (ii) considering the relationship between these practices. The follow-up study, Study 2 (Chapter 6) considered how

the curriculum unit could accommodate learners with varying experiences of multimodal text construction and programming. It aimed to achieve this by considering what iterations were necessary to arrive at a closer, more successful enactment of the intended curriculum. Finally, Study 3 sought to consider how the unit could be embedded within a formal classroom context and how it could be used to support teachers' curriculum needs. It aimed to do so by (i) considering the experiences of the participants (teacher and students) taking part, (ii) investigate what perceived factors enhanced or inhibited the aims of the curriculum unit, and from this (iii) consider how the curriculum unit could be adapted to meet the needs of all stakeholders. Summaries of the results of these studies are discussed briefly below. Thus, it can be said that thesis has been concerned with three main issues:

- Exploring how MMC activities can facilitate learners' programming and storytelling practices
- Theoretical and practical insights into the process of facilitating the curriculum unit
- Defining and testing a methodological approach for assessing the enacted learning design

The final chapter of this thesis summarises the results and implications of this work. In the first section, the main research findings are briefly summarised. This section illustrates how the empirical work addressed the overall research objectives of the thesis. The next section more substantially discusses the design work undertaken in formulating and implementing the curriculum unit and how this was iterated and developed throughout the process. In doing so, it contextualises the research undertaken against existing scholarship, both research that informed the design and development of the curriculum unit, and emerging research that has been published since the implementation period. It also considers whether and how aspects of the curriculum supported the intended pedagogical goal and how these were adapted across successive iterations. This served to illustrate where findings overlap or offer new insights, and what contributions to knowledge have been made. This section also discusses the implications for theory and practice and where the research undertaken has contributed to the

development of both. As the empirical work centred on research-practice partnerships and enacting theory-derived principles in real-world contexts, findings in terms of the design implications on future theoretical, empirical and methodological work are discussed. Also, trajectories of this research for both the research and practitioner communities are offered. The final section discusses the limitations of the empirical studies and analytical approach and argues where future work could seek to address these.

8.2 Research aims and objectives

This section details an overview of the main research aims and objects of this thesis. It does so by briefly reviewing the individual research questions (**RQ1-RQ3**) corresponding to each study and discussing the main research findings of each. These findings provide a means of answering the main research question (**MRQ**) of the thesis, that is, whether and how multimodal composition activities can be used to support primary students' programming and storytelling practices.

8.2.1 Study 1

The first empirical chapter (Study 1) sought to answer the first main research question (**RQ1**: Which multimodal composition activities promote storytelling and programming goals?). It did so by considering which forms of programming and storytelling practices are fostered through the designed multimodal composition activities (**SQ1**). Results indicated that MMC activities provided learners with opportunities to explore and embed foundational programming concepts in their story-based projects. These included sequencing (deciding on the order of events), algorithm design (deciding on how a programme should be executed), initialisation (deciding on the initial state of the programme) and parallelization (executing multiple processes simultaneously). Learners were generally successful in programming events in correct sequence, with correct timings, and executed according to their narrative goals. They also reported feeling satisfied with their performance and motivated to continue creating narrative-based projects after the designated sessions.

Limited experiences of computing generated difficulties for some learners in achieving some of the more advanced learning goals. For instance, in earlier sessions, many learners struggled with using text-based blocks to generate speech or dialogue, instead opting for sound-based blocks. Adopting broadcast messages also proved difficult as most learners interpreted a broadcast 'message' as analogous to a text-based command. Finally, initialising sprites by ascribing initial values (e.g., size, position or orientation) proved difficult as learners were unable to automate some of these processes, instead opting to manually control these attributes.

Activities also challenged learners in both planning and executing their respective narrative goals. While learners found success during the structured activities, they were less successful planning and writing their own story. In some cases, they reported being unhappy with the level of resulting projects relative to their respective goals. In a debriefing discussion, learners commented on the difficulties in thinking of successive narrative events and satisfying conclusions. These discussions prompted the planning activities to be reconfigured to support learners in generating story ideas. These modifications were implemented in Study 2 and found to provide learners with more appropriate learning supports in achieving their goals.

Composing multimodal texts in Scratch also provided learners with opportunities to use context-specific representations (e.g., sprites, backdrops, code blocks) to structure their texts. For instance, participants were found to have successfully chosen and manipulated environment-specific representations (e.g., sprites, backdrops, text-based blocks) to depict narrative features (i.e., characters, settings, dialogue). Data indicated how learners mostly used text- and image-based representations to represent their narratives and often demonstrated an implicit understanding of the effect of certain representations (e.g., a transparency effect) on meaning. However, many struggled to programme more advanced multimodal effects in their final projects, opting to represent their stories through image and text alone. Limited prior experiences of programming also meant that their final projects never achieved the narrative

goals set. This was due to most participants indicating a preference for embedding animation into their final projects.

Results also demonstrated that composing multimodally did not result in the widespread use of certain programming features (e.g., loops and conditionals). In the case of event-based programming, where the speed of execution is programmed automatically, this was achieved by setting the speed and order of sprites and backdrops (e.g., characters, setting, dialogue) using *wait* blocks to sync within and across sprites (inter- and intra-sprite synchronisation). This resulted in stories that were generally sequenced correctly, ensuring texts were coherent and comprehensible by other users.

The study considered how these practices were related to one another in the production of multimodal texts (**SQ2**). Results suggested that learners could productively engage in MMC activities to focus on specific programming (e.g., algorithm design) and storytelling (representing narrative events) practices. To encourage stronger integration between storytelling and programming, analyses highlighted areas where storytelling and programming conjectures were more prevalent though not employed widely (e.g., using multimodal text features for narrative effect). This suggested that more explicit instruction or scaffolding in composing multimodally might support learners in better achieving their intended programming and storytelling goals in subsequent iterations.

8.2.2 Study 2

The second empirical chapter (Study 2) sought to answer the second research question (**RQ2**: How can learners with varying experiences of programming and storytelling be supported during multimodal text construction?). It aimed to do so by iterating on the previous curriculum design before considering what forms of programming and storytelling practices are fostered through multimodal composition (**SQ3**) and how these practices relate to one another. This design cycle aimed to iterate on the previous curriculum design and address the weaknesses identified by the task-based analyses. These iterations included adding additional examples of multimodal text effects, sample texts and brainstorming activities, and allowing

learners to start work on their final projects in earlier sessions. Additionally, as learners had variable experiences in programming and creating digital texts, scaffolding measures were differentiated to provide adequate support to the group.

Findings from this study indicated that the revised unit provided learners with multiple opportunities to apply their programming and storytelling practices in more meaningful ways. In terms of programming, evidence was provided of where learners were more successful during structured and independent activities, including creating a complete multimodal story and satisfying their respective goals. For instance, composing texts promoted learners to consider the sequencing, algorithm design, initialisation, and parallelization (or sequential execution). These findings align with previous research into project-based activities (e.g., Adams & Webster, 2012; Grover et al., 2018) demonstrate how loops, conditional logic and broadcasting messages were less commonly found in text-based projects. Like Study 1, findings also aligned with research on learners' use of event-based programming strategies (Weintrop et al., 2018) to ensure their texts were comprehensible by other users.

In terms of storytelling, participants were more successful in generating and programming complete narrative texts. Analyses demonstrated how final project work was characterised by more complex structures with most students producing a text with multiple episodes and a conclusion. Employing more complex multimodal features enabled learners to embellish their stories with more engaging representations for the audience (e.g., dragons flying, floating characters). Participants were also encouraged to draw upon 'unofficial' literacy resources (Dyson, 2008) to plan and construct their texts. That is, they were encouraged to bring their favourite texts from popular culture, other media forms (e.g., games, comics, YouTube characters) to encourage integration between formal and informal literacy practices. This was deemed successful as it resulted in rich, complex stories with participants adapting existing stories to suit their own needs. Though privileging learner agency when choosing which stories to tell was successful in this instance, its function was challenged in Study 3 where the demands of the school curriculum limited the ability to set open outcomes.

Study 2 also yielded evidence that learners with prior experiences of programming were able to use more advanced programming features in the pursuit of narrative goals. For instance, learners additionally explored features of programme initialisation—or assigning initial values and characteristics to sprites and objects—by considering the initial state of their programmes. Analyses highlighted learners who adopted multimodal effects including motion/graphic effects/animation and were able to successfully reset sprites to their original size/position/characteristic allowing their projects to be executed multiple times with the same outcome. However, this effect was only achieved by students with more programming experience as the others struggled to achieve these goals. Learners also found it challenging to programme motion-based events and reset these values, opting instead to manually move sprites before running their programmes.

Like Study 1, findings indicated that learners employed many of the basic programming and storytelling constructs that had been introduced in the structured activities during their independent project work. Unlike the previous study, however, most participants were successful in achieving their intended narrative goals and creating a complete multimodal text. Adaptations made following Study 1 provided additional time for learners to work on their independent project—as well as the ability to start earlier—and enabled greater success throughout. In addition, as specific representations (e.g., character motion) were made explicit during instruction where appropriate, that is, relative to their level of experience, learners were found to have employed these features independently in their final projects. The results of this study highlighted the challenges of successfully supporting learners with varied experiences of programming and digital storytelling though highlighted how MMC activities could be productively employed to achieve these goals simultaneously. This is discussed further in the next section (Implications).

8.2.3 Study 3

The third and final empirical chapter (Study 3) sought to answer the final research question (**RQ3**: How can multimodal text construction that support students' programming and

storytelling goals be embedded within regular classroom instruction?). By focusing on teacher-led implementation, it aimed to explore the experiences of teaching the curriculum unit in a formal classroom setting (**SQ5**). There was also a need to investigate the perceived factors that enhanced or inhibited the intervention goals (**SQ6**) to ultimately consider how the curriculum could be adapted to further meet teachers' professional needs (**SQ7**). As noted in Chapter 7, though COVID-19 meant that the study could not proceed as originally planned, empirical work was still achieved prior to national lockdown. Results of this study indicated that the curriculum could be embedded into formal classroom time and provide opportunities for the participant teacher and learners to productively engage in mutual programming and storytelling activities. Results from interviews suggested that the experience of the teacher facilitating the curriculum was a positive one. He reported viewing the curriculum activities as applicable and useful in his own future teaching practice. However, he emphasised the need for greater structure to both accommodate students with varying experiences in computing and multimodal storytelling and provide learning supports to ease the process of facilitating (e.g., sample completed projects, lesson plans).

Results also suggested that enacting the designed curriculum posed professional (i.e., required subject knowledge and skills, preferred teaching strategies) and practical (time management, availability of resources, intervention status) challenges when facilitating. As it proved challenging in achieving the intended pedagogical goals, the experience provided opportunities to reflect on the teacher's beliefs and practices and the context in which he was teaching. Though some of these were context-specific (e.g., lack of resources), others were more general. For instance, one of the perceived inhibitors affecting future uptake was how the curriculum was proposed. As it sought to satisfy curricular goals in both primary computing and literacy education, it proved challenging in discussing where and when it would fit within the existing timetable. Moreover, the professional demands placed on the participant teacher and students to produce 'quality work' meant that the value of the curriculum unit, at least in this reported context, was challenging to articulate. In short, the participant teacher

viewed the activities as providing opportunities to explore features of programming and storytelling in a meaningful context. However, he noted that focusing on separate learning strands before approaching an integrated project would be easier to implement in future practice.

These studies represent the three cycles of DBR undertaken to consider how multimodal composition activities could support young learners' programming and storytelling practices. The approach taken naturally constitutes just one example of how a curriculum could achieve this goal—in this case, through MMC. While the data in this project does not allow the claim that integrating storytelling and programming is superior to separate forms of instruction, particularly where MMC is concerned, they do present insights into the novel forms of programming and storytelling practices unique to this context. Further considerations of the limitations of the data generation and analysis undertaken is discussed in section 8.4.

8.3 Implications

Having reviewed the research aims and objectives of this thesis and provided a summary of the main research findings, the implications of these on future research are now discussed. This section is divided into two parts: the first part (3.1) contextualises these findings (the 'what') against existing research. Findings are also discussed in terms of the underlying design principles (the 'why') that were used to achieve the intended pedagogical goals. This section then offers insights and considerations for future design research in this area. The second part (3.2) discusses how findings were derived and interpreted (the 'how') and how this approach yielded a novel contribution to analytical strategies in DBR.

8.3.1 Design principles

This section details the design work undertaken and contextualises research findings against the underlying design principles, both intended and enacted, that provide insights into the orienting role played by these design principles when designing the curriculum unit. In the

following sections, each principle is briefly summarised and implications for future theoretical and practical work are discussed.

Four design principles were proposed to guide the development, implementation, and evaluation of the curriculum unit. These drew on relevant theories of learning, as well as research relevant to each learning strand, to derive guiding principles that were instantiated in multiple ways to serve the pedagogical goal. The level of design modifications necessary for each iteration capture the complexity of each context in which the intervention was trialled.

8.3.1.1 Building on learners' prior knowledge and skills (DP1)

The first design principle focused on building on learners' prior knowledge and skills. Drawing on theories of representation construction (e.g., diSessa, 2004), it was proposed that by encouraging learners to draw upon their representational knowledge and engage in self-directed representational construction through multimodal storytelling activities, they would be encouraged to draw upon a 'bank' of existing representational resources and conventions for constructing and representing their current knowledge. This was based on the assumption that learners should be provided opportunities for open-ended learner-generated construction with little regard for inducting within "semiotic conventions or traditions" (Prain & Tytler, 2012, p. 2769).

Findings from this thesis indicated that learners' implicit understanding of the functions of multimodal resources and their effect on meaning. Final project data provided evidence that learners could use, adapt and create representations based on existing resources (e.g., characters, costumes, backdrops, and objects) in the Scratch programming tool. In addition, when learners were unable to source representations based on their favourite texts, they sought alternative solutions. By encouraging learners to draw on existing frames of reference (e.g., favourite picture books, comics, visual texts), the unit pointed to the potential of bridging home and school literacy practices. This aligned with work by MMC researchers (e.g., Millard, 2006) who stress that 'hybridising' classic and digital texts (e.g., comic books, animation, games) can provide a rich and motivating context for learners to practice their

emerging storytelling (and programming) skills. These findings proved insightful in investigating learners' implicit though often-times mature understanding of multimodal conventions and their effects on meaning. They also align with extant literature (e.g., Bazalgette & Buckingham, 2013) that suggests learners bring to the classroom a wide knowledge and understanding of the conventions of visual texts and mediums that these should be exploited where necessary.

Some limitations of this approach are noted. For instance, learners' breadth of familiarity with multimodal texts—particularly those encountered in informal learning contexts (e.g., afterschool clubs, out-of-school literacy experiences)—prompted a high expectation on their resulting project work. Learners were dissatisfied, in some cases, that the graphical features of their projects were somewhat basic, and that richer multimodal effects (e.g., character animation, transitions) were difficult to achieve. Though research continues to underscore the value of drawing upon learners' rich set of existing representational resources (Ainsworth et al., 2011; diSessa, 2004) in practical terms, providing existing template resources and opportunities to manipulate and repurpose existing representational resources was found to be more suitable in satisfying curricular goals within a limited timeframe.

This tension was also noted in Study 3 as the facilitator faced challenges in providing agency through open-ended activities while also adding structure through carefully selected resources and examples. This tension, he noted, emerged from a need to understand and feel comfortable troubleshooting and providing adequate support for learners. In earlier studies (e.g., Study 2), providing learners with agency in choosing their own resources based on popular cultural texts was somewhat impractical though it had furnished them with more engaging and rich material resources. This finding points towards the challenge of 'fusing' home and school literacy practices through encouraging 'hybridisation' of popular cultural texts (e.g., games and comics) and classic texts (e.g., children's literature). As Dyson (2008) notes, learners may draw upon official and unofficial literacy resources to plan and write stories but their function within the school curriculum may be limited to planning strategies as opposed to desired outcomes.

8.3.1.2 Engaging learners in constructive activities (DP2)

The second design principle focused on engaging learners in constructive activities. This principle drew upon the theory and practice of constructionism (Harel & Papert, 1991) by proposing that learners should be afforded opportunities to construct and externalise "objects-to-think-with" through constructive activities. In other words, by externalising their current understanding, learners (and facilitators) can reflect, act upon and transform these ideas which, in turn, contributes to greater understanding and insights.

One central insight across multiple studies was the degree to which learners used the inherent features in Scratch—that of executing, testing and retesting their projects—to enable greater success in their programming and storytelling. Executing the code of their multimodal stories was found to have encouraged learners to reflect on how their projects were executed and presented by the visual programming environment, enabling both learners and facilitator to diagnose or self-diagnose issues with projects. As learners were encouraged to routinely test that their projects achieved their desired effects, it ultimately enabled them to engage deeply with their externalised representations of stories and, through debugging activities, arrive at a closer approximation to their intended narrative goals. This finding resonated with work by Collins et al. (1991) who argued that learners viewing their own self-generated representations can more easily diagnose current misunderstandings, discover errors, and revise work. Moreover, in Study 3, these reflective artefacts served as a means of diagnosing group-wide issues and furnishing the facilitator with relevant and timely examples for modelling and demonstration activities.

Using open-ended (i.e., learner-directed) activities also proved successful in providing learners with opportunities to construct personally-meaningful artefacts (Harel & Papert, 1991). Initially, it was proposed that learners engage in a series of closed activities before a self-directed project-based activity. In this sense, learners were first tasked with a series of structured, closed activities with largely fixed outcomes before being given opportunities to embed their personal preferences and ideas into their final narrative projects.

Findings from Study 1 highlighted where learners sought to focus their attention on self-directed activities, including designing their own stories from the beginning, as opposed to seeking the same outcome as others through closed activities. In doing so, some of the activity-related goals were not widely met.

The revised curriculum unit in Study 2 sought to promote greater learner agency through open-ended narrative activities. Though many of the desired programming goals were similar, modifying the context in which these took place was found to benefit learners' agency. It also compensated for the lack of opportunities for purposeful expression through creative activities. Activities were more in line with what Bearne & Reedy (2017) argued as a desirable goal in literacy, encouraging learners to find a communicative purpose for their compositions. However, this finding was tempered in Study 3 as a tension emerged between managing learners' expectations while providing sufficient agency. In particular, the balance of curriculum demands against learner expectations was found to present challenges in deciding on and implementing activities. In Studies 1-2, the luxury of an informal learning context had precluded the need to consider the demands of the formal curriculum but this was more acutely felt when planning the final study.

Future work should address the balance between structure and agency by considering how a model of instruction could serve both teachers and students. A more 'balanced' model could add some prescriptive elements to support novice teachers managing variation across projects while aiming to provide some opportunities for learners to creatively use the resources given. As generating representations (e.g., drawing a sprite or backdrop) proved unmanageable within the limited timeframe, evaluating, negotiating and refining existing representations served as a more practical form of representation construction. Tytler et al. (2013) argue that "constructing [one's] own representations" is merely one key element within a representation construction approach to learning. Further research could also consider more closely the "role of representation in learning and knowing" (p. 51) within integrated contexts with the explicit goal of understanding the role and 'appropriateness' of representations in narrative construction in mind.

8.3.1.3 Scaffolding student-generated construction (DP3)

The third design principle focused on scaffolding student-generated construction. Drawing on relevant literature in computer science education that points to the challenges in supporting novice programmers (e.g., Lye & Koh, 2014), this proposed that adopting scaffolding measures, including classroom prompts and overhead displays (e.g., PowerPoint) to mark critical features, would better support learners in achieving their intended goals.

Enacting this principle was necessary in supporting learners during challenging activities. For instance, learners demonstrated limited experiences of programming and had moderate success in creating story-based projects in Study 1. Results highlighted their difficulties in generating story ideas—or story endings—which resulted in instructional supports in Study 2 being redesigned to focus more strongly on the demands of storytelling. This was unexpected as it was initially anticipated that learners would face more difficulty in programming and so task supports were designed accordingly. Following these results, task supports were designed to support learners equally in both challenging aspects of programming and storytelling. The scaffolding materials (e.g., model texts, demonstration activities) were revisited and found to have supported storytelling practices well.

Operationalising the principle differently required the nature of the structured activities, alongside the challenges encountered by learners using Scratch to achieve a narrative goal, to be considered. As noted by Howell (2018) in her work on scaffolding multimodal construction, this requires a careful consideration of the selection and adaptation of instructional supports and materials. In the context of multimodal storytelling, a focus on narratives coherence through episodic structures was chosen and demonstrated through modelled examples. Adopting the structures of a classic narrative (or 'complete' story) (McCabe & Peterson, 1984) was useful in provided learners with opportunities to reflect on the structure and sequence of their stories. This included discussing multiple narrative episodes, story endings, and which programming concepts were necessary to represent these multimodally. Results from Study 2 found that most learners achieved a complete story

relative to their initial narrative goals, suggesting that instructional supports focusing on narrative structures were necessary in producing better stories. Indeed, in follow-up interviews during Study 3, similar observations were noted in the provision and use of story-centred scaffolding materials in providing learners with a richer set of narrative structures and examples in which to apply to their respective projects. As this process is context-dependant, its success has implications for project-based activities.

One implication of the research undertaken is the role of scaffolding during programming activities. As findings from Studies 1-2 demonstrate, learners' prior experiences of computing (or the lack thereof) impacted on their ability to engage successfully in certain activities. More advanced programming features, such as applying broadcast messages to coordinating project resources, proved difficult and resulted in many seeking alternative solutions. For instance, many learners struggled to adopt more complex programming features (e.g., motion, initial sprite states, final sprite position) instead opting to express narrative events through dialogue alone. Explicit instruction on the practice of animating sprites for different purposes provided learners with more narratively rich examples to be used. Previous findings (e.g., Franklin et al., 2017) also found that novice learners found difficulty in setting initial values for sprite attributes, including sprite size, position, and graphic effect. Where they found difficulty to integrate motion and its associated programming attributes into their final projects, they instead primarily used text- and image-based blocks to represent their texts.

Evidence also points to learners widely employing broadcasts unsuccessfully as they generally interpreted the function as analogous to a sprite 'speaking'. This finding was consistent previous research on learners' understanding of the natural semantics of visual programming languages (e.g., using words like 'say' and 'broadcast') and their impact on understanding (Pea et al., 1987; Sorva, 2013). An unintended consequence of providing analogous examples using everyday language to aid understanding is that learners formed misconceptions of the purpose and function of certain programming features. This was significant as it suggested that examples given during instruction should be carefully considered to ensure that learners better understand the function of different programming

features. For instance, demonstrating how to programme a sprite to relay a message to another sprite using broadcast messages caused some confusion.

Future research in this area—that is, novice programming contexts—should be mindful of these misconceptions as they slip from everyday language into discussions and support materials. It should also consider the nature of the intended literacy project and the required structures and supports needed to highlight critical features relative to the literacy goal. It could consider how a more structured approach could address the instructional needs of a larger group of learners with varying experiences in programming and digital storytelling and provide reassurance to teachers unfamiliar with either or both learning strands.

Scaffolding, coupled with more explicit-guided instruction and opportunities for independent exploration for students who are ready, is arguably even more important when teaching classrooms composed of students with varying levels of experience. As such, engaging with inclusive instructional strategies that scaffold instruction and afford access and modified levels of challenge for learners with varying experiences of programming is a challenging yet important goal for computing education. Additionally, these goals become increasingly relevant as computing education is integrated more widely into primary and secondary school formal contexts where learners with varying experiences and needs and schools with varying resources should be considered.

By arguing for conjecture-based forms of assessment with multiple conjectures, the results of this study demonstrated how explicit instruction can be designed to provide learners with more complex and engaging activities. In particular, offering learners multiple pathways for success, by identifying appropriate concepts and applications for learners with different experiences, was found to engage learners more productively in desirable learning goals.

Though the approach taken was relatively successful in Study 2, addressing the instructional needs of for all the students in the class was a noteworthy challenge in Study 3. As participants in the subsequent study was notably larger (n=28 vs n=10) and the intervention instead facilitated by a classroom teacher, resourcing and facilitating the sessions provided insight into some of the challenges faced by the participant teacher. The use of the intervention

materials, including the annotated text and lesson plans (see Appendix 2), was also found to have supported the participant teacher in approaching the novel learning content.

8.3.1.4 Engaging learners in authentic and purposeful activities (DP4)

The fourth design principle focused on engaging learners in authentic and purposeful activities. Drawing on theory within computer science (Guzdial & Forte, 2005; Tissenbaum et al., 2019), it was proposed that by engaging learners in authentic practices—that is, by contextualising learning within relevant real-world contexts and examples—it would benefit learners' understanding of how and when to apply certain programming concepts (e.g., sequencing, events, and conditionals) in project-driven activities. It was suggested that multimodal construction could provide an authentic context to invoke storytelling practices and consider if and to what extent profound links between computing and literacy are found. Moreover, the curriculum designed employed Scratch as an 'empowering' (as opposed to 'teaching) system (Kelleher & Pausch, 2005)—that is, one where programming is taught in pursuit of another goal, multimodal storytelling.

Existing literature has indicated the role of visual programming in supporting narrative construction. Some researchers (Buckingham & Burn, 2007) have suggested that narrative framing serves to highlight analogous textual and visual elements across different media (e.g., character types, narrative events, story settings). Indeed, Howells and Robertson (2012) found that visual programming prompted learners to situate a narrative visually through story 'locations' to map and sequence a story. As an alternative endpoint to CS, the results of the three studies highlighted where learners could be provided with opportunities to tell stories multimodally and how this process was augmented within the visual programming environment. Being context-specific, learners were required to use and adapt domains-specific representations (e.g., sprites, backdrops, stage), manipulate and sequence these using programming techniques, and consider multimodal features for narrative effect. As this was, in most cases, a novel experience for learners, their use of representations was largely limited to text- and image-based resources in constructing their texts. Nonetheless, the integration of

visual programming constituted a rich multimodal experience for learners as coordinating and integrating multiple representational forms for different purposes and goals presented many challenges throughout.

Results also pointed to specific activities that could meaningfully integrate programming and storytelling through authentic contexts and applications. In Study 1, findings from the task-oriented analyses indicated that learners found greater use and value in programming constructs that resulted in richer multimodal text effects (e.g., loop functions) as opposed to programming constructs with little visual difference (e.g., broadcast messages over wait blocks). Where learners had faced challenges in adopting more complex programming features and sought alternative strategies, the purpose of several structured activities and their desired outcomes became somewhat *inauthentic* and lacked immediate relevance.

In Studies 2-3, this design principle was re-operationalised through the addition of activities designed to exploit more meaningful and relevant programming constructs for learners' storytelling and programming goals. This included embedded examples in model texts, facilitator demonstrations, and reflective discussions on the impact of certain multimodal design decisions on participants' projects. Findings from the revised curriculum unit suggested that learners were more successful in applying relevant and meaningful multimodal text features (e.g., animations, movement) to embellish and enhance their stories. As the specific purposes of these programming activities were authentic and provided practical applications for independent work, that learners chose to apply a greater set of multimodal text effects in their final projects was encouraging.

Findings from this study contrasted with previous research (e.g., Adams & Webster, 2012; Grover et al., 2018) that indicated the lack of more complex programming features (e.g., loops and nested loops, the use of repeated program execution, defining initial sprite state and motion blocks) in story-based projects over other forms (e.g., games, music videos, animations). Where modelled examples were designed to provide authentic, relevant and meaningful applications (e.g., using motion-based blocks to create animation and control the

positioning of sprites and objects), learners were found to have successfully employed more complex and demanding features in the pursuit of their narrative goals.

Prado et al. (2021) argues that existing literature on approaches to computational thinking centre largely on the identification of CT concepts, practices and perspectives with less regard for the development of alternative approaches that may be appealing to more learners. This thesis has argued that foregrounding activities and approaches that engage learners with varying experiences and understandings of computing and multimodal storytelling is a meaningful and worthwhile goal. As such, the results of this thesis are promising for proponents of applied CT/computational action who seek "alternative endpoints" (Tissenbaum et al., 2021) within and across the domain as they point toward the success in embedding computational ideas through creating and revising authentic products.

As findings highlighted where targeted examples (loops, parallelism and event-based scripts) could be employed to create multimodal effects to tell more engaging stories or used to provide structure to story-based projects (e.g., using initialising blocks to manage events). However, they also indicated the limits of the designed curriculum unit in employing a wider variety of complex multimodal features. While computational action is arguably a more meaningful and relevant goal, both within computing and across the curriculum, the desire to encourage more explicit integration across learning strands places both practical (i.e., resources needed) and professional (i.e., subject knowledge and skills) demands on teachers to implement successfully.

As this thesis did not consider the integration of CT outside of the domain of storytelling, or narrative within the literacy curriculum, future research could establish whether and how further concepts and practices in programming could be meaningfully integrated across a wider range of literacy-based applications. Indeed, more recent research (e.g., Ballard & Haroldson, 2021) has begun to explore the use of children's literature (e.g., picture books and graphic novels) as rich "unplugged" resources to derive and explore representations of programming constructs (e.g., algorithmic thinking, abstraction, sequencing). Likewise, Proctor (2017) found that interactive (i.e., choose-your-own-adventure) storytelling was

productive in developing middle school students' debugging skills. The findings presented in this thesis align with these by highlighting how such rich and varied applications can be both rewarding and demanding.

Future research on similar approaches could more sensitively consider the time and resources necessary for both teachers and learners to engage successfully and derive positive outcomes in applied contexts. Indeed, though the curriculum unit proposed in this thesis was designed for a six-hour period, the potential for integrated curricular content in a limited context was identified. Research could also explore further applications of integrated activities across diverse learning strands to further investigate their pedagogical value in other contexts.

8.3.2 Analytical approach

This section details a further contribution made by this thesis in the form of a novel analytical strategy. Assessing multiple learning strands during a single intervention required existing DBR approaches to be adapted for this context. One contribution therefore was the development of a novel analytical method to assess student performance during the activities. In particular, the approach taken employed and adapted existing techniques—at least those within the DBR tradition—primarily focused on means of assessing performance within a single learning strand (e.g., Sandoval, 2004). The analytical strategy offered a means of characterising and assessing learners' progressions across multiple learning strands. This was achieved by employing the task-oriented analytical strategy developed by Dierdorp et al. (2011) for assessing intended against enacted learning, and adapting it for multiple learning strands within a singular context.

Employing a task-oriented approach served as a form of conjecture-based analysis (Sandoval, 2014), focusing on deriving analytical categories to evaluate learning designs conjectures. As Reinking & Bradley (2007) argue, defining analytical categories such as conjectures embodies the expectations, values, actions and interactions of learners in a given context. As these conjectures—or theories of anticipated learning—were generated and tested against student performance during each intervention, several insights were generated into the

perceived strengths and weaknesses of either the design or implementation of the curriculum unit. In short, results of these analyses provided focus areas when redesigning activities, reconsidering learning goals, and adapting the curriculum unit to meet the needs of others (i.e., teachers and facilitators). Where 'gaps' were found—that is, where conjectures not widely met—this enabled a warrant for follow-up studies to refine and re-test the design.

Though visual representations such as conjecture maps or hypothetical learning trajectories are common within the DBR tradition (e.g., Sandoval, 2004; Simon & Tzur, 2004), understanding the relative strengths and weaknesses of the curriculum unit as an integrated entity required existing tools to be adapted. From multiple, simultaneous conjectures, a picture emerged of learners' emergent programming and storytelling practices and how these were supported through multimodal narrative construction.

8.4 Current limitations and suggested future research

The results of the three studies suggest that applying a multimodal composition framework to integrate concurrent learning goals in programming and storytelling can be mutually supportive. However, there were some limitations noted throughout the process. These are discussed below in terms of their impact on claims made, and their implications for future research.

8.4.1 Recruiting multiple teachers

One limitation in the research undertaken was the nature of the co-design partnership with participant teachers. The impact of COVID-19 posed some limitations to research in this thesis in Study 3. The intended research procedure had involved recruiting multiple teachers with a breadth of experience in facilitating computing lessons and integrating multimodal composition within their practice. This was partly done to disentangle the results of Studies 1-2 from the lead researcher serving as facilitator. It was also done to understand the process of successfully implementing the designed unit taking into account the knowledge, skills and understanding necessary to facilitate. It was rightly assumed that participant teachers would require an understanding of the curriculum requirements and experience with visual

programming tools in order to facilitate the curriculum unit successfully. Moreover, the aim had been to ask classroom teachers to integrate the curriculum unit within their existing curricular plans. Though some work had been started on this, the threat of school closures prevented the sessions from taking place.

The research plan was amended to accommodate the national lockdown as the original plan of working with multiple teachers was unable to be completed. Moreover, as lockdown was approaching, the revised procedure contracted the six-week period of classroom observations into a two-week period. As the frequency of lessons were intensified, this had the effect of shortening between-session design meetings, limiting the ability to capture immediate reactions and reflections. To accommodate this shift in focus, the revised interviews conducted remotely after the sessions were completed drew heavily on artefacts collected during data generation to ground discussions in evidence and prompt relevant questions.

The research-practice partnership achieved despite these restrictions involved a co-design process between the lead researcher and school computing coordinator. This enabled insights to be gained into the role of facilitating and leading teachers to implement the computing curriculum. However, as computing coordinator—whose role as computing teacher in most classes was limited to facilitating the computing curriculum—his experience at serving the needs of the wider curriculum was limited. As the role of classroom teachers in Studies 1-2 were limited to gatekeepers, and the lessons enacted informally during an after-school setting, the original aim of Study 3 had been to consider the role and value of integrated practices across multiple curriculum strands. However, while these insights were not able to be found, collaborating with the computing coordinator as facilitator provided a useful opportunity to understand other implementation issues (Penuel et al., 2018).

Further work in this area could focus more closely on the research-practice partnership between researchers and classroom teachers. Though work was begun on developing a series of CPD sessions that focus on the integration of computing and CT within existing curricular areas, this was unable to be implemented during lockdown restrictions. As

such, these research activities are argued to greater align with the aims of co-design within the DBR tradition and provide insights into the authentic integration of CT within curricular contexts. Moreover, this would serve to not only to embrace long-term partnership to improve practice over time (O'Neill, 2016) but also align with the aims and objectives of computational action. As computational action focuses on embedding experiences of computation across multiple and varied authentic learning experiences, then identifying and co-designing such experiences with classroom teachers would arguably serve to adapt for individual teaching goals. In this thesis, an integrated curriculum unit based around multimodal composition activities was proposed and found to offer several benefits for learners' storytelling and programming practices. Discussing these activities with prospective teachers could enable other authentic and contextualised projects to be considered based on previous work.

Finally, implementing the curriculum unit with multiple teachers with diverse needs and contexts would provide further insights into the role of visual programming tools as empowering systems, or where programming is taught in pursuit of another goal (e.g., project-based learning). Though research has often indicated the challenges of embedding and promoting digital technologies in the classroom (Hall et al., 2013) employing a tool such as Scratch to facilitate "learners to build things that are tailored to their own needs" (Kelleher & Pausch, 2005, p. 109) and teachers to facilitate experiences that align with their curricular needs would enable multiple insights to be gained. By discussing the aims and outcomes of the curriculum unit and taking into account the previous successes and failures found in this thesis, would serve to support teachers to adapt the unit to fit their needs. For instance, reports (e.g., National Research Council, 2011) have suggested that research into the benefits of computational experiences within integrated over stand-alone contexts is limited but growing. Though challenging to implement, more work on the benefits of efficiency is still needed.

8.4.2 Broadening computing for multiple purposes

Since empirical work was completed in this project, more resounding calls for alternative pathways in computing educations, including forms of multimodal composition, have been

made. For instance, Tissenbaum et al. (2021) argue that by promoting multiple and alternative endpoints in computing education through constructionist-based curricula, computing can be positioned as a more valuable and relevant experience for more learners. They argue that one endpoint, personal and social creative expression, offers a means to support learners in expressing themselves through digital means. As a curricular goal, this resonates both with the purpose and practices of multimodal composition—as argued by Bearne (2017) among others—and the wider goals of constructionism. This goal notably resulted in the development of Logo, the programming language, to provide learners tools to generate computer-based graphics not dissimilar to those in then contemporaneous video and computer-based systems (Harel & Papert, 1991). The more recent proliferation of Scratch(Resnick et al., 2009)—the software used during empirical work—was equally motivated by a desire to encourage and support learners to create "interactive stories, games and animations" (p. 60) through visual- and code-based media. These 'outputs' position Scratch, like other visual programming languages, as a "digital canvas" (Tissenbaum et al., 2021, p. 8) to explore alternative uses of computational media that are both personally meaningful and tied to powerful concepts within the domain. In fact, during the course of the empirical work in this thesis, additional experimental blocks in the Scratch programming language were also added (MIT Scratch Team, 2020), adding the ability to display text (embedded within a sprite; not a costume), add animated text effects, and manipulate font choice, colour, and size (see Figure 27). Though limited to beta testing—that is, not available on the regular online- or download- version of Scratch—the features demonstrate the additional multimodal (esp. typographical) resources available to users.

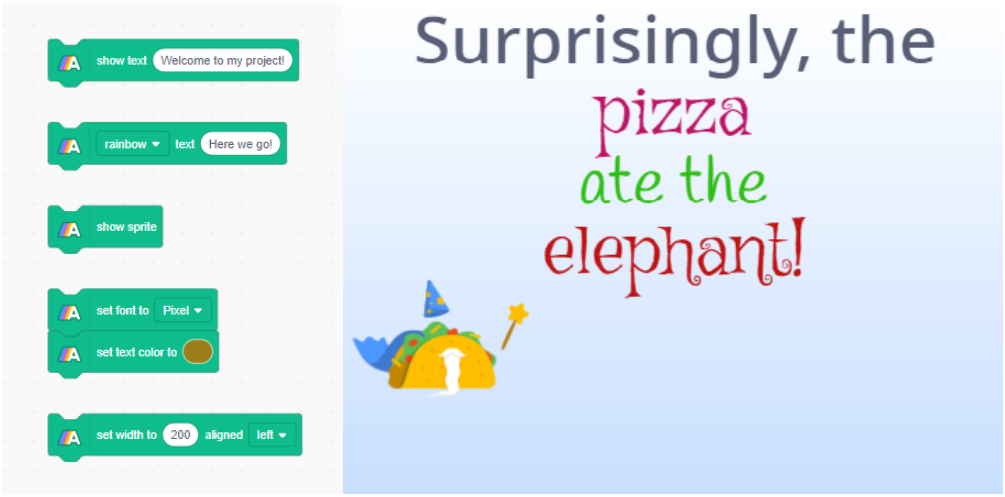


Figure 27: Screenshot of Scratch animated text features (beta)

The implications of the additional features for storytelling purposes are many; for one, participants in the three studies routinely opted to use text (i.e., speech and thought) blocks to express desires and intentions, illustrate narrative events, and develop characterisation in their sprites. The additional animated text blocks afford users the option of using text for prolonged descriptive text or exposition. Previously, participants in two of the empirical studies programmed a *narrator* sprite, separate to their story, to deliver more conventional text-based story features (e.g., descriptive text). In addition, the use of colour, font choice, size and animated effects (e.g., flashing colours) poses opportunities to discuss the affordances of multimodal choices on how meaning is conveyed.

8.4.3 Conjecture-based analyses

A third limitation was noted in the methodological procedures undertaken across all studies. This refers to the nature of the conjecture-based analyses and their purported claims to illustrate learning; while claims of participant learning are limited, data analyses of this sort provide useful insights into the design and implementation of learning environments based on what is observed through authentic practices. As noted earlier, though the use of task-oriented analyses and similar analytical techniques is relatively common in DBR, the nature of claims made differs from study to study. In this case, the comparison of the "actual" learning conjectures given in the task-oriented analysis with the "impression" of learning noted through

the data generated allows only for an "impression" of learning to be made. Dierdorp et al. (2011) suggests that the veracity of these claims to support learning are strengthened by their occurrence across cases and studies.

In order to give an impression of how participants' learning was supported by the curriculum unit and how well this learning was predicted in the HLT, actual learning (or the ALT) was compared with the hypothetical learning (or the HLT) for each activity. To find evidence and counter-evidence for the conjectures formulated in the HLT, the transcripts, field notes, interviews, and final assignment were analysed with the help of a conjecture-oriented analysis.

Dierdorp et al. (2011) further argues that interventionist studies conducted with a small sample (*n*=12) can only suggest a "proof of principle" (p. 148). Applying that point to this context suggests that implementing an intervention to support integrated programming and storytelling practices is *possible* but wider claims about intervention *effectiveness* are limited. However, the ecological validity of the research is helped by the decision to conduct the research in a formal classroom setting (see Study 3) as it meant that the designer of the intervention was not the facilitator in all cases. By involving other stakeholders as facilitators, claims about the value of the curriculum unit in support existing classroom practices were strengthened.

8.4.4 Generalisation of findings

A final point centres on the limitations of claims made in DBR. In view of the small number of participants (*n*=12/10/28), results are understandably specific to the local contexts of each study and, as such, not immediately applicable to other contexts. However, the aim of this thesis is arguably not to attempt to generalise too widely. The findings reported in this thesis, though specific to each individual context, aim to generalise the design processes and insights learned from each case such that they could be adapted for other situations. Therefore, the aim of DBR in this context is to better understand the challenges and opportunities of integrating

programming and storytelling through an integrated context, observing the enacted curriculum and assessing learning through suitable analytical tools.

Building upon the findings of this thesis would enable further insights to be revealed. For instance, in the context of DBR, similar findings reported over multiple iterative studies can contribute to the trustworthiness of the data and dependability of findings (Dierdorp et al., 2011). Future iterations of this research could seek to mitigate these threats. Areas for future direction are noted in the previous sections (8.4.1-8.4.3) that would involve more teachers, students and other content areas within the specific learning strands. Though these could draw more general conclusions, the context of small-scale design research with a single PhD researcher should be noted. In this sense, the findings of DBR lie closer to what Stake (1995) refers to as 'petit generalisations' or findings that furnish other researchers with insights into the process of conducting similar research. These include rich descriptions of designs enacted in practice and theory-driven conjectures tested in different contexts. For instance, Cobb et al. (2003) argues that the research context should be provided in greater detail to inform other researchers on the procedures undertaken to allow research to be more easily adapted to local contexts and needs.

This thesis arguably provides value by highlighting the challenges and opportunities of designing similar research while recognising the messy nature of real-world practice in the claims made. Moreover, it provides strategies and tools to approach designing similar research. Though the research was guided by grand theories, the development of humble theories around integrated contexts for programming and storytelling was ultimately grounded in authentic pedagogical contexts that aligned closer—that is, they were arguably more ecologically valid—with local instructional practices.

The situated nature of the type of retrospective analyses employed in this thesis is arguably a strength of the methodology, providing an overall goal of engineering new forms of learning. Cobb et al. (2003) argues that the tendency to seek "'high' theory [which tends] to pass over what may be important details in an effort to paint phenomena in uniform terms"

(p. 13) contrasts with the commitment of DBR to seek theories that "do real work" (p. 10). As the resulting accounts of learning are intimately tied to the context in which they are generated, the need to develop testable conjectures about how those means of support and how an instructional design serves particular goals is important. The process of detailing the theory employed throughout as well as detailed accounts of theory-in-action serves as a methodological strength to the process and products of DBR.

8.5 Conclusion

Existing research has primarily positioned computer science education as problem-solving; this thesis asked whether it could also be a creative way to engage with literacy. A curriculum unit was designed, and three empirical studies were conducted with primary school children. Findings suggested that multimodal resources within Scratch provided multiple opportunities for learners to pursue storytelling and programming goals that were personally meaningful and authentic. To achieve this outcome, the unit was iteratively improved to develop additional structures to help learners whose prior experiences in programming and storytelling were limited. Embedding the designed curriculum unit in a formal classroom setting revealed additional challenges, including technical difficulties and the level of subject knowledge required across the chosen learning strands. Nonetheless, the participating teacher appreciated the value of the approach, finding it useful for his future teaching practices.

Some researchers have been concerned that storytelling and computing are not happy bedfellows: learners' insufficient programming knowledge could limit stories or that the linearity of storytelling will inhibit opportunities to develop crucial programming skills. This thesis suggests that these concerns are overstated. While learners will continue developing their skills in these areas separately, there is also value to be found in integrating them as multimodal composition.

References

Ackermann, E. K. (2004). *Constructing Knowledge and Transforming the World*. https://web.media.mit.edu/~edith/publications/2004-Constructing_Knowledge.pdf

Adams, J. C., & Webster, A. R. (2012). What do students learn about programming from game, music video, and storytelling projects? *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*, 643. https://doi.org/10.1145/2157136.2157319

Ainsworth, S., Prain, V., & Tytler, R. (2011). Drawing To Learn in Science. *Science*, *333*(6046), 1096–1097. https://doi.org/10.1126/science.1204153

Alexander, R. (2009). *Children, their world, their education: Final report and recommendations of the Cambridge Primary Review*.

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research? *Educational Researcher*, *41*(1), 16–25. http://journals.sagepub.com/doi/pdf/10.3102/0013189X11428813

Andrews, R., & Smith, A. (2011). *Developing Writers: Teaching and Learning in the Digital Age*. McGraw-Hill Education.

Bakker, A. (2018). *Design Research in Education: A Practical Guide for Early Career Researchers*. Routledge. https://doi.org/10.4324/9780203701010

Bakker, A., & van Eerde, D. (2015). An Introduction to Design-Based Research with an Example From Statistics Education. In *Approaches to Qualitative Research in Mathematics Education* (pp. 429–466). Springer Netherlands. https://doi.org/10.1007/978-94-017-9181-6_16

Balch, C., Chung, M., & Brennan, K. (2014). *Creative Computing Curriculum*. https://creativecomputing.gse.harvard.edu/

Ballard, E. D., & Haroldson, R. (2021). Analysis of Computational Thinking in Children's Literature for K-6 Students: Literature as a Non-Programming Unplugged Resource. *Journal of Educational Computing Research*, 073563312110040.

https://doi.org/10.1177/07356331211004048

Bannan-Ritland, B. (2003). The Role of Design in Research: The Integrative Learning Design Framework. *Educational Researcher*, *32*(1), 21–24. https://doi.org/10.3102/0013189X032001021

Barab, S. (2014). Design-based research: A methodological toolkit for engineering change. *The Cambridge Handbook of the Learning Sciences, Second Edition*, 151–170. https://doi.org/10.1017/CBO9781139519526.011

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *The Journal of the Learning Sciences*, *13*(1), 1–14. https://doi.org/10.1207/s15327809jls1301_1

Bazalgette, C., & Buckingham, D. (2013). Literacy, media and multimodality: A critical response. *Literacy*, *47*(2), 95–102. https://doi.org/10.1111/j.1741-4369.2012.00666.x

Bearne, E. (2017). Assessing children's written texts: a framework for equity. *Literacy*, *51*(2), 74–83. https://doi.org/10.1111/lit.12105

Bearne, E., & Bazalgette, C. (2010). *Beyond Words: Developing children's response to multimodal texts*. http://www.ukla.org/extracts/Beyond_Words. pdf

Bearne, E., & Reedy, D. (2017). *Teaching Primary English: Subject Knowledge and Classroom Practice*. Routledge.

Bearne, E., & Wolstencroft, H. (2007). *Visual Approaches to Teaching Writing: Multimodal Literacy 5-11*. Sage Publications.

Ben-David Kolikant, Y. (2004). Learning concurrency: Evolution of students' understanding of synchronization. *International Journal of Human Computer Studies*, *60*(2), 243–268. https://doi.org/10.1016/j.ijhcs.2003.10.005

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2016). Building mathematical knowledge with programming: insights from the ScratchMaths project. *Constructionism in Action 2016*, 26–33. http://e-school.kmutt.ac.th/constructionism2016/Constructionism 2016 Proceedings.pdf

Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning

mathematics through programming: A case study of pupils engaging with place value. *International Journal of Child-Computer Interaction*, *16*, 68–76. https://doi.org/10.1016/j.ijcci.2017.12.004

Bers, M. U. (2020). Playgrounds and Microworlds: Learning to Code in Early Childhood. *Designing Constructionist Futures*, 31–38. https://doi.org/10.7551/mitpress/12091.003.0006

Blikstein, P. (2018). *Pre-College Computer Science Education : A Survey of the Field*. https://goo.gl/gmS1Vm.

Blumenfeld, P. C., Kempler, T. M., & Krajcik, J. S. (2005). Motivation and Cognitive Engagement in Learning Environments. In *The Cambridge Handbook of the Learning Sciences* (pp. 475–488). Cambridge University Press. https://doi.org/10.1017/CBO9780511816833.029

Brennan, K. (2014). Constructionism in the Classroom: Three experiments in Disrupting Technocentrism. *Constructionism*, 1–8.

Brennan, K. (2015). Beyond technocentrism: Supporting constructionism in the classroom. *Constructivist Foundations*, *10*(3), 289–296.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *annual American Educational Research Association meeting* (pp. 1–25). https://doi.org/10.1.1.296.6602

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *The Journal of the Learning Sciences*, *2*(2), 141–178. https://doi.org/10.1207/s15327809jls0202_2

Brown, N. C. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart. *ACM Transactions on Computing Education*, *14*(2), 1–22. https://doi.org/10.1145/2602484

Bruce, D. L. (2016). Writing with Visual Images: Examining the Video Composition Processes of High School Students. *Research in the Teaching of English*, *43*(4), 426–450.

Bruner, J. S. (1986). *Actual minds, possible worlds*.

Bruner, J. S. (1990). *Acts of Meaning*. Harvard University Press.

Bruner, J. S. (1991). The Narrative Construction of Reality. *Critical Inquiry*, *18*(1), 41–62.

BT, & Ipsos MORI. (2016). *A New Cornerstone of Modern Primary School Education*.

Buckingham, D. (2007). Media education goes digital: An introduction. *Learning, Media and Technology*, *32*(2), 111–119. https://doi.org/10.1080/17439880701343006

Buckingham, D. (2008). Youth, Identity, and Digital Media. In *Digital Media*. https://doi.org/10.1162/dmal.9780262524834.vii

Buckingham, D., & Burn, A. (2007). Game Literacy in Theory and Practice. *Journal of Educational Multimedia and Hypermedia*, *16*(3), 323–349. http://eprints.ioe.ac.uk/4224/

Buckingham, D., Burn, A., & Cranmer, S. (2005). The Media Literacy of Children and Young People. *Ofcom*, 1–75.

Burke, Q. (2012). The Markings of a New Pencil: Introducing Programming-as-Writing in the Middle School Classroom. *Journal of Media Literacy Education*, *4*(2), 121–135.

Burke, Q., & Kafai, Y. B. (2010). Programming & Storytelling: Opportunities for Learning About Coding & Composition. *Proceedings of the 9th International Conference on Interaction Design and Children - IDC '10*, 348. https://doi.org/10.1145/1810543.1810611

Burke, Q., & Kafai, Y. B. (2012). The writers' workshop for youth programmers. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*, 433. https://doi.org/10.1145/2157136.2157264

Burke, Q., O'Byrne, W. I., & Kafai, Y. B. (2016). Computational Participation: Understanding Coding as an Extension of Literacy Instruction. *Journal of Adolescent and Adult Literacy*, *59*(4), 371–375. https://doi.org/10.1002/jaal.496

Burn, A., Buckingham, D., Parry, B., & Powell, M. (2010). Minding the gaps: teachers' cultures, students' cultures. In D. E. Alvermann & P. Lang (Eds.), *Adolescents' online literacies: Connecting classrooms, digital media, and popular culture*. https://books.google.co.uk/books?id=TLbzcPp91JMC&lr=&source=gbs_navlinks_s

Burnett, C., & Merchant, G. (2018). *New Media in the Classroom: Rethinking Primary Literacy*. SAGE Publications Ltd. https://doi.org/10.4135/9781529714906

Cellan-Jones, R. (2017). *Computing in schools - alarm bells over England's classes*. BBC News. https://www.bbc.co.uk/news/technology-40322796

Cellan-Jones, R. (2019). *Computing in schools in "steep decline."* BBC News. https://www.bbc.co.uk/news/technology-48188877

Clark, C. (2016). *Children's and Young People's Reading in 2015*. https://literacytrust.org.uk/documents/171/2015_01_01_free_research_-_young_peoples_reading_final.pdf_dCLScPc.pdf

Cobb, P. (1994). Constructivism in Mathematics and Science Education. *Educational Researcher*, *23*(7), 4. https://doi.org/10.3102/0013189X023007004

Cobb, P., Confrey, J., diSessa, A. A., Lehrer, R., & Schauble, L. (2003). Design Experiments in Educational Research. *Educational Researcher*, *32*(1), 9–13. https://doi.org/10.3102/0013189X032001009

Cohen, L., & Manion. (2018). *Research Methods in Education*. Routledge. https://doi.org/10.4324/9780203720967

Collins, A. (1992). Toward a Design Science of Education. In *New Directions in Educational Technology* (pp. 15–22). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-77750-9_2

Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. *American Educator*, *15*(3), 6–11.

Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design Research: Theoretical and Methodological Issues. *The Journal of the Learning Sciences*, *13*(1), 15–42. https://doi.org/10.1207/s15327809jls1301_2

Cviko, A., McKenney, S., & Voogt, J. (2014). Teacher roles in designing technology-rich learning activities for early literacy: A cross-case analysis. *Computers & Education*, *72*(4), 68–79. https://doi.org/10.1016/j.compedu.2013.10.014

Department for Education. (2013, September). *The National Curriculum in England: Key*

Stages 1 and 2 framework document. https://www.gov.uk/government/publications/national-curriculum-in-england-secondary-curriculum

Department for Education and Employment. (1999). *The National Curriculum: Handbook for primary teachers in England. Key stages 1 and 2*. http://www.educationengland.org.uk/documents/pdfs/1999-nc-primary-handbook.pdf

Dierdorp, A., Bakker, A., Eijkelhof, H., & van Maanen, J. (2011). Authentic practices as contexts for learning to draw inferences beyond correlated data. *Mathematical Thinking and Learning*, *13*(1–2), 132–151. https://doi.org/10.1080/10986065.2011.538294

diSessa, A. A. (2000). *Changing Minds: Computers, Learning, and Literacy*. MIT Press.

diSessa, A. A. (2004). Metarepresentation: Native Competence and Targets for Instruction. *Cognition and Instruction*, *22*(3), 293–331. https://doi.org/10.1207/s1532690xci2203_2

diSessa, A. A., & Abelson, H. (1986). Boxer: A reconstructible computational medium. *Communications of the ACM*, *29*(9), 859–868. https://doi.org/10.1145/6592.6595

diSessa, A. A., & Cobb, P. (2004). Ontological Innovation and the Role of Theory in Design Experiments. *The Journal of the Learning Sciences*, *13*(1), 77–103. https://doi.org/10.1207/s15327809jls1301_4

diSessa, A. A., & Sherin, B. L. (2000). Meta-representation: An introduction. *Journal of Mathematical Behavior*, *19*(4), 385–398. https://doi.org/10.1016/S0732-3123(01)00051-7

Dowdall, C. (2009). Impressions, improvisations and compositions: reframing children's text production in social network sites. *Literacy*, *43*(2), 91–99.

Dredge, S. (2014, September 22). *Kids coding at school: "When you learn computing, you're thinking about thinking."* The Guardian. https://www.theguardian.com/technology/2014/sep/22/computing-bcs-uk-computing-curriculum

Dyson, A. H. (2008). Staying in the (curricular) lines: Practice constraints and possibilities in childhood writing. *Written Communication*, *25*(1), 119–159.

https://doi.org/10.1177/0741088307309552

Easterday, M. W., Rees Lewis, D. G., & Gerber, E. M. (2018). The logic of design research. *Learning: Research and Practice*, *4*(2), 131–160. https://doi.org/10.1080/23735082.2017.1286367

Eidman-Aadahl, E., Blair, K., Devoss, D. N., Hochman, W., Jimerson, L., Jurich, C., Murphy, S., Rupert, B., Whithaus, C., & Wood, J. (2013). Developing Domains for Multimodal Writing Assessment: The Language of Evaluation , the Language of Instruction Multimodal Assessment Project (MAP) Group. In H. A. McKee & D. N. Devoss (Eds.), *Digital Writing: Assessment and Evaluation*. Computers and Composition Digital Press.

Elbow, P. (1998). *Writing without teachers*. Oxford University Press, USA.

Euler, D. (2017). Design principles as bridge between scientific knowledge production and practice design. *EDeR. Educational Design Research*, *1*(1). https://doi.org/10.15460/eder.1.1.1024

Flower, Linda, & Hayes, John R. (1984). Images, plans, and prose: The representation of meaning in writing. *Written Communication*, *1*(1), 120–160.

Flower, L., & Hayes, J. R. (1981). A Cognitive Process Theory of Writing. *College Composition and Communication*, *32*(4), 365–387. https://doi.org/10.2307/356600?ref=search-gateway:e37adfc5274fe1c8bc59be1009904e52

Forte, A., & Guzdial, M. (2004). Computers for Communication , Not Calculation : Media as a Motivation and Context for Learning Andrea Forte. *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, *4*(C), 40096.1.

Franklin, D., Conrad, P., Boe, B., Nilsen, K., Hill, C., Len, M., Dreschler, G., Aldana, G., Almeida-Tanaka, P., Kiefer, B., Laird, C., Lopez, F., Pham, C., Suarez, J., & Waite, R. (2013). Assessment of computer science learning in a scratch-based outreach program. *SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer*

Science Education, 371–376. https://doi.org/10.1145/2445196.2445304

Franklin, D., Skifstad, G., Rolock, R., Mehrotra, I., Ding, V., Hansen, A., Weintrop, D., & Harlow, D. (2017). *Using Upper-Elementary Student Performance to Understand Conceptual Sequencing in a Blocks-based Curriculum*. 231–236. https://doi.org/10.1145/3017680.3017760

Fu, D., & Shelton, N. R. (2007). Including Students with Special Needs in a Writing Workshop. *Language Arts*, *84*(4), 325–336. http://www.jstor.org/stable/41962201%0Ahttp://about.jstor.org/terms

Gelmini-Hornsby, G. (2012). *Scaffolding children's collaborative story-telling through constructive and interactive story-making*. https://doi.org/10.1113/JP271498

Graham, S., McKeown, D., Kiuhara, S., & Harris, K. R. (2012). A meta-analysis of writing instruction for students in the elementary grades. *Journal of Educational Psychology*, *104*(4), 879–896. https://doi.org/10.1037/a0029185

Graves, D. H. (1983). *Writing: Teachers and children at work*. Heinemann.

Grover, S., Basu, S., & Schank, P. (2018). What We Can Learn About Student Learning From Open-Ended Programming Projects in Middle School Computer Science. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE '18*, 999–1004. https://doi.org/10.1145/3159450.3159522

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Grover, S., & Pea, R. (2018). Computational Thinking: A Competency Whose Time Has Come. *Computer Science Education: Perspectives on Teaching and Learning*, *December*, 1997–2004. http://hub.mspnet.org/index.cfm/33300

Guzdial, M. (2008). Paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25–27. http://dl.acm.org/citation.cfm?id=1378713

Guzdial, M. (2019). Computing Education as a Foundation for 21st Century Literacy. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education -*

SIGCSE '19, 502–503. https://doi.org/10.1145/3287324.3290953

Guzdial, M. (2017). Using Learning Sciences Research To Improve Computing Teaching: Predictions, Subgoals, And Parsons. *Computing At School 9th Conference for Teachers*, 1–30.

Guzdial, M., & Forte, A. (2005). Design process for a non-majors computing course. *ACM SIGCSE Bulletin*, *37*, 361. https://doi.org/10.1145/1047344.1047468

Hall, K., Cremin, T., Comber, B., & Moll, L. C. (2013). *International Handbook of Research on Children's Literacy, Learning and Culture* (K. Hall, T. Cremin, B. Comber, & L. C. Moll (eds.)). John Wiley & Sons. https://doi.org/10.1002/9781118323342.index

Harel, I., & Papert, S. (1991). *Constructionism*. Ablex Publishing.

Herrington, J., McKenney, S., Reeves, T., Oliver, R., Jerry, J Herrington, S. M. T. R., & Oliver, R. (2007). Design-based research and doctoral students: Guidelines for preparing a dissertation proposal. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 4089–4097. https://doi.org/10.1017/CBO9781107415324.004

Howell, E. (2018). Scaffolding multimodality: writing process, collaboration and digital tools. *English Teaching*, *17*(2), 132–147. https://doi.org/10.1108/ETPC-05-2017-0053

Howell, E., Butler, T., & Reinking, D. (2017). Integrating multimodal arguments into high school writing instruction. *Journal of Literacy Research*. https://doi.org/10.1177/1086296X17700456

Howells, C., & Robertson, J. (2012). Children as game designers: New narrative opportunities. *Virtual Literacies: Interactive Spaces for Children and Young People*, 142–160. https://doi.org/10.4324/9780203096468

Hubber, P., Tytler, R., & Haslam, F. (2010). Teaching and learning about force with a representational focus: Pedagogy and teacher change. *Research in Science Education*, *40*(1), 5–28. https://doi.org/10.1007/s11165-009-9154-9

Hull, G. A., & Nelson, M. E. (2005). Locating the semiotic power of multimodality. *Written Communication*, *22*(2), 224–261. https://doi.org/10.1177/0741088304274170

Israel, M., & Lash, T. (2020). From classroom lessons to exploratory learning progressions: mathematics + computational thinking. *Interactive Learning Environments*, *28*(3), 362–382. https://doi.org/10.1080/10494820.2019.1674879

Jewitt, C. (2008). Multimodality and Literacy in School Classrooms. *Review of Research in Education*, *32*(1), 241–267. https://doi.org/10.3102/0091732X07310586

Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. Routledge.

Kafai, Y. B. (2005). Constructionism. In *The Cambridge Handbook of the Learning Sciences* (pp. 35–46). Cambridge University Press. https://doi.org/10.1017/CBO9780511816833.004

Kafai, Y. B., & Burke, Q. (2014). Connected Code: Why Children Need to Learn Programming. In *MIT Press* (p. 200). The MIT Press.

Kafai, Y. B., & Burke, Q. (2015). Constructionist Gaming: Understanding the Benefits of Making Games for Learning. *Educational Psychologist*, *50*(4), 313–334. https://doi.org/10.1080/00461520.2015.1124022

Kafai, Y. B., Franke, M. L., Ching, C. C., & Shih, J. C. (1998). Game design as an interactive learning environment for fostering students' and teachers' mathematical inquiry. *International Journal of Computers for Mathematical Learning*, *3*(2), 149–184. https://doi.org/10.1023/A:1009777905226

Kelleher, C., & Pausch, R. (2005). Lowering the Barriers to Programming : a survey of programming environments and languages for novice programmers. *Science*, *37*(2), 83–137. https://doi.org/10.1145/1089733.1089734

Kelleher, C., & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, *50*(7), 58–64. https://doi.org/10.1145/1272516.1272540

Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling Alice motivates middle school girls to learn computer programming. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '07*, 1455. https://doi.org/10.1145/1240624.1240844

Kellogg, R. T. (2008). Training writing skills: A cognitive developmental perspective. *Journal of Writing Research*, *1*(1).

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why Minimal Guidance During Instruction Does Not Work. *Educational Psychologist*, *41*(2), 87–98. https://doi.org/10.1207/s15326985ep4102

Kress, G. (1997). Before writing. Rethinking the paths to literacy. In *Routledge*. https://doi.org/drake LB1576.K67 1997; uofr rhees

Kress, G. (2003). *Literacy in the New Media Age*. Routledge.

Labov, W. (1972). *Sociolinguistic Patterns*. University of Pennsylvania Press. https://doi.org/10.2307/412715

Langer, J. A., & Applebee, A. N. (1987). *How Writing Shapes Thinking: A Study of Teaching and Learning*.

Larke, L. R. (2019). Agentic neglect: Teachers as gatekeepers of England's national computing curriculum. *British Journal of Educational Technology*, *50*(3), 1137–1150. https://doi.org/10.1111/bjet.12744

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation* (Issue 1).

Lee, H. S., & Anderson, J. R. (2013). Student learning: What has instruction got to dowith it? *Annual Review of Psychology*, *64*, 445–469. https://doi.org/10.1146/annurev-psych-113011-143833

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

Lehrer, R., Lee, M., & Jeong, A. (1999). Reflective Teaching of Logo. *Journal of the Learning Sciences*, *8*(2), 245–289. https://doi.org/10.1207/s15327809jls0802_3

Lewis, D. G. R., Carlson, S. E., Riesbeck, C. K., Lu, K. J., Gerber, E. M., & Easterday, M. W. (2020). The Logic of Effective Iteration in Design-Based Research Iteration Process and Challenges in Education Design Research. *Proceedings of the International*

Conference of the Learning Sciences (ICLS 2020), 1149–1156.

Lincoln, Y. S., & Guba, E. G. (1986). But is it rigorous? Trustworthiness and authenticity in naturalistic evaluation. *New Directions for Program Evaluation*, *1986*(30), 73–84. https://doi.org/10.1002/ev.1427

Liu, C. C., Chen, H. S. L., Shih, J. L., Huang, G. T., & Liu, B. J. (2011). An enhanced concept map approach to improving children's storytelling ability. *Computers and Education*, *56*(3), 873–884. https://doi.org/10.1016/j.compedu.2010.10.029

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

Maloney, J., Peppler, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. *SIGCSE '08 Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, 367–371. https://doi.org/10.1145/1352135.1352260

Marsh, J. (2004). Popular culture, new media and digital literacy in early childhood. In *Popular Culture, New Media and Digital Literacy in Early Childhood*. https://doi.org/10.4324/9780203420324

Marsh, J. (2010). *Creativity, Culture and Education Series Childhood, culture and creativity: a literature review Creativity, Culture and Education (CCE) Literature Reviews*. http://www.creativityculture education.org/research-impact/literature-reviews/%5Cnwww.hpm.uk.com%5Cnwww.tangerinelimited.com

Mayer, R. E. (2004). Should There Be a Three-Strikes Rule against Pure Discovery Learning? The Case for Guided Methods of Instruction. *American Psychologist*, *59*(1), 14–19. https://doi.org/10.1037/0003-066X.59.1.14

McCabe, A. (1997). *Developmental and cross-cultural aspects of children's narration.*

McCabe, A., & Peterson, C. (1984). *What Makes A Good Story? 13*(6), 457–480. https://doi.org/10.1007/BF01068179

McCabe, A., & Peterson, C. (1991). *Developing Narrative Structure*.

https://doi.org/10.1037/032742

McGrail, E., & Behizadeh, N. (2017). K-12 multimodal assessment and interactive audiences: An exploratory analysis of existing frameworks. *Assessing Writing*, *31*, 24–38. https://doi.org/10.1016/j.asw.2016.06.005

McLuhan, M. (1964). *Understanding media: The extensions of man*. McGraw-Hill.

Medwell, J., Wray, D., Minns, H., Griffiths, V., & Coates, L. (2009). *Primary English: Teaching Theory and Practice*. Learning Matters. http://books.google.co.uk/books?id=OSzbDgAAQBAJ&printsec=frontcover&dq=intitle:Primary+English+Teaching+Theory+and+Practice&hl=&cd=1&source=gbs_api

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (Moti). (2013). Learning computer science concepts with Scratch. *Computer Science Education*, *23*(3), 239–264. https://doi.org/10.1080/08993408.2013.832022

Merchant, G. (2007). Writing the future in the digital age. *Literacy Theory and Research*, *41*(3), 118–128.

Millard, E. (2006). Transformative pedagogy: Teachers creating a literacy of fusion. In *Travel Notes from the New Literacy Studies* (pp. 234–253). Multilingual Matters.

Millard, E., & Marsh, J. (2001). Words with Pictures: The Role of Visual Literacy in Writing and its Implication for Schooling. *Literacy (Formerly Reading)*, *35*(2), 54–61. https://doi.org/10.1111/1467-9345.00162

Mills, K. A. (2010). A Review of the "Digital Turn" in the New Literacy Studies. *Review of Educational Research*, *80*(2), 246–271. https://doi.org/10.3102/0034654310364401

Mintrop, R. (2020). *Design-based school improvement: A practical guide for education leaders*. Harvard Education Press.

MIT Scratch Team. (2020). *Scratch Lab*. https://lab.scratch.mit.edu/text/

Moje, E. B. (2008). Foregrounding the Disciplines in Secondary Literacy Teaching and Learning: A Call for Change. *Journal of Adolescent & Adult Literacy*, *52*(2), 96–107. https://doi.org/10.1598/JAAL.52.2.1

Moje, E. B., Ciechanowski, K. M., Kramer, K., Ellis, L., Carrillo, R., & Collazo, T. (2004).

Working toward third space in content area literacy: An examination of everyday funds of knowledge and Discourse. *Reading Research Quarterly*, *39*(1), 38–70. https://doi.org/10.1598/RRQ.39.1.4

Myhill, D., & Cremin, T. (2013). Writing voices: Creating communities of writers. In *Writing Voices: Creating Communities of Writers*. https://doi.org/10.4324/9780203803332

National Research Council. (2011). Report of a Workshop on the Pedagogical Aspects of Computational Thinking. *Report of a Workshop on the Pedagogical Aspects of Computational Thinking*, 176. https://doi.org/10.17226/13170

Nieveen, N., Akker, J. Van Den, Gravemeijer, K., McKenney, S., Van Den Akker, J., Phillips, D., Musta'amal, A., den Akker, J., Gravemeijer, K., McKenney, S., & Nieveen, N. (2006). Educational design research. *Educational Design …*, 67–90. https://doi.org/10.1111/j.1467-8535.2008.00855_1.x

Nixon, R. (2012). Teaching narrative writing using comics: Delainey and Rasmussen, the creators of Betty, share their composing strategies as rich literacy resources for elementary teachers. *Literacy*, *46*(2), 81–93. https://doi.org/10.1111/j.1741-4369.2011.00580.x

Noss, R., & Hoyles, C. (1996). Windows on mathematical meanings: learning cultures and computers. In *Mathematics education library* (Issue 17). https://doi.org/10.1007/978-94-009-1696-8

O'Neill, D. K. (2016). Understanding Design Research–Practice Partnerships in Context and Time: Why Learning Sciences Scholars Should Learn From Cultural-Historical Activity Theory Approaches to Design-Based Research. *Journal of the Learning Sciences*, *25*(4), 497–502. https://doi.org/10.1080/10508406.2016.1226835

Olson, D. R., & Torrance, N. (2010). *The Cambridge Handbook of Literacy*. Cambridge University Press.

Owston, R., Wideman, H., Ronda, N. S., & Brown, C. (2009). Computer game development as a literacy activity. *Computers and Education*, *53*(3), 977–989.

https://doi.org/10.1016/j.compedu.2009.05.015

Pantaleo, S. (2017). Exploring the artwork of young students' multimodal compositions. *Education 3-13*, *45*(1), 17–35. https://doi.org/10.1080/03004279.2015.1033438

Papert, S. (1980). Mindstorms: children, computers, and powerful ideas. In *New Ideas in Psychology* (Vol. 1). Basic Books Inc. https://doi.org/10.1016/0732-118X(83)90034-X

Papert, S. (1987). Information technology and education: Computer criticism vs. technocentric thinking. *Educational Researcher*, *16*(1), 22–30.

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology*, *2*(2), 137–168. https://doi.org/10.1016/0732-118X(84)90018-7

Pea, R. D., Soloway, E., & Spohrer, J. C. (1987). *The Buggy Path to the Development of Programming Expertise*. *9*(1).

Pelletier, C., Burn, A., & Buckingham, D. (2010). Game design as textual poaching: Media literacy, creativity and game-making. *E-Learning*, *7*(1), 90–107. https://doi.org/10.2304/elea.2010.7.1.90

Penrod, D. (2005). *Composition in convergence: The impact of new media on writing assessment*. Routledge.

Penuel, W. R., Reiser, B. J., Novak, M., McGill2, T., Frumin, K., Horne, K. Van, Sumner, T., & Watkins, D. A. (2018). Using Co-Design to Test and Refine a Model for Three-Dimensional Science Curriculum that Connects to Students' Interests and Experiences. *Annual Meeting of the American Educational Research Association*.

Penuel, W. R., Roschelle, J., & Shechtman, N. (2007). Designing Formative Assessment Software With Teachers: an Analysis of the Co-Design Process. *Research and Practice in Technology Enhanced Learning*, *02*(01), 51–74. https://doi.org/10.1142/s1793206807000300

Peppler, K. A., & Kafai, Y. B. (2007). From SuperGoo to Scratch: Exploring creative digital media production in informal learning. *Learning, Media and Technology*, *32*(2), 149–166. https://doi.org/10.1080/17439880701343337

Peterson, C., & McCabe, A. (1983). *Developmental Psycholinguistics*. Springer US. https://doi.org/10.1007/978-1-4757-0608-6

Piaget, J. (1926). *The language and thought of the child*. Kegan Paul, Trench & Trubner.

Plomp, T., & Nieveen, N. (2013). *Educational design research: Part B: Illustrative cases*. SLO—Netherlands Institute for Curriculum Institute. http://international.slo.nl/edr

Prado, Y., Jacob, S., & Warschauer, M. (2021). Teaching computational thinking to exceptional learners: lessons from two inclusive classrooms. *Computer Science Education*, *00*(00), 1–25. https://doi.org/10.1080/08993408.2021.1914459

Prain, V., & Tytler, R. (2012). Learning Through Constructing Representations in Science: A framework of representational construction affordances. *International Journal of Science Education*, *34*(17), 2751–2773. https://doi.org/10.1080/09500693.2011.626462

Proctor, C., & Blikstein, P. (2017). Interactive fiction. *Proceedings of the 2017 Conference on Interaction Design and Children*, 555–560. https://doi.org/10.1145/3078072.3084324

QCA/UKLA. (2004). *More than Words: Multimodal texts in the classroom*. Qualifications and Curriculum Authority.

QCA/UKLA. (2005). *More than words 2: Creating stories on page and screen*. Qualifications and Curriculum Authority.

Qian, Y., & Lehman, J. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Trans. Comput. Educ.*, *18*(1), 1:1--1:24. https://doi.org/10.1145/3077618

Reeves, T. (2006). Design research from a technology perspective. In *Educational design research* (pp. 64–78). Routledge.

Reinking, D., & Bradley, B. A. (2004). Connecting research and practice using formative and design experiments. In M. H. Mallette & N. K. Duke (Eds.), *Literacy research methodologies* (pp. 149–169). The Guilford Press.

Reinking, D., & Bradley, B. A. (2007). *On Formative and Design Experiments: Approaches to Language and Literacy Research*. Teachers College Press.

Reinking, D., & Watkins, J. (2000). A Formative Experiment Investigating the Use of Multimedia Book Reviews to Increase Elementary Students' Independent Reading. *Reading Research Quarterly*, *35*(3), 384–419. https://doi.org/10.1598/rrq.35.3.4

Reiser, B. J., & Tabak, I. (2014). Scaffolding. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (pp. 44–62). Cambridge University Press. https://doi.org/10.1017/CBO9781139519526.005

Resnick, M. (2002). Rethinking Learning in the Digital Age. In *The Global Information Technology Report 2001-2002: Readiness for the Networked World*. https://doi.org/10.1353/cj.2007.0001

Resnick, M., Silverman, B., Kafai, Y. B., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., & Silver, J. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60. https://doi.org/10.1145/1592761.1592779

Robertson, J. (2005). Adventure Author: An Authoring Tool for 3D Virtual Reality Story Construction. *Proceedings of the AIED-05 Workshop on Narrative Learning Environments*, 63–69.

Robertson, J., & Good, J. (2003). Using a Collaborative Virtual Role-Play Environment to Foster Characterisation in Stories. *Journal of Interactive Learning Research*, *14*(1), 5–29.

Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers and Education*, *50*(2), 559–578. https://doi.org/10.1016/j.compedu.2007.09.020

Robson, C., & McCartan, K. (2016). *Real World Research* (Vol. 2016). Wiley. https://doi.org/10.1002/rwm3.20329

Rose, J. (2009). *Independent Review of the Primary Curriculum: Final Report*. Department for Children, Schools and Families.

Roth, F. P. (2000). Narrative Writing. In *Topics in Language Disorders* (Vol. 20, Issue 4, pp. 15–28). https://doi.org/10.1097/00011363-200020040-00004

Rubin, M. J. (2013). The effectiveness of live-coding to teach introductory programming. *SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 651–656. https://doi.org/10.1145/2445196.2445388

Rumelhart, D. E. (1980). On Evaluating Story Grammars. *Cognitive Science*, *4*(3), 313–316. https://doi.org/10.1207/s15516709cog0403_5

Saddler, B., Moran, S., Graham, S., & Harris, K. R. (2004). Preventing Writing Difficulties: The Effects of Planning Strategy Instruction on the Writing Performance of Struggling Writers. *Students Who Are Exceptional and Writing Disabilities*, *2835*, 3–17. https://doi.org/10.1207/s15327035ex1201_2

Saddler, B., Moran, S., Graham, S., & Harris, K. R. (2018). Preventing Writing Difficulties: The Effects of Planning Strategy Instruction on the Writing Performance of Struggling Writers. *Students Who Are Exceptional and Writing Disabilities*, *September 2013*, 3–17. https://doi.org/10.4324/9780203063293-1

Sandoval, W. A. (2004). Developing Learning Theory by Refining Conjectures Embodied in Educational Designs. *Educational Psychologist*, *39*(4), 213–223. https://doi.org/10.1207/s15326985ep3904_3

Sandoval, W. A. (2014). Conjecture Mapping: An Approach to Systematic Educational Design Research. *The Journal of the Learning Sciences*, *23*(1), 18–36. https://doi.org/10.1080/10508406.2013.778204

Scardamalia, M., & Bereiter, C. (1987). Knowledge telling and knowledge transforming in written composition. *Advances in Applied Psycholinguistics*, *2*, 142–175.

Scardamalia, M., & Bereiter, C. (2014). Knowledge Building and Knowledge Creation. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (pp. 397–417). Cambridge University Press. https://doi.org/10.1017/CBO9781139519526.025

Schank, R. C. (1995). *Tell me a story: Narrative and intelligence*. Northwestern University Press.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A

theoretical framework. *Educ Inf Technol*, *18*(2), 351–380. https://doi.org/10.1007/s10639-012-9240-x

Sengupta, P., Voss Farris, A., & Wright, M. (2012). From agents to continuous change via aesthetics: Learning mechanics with visual agent-based computational modeling. *Technology, Knowledge and Learning*, *17*(1–2), 23–42. https://doi.org/10.1007/s10758-012-9190-9

Sentance, S., Barendsen, E., & Schulte, C. (2018). *Computer Science Education : Perspectives on Teaching and Learning in School*. Bloomsbury Publishing.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469–495. https://doi.org/10.1007/s10639-016-9482-0

Sentance, S., Dorling, M., McNicol, A., & Crick, T. (2012). Grand challenges for the UK. *Proceedings of the 7th Workshop in Primary and Secondary Computing Education on - WiPSCE '12*, *April 2016*, 82. https://doi.org/10.1145/2481449.2481469

Sentance, S., Waite, J., & Kallia, M. (2019). *Teachers ' Experiences of using PRIMM to Teach Programming in*. 476–482.

Severance, S., Penuel, W. R., Sumner, T., & Leary, H. (2016). Organizing for Teacher Agency in Curricular Co-Design. *Journal of the Learning Sciences*, *25*(4), 531–564. https://doi.org/10.1080/10508406.2016.1207541

Simon, M. A. (1995). Reconstructing Mathematics Pedagogy from a Constructivist Perspective. *Journal for Research in Mathematics Education*, *26*(2), 114–145. https://doi.org/10.5951/jresematheduc.26.2.0114

Simon, M. A., & Tzur, R. (2004). Explicating the Role of Mathematical Tasks in Conceptual Learning: An Elaboration of the Hypothetical Learning Trajectory. *Mathematical Thinking and Learning*, *6*(2), 91–104. https://doi.org/10.1207/s15327833mtl0602_2

Smith, B. E. (2017). Composing across modes: a comparative analysis of adolescents' multimodal composing processes. *Learning, Media and Technology*, *42*(3), 259–278. https://doi.org/10.1080/17439884.2016.1182924

Smith, S., & Burrow, L. E. (2016). Programming Multimedia Stories in Scratch to Integrate Computational Thinking and Writing with Elementary Students. *Journal of Mathematics Education*, *9*(2), 119–131. http://www.educationforatoz.net/images/2016_Commentary_6.pdf

Sorva, J. (2013). Visual Program Simulation in Introductory Program Education. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9). https://doi.org/10.1007/s13398-014-0173-7.2

Stake, R. E. (1995). *The art of case study research*. Sage Publications. http://books.google.com/books?id=sIMOAQAAMAAJ

Stein, N. L., & Glenn, C. G. (1979). An analysis of story comprehension in elementary school children. *New Directions in Discourse Processing*, *January 1979*, 53–120.

Stein, N. L., & Trabasso, T. (1981). What's in a Story: An Approach to Comprehension and Instruction. In *Center for the Study of Reading Technical Report*.

Tabak, I. (2004). Synergy: A complement to emerging patterns of distributed scaffolding. *Journal of the Learning Sciences*, *13*(3), 305–335. https://doi.org/10.1207/s15327809jls1303_3

Tatar, D. (2007). The design tensions framework. *Human-Computer Interaction*, *22*(4), 413–451. https://doi.org/10.1080/07370020701638814

The Design-Based Research Collective. (2003). Design-Based Research: An Emerging Paradigm for Educational Inquiry. *Educational Researcher*, *32*(1), 5–8. https://doi.org/10.3102/0013189X032001005

The Royal Society. (2012). *Shut down or restart? The way forward for computing in UK schools*. The Royal Society. https://doi.org/10.1088/2058-7058/25/07/21

The Royal Society. (2017). *After the reboot: computing education in UK schools*. The Royal Society.

Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). From computational thinking to computational action. *Communications of the ACM*, *62*(3), 34–36. https://doi.org/10.1145/3265747

Tissenbaum, M., Weintrop, D., Holbert, N., & Clegg, T. (2021). *The case for alternative endpoints in computing education*. *January*, 1–14. https://doi.org/10.1111/bjet.13072

Tracy, S. J. (2013). *Qualitative Research Methods*. John Wiley & Sons.

Turkle, S. (2007). *Evocative objects*. MIT Press. https://doi.org/10.1017/CBO9781107415324.004

Tyner, K. (2014). *Literacy in a digital world: Teaching and learning in the age of information*. Routledge.

Tytler, R., & Hubber, P. (2013). *Constructing representations to learn science*. Sense Publishers. https://doi.org/10.1007/978-3-319-16450-2_9

UKLA. (2004). *More than Words: Multimodal texts in the classroom*. Qualifications and Curriculum Authority.

Ukrainetz, T. A. (2006). Teaching Narrative Structure: Coherence, Cohesion, and Captivation. *Contextualized Language Intervention: Scaffolding PreK-12 Literacy Achievement*, 195–246.

Ukrainetz, T. A., Justice, L. M., Kaderavek, J. N., Eisenberg, S. L., Gillam, R. B., & Harm, H. M. (2005). The development of expressive elaboration in fictional narratives. *Journal of Speech, Language, and Hearing Research*, *48*(6), 1363–1377. https://doi.org/10.1044/1092-4388(2005/095)

Van den Akker, J. (2006). *Educational Design Research* (J. van den Akker, K. Gravemeijer, S. McKenney, & N. Nieveen (eds.); 1st ed.). Routledge. https://doi.org/10.4324/9780203088364

Vee, A. (2013). Understanding Computer Programming as a Literacy. *Literacy in Composition Studies*, *1*(2), 42–64. https://doi.org/10.21623/1.1.2.4

Vincent, J. (2006). Children writing: Multimodality and assessment in the writing classroom. *Literacy*, *40*(1), 51–57. https://doi.org/10.1111/j.1467-9345.2006.00426.x

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.

Wahleithner, J. M. (2014). The national writing project's multimodal assessment project:

Development of a framework for thinking about multimodal composing. *Computers and Composition*, *31*(1), 79–86. https://doi.org/10.1016/j.compcom.2013.12.004

Walsh, C. (2010). Systems-based literacy practices: Digital games research, gameplay and design. *Australian Journal of Language and Literacy*, *14*(1), 1. https://doi.org/Article

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Weintrop, D., Hansen, A. K., Harlow, D. B., & Franklin, D. (2018). Starting from Scratch: Outcomes of Early Computer Science Learning Experiences and Implications for What Comes Next. *Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER '18*, 142–150. https://doi.org/10.1145/3230977.3230988

Wing, J. M. (2006). Computational Thinking. *Viewpoint*, *49*(3), 33–35. http://portal.acm.org/citation.cfm?doid=1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (2011). Computational Thinking and Expository Writing in the Middle School. *ACM Transactions on Computing Education*, *11*(2), 1–22. https://doi.org/10.1145/1993069.1993073

Wood, D., Bruner, J. S., & Ross, G. (1976). The Role of Tutoring in Problem Solving. *Journal of Child Psychology and Psychiatry*, *17*(2), 89–100. https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Wyatt-Smith, C., & Kimber, K. (2005). Valuing and evaluating student-generated online multimodal texts: Rethinking what counts. *English in Education*, *39*(2), 22–43. https://doi.org/10.1111/j.1754-8845.2005.tb00615.x

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching

science. *Computer Science Education*, *28*(4), 371–400.

https://doi.org/10.1080/08993408.2018.1560550

Appendix 1: Intervention lesson plans

Lesson 1: Programming characters

Learning goals

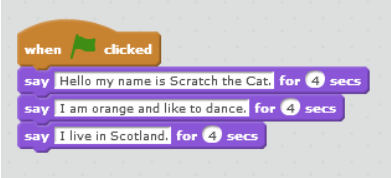
- Explore the concepts of sequencing (i.e., creating sequences) and execution (i.e., wait blocks) by programming sprites
- Use sprites, backdrops, and blocks to establish character(s) and setting(s)

Activities

Discussion (5 minutes)

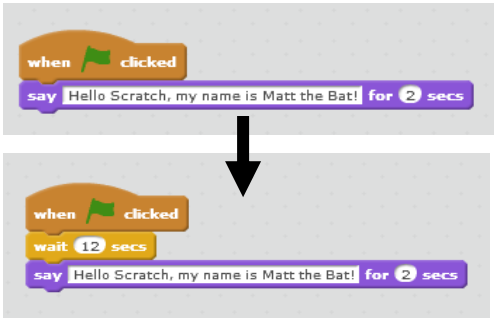
1. Discuss students' prior uses of programming and using computers to create digital projects (e.g., making games, animations, presentations). Explain that our project over the next six weeks will be to use Scratch to create different stories and learn how to program and tell better stories.

Programming a character (20 minutes)



2. Start off by modelling adding a sprite to Scratch and writing a sequence to enable him/her to speak. Have students (in pairs) create their own sprite and program a simple introduction (e.g., "My name is…")
3. Demonstrate adding additional lines of dialogue (seen above) and discuss *readability* of each line (i.e., how long should the sprite speak for/how long it takes the audience to read the text/how much text is on each line/etc.) Have students create their own and assist where necessary.

Programming multiple characters (20 minutes)

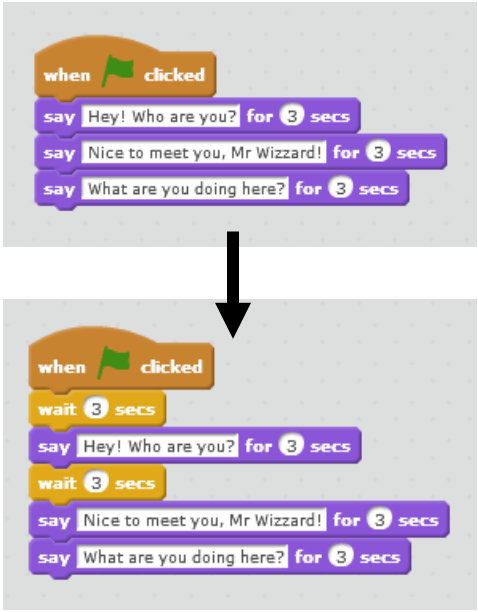


4. Model adding another sprite and add dialogue to create a conversation. Run the program and ask students to note problems (e.g., characters speak at the same time/characters don't respond to one another/etc.)
5. Explain how the second sprite needs to react to the first sprite (by waiting/by 'listening' to the other sprite/by answering the other sprite's questioning). Model adding a wait block and changing the seconds to match the first sprite (illustrated on the right) so that no sprite interrupts one another. Have students create a conversation between their two sprites. Encourage them to create a realistic conversation where sprites respond to one another appropriately (e.g., "I'm fine!" in response to "How are you today?") Assist students where necessary. Discuss any issues that arise.
6. Challenge students to create multiple lines of dialogue and to time their accordingly using multiple wait blocks.

Discussion (10 minutes)

7. Ask students to think about some of the challenges they encountered any why it was necessary to work hard on the wait blocks in their code. Discuss examples of student practice or ask for suggestions from students. Ask students to think about their favourite comic book, picture book, film TV show or any story that is told through images.

Lesson 2: Sequencing story elements



Learning goals

- Debug an existing Scratch project
- Extend a story by creating additional episodes
- Generate ideas for an original multimodal story

Activities

Introduction (5 minutes)

1. Discuss last week's lesson. Make a conversation from scratch and ask students to look for problems (bugs). Model adding wait blocks to correct the sequence.
2. Have students find a problem in the boy's reaction (he says, 'where did you come from?' but the wizard was always on the stage). Model using 'show' blocks to achieve this. Let students discover that a 'hide' block also needs to be used at the beginning of the programme.

Introduction (15 minutes)

3. Ask students if they had brought any visual stories (e.g., comics, animated films, picture books) that they wished to use in their stories. Using the PowerPoint as a stimulus, complete the first section of the storyboard. Model completing the section and generating story ideas.

Adding backdrops (12 minutes)

4. Display the sample project again (without bugs). Discuss what might happen in the story next (e.g., Why is the wizard there? Where will he take the boy? How can we show this?). Model adding additional speech blocks and a change backdrop block to change setting. Ask 'What effect does the 'change backdrop' block have on our story? (e.g., change setting, new episode in story). Have students use different backdrop(s) to continue the story.

Adding costumes (12 minutes)

5. Model costumes to produce different effects (e.g., a new story setting, a character reaction). Discuss, for example, how the costume blocks can show us a character's emotions without telling us. Ask 'What effect does the 'change costume block have on our story? (e.g., show character reaction, change expression). Ask students to use these in their projects or to choose other sprites with similar costumes.

Resetting scripts (12 minutes)

6. When students have programmed their own examples, demonstrate the program not resetting (e.g., the backdrop not resetting to the first location). Challenge students to figure out the solution. Model an example solution at the end (illustrated on the right).

Discussion (5 minutes)

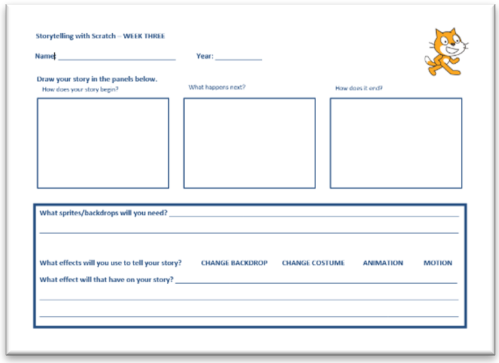
7. Discuss students' extensions to the story project and where they took their stories. Ask students what they found easy/difficult to do and whether they had to change any of their ideas/why (e.g., couldn't achieve a particular effect/came up with a better story idea). Complete the questions on the storyboard.

Lesson 3: Animating story elements

Learning goals

- Explore the concept of repetition and program reset through animating and moving sprites
- Create additional narrative episodes using Scratch tools
- Plan an original multimodal story

Activities


Storytelling with Scratch – WEEK THREE

Name: _____ Year: _____

Draw your story in the panels below.

How does your story begin? | What happens next? | How does it end?

What sprites/backdrops will you need? _____

What effects will you use to tell your story? CHANGE BACKDROP CHANGE COSTUME ANIMATION MOTION

What effect will that have on your story? _____

Introduction (15 minutes)

1. Discuss lessons to date (e.g., what was achieved/what blocks have been used/what effects have been achieved/what stories students would like to tell visually). Model showing a complete Scratch story (3 episodes) and have students think about their story and how to break it up episodically. Complete the first section of the storyboard (top half).

Using motion blocks (15 minutes)

2. Explain how Scratch can used to animate characters and objects through movement. Model programming the character to run to the basketball hoop, jump towards it and land back on the ground (using the glide block). Have students create a similar effect on their projects.

Using a looping costume change (15 minutes)

3. Discuss the use of changing costumes and how it can create animated effects (e.g., movement, talking, running). Model adding a loop of changing costumes to animate the character running.
4. Demonstrate how to flip a character (costume change) before walking back to the right of the court. Model programming Avery so that he walks back to the starting point automatically when you click the green flag. Stop the running animation when Avery reaches the end.

Extension (10 minutes, if required)

5. Discuss what might happen in the story next (e.g., What complication will happen in this story? What obstacle will he encounter?). Model using change backdrop/costume blocks, an additional character, and a plot complication (e.g., Avery meets another boy who challenges him to a basketball game). Have students experiment with the two effects above to continue the story.

Discussion (15 minutes)

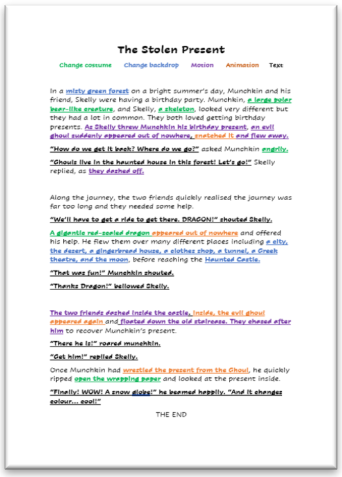
6. Have students look back on their story plans and think about where they could use animation, motion and other visual effects to enhance their stories. Question them on what effect they would like to achieve and whether certain effects are more suited to their story (e.g., What does that effect tell the reader? Is it better to tell or show them? etc.) Ask students to consider where animation/motion could be used to enhance their extended story.

Lesson 4: Creating an original multimodal story

Learning goals

- Outline the multimodal elements of an original story
- Consider the use of multimodal choices and their effect on story
- Revisit core computing concepts

Activities



The Stolen Present

Change costume Change backdrop Motion Animation Text

In a misty green forest on a bright summer's day, Munchkin and his friend, Skelly were having a birthday party. Munchkin, a large polar bear-like creature, and Skelly, a skeleton, looked very different but they had a lot in common. They both loved getting birthday presents. As Skelly threw Munchkin his birthday present, an evil ghoul suddenly appeared out of nowhere, snatched it and flew away.

"How do we get it back? Where do we go?" asked Munchkin angrily.

"Ghouls live in the haunted house in this forest! Let's go!" Skelly replied, as they dashed off.

Along the journey, the two friends quickly realised the journey was far too long and they needed some help.

"We'll have to get a ride to get there, DRAGON!" shouted Skelly.

A gigantic red-scaled dragon appeared out of nowhere and offered his help. He flew them over many different places including a city, the desert, a gingerbread house, a clothes shop, a tunnel, a Greek theatre, and the moon, before reaching the Haunted Castle.

"That was fun!" Munchkin shouted.

"Thanks Dragon!" bellowed Skelly.

The two friends dashed inside the castle. Inside, the evil ghoul appeared again and flooted down the old staircase. They chased after him to recover Munchkin's present.

"There he is!" roared munchkin.

"Get him!" replied Skelly.

Once Munchkin had wrestled the present from the Ghoul, he quickly ripped open the wrapping paper and looked at the present inside.

"Finally! WOW! A snow globe!" he beamed happily. "And it changes colour... cool!"

THE END

Introduction (20 minutes)

1. Discuss lessons to date and ask students what we've been doing ('storytelling' or 'programming to tell stories'). Read students the sample text ('The Stolen Present') and show them its features (words/no pictures, a title, paragraphs to show each episode, language to describe the character's emotions and state, dialogue, etc.) Then, show them the Scratch version and ask students to think about the differences between the two (despite being the same story). Write a list on the whiteboard of their ideas.

Outline story elements (30 minutes)

2. Model saving a project under a student's name using the PowerPoint guide (using Scratch Online). Ask students to draft the outline of their project using a series of backdrops, sprites and, if time, programme the first episode using the blocks previously covered. Support students where necessary.

Discussion (10 minutes)

3. Open the last slide on the PowerPoint ('Special effects'). Ask students to think about the four effects we've looked at and give an example of each one. Show them the sample project again and have them look out for each of the effects, one at a time.
4. Use the annotated story (shown on the right) to question students on how you might represent certain aspects of their stories in Scratch. For example:
    - Use **Backdrop** blocks to show setting change
    - Use **Costume** blocks to show character, character actions, emotions, reactions
    - Use **Motion** blocks to show movement, chasing, running, walking, a journey
    - Use **Animation** ('show/hide', looping costumes) to show movement, introduce or remove characters or objects
    - Use **Text** to show dialogue
5. As they spot each one, ask them to think about the effect it had on them as a reader. What did it tell/show you? How is that different to when you write a story using words? Which do you think is more effective? Why?.

Lesson 5: Creating an original multimodal story

Learning goals

- Revisit the concepts of sequencing, repetition and concurrency
- Draft episodes in a multimodal text
- Consider the use of multimodal choices and their effect on story

Activities

Introduction (10 minutes)

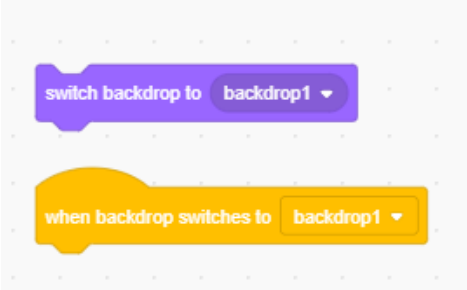
6. Introduce tasks for the lesson (i.e., to code the first and second episodes of their story). Review the multimodal effects using the Scratch project (change costumes/backdrops/animation/ motion/text). Ask students to think of where they could use these effects in their stories. Show the first task/blocks using the PowerPoint.

Writing the first episode (20 minutes)

7. Ask students to write the first episode of their story using blocks of code and the sprites and backdrops already in their projects. Support students where necessary. Where students encounter issues/misconceptions/errors, record these for a whole-class teaching moment later.

Managing different episodes (10 minutes)

8. As students begin to write their second episode, introduce and model the 'When backdrop changes to X' block and show how it can be used to manage sprites and backdrops across multiple episodes. Ask students to use the block to write code for their second episode that doesn't get mixed up with their first.



Writing the second episode (20 minutes)

9. Ask students to write their second episode and support them to use the backdrop event block to manage their story episodes. Check students' progress and plan for the following lesson.

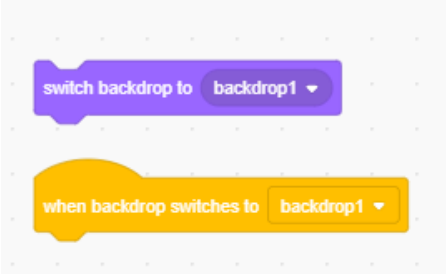
Lesson 6: Revising an original multimodal story

Learning goals

- Embed the concepts of sequencing, repetition and concurrency to a programme
- Complete a multimodal story
- Consider the use of multimodal choices and their effect on story

Activities

Introduction (5 minutes)



1. Recap the use of backdrop blocks to manage and coordinate episodes in a story. Correct any common misconceptions students may have (e.g., timing using wait blocks). Introduce tasks for the lesson (i.e., to code third and final episode of their story).

Writing the final episode (20 minutes)

2. Ask students to write the final episode of their story using blocks of code and the sprites and backdrops already in their projects. Support students where necessary. Where students encounter issues/misconceptions/errors, record these for the debugging activity.

Debugging (15 minutes)

3. Students may encounter bugs in their code that prevent their stories from running correctly. For this, use the demo project to model an example (e.g., timing error using wait blocks/no blocks to reset their characters/setting at the beginning of their stories).

Writing the second episode (15 minutes)

4. Discuss students' work and their experiences of the project.

Appendix 2: Intervention scaffolding materials

Sample monomodal text

In a misty green forest on a bright summer's day, Munchkin and his friend, Skelly were having a birthday party. Munchkin, a large polar bear-like creature, and Skelly, a skeleton, looked very different but they had a lot in common. They both loved getting birthday presents. As Skelly threw Munchkin his birthday present, an evil ghoul suddenly appeared out of nowhere, snatched it and flew away. "How do we get it back? Where do we go?" asked Munchkin angrily. "Ghouls live in the haunted house in this forest! Let's go!" Skelly replied, as they dashed off.

***

Along the journey, the two friends quickly realised the journey was far too long and they needed some help. "We'll have to get a ride to get there. DRAGON!" shouted Skelly. A gigantic, red-scaled dragon appeared out of nowhere and offered his help. He flew them over many different places including a city, the desert, a gingerbread house, a clothes shop, a tunnel, a Greek theatre, and the moon, before reaching the Haunted Castle. "That was fun!" Munchkin shouted. "Thanks Dragon!" bellowed Skelly.

***

The two friends dashed inside the castle. Inside, the evil ghoul appeared again and floated down the old staircase. They chased after him to recover Munchkin's present. "There he is!" roared munchkin. "Get him!" replied Skelly. Once Munchkin had wrestled the present from the Ghoul, he quickly ripped open the wrapping paper and looked at the present inside. "Finally! WOW! A snow globe!" he beamed happily. The snowball began flashing. "And it changes colour… cool!"

THE END

Annotated monomodal text

Blue=Backdrop, Green=Costume, Purple=Motion, Orange=Animation, Red=Dialogue

In a **misty green forest** on a bright summer's day, Munchkin and his friend, Skelly were having a birthday party. Munchkin, **a large polar bear-like creature,** and Skelly, **a skeleton**, looked very different but they had a lot in common. They both loved getting birthday presents. As **Skelly threw Munchkin his birthday present, an evil ghoul suddenly appeared out of nowhere**, **snatched it and flew away. "How do we get it back? Where do we go?" asked Munchkin angrily. "Ghouls live in the haunted house in this forest! Let's go!" Skelly replied,** as **they dashed off.**

***

Along the journey, the two friends quickly realised the journey was far too long and they needed some help. **"We'll have to get a ride to get there. DRAGON!" shouted Skelly**. A **gigantic red-scaled dragon** appeared out of nowhere and offered his help. He flew them over many different places including **a city, the desert, a gingerbread house, a clothes shop, a tunnel, a Greek theatre, and the moon**, before reaching the **Haunted Castle**. **"That was fun!" Munchkin shouted. "Thanks Dragon!" bellowed Skelly.**

***

The two friends dashed inside the castle. Inside, the evil ghoul appeared again and floated down the old staircase. They chased after him to recover Munchkin's present. "There he is!" roared munchkin. "Get him!" replied Skelly. Once Munchkin had wrestled the present from the Ghoul, he quickly ripped open the wrapping paper and looked at the present inside. "Finally! WOW! A snow globe!" he beamed happily. The snowball began flashing. "And it changes colour… cool!"

THE END

Appendix 3: Task-oriented analyses

MMC tasks (Bearne & Reedy, 2017)		Storytelling conjectures	Programming conjectures
Decide on mode and content for specific purpose(s) and audience(s)	Chose which mode will best communicate meaning for narrative purpose(s)	Add character(s) or object(s) using sprite(s)	Add sprite(s)
		Illustrate characterisation (roles, personality, appearance) through dialogue (e.g., *say* blocks)	Create sequence using *say* block(s)
		Maintain readability using multiple *say* block(s)	
		Establish setting(s) using backdrop(s)	Adds backdrop(s)
		Use varied text block(s) (e.g., *say*, *think*) to represent dialogue/thoughts	Use multiple *looks* block(s) for multiple purposes (e.g., say, think)
	Use multimodal effects to engage and hold a reader's attention (e.g., line, colour, movement, expression and language)	Use *show/hide* block(s) as cohesive device(s)	Add *show/hide* block(s) to sequence(s)
		Illustrates story episode(s) through text blocks	Create sequence using *say* block(s)
		Illustrates story episode(s) through motion	Use *motion* blocks to illustrate movement (e.g., glide/go to X: Y:)
		Illustrate characterisation (behaviour, appearance) through costume(s)	Use *change costume* block(s) to change and reset costume(s)
			Use loops to create animation or repeating motion (e.g., looping costume changes)
	Adapt, synthesise and shape content to suit personal intentions	Add or manipulate costume(s) to illustrate action(s) (e.g., looking around)	Use change costume block(s)
		Adapt graphic or text elements (e.g., dialogue, character or object) for narrative effect	Edit sprite(s) or backdrop(s)
			Use change effect block(s)
	Select appropriate content to express personal intentions, ideas and opinions	Define a narrative goal	
Structure texts	Pay conscious attention to design and layout of texts, use structural devices (pages, sections, frames, paragraphs, blocks of text, screens, sound sequences) to organise texts	Organise narrative event(s) using backdrop(s), hat block(s) or broadcast message(s)	Uses change backdrop block(s) to change and reset backdrop(s)
	Structure text(s) using cohesive device(s)	Use structural device(s) to ensure cohesion (e.g., yellow 'events' blocks)	Initialise sprite using *hat* block (e.g., when green flag is pressed)
	Story is structured sequentially (i.e., in correct order)	Use *wait* block(s) as cohesive device(s)	Use *wait* block(s) to synchronise events correctly
		Use *broadcast message* block(s) as cohesive device(s)	Use *broadcast message(s)* to coordinate multiple processes
	Use background detail to create mood and setting	Establish setting (surroundings, and habitual states) using backdrop(s)	Add backdrop(s)
			Edit or manipulate backdrop(s)
		Illustrate setting change(s) using *change backdrop* block(s)	Add *change backdrop* block(s) to sequence(s)

Use technical features for effect	Handle technical aspects and conventions	Coordinate narrative element(s) to execute correctly	Use initialising block(s) (e.g., *when* blocks) on all scripts
			Initialise sprite using hat block (e.g., when green flag is pressed)
			Execute two or more independent sprites concurrently
		Consider audience effect when viewing project	Set initial program state(s) (e.g., sprite size, position, graphic effect)
			Ensure all scripts have hat block(s) (i.e.no 'dead code')
	Choose and use a variety of structures for specific purposes	Illustrate event(s) resulting from actions (e.g., setting change)	Add *change costume/backdrop* block(s) to sequence(s)
		Create title cards to orient reader(s)	Create text-based sprites (e.g., title cards)
		Illustrates basic narrative structure (e.g., beginning-middle-end)	Use text block(s) (e.g., say, think)
Reflect	Improve own composition or performance, reshaping, redesigning and redrafting for purpose and readers'/viewers' needs	Check project against plans	Run programme to check for errors
		Revise project in light of plans	Debug errors

Appendix 4: Ethics documents

Participant teacher's consent form

Project title	Integrating computing and digital storytelling
Researcher's name	Robert Whyte
Supervisor's name	Shaaron Ainsworth

- I have read the Participant Information Sheet and the nature and purpose of the research project have been explained to me. I understand and I agree for to take part.
- I understand the purpose of the research project and my involvement in it.
- I understand that I will be teaching an integrated computing and literacy curriculum unit during my regular computing lessons.
- I understand that I will attend debriefing sessions with the researcher after the lessons.
- I understand that I may withdraw from the research project at any stage and that this will not affect my status now or in the future.
- I understand that the information gained during the study may be published though all personal data—including school name, my name, and any data generated—will be anonymized to maintain confidentiality.
- I understand that all data obtained during this study will remain confidential and will be used solely for research purposes and that it will be deleted 7 years after any publication.
- I understand that the raw data, which will be kept confidentially, will only be accessible to the researcher and his supervisor.
- I understand that I may contact the researcher or his supervisors if I require further information about the research and that I may contact the Research Ethics Coordinator of the School of Education, University of Nottingham if I wish to make a complaint relating to my involvement in the research.

Print name ……

Signed ……..…… Date ……

Contact details

Researcher: Robert Whyte, ttxrw49@nottingham.ac.uk, 0115 846 7672

Supervisor: Shaaron Ainsworth, shaaron.ainsworth@nottingham.ac.uk; 0115 846 7671

School of Education Research Ethics Coordinator: educationresearchethics@nottingham.ac.uk

Participant teacher's information sheet

My name is Robert Whyte and I am an PhD student at the Learning Sciences Research Institute, School of Education at the University of Nottingham. I am currently researching the new 2014 Computing curriculum and its applications across the primary English curriculum. More specifically, I am developing a series of activities to explore how integrating computing and digital storytelling activities could support learning in both subject areas.

I have recently been provided permission to organise a series of computing lessons with Year 5 to support children's computational thinking through digital storytelling activities. Computational thinking is a series of concepts and skills, including coding, that are regarded as important 21st century skills and are now required to be taught from the beginning of Key Stage 1. I believe it benefits children in the construction of meaning and development of understanding. Therefore, the purpose of the study is to investigate how we can support learners in the computing classroom as they create different projects (e.g., digital stories).

The research project will involve you and your class group in weekly computing lessons over one half-term period [INSERT DATES HERE]. By participating in this study, you agree to teach six in-class sessions (60 minutes each) based on the intervention. Tasks involve students exploring several small digital stories using the computer, structured tasks with specific learning outcomes, and independent tasks where students create their own digital stories. Students will also have a chance to ask questions to their friends or to you. You will act as a guide during the teaching sessions, answering students' questions and providing support where necessary. Then, we will have a debriefing session after each teaching session to discuss the discuss the intervention and how we could improve it to support learners. Interactions between you and your students will be audio recorded, and all children's work—including a recording of their computer screens—will be collected to assess how the intervention supports the learning outcomes. The sessions are voluntary and have been designed to generate data for my study, but all participants are free to stop the sessions at any point or to not answer any question. Furthermore, should any student appear uncomfortable, they have the right to withdraw their consent and any information collected would be withdrawn from the study and destroyed.

The proposed research project has been approved by the University of Nottingham Ethics committee in compliance with the University's Code of Research Conduct and Research Ethics and follows the

ethical protocols outlined by the British Educational Research Association. This means that your identitiy will be anonymised from the outset and all materials relating to you would be stored on a secure password-protected University server. This would only available to my supervisors and me. You can withdraw from the study at any point before, during or after the intervention. Once I have analysed the data, I will use these for my PhD thesis and for publication. I hope that these findings will help to improve the teaching of computing to young children and enable us to better understand the application of this subject area across the wider curriculum.

I have a clean DBS background check and taught at this school during the 2015-2016/2018-19 school year. Having worked closely with pupils in the past, I understand the practices and protocols of working within the school environment and with young children.

If you are happy with the nature and purpose of the research project, and agree to taking part, then please complete and return the attached form by [INSERT DATE HERE]. If you have any questions or require more information about the project please contact me, my supervisors or the Ethics committee, using the contact details attached.

Yours Sincerely,

Robert Whyte

Learning Science Research Institute,

School of Education, University of Nottingham

Contact details

Researcher: Robert Whyte, ttxrw49@nottingham.ac.uk, 0115 846 7672

Main Supervisor: Shaaron Ainsworth, shaaron.ainsworth@nottingham.ac.uk; 0115 846 7671

School of Education Research Ethics Coordinator: educationresearchethics@nottingham.ac.uk

Participant students' consent form

Project title	Integrating computing and digital storytelling
Researcher's name	Robert Whyte
Supervisor's name	Shaaron Ainsworth



I have **read** the Information Sheet and understand the research project and what I will be doing.



I **agree** to take part in the research project.



I **talked** about the research project with my parent/guardian.



I **understand** that if I don't want to participate in the research project, I **can withdraw my consent** at any time and I won't get into any trouble.





I am happy that there will be a **microphone** used in the research project and that Mr Whyte will record my computer screen.



If anything is written about me, I will get a new **secret name** so that nobody will know who I am.

Print name (child) ……..……..……

Signed (child) ……..… **Date** ……

Contact details

Researcher: Robert Whyte, ttxrw49@nottingham.ac.uk, 0115 846 7672

Main Supervisor: Shaaron Ainsworth, shaaron.ainsworth@nottingham.ac.uk; 0115 846 7671

School of Education Research Ethics Coordinator: educationresearchethics@nottingham.ac.uk

Participant students' information sheet

Hello! On this page, I will tell you a little bit about my project, what I want to do, and what you will be doing in the project if you take part.

My project

With your teacher, I am organising afternoon computing lessons at your school every week. I want to see if computing can help us to create digital stories. I also want to see if the kinds of stories we create on the computer can help us learn more about computing.

Activities

Each week, we will have our computing lessons in the computer room. In the lessons, we will work together to make digital stories using the computer and your ideas. You will read some of my digital stories on the computer and then create your own stories using your own ideas. If you want, you can share your stories with the rest of the class, using your name or a secret name. During the lessons, we will have a little chat about what you are making on the computer and how you're making it.

Things you need to know



When we have a chat during the sessions, I will ask you a few questions and record our conversation with my special microphone.



When you are making your digital stories, I will record a video of your computer screen but **NOT** you.



You can **always** ask any questions or put your hand up if you are uncomfortable.



I will keep a copy of the projects you create so I can study them. If I write anything about you, I will give you a different name so that nobody will know who you are.



You can **stop** at any time. You don't have to give a reason and nothing will happen to you as a result.

If you are happy with this and want to take part in the project, have a chat with your parents/guardians and write your name on the **Consent Form.** Thank you for looking at this letter and listening to me today — **Mr. Whyte**

Parents' consent form

Project title	Integrating computing and digital storytelling
Researcher's name	Robert Whyte
Supervisor's name	Shaaron Ainsworth

- I have read the Participant Information Sheet and the nature and purpose of the research project have been explained to me. I understand and I agree for my child to take part.
- I understand the purpose of the research project and my child's involvement in it.
- I understand that I may withdraw my child and my child may withdraw from the research project at any stage and that this will not affect her/his status now or in the future.
- I understand that the information gained during the study may be published.
- I understand that my child will not be identified, that his/her information will remain confidential, and that her/his identity will be anonymized for the writing of the publications.
- I understand that all data obtained during this study will remain confidential and will be used solely for research purposes and that it will be deleted 7 years after any publication.
- I understand that the raw data, which will be kept confidentially, will only be accessible to the researcher and his supervisors.
- I understand that I may contact the researcher or his supervisors if I require further information about the research and that I may contact the Research Ethics Coordinator of the School of Education, University of Nottingham if I wish to make a complaint relating to my child's involvement in the research.

Print name (child) ……

Signed (parent/guardian) …….. Date ……

Contact details

Researcher: Robert Whyte, ttxrw49@nottingham.ac.uk, 0115 846 7672

Main Supervisor: Shaaron Ainsworth, shaaron.ainsworth@nottingham.ac.uk; 0115 846 7671

School of Education Research Ethics Coordinator: educationresearchethics@nottingham.ac.uk

Parents' information sheet

My name is Robert Whyte and I am an PhD student at the Learning Sciences Research Institute, School of Education at the University of Nottingham. I am currently researching the new 2014 Computing curriculum and its applications across the primary English curriculum. More specifically, I am developing a series of activities to explore how integrating computing and digital storytelling activities could support learning in both subject areas.

I have recently been provided permission to organise a series of computing lessons with Year 5 and their teacher to support children's computational thinking through digital storytelling activities. Computational thinking is a series of concepts and skills, including coding, that are regarded as important 21st century skills and are now required to be taught from the beginning of Key Stage 1. I believe it benefits children in the construction of meaning and development of understanding. Therefore, the purpose of the study is to investigate how we can support learners in the computing classroom as they create different projects (e.g., digital stories).

The research project will involve children in weekly computing lessons over one half-term. The classroom teacher will teach children how to create digital stories using the computer, offering feedback and giving suggestions. Children will plan and create their own digital stories before presenting their work to the group. Activities in the computing club will include whole-class discussion, individual reflection, and programming on the computer. I will also interview children during the intervention so that we can discuss their projects and difficulties that are having, in order to better support them. Interactions between the classroom teacher and students will be audio recorded, and copies of all children's work will be made to assess what was learned in the sessions. The classroom teacher will be present during ALL sessions if any child has any questions or concerns. By attending the lessons, you agree, and your child agrees (as a participant), to being audio-recorded and for a recording to be taken of their computer screens when they are making their projects. The sessions are voluntary and have been designed to generate data for my study, but I want to reassure you that your child would be free to stop the sessions at any point or to not answer any question. Furthermore, should your child appear uncomfortable, they have the right to withdraw their consent immediately and any information collected would also be withdrawn from the study and destroyed. The proposed research project has been approved by the University of Nottingham Ethics committee in compliance with the University's Code of Research Conduct and Research Ethics and follows the

ethical protocols outlined by the British Educational Research Association. This means that your identity will be anonymised from the outset and all materials relating to you would be stored on a secure password-protected University server. This would only available to my supervisors and me. You can withdraw your child from the study at any point before, during or after the intervention. Once I have analysed the data, I will use these for my PhD thesis and for publication. I hope that these findings will help to improve the teaching of computing to young children and enable us to better understand the application of this subject area across the wider curriculum.

I have a clean DBS background check and taught at this school during the 2015-2016/2018-19 school year. Having worked closely with pupils in the past, I understand the practices and protocols of working within the school environment and with young children.

If you are happy with the nature and purpose of the research project, and agree to your child taking part, then please complete and return the attached form. If you have any questions or require more information about the project please contact me, my supervisors or the Ethics committee, using the contact details attached.

Yours Sincerely,

Robert Whyte

Learning Science Research Institute,

School of Education, University of Nottingham

Contact details

Researcher: Robert Whyte, ttxrw49@nottingham.ac.uk, 0115 846 7672

Main Supervisor: Shaaron Ainsworth, shaaron.ainsworth@nottingham.ac.uk; 0115 846 7671

School of Education Research Ethics Coordinator: educationresearchethics@nottingham.ac.uk