



The
University
Of
Sheffield.

**Exploring Coding Tools and Their Contribution to Children's
Learning in Early Childhood Education:
A Systematic Literature Review and Parent Interviews**

By:

Melike Aslan

180132498

The University of Sheffield
Faculty of Social Sciences
School of Education

Dissertation submitted in part requirement for the MA in Early Childhood
Education of the University of Sheffield

Word count: 19793

August 2019

Abstract

A range of digital coding tools has recently been designed to teach young children coding skills through play and fun (García-Peñalvo *et al.*, 2016). However, we know little about the characteristics of these apps and how effective they are in children's learning. Therefore, as one of the aims of this research, I systematically reviewed the existing literature to examine the current digital coding tools designed for young children (aged 0-8) and their contribution to young children's learning. It is also known that children's interest in technology is usually aroused by their family members' interaction with technology tools (Donohue 2015 cited in Ihmeideh and Alkhawaldeh, 2017). To better understand the actual experiences of children using the digital coding tools at an early age, it is necessary to consider the attitudes and views of the parents regarding coding and coding tools. Therefore, another aim of this research was to investigate two parents' perspectives towards coding by using semi-structured interviews. This thesis addressed some gaps and made some important contributions to knowledge by using mixed methods. Firstly, some of the main issues raised by the reviewed studies were discussed. Secondly, the skills promoted by digital coding tools in the reviewed studies were presented and interpreted. Thirdly, the outcomes of the interview with parents were discussed.

Key words: coding, programming, digital coding tools, early childhood education, young children

Acknowledgements

I am deeply indebted to my family for their unconditional love and constant support. To my parents, **Arif** and **Niymet Aslan**, your love and guidance have always been with me in whatever I have pursued. To my siblings, **Meltem** and **Melih Aslan**, you have never let me fail. A special thanks goes to my brother **Melih** who has always been with me in the biggest exams and changes of my life.

I would like to thank my perfect other half, **Kemal Ünlü**, who has been my constant companion and emotional support during this process.

Having you all has been my reward in this life and has made me the person who I am today. For everything you have all done for me, I am so grateful.

Throughout this thesis, I have received a great deal of support and assistance from my supervisor, **Dr Fiona Scott**, who has always been kind and supportive. I would like to express my deepest appreciation to her.

I would also like to thank my supervisor in Turkey, **Prof Dr Birsen Güzel**, for sharing valuable ideas with me during this process.

This study has been entirely sponsored by **The Republic of Turkey**. None of these opportunities would have been possible for me without the support and sponsorship of the Turkish Government. I am truly grateful.

August 2019.

TABLE OF CONTENTS

TABLE OF CONTENTS.....	4
List of Tables	5
CHAPTER 1: INTRODUCTION	6
1.1. Context of the research	6
1.2. Research Questions	10
1.3. Personal Motivation for the Study.....	10
CHAPTER 2: LITERATURE REVIEW	11
2.1. Introduction	11
2.2. Defining Coding as a New Literacy.....	11
2.3. Why are we teaching coding?	12
2.4. Coding as an educational tool.....	14
2.5. A developmental glance at coding.....	16
2.6. Some recent developments and related policy in coding	18
2.7. What skills do children develop via digital coding tools?	19
2.8. Parents' views and mediation.....	21
CHAPTER 3: METHODOLOGY	25
3.1. My Methodological Approach	25
3.2. Data Collection Methods	25
3.2.1. Systematic Literature Review	26
3.2.2. Semi-structured Interviews with Parents	29
3.3. Ethical dimensions	31
CHAPTER 4: DATA ANALYSIS and DISCUSSION	33
4.1. Analysis and Interpretation of Systematic Literature Review	33
4.1.1. Answering the RQ1: The existing digital coding tools designed for young children.....	41
4.1.1.1. The importance of the design of digital coding tools	41
4.1.1.2. The differentiation of the promoted skills of children at different ages	43
4.1.1.3. The key role of adults in facilitating and enhancing children's learning.....	43
4.1.1.4. The gender stereotype in children's computer science learning	45
4.1.1.5. The significance of how we introduce coding tools to children	45
4.1.2. Answering the RQ2: The skills promoted by digital coding tools	46
4.1.2.1. Basic Programming Skills.....	51
4.1.2.2. Cognitive Skills	52
4.1.2.3. Social skills.....	57

4.1.2.4. Language skills	58
4.2. Analysis and Interpretation of Interviews with Parents	59
4.2.1. Type of digital devices used by children	60
4.2.2. The features of children's engagement with digital devices	60
4.2.3. The selection priorities for apps	60
4.2.4. Answering the RQ3: The perspectives of parents towards coding and coding tools	61
4.2.4.1. Awareness of what coding is.	61
4.2.4.2. Too young to learn coding	62
4.2.4.3. Coding as brainstorming	63
CHAPTER 5: CONCLUSION.....	65
BIBLIOGRAPHY	69
APPENDICES	79
Appendix 1. Example participant information document	79
Appendix 2. Example consent form	81
Appendix 3. Ethics approval letter.....	83
Appendix 4. Interview Questions.....	84

List of Tables

Table 1. Data Collection Methods.....	26
Table 2. Descriptive Table for Reviewed Studies.....	34
Table 3. Matching table for digital coding tools and the skills promoted by them	47
Table 4. The Features of the Participants and Their Children	59

CHAPTER 1: INTRODUCTION

In this chapter, I explain my research topic and introduce my overall aim to readers. This chapter also lays the groundwork for a more comprehensive review of the literature and allows me to articulate my research questions more thoroughly.

I begin by discussing the increasing impact of technology on the lives of people (Miller, 2001). Primarily, I focus on coding which has become an important skill in early childhood education (Balanskat and Engelhardt, 2014) and I evaluate the parents' perspective on children's use of technology tools, more specifically in terms of coding applications. Then, I present my research questions. Finally, I conclude this chapter by exploring my personal motivations for this study.

1.1. Context of the research

The world is seeing a significant change in the 21st century, and digital technologies are likely to be a contributing factor to this change (Palaiologou, 2016). Likewise, in light of the findings of the OECD International Futures Programme, Miller (2001) argues that the daily life of many of the world's population will be radically different in the 2030s compared to the last ten years of the 20th century. Regarding this prediction, he draws attention to the key role that the advances in technology have played and the large-scale changes that it has produced. These changes can be seen in the many aspects of life, such as using robots in factories rather than employees, communicating with others using online platforms instead of using paper communication, and shopping online instead of physically going to malls to name a few examples.

The rapid development of new digital technologies has influenced not only science and economy but also education. Silver-Pacuilla (2006 cited in Parette *et al.*, 2010) noted that there has been extreme innovation in technology which has become user-friendly, available for everyone with lower cost, and more powerful. These improvements to technology make its usage more attractive and beneficial to the education sector as a means to enhance the children's learning. However, the latest research from Ofcom (2018) showed that although the internet seems ubiquitous, not everyone has the same online experience. 12 % of adults in the UK do not use the internet that this relates to age and socioeconomic status. According to the findings, the percentage of the population who do not utilise the internet increases with age.

22% of adults in DE households (unskilled/uneducated workers and unemployed persons) who are not online are nearly double the UK average (12%). Certain groups may have limited access to the Internet due to their preferences or existing demographic conditions. Therefore, unlike Silver-Pacuilla (2006), it may not be possible to generalise the accessibility of the internet for everyone.

From the educational perspective, Lee *et al.* (2013) emphasise that the struggle with finding the best way to integrate technology into the classroom is an ongoing challenge. Likewise, Parette *et al.* (2010) suggest that implementing digital technologies to preschool settings has a history of difficulties. These difficulties were associated with concerns regarding the child's ability to learn using digital technologies. By contrast, digital technologies are considered as an important part of children's learning modalities today (Edwards and Bird, 2017).

Unlike their parents, today's children are born into a media-saturated world where cutting-edge technologies such as smartphones, iPads and computers are ubiquitous, and they play a major role in everyday life. Children are exposed to these technologies and become aware of these tools at a very early age (Ihmeideh and Alkhawaldeh, 2017). This in turn has a major impact on their upbringing in terms of play, communication, learning and education. Conn (quoted in Parette *et al.*, 2010, p.336) likened the technology to "like the air" for the new generation of children, and indeed technology is as important as breathing air in the new generations' lives today.

Through the increasing integration of technology into our lives, numerous new terms have emerged such as digital literacy, virtual reality, digital play, computer programming and coding. The current generation of children, Prensky (2001) are called "digital natives" and they are much more likely to learn and use these ubiquitous digital technologies. According to Prensky (2001, p.2), today's students are "all native speakers of the digital language of computers, video games and the Internet". In contrast, the term "digital immigrants" is used to describe persons who were not born into the digital world. He explains the adaptation process of digital immigrants to the digital world with the following metaphor. Digital immigrants always retain their original 'accent' which comes from their upbringing just as new language learners retain the accent of their heritage language. In addition, he adds "a language learned later in life, ..., goes into a different part of the brain" (Prensky, 2001, p.3). This information

can explain the accent of digital immigrants. In order to understand what he means by accent; the behaviours of digital immigrants can be observed in many ways such as choosing books for information instead of the Internet or needing a printed version of a paper instead of editing on the screen. In contrast, the activities of the new generation are mostly screen-based, online and quick. Therefore, Prensky (2001) emphasises that the difference in language between digital immigrant educators and the new generation leads to problems in education. However, there have been attempts to solve this problem since 2001. For example, Brush *et al.* (2003) launched 'The Preparing Tomorrow's Teachers to Use Technology (PT3)' initiative to improve and enhance the technology preparedness of preservice teachers.

Moreover, digital competency has gained an important place in employment. Pila *et al.* (2019) suggest that graduates cannot fill the science, technology, engineering and math (STEM) jobs despite the growing significance of these fields globally. Furthermore, some criticise schools for insufficient graduates to meet this need (Rothwell, 2014). To address this shortfall, policy changes are attempting to increase the number of students pursuing STEM education by placing greater emphasis on STEM initiatives which begin at the preschool level of education (Shilling, 2016). Recently, early STEM education mainly focuses on teaching branches of computer science such as coding and programming which are likely to meet the need of 'T' (Technology) and 'E' (Engineering) of STEM (Pila *et al.*, 2019). To this end, a range of digital apps has recently been designed to teach young children coding skills through play and fun (García-Peñalvo *et al.*, 2016).

However, we know little about the characteristics of these apps and how effective they are in children's learning. Therefore, as one of the aims of this research, I systematically review the existing literature to examine the current digital coding tools designed for young children (aged 0-8) and their contribution to young children's learning. This thesis also seeks to create a list of different digital coding tools that were specifically designed for pre-school children and the skills promoted by these tools are also investigated.

Another key fact to remember is that children's interest is usually aroused by their family members' interaction with technology tools. Hence, parents and family members have an important role in the introduction of technology and digital media to the children (Donohue 2015 cited in Ihmeideh and Alkhawaldeh, 2017). Some parents may be overprotective and

controlling of their children's use of technology. Research has shown that although 99% of the surveyed parents had their own computers, tablets or smartphones, less than half of them (46.51%) allow their children to access these devices at home (Preradović *et al.*, 2016). Researchers have suggested a possible reason for this restriction: due to negative reports and studies, parents may have several misconceptions and stereotypes about the potential risks and harmful impacts of digital technologies on young children. The parents' main concern was that they believed their children would be aggressive or bully due to some digital games. Because of such reports, parents may prohibit their children's access to digital technologies.

Some parents, on the other hand, may be indifferent to their child's technology usage. Research has shown a lack of parental awareness of the use of digital media and technology by children (Vittrup *et al.*, 2016). Furthermore, the review of the literature addresses a need to understand the parents' awareness and active participation in the use of technology and digital media which are increasingly more available to preschool children (Vittrup *et al.*, 2016).

Furthermore, Vittrup *et al.* (2016) suggest that the willingness of parents to allow their child to access technology tools depends on how they value these tools. Likewise, Preradović *et al.* (2016)'s research indicates that despite the potential hazards that may result from the usage of computer by young children, most parents believe that their child should be trained in how to use digital technology and acquire digital skills, which they believe to be valuable for their child's development. To better understand the actual experiences of children using technology at an early age, it is necessary to consider the attitudes and views of the parents regarding technology. Today, schools, for-profit and non-profit organisations provide coding courses for children. Children's attendance in these courses mostly depends on their parents' views. If parents see coding as valuable, they are likely to support their children to learn coding.

To that end, for another aim of this research, I investigate parents' perspectives towards coding. It allows me to understand how they consider coding as a viable educational tool and how likely they are to include coding games as a modality in their child's learning methods.

1.2. Research Questions

In evaluating past studies, I decided to focus my research on the following three questions regarding the coding applications in early years. These are:

1. What digital tools can be used to teach coding to young children in early years?
2. What skills does the literature suggest digital coding tools promote?
3. What are the perspectives of parents regarding the use of digital technology, specifically coding?

1.3. Personal Motivation for the Study

As a student, I have been interested in digital technology and games since my childhood. As a teacher, I believe that learning emanates from interest and curiosity. During my professional life, I have seen that technology plays a facilitating role in my students' learning. Based on my experience and observations, I think that children find technology-focused education fun and are willing to explore it. Therefore, we should include exciting and valuable aspects of technology in children's learning. My interest in coding apps makes me excited about researching this topic.

CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

This thesis examines the digital coding tools designed for young children, the skills promoted by them, and the views of families about these tools and coding. It was designed to answer the questions of the study given in detail in the introduction part. Therefore, research questions were considered in the selection of the materials to be included in the literature review. In simple terms, the priorities for inclusion in the literature focused on the following topics;

The history and place of coding tools in education

Young children's engagement with tangible and digital coding tools

Parents' views on digital tools and coding tools.

In line with the aims of this review, I define 'young children' as children from birth to 8 years according to Farrell *et al.* (2015)'s definition of 'early childhood' (Scott, 2018).

This literature review focuses on research addressing coding tools designed for children, but it also encompasses research in relation to a variety of digital devices. It includes the definition of coding, the reasons to teach coding, the development process of coding as an educational tool and the place of coding tools in early childhood education. This is followed by a developmental review of coding and programming tools, and some new developments and related policies related to coding. Then, the potential skills that children may enhance in coding tools are discussed. Finally, parents' difficulties in mediating children's use of technology and their views on media and technology tools are discussed.

2.2. Defining Coding as a New Literacy

Literature has several definitions for coding. McLennan (2017, p.1) defines “coding (or programming) is a basic language of the digital age”. According to Bers (2017, p.11) coding can be defined “...as epistemology,... as a way of knowing....”. In brief, coding provides a meaningful interactive process in children's learning. Each interaction between people and computers is governed by code whether creating a website or following GPS directions while driving. In order to understand the hyper-connected world, coding is the fundamental literacy

which helps to practise 21st century skills such as analytical thinking, problem-solving and working as a team (EU Digital Single Market, 2018).

Bers (2018b) delineates literacy as a skill to use a symbol system (the language of programming or the language of natural written) and a tool (paper and pencil, or an iPad) so as to understand, communicate, generate and transfers opinions and thoughts by creating a shareable product such as a text, a robot or a video that enables others' interpretation. This definition contains both textual and coding literacy. As more coding and programming languages spread to other professions and different thinking styles, they become a new literacy. However, according to Kress (2003), while the meaning of traditional written material such as a book is waiting to be filled with readers, the meaning of image on screens is already filled, thus 'semantically closed'. Therefore, Kress (2003) argues that traditional literacy (print-based) and new media literacy (screen-based) should not be regarded as the same because they are fundamentally different. Resnick (2015), on the other hand, clarifies coding as an expanded version of writing. To give an illustration of what he means, he suggests that as a birthday present for a friend, people can use coding to create an interactive birthday card just as they might write a letter. He notes that although a few people grow up to become professional writers, everyone is taught how to write since it is a way to communicate with others, to organise and express our thoughts. Similarly, when people learn how to code, they learn how to organise, express and share their ideas in a new environment (Merrill, 2017). However, according to the UNESCO Institute for Statistics (2016) report, 750 million adults still lack basic reading and writing skills. What is more, 102 million youth (aged 15-24) account for a remarkable part of the illiterate population. In this case, it is disputable to make a generalisation for literacy or to give priority to coding (new literacy) on behalf of everyone while struggling with existing illiteracy problem.

2.3. Why are we teaching coding?

There are conflicting views on the purpose of teaching coding to children. Rushkoff (2010) emphasises the importance of controlling technology and learning programming since almost all areas will be digital and be coded that human-beings will use or consume in future. However, García-Peñalvo *et al.* (2016, p.2) interrogate: Do we create a labour force for the future and try to address the existing skill shortcomings in the labour market? or “...is it an academic subject geared to intellectual development or simply a necessary life skill?”. Coupled with these questions, the review of the literature has revealed three main aims to teach coding. These are; raising concerns about the needs for future tech-savvy employment, acquisition of

the emerging new literacy skill of the next century and believing in the importance of coding as a future skill regardless of occupations. For instance, even today a lot of software is designed so that people can be creative without the need for complex coding skills such as website design industry works.

Some researchers estimate that due to the rapid change in the technological community, 65% of students entering school today may have future professions that do not yet exist (Davidson, 2011 cited in Kazakoff *et al.*, 2013). Therefore, the idea that preparing children for future employment arises. Eisenberg *et al.* (2017, p.1) claim that “...of course children should learn programming...How will a child otherwise be able to get a job in the coming century?”. Similarly, The European Coding Initiative (2014) drew attention to the influence of digitalisation on the labour market, and noted that “whether you want to be an engineer or a designer, a teacher, nurse or web entrepreneur, you will need digital skills” (p.3). Although digital skills are not limited to coding, it is considered one of the essential digital skills. Therefore, it is highly associated with future employment alongside a new literacy.

However, some contrary-minded suggests how reasonable is it to prepare a young child for a particular job? (Duncan *et al.*, 2014). They also emphasise that it is more valued to allow children to enjoy their childhood without pressure to prepare for certain future employment. Otherwise, children become an investment instrument. The idea of seeing children as an investment and a more skilled workforce for the future can be associated with the neoliberal approach. Moss (2014 quoted in Dýrfjörð and Magnúsdóttir, 2016, p.81) defines neoliberalism is based on the idea that “every aspect of human life can and should be reduced to a set of economic relationships and values”. This is done by altering the vocabulary of the world of education: from pedagogy and philosophy to industry; by talking about child-care industries rather than early years education (Moss 2008 cited in Dýrfjörð & Magnúsdóttir 2016). Similarly, in their study, Sonu and Benson (2016) argue that children have been constructed as a subject of the neoliberal globe via school politics and practices that “...worked on the child, rather than with the child” (p.230). However, this view contradicts the deep-rooted early childhood practices that operate from the strengths of children (Brown, 2015).

Some argue that coding is just another language and tool to teach computational thinking. Therefore, it should not be the only emphasis. García-Peñalvo *et al.* (2016) suggest that there is a need to teach more important transferable skills such as problem-solving, logical thinking

and creativity that can be applied not only to programming process but also to real life conditions. Computational thinking enables children “to be not only technology-literate but also creators of computational artefacts” (Bocconi *et al.*, 2016, p.36), therefore; it is regarded as a significant skill in the 21st century. The inclusion of computational thinking skill and other relevant concepts such as coding, programming and algorithmic thinking in compulsory education ought to improve children's self-expression, problem-solving and creativity skills by using computational tools.

Crow (2014, p.1) has noted that not every job will involve programming in the future, however, it is still necessary to teach every child how to code because “this is not primarily about equipping the next generation to work as software engineers, it is about promoting computational thinking”. Wing (2006) delineates some of the characteristics of computational thinking (CT). CT is “conceptualising, not programming”, it requires more than programming. It is “fundamental, not rote skill”, therefore people should have it to be a part of modern society, not a set of mechanical routines. It is a “a way that humans, not computers, think...”, it is necessary to use as a way to solve problems, not to make people think like computers. CT is “ideas, not artifacts”; it allows people to approach and solve problems and to communicate and interact with others beyond producing software and hardware. It performs an exhaustive analysis of problems (p.35). Therefore, Sanz (2015 cited in García-Peñalvo *et al.*, 2016) pointed out that this skill plays a key role not only in coding but also in thoroughly analysing and understanding problems, recognising patterns and estimating solutions (Further discussions of computational thinking are stated in section 4.1.2.).

2.4. Coding as an educational tool

The history of educational coding or computer programming tools is as old as programming as a discipline (Kölling, 2016). The first initiatives to introduce computer programming to students date back from the late 1960s (Bers, 2018). Seymour Papert, who is known as the father of constructionism, and his colleagues developed the first version of the LOGO at MIT in 1967. LOGO has an important place in the history of educational tools since it was the first programming language designed for children (Bers, 2018). In the LOGO, children could direct a robotic turtle by typing commands at the computer, so the turtle could go around and draw some geometrical shapes with the attached pen. LOGO has gained popularity all over the world

through its success. Today, there are numerous updated versions of the LOGO in use such as UCB Logo, MSW Logo, FMS Logo and Kinder Logo (The Logo Foundation, 2015).

During the last decade, a significant number of new programming environments inspired by LOGO were developed (Bers, 2018). They were namely DrJava (2002), Jeroo (2003), Scratch (2005), StarLogo TNG(2007), Kodu (2009), Blockly (2013), Pencil Code (2015) (Kölling, 2016). For instance, the Scratch, a graphical programming environment where children at age 8 and older can create and share their own animated stories, games, and artistic presentations (Maloney *et al.*, 2008). Some others, such as Kodu, Etoys and Storytelling Alice provide computer programming for children in more accessible ways, but like Scratch, they are designed for the children aged 8 and above (Resnick *et al.*, 2009). CHERP designed for the use of pre-school children as a tangible and graphical programming language (Kazakoff *et al.*, 2013).

Some argue that coding education has become a more attractive experience for young learners through a large number of child-friendly programming environments (Papavaslopoulou *et al.*, 2018). Yet it does not mean that children can use those programming environments efficiently. The 'digital native' children discourse of Prensky (2001) is true, but merely to some extent (Chaudron, 2014). Though many children acquire basic operational skills without a problem and some of them acquire more higher-level online competencies, they also encounter difficulties that they cannot manage. These difficulties may result from their limited capabilities of cognitive development, reading and writing skills (Chaudron, 2014). Therefore, it is significant to make sure that children gain meaningful experiences in those environments. Likewise, Coravu *et al.* (2015) suggest that children should be encouraged to become creative thinkers and active learners that enable them to escape from the trap of simply being end-users of a program.

Some have pointed to ongoing debate for the children's engagement with digital technology. Plowman *et al.* (2010) suggest that the growing pervasiveness of technology has created a public concern about its role in young children's lives. As the main concern, they highlight the transformation of childhood by technology and draw attention to the threat to children's cognitive, emotional and social development. Cordes and Miller (2000) stated that there is an immediate need to end the advanced introduction of the computer in early childhood except for children with disabilities. They highlighted the importance of hands-on experiences, play and

reading books as the principals of a healthy childhood. Likewise, García-Peñalvo *et al.* (2016) argue that there is still disagreement about whether children should be allowed to spend time on computer screens, let alone taught how to programme. According to Papavlasopoulou *et al.* (2018), although cutting-edge technology, innovative pedagogies, guides and sources are available in computing education, there is still uncertainty as to how to decide, design and apply the appropriate learning activities for children. Many young students are engaged with the Internet, digital games, robots and phones for fun. Their motivation may decrease when they are required higher level programming language in those platforms. In Romania, Coravu *et al.* (2015) compared the influence and outcome of two different (visual and non-visual) programme languages on primary and secondary students' progress in their 2-month experimental study. They found that the interest and comprehension of secondary school children were low when they used non-visual computer programming language, whereas the experimental group had better results and higher motivation to progress in writing and developing programs and applications in a visual programming language (Scratch). When students are asked for feedback, they replied that they imagine interactive apps with good design and easy to accomplish.

As a result, if we are to introduce coding to children at an early age, we need age-appropriate technologies and pedagogical approaches that consider the cognitive competency and abilities of young children. For the age-appropriate apps, National Association for the Education of Young Children (2009) stated that apps should be designed in the interest of children, allow children to make different choices, and meet the learning needs of children in their current developmental stage meanwhile challenging them to reach further goals.

2.5. A developmental glance at coding

Considering physical objects as a key factor in children's learning is a relatively new opinion (Resnick *et al.*, 1998). Pestalozzi (1803), one of the advocators for hands-on learning, claimed that children's learning should occur through their senses and physical activities, arguing for "things before words, concrete before abstract" (quoted in Resnick *et al.*, 1998, p.281). Froebel was influenced by Pestalozzi's views and developed a specific set of 20 gifts involving physical objects (balls, blocks, and sticks) for kindergarten children. These objects allowed children to model the structures they saw in their environment and to create their own designs (Strawhacker and Bers, 2015). Maria Montessori expanded Froebel's ideas and inspired a school network where manipulative materials played a central role. She aimed to use these

materials for the development of children's sensory abilities by enabling them to learn through exploration of these materials.

Piaget (1997) theorised that children first construct knowledge in 'concrete operations', then move on to 'formal operations'. However, Turkle and Papert (1990) have asserted that Piaget understated the significance of concrete operations stage. They, then, argued that abstract thinking should not be regarded as superior to concrete manipulations since the movement of an item on a computer screen can be defined by mathematical and abstract approaches, but at the same time, this movement is visible, almost concrete, and thus allowing the concrete approaches. Indeed, children do not leave their concrete thinking when they are required abstract thinking, so they need both. Resnick *et al.* (1998) drew attention to the need for digital manipulatives emerged from the difficulties of discovering some concepts requiring manipulation of abstract icons with traditional manipulative materials. Resnick *et al.* (1998, p.281), then, introduced digital manipulatives as "computationally enhanced versions of traditional children's toys" which enable children to explore advanced concepts, coding and programming tools are some of them. In programming environments, children are encouraged to think about their own thinking (Papert, 1980) and use coding as a language to invent or to inquire.

Piaget's thoughts about children's cognitive development formed the basis of Papert's work, but Papert applied them to computer technology (Goldstein and Papert, 1977 cited in Kazakoff *et al.*, 2013). Piaget is known as the pioneer of constructivism: the view that knowledge is constructed through the interactions of one's environment. Papert composed his own term 'constructionism'. The focus of Papert's constructionism is the things that are made during the learning process, whereas Piaget's constructivism provides a window for children's interests and abilities in developmental goals which can be achieved at different stages of child development (Ackermann, 2001).

Vygotsky's theory of development emphasises socio-cultural impacts on children, particularly the role of adults in leading and enhancing children's learning and cultural artefacts as educational tools. Regarding his ideas today, children (under the age of 8) do not buy their own technological tools, and therefore the role of teachers and parents in introducing children to these tools is key (Kazakoff, 2015). Moreover, some certain technological tools can support children's social and prosocial development with their design features (Bers 2012 cited in

Kazakoff, 2015). Early studies about technology and young children have shown that computers can have a catalyst role in social interaction in early childhood education classes (Clements, 1999) and that children have twice as much interaction when they are involved in computer activities as compared to other classroom activities (Svensson, 2000).

Research in developmental psychology has also focused on the manipulation and motor development of children regarding the use of the object, whereas the place of objects in sociology is their role in the social lives of young children (Bateman and Church, 2017). In some studies, for example, children have been observed orienting the objects as a way of approaching a peer group object (Bateman and Church, 2017) and making rules around the object they have in their play (Corsaro, 1979). However, Corsaro (2015) argues that the objects that children use in their plays have altered over the years from everyday objects to the mass production toy industry. Such a critique may remind uncontrolled increase in app stores (Chau, 2015) to make money from in-app purchases by children (Discussed further in 2.8).

2.6. Some recent developments and related policy in coding

The inclusion of computational skills in the formal education of primary and secondary students has been recognised by many countries. Therefore, there are various initiatives across the globe to implement coding framework in children's learning. For instance, computational thinking and computer programming were formally included as part of the UK national primary and secondary education curriculum in the academic year 2014-2015 (Department for Education, 2013). In order to introduce computer programming to K-12 students, policies and frameworks have been developed by Australia, Singapore, and Malaysia (Australian Curriculum and Assessment Authority 2015 cited in Bers, 2018). Coding education has been integrated into the curriculum by 16 European countries (Balanskat and Engelhardt, 2014) and the US launched 'Computer Science for All' initiative to teach programming in all educational levels (Smith, 2016). According to Bers (2018), many of these changes have caused by the demand to educate the workforce of the next century's automatic economy and the notice of needs for technology-savvy workers in the sector.

Research has shown that educational initiatives which begin in early years have lasting effects and reduce costs both economically and developmentally (Cunha and Heckman, 2007). With this awareness, a range of new programming tools was designed for young children and many coding initiatives have been launched. In Asia, Singapore has launched an initiative called

PlayMaker, which offers robotics and other coding applications to young children's learning environments. Likewise, in Chile, the Kodea Foundation has drawn intense interest. Some developing countries such as Ghana, have chosen non-profit organisations such as Ghana Code Club to introduce coding (Nguyen, 2016). ProgeTiger program, an initiative to improve the digital skills of children and young people in Estonia, aimed to integrate technology education into preschool, primary school and vocational education curricula while providing teachers with training opportunities and providing financial support to kindergartens and schools in obtaining various programmable devices (HITSA, 2015).

2.7. What skills do children develop via digital coding tools?

Past research has shown that children exposed to STEM (Science, Technology, Engineering, and Mathematics) curriculum in early childhood, including programming languages, demonstrate a less gender-based stereotype related to STEM careers (Steele, 1997) and have less difficulty entering these fields in the future (Markert, 1996). However, there may be a misunderstanding about coding for young children. At first glance, coding may seem difficult and complicated for children's comprehension. However, coding for young children means that learning basic coding skills which can be achieved by them. Besides, there is a number of offline, unplugged and practical activities that can support problem-solving and logic without touching a computer. Thus, coding or programming activities, not every time require high priced complicated digital tools. Coding is engaging for children since they are in on the process and willing to try and find out things. Moreover, there is no certain correct or wrong method. If they face a problem to work their code, they can just keep going to try other ways until achieving what they want to achieve. Children can either generate their game or perform basic instructions such as moving a game character on the screen (Bocconi *et al.*, 2016). Furthermore, they can become the creators of their own digital applications with coding rather than being passive consumers of the digital world (European Schoolnet, 2016).

Coding process consists of a range of basic steps: “sequencing, conditions, and loops (repetition)” (Pila *et al.*, 2019, p.53). Bers (2012 cited in Pila *et al.*, 2019) defines sequencing as a set of directives taken place in a program and given by users to operate a robot or a computer. Each block has its own certain meaning; therefore, the alignment of blocks is significant. Conditions are statements such as ‘if, then, else’ that lead the program what actions to take next. For instance, ‘if the road hits green, then up’. Loops are the sequence of directives that can be adjusted to occur again and again. Therefore, coding is not a simple technical skill,

but one of the ways to achieve the literacy of the 21st century. Kazakoff *et al.* (2013) highlight the importance of sequencing in early years as it is crucial to understand and develop computer programs. Likewise, Nam *et al.* (2019) suggest that to develop a program, a child needs sequential thinking and planning ability for what action to take next, before, or until another one.

An online survey of 2000 parents or carers of children aged 0-5 years in the UK showed that half of the children in the 0-2 age group and two-thirds of the children in the 3-5 age group were capable of opening apps independently, swiping and flipping the pages of an e-book, and tracking shapes on the screen by using their finger (Marsh *et al.*, 2015). This large-scale study reveals the fact that children even at a very young age have abilities to use digital tools that they may need in coding tools. The question, however, is whether they are conscious users of these tools.

Neumann and Neumann (2017)'s review pointed to the importance of the quality of apps to foster children's literacy development through tablets. However, testing the quality of apps might be difficult. In the study of the selection of educational apps for young children with disabilities, More and Travers (2013) noted that producers market apps as educational for young children by designing them with pictures, shapes and colours. However, the title and appearance of these apps could be easily misleading in terms of their age appropriateness. They, therefore, pointed out that numerous so-called educational apps in the marketplace do not have the due design, direction, content and other significant qualities that should be in high-quality educational software.

Although the selection of high-quality apps for children is challenging, there are several positive examples of the usage of coding and programming tools in children's learning. Research indicates that children can "learn and apply concepts such as abstraction, automation, analysis, decomposition and iterative design" through age-appropriate computer programming environments (Kazakoff, 2015, p.45). Kanbul and Uzunboylu (2017)'s review of the literature stated that putting in order, creating order and separating into small pieces can be taught to pre-school children through Scratch JR, Code.org, the first two periods of Kodable, The foos software programs and BeeBots, KIBO robotic application tools. Additionally, Maloney *et al.* (2010) note that the main purpose of these environments is to develop self-directed learning through tinkering and motivation of students through collaboration with peers rather than

teaching coding itself. Moreover, such a learning environment can be used in the learning of children with different level of competencies and children with learning difficulties or disabilities. In a case study in New Zealand where Scratch programming was introduced to 9-10-year-old children, Taylor *et al.* (2010) investigated the impact of programming on the development of children's mathematical and computational thinking. They grouped children with low and high mathematics ability and found that Scratch programming education was effective in teaching mathematics, enhancing problem-solving strategies and providing opportunities for systematic and creative thinking. Scratch programming also fostered children's several social skills such as increasing involvement, interaction and collaboration. A meta-analysis in an earlier work showed that computer programming has slightly positive effects on the children's cognitive skills such as reasoning, logical thinking, planning and general problem-solving (Liao and Bright, 1991).

Clements and Nastasi (1999)'s review of empirical studies indicated that engaging in robot programming promoted children to express their thoughts and clarify their thinking process, observe the results and receive feedback from their educators while trying to solve a problem. In an experimental study with 53 children (aged 5-6) from two classes in Korea, Nam *et al.* (2019) found that children who participated in an 8-week card-coded robotics curricular intervention and following activities showed better performance on sequencing and mathematical problem-solving tests.

Bocconi *et al.* (2016) emphasise another opportunity offered by both tangible and virtual coding tools to create an inclusive educational environment in which all students in the classroom participate in programming activities. This implication also allows the participation of students with special needs or learning difficulties (This title is discussed further in Chapter 4).

2.8. Parents' views and mediation

Although there is no specific research on parents' views on coding or programming tools, relevant studies examining parents' views on digital technologies has been reviewed in this title. Children (aged 0-8) increasingly use technologies at home. Holloway *et al.* (2013) note that new products and apps for this age group are released almost every week. Besides, cutting-edge technologies such as smartphones and tablets are easy to be used by babies or toddlers because they do not require complicated and higher motor skills or difficult process such as

using the keyboard or clicking the mouse. Such technologies are usually turned on at the touch of a button. Toddlers and other pre-schoolers love to play with digital tools and often do it as part of their interaction with adults. For some parents, their children's digital competency at a very young age might be very proud, whereas, for others, it might be something worrying or needs to be controlled.

A survey of 109 Australian parents showed that almost two-thirds of aged 3-5 children have one or more tablets at their home, and their average tablet usage is 20 minutes per day. 70% of the parents in the research believed that tablets could support the early literacy development of children and 69% reported that tablets were easy to use by their children. 53% supported the idea that the children should have access to tablets at preschool (Neumann, 2014). Another survey of 152 Croatian parents of children (3-7) showed that although participants' homes were well-equipped with digital technologies (around 99% of the surveyed parents had a computer, tablet or smartphone), they were anxious and not always willing to allow their children to use digital technologies (Mikelić Preradović *et al.*, 2016). Parents' common concerns and fears were the potential negative effects of digital technologies on their children's social lives and health such as the increased risk of obesity. In Vittrup *et al.* (2016)'s research, interviews with 101 US parents of children (aged 2-7) and 39 children (aged 3-6) showed that parents had generally positive attitudes toward technology and media, and their demographic features had no impact on their media and technology attitudes. Furthermore, there was a positive correlation between parent's attitudes and children's exposure to media and technology. More importantly, researchers noted that parents' willingness to allow their children to access the media and technology may be associated with how valuable they consider these technologies. It is, therefore, important to discover how valuable the coding tools for parents.

The dramatic increase in the usage of the internet by both pre-school and young children leads an uncertainty for parents and policymakers related to its advantages and disadvantages and how best to assist children's engagement with the Internet in a riskless way (Holloway *et al.*, 2013). Moreover, Zaman *et al.* (2016) noted that parents are increasingly struggling with the digital world they are not familiar and are forced to seek a balance between minimising children's exposure to digital risks and providing digital opportunities equally with them. This struggle stems from their efforts to assist children's media use, to provide children with opportunities to play, learn and socialise while trying to limit other factors (Zaman *et al.*, 2016). However, they are often less knowledgeable (digital immigrants with Prensky (2001)'s

discourse) than their children and may not know how to control or use digital tools. Given this lack of information, some experts argue that parents should restrict the exposure of preschool children to digital technologies. However, instead of making digital technology a forbidden place in children's lives, it may be more appropriate to mediate children's technology use with the active participation of their parents and older siblings (Holloway *et al.*, 2013). Therefore, Holloway *et al.* (2013) pointed to the importance of evidence base in order to enlighten the relevant policy, raise public awareness, support safety education and parents in the efficient mediation of their children's these platforms use.

Debates on digital technologies may make parents concerned. There is an ongoing process to find a scapegoat for children's safety issues online. Some lays the burden on the app producers. Chau (2015) noted that Apple, one of the app providers for children through the Apple store platform, offers inadequate information or guiding for the apps designed for young children that undergo a minimum content inspection aside from a superficial control to protect children's privacy and preventing in-app purchases. For some others, parents have the responsibility to monitor and protect their children, hence they should be conscious and watchful of their children's engagement with digital tools and apps. Therefore, parents must develop some strategies to prevent their children from unwanted issues. However, their strategies have an important role in child's development because parents not only apply their offline mediation strategies to children but also adapt their own online mediation strategies to the child's developmental stage (Nikken and Jansz, 2014).

On the other hand, 'How well do parents know about their child's interaction with technological tools?' is an important starting question to discover the views of parents about technology. To answer this question, Marsh *et al.* (2015) examined what factors affect parents' app choices for their children. Findings showed that most of the parents were conscious while looking for new apps. For their app searching process, parents reported that their priorities were that the apps were perceived should be fun, educational, and easy for their child to use. In addition, app downloading barriers for parents were also investigated. The main barriers for parents were financial; these were high priced apps (35%) and in-app payment concerns (34%). The third barrier was noted by 29% of parents as the quality concern of the apps.

Apple Store and Google play store operate the apps market as two dominant players with more than one million apps per each (Chau, 2014). The number of apps increases rapidly since app

developers constantly upload new ones. Therefore, making the correct selection of apps becomes potentially overwhelming for parents and educators. More and Travers (2013) suggest that this process is challenging as the Apple store and Google play store do not conduct meticulous control and evaluation of apps which target young children. In 2013, Apple Store launched the category for 'Kids' so that app manufacturers can assign a targeted age group such as 5-years-old and under or 3 to 5 years old that makes easier to discover content on the virtual marketplace of Apple. In his PhD thesis, (Chau, 2014) examined how appropriate the selected 100 educational apps from Apple store for iPad in terms of promoting optimal development of the targeted pre-school children (aged 3 to 5) by using content analysis method. The research findings showed that just above than half (58%) of the apps were developmentally meaningful, many of them offered limited content.

Summary

In this chapter, I reviewed the existing literature regarding the questions of this thesis. This review has addressed the gaps in knowledge and debates on the current situation. In particular, the need for the investigation of the perception and knowledge of parents about digital coding tools and the examination of the contribution of these tools to targeted age group children has been established. The first goal of this thesis, therefore, is to review the literature systematically to explore current digital coding tools designed for children aged 0 to 8 and what skills of young children can be promoted through these tools. The second goal is to discover parents' perspectives and knowledge about coding tools through interviews.

CHAPTER 3: METHODOLOGY

So far, I have explained the context of the research and reviewed some relevant literature to shed light on the present study. The review of the literature in the previous chapter showed that coding is an emergent topic which needs to be understood and explored more in early childhood. Therefore, this thesis aims to examine digital coding tools designed for young children (aged 0-8) in literature, the contribution they make to children's learning and the views of parents towards these tools. The research questions of this study were placed in section 1.2. In this chapter, I aim to provide a detailed account of the methodological approach of the study.

3.1. My Methodological Approach

I believe that knowledge is as valuable as it serves human purposes (Moon and Blackman, 2014) and thus adopting a pragmatic approach in this research. Pragmatically speaking, I chose systematic literature review and interview methods in this research context since I believed that they were appropriate to answer my research questions and were complementary by elaborating, enhancing, demonstrating and clarifying the results obtained from systematic literature review method with the results obtained from interview method (Onwuegbuzie and Leech, 2005). Greene *et al.* (1989)'s characterisation of complementarity design was largely based on the uses of the findings of relatively similar questions obtained from qualitative data and quantitative data. However, I used complementarity design in this research to enrich the findings of different questions obtained from two different qualitative methods. Creswell (2013) notes that whether we are aware or not, we always have certain beliefs and philosophical assumptions for our research. The challenge lies first in recognising these assumptions and beliefs, and secondly in deciding whether we actively involve them in our research. In this research, I interpreted and shaped the data that I found through my beliefs and experiences. Therefore, as Greenbank (2003) notes, this process was not value-free.

3.2. Data Collection Methods

In alignment with my methodological approach, this research was undertaken using a mixed methods approach. I reviewed previous studies conducted by other researchers as the first method in this study. It is important to acknowledge that, as an individual, I have already constructed some beliefs and ideas about digital coding tools. However, these may change in

the course of the research. Arguably, the more I read others' studies, the more my existing knowledge will be reconstructed. As the second method, I examined parents' views on digital coding tools via semi-structured interviews. I want to repeat once again that I have already had some expectations about the views of families, but they may change.

Table 1. Data Collection Methods

Method	Digital coding tools designed for young children	The contribution of digital coding tools to children development	Parents' views
Systematic literature review	✓	✓	
Semi-structured interview			✓

Although conducting an empirical study on children's engagement with coding tools are widely acknowledged, it could not be possible in this study because of the time and some other limitations. However, doing a systematic literature review allowed me to discover more information about coding tools and their contribution to young children's learning than those would be discovered in a single experimental study involving only 1 or 2 coding tools.

3.2.1. Systematic Literature Review

Literature reviews may have varied purposes such as to investigate old theories and propose new ones, consider the evidence of a particular topic, provide a base for suggestions for effective interferences (Petticrew and Roberts, 2006) or provide an explicit, targeted answer to certain research questions and enable for reproducing in a way which minimises prejudice (O'Brien and Mc Guckin, 2015). Wellington (2015) suggests that doing literature review research has various advantages in education. It may provide a historical perspective and a great source of additional data on a specific field. In addition, it may be cost-effective and productive while enhancing the trustworthiness and validity of the research.

Literature review (Chapter 2) showed that there is a lack of knowledge of the contribution of digital coding tools to young children's learning. In this study, therefore, I used systematic literature review method to systematically assess what is already known about this topic across a wide body of literature. In particular, I did this by creating a database of existing evidence for the skills promoted by digital coding tools in the previous works.

Bettany-Saltikov (2012) suggests there are three different types of systematic literature review research methods: quantitative, qualitative and mixed method. Systematic reviews usually depend on results from quantitative studies. However, the advantages of the inclusion of qualitative studies are increasingly recognised (Tacconelli, 2010) and may improve the quality of the review (Kachouie *et al.*, 2014). Investigating digital coding tools' contribution to young children's learning in early childhood might be multifaceted and complicated, therefore, I believe that using qualitative sources as well as quantitative sources provide me with a wider window in the literature review.

A systematic review requires a specific protocol that indicates how the methods will be applied (Petticrew and Roberts, 2006). In this study, I followed the principles laid down in the book 'Systematic Reviews in the Social Sciences: A practical guide' (Petticrew and Roberts, 2006) to establish a reliable literature review. The book that I followed suggests consideration of working with others as a steering or advisory group. However, in this study, I conducted my research on my own, therefore, creating such a group was not possible. According to the guide book, there are six stages in carrying out a systematic literature review. My research followed them step by step. These stages are;

Clearly defining the question that review purposes to answer

In this study, my questions to be answered by systematic literature review were;

- What digital tools can be used to teach coding to young children in early years?
- What skills does the literature suggest digital coding tools promote?

Determining what type of studies will be located to answer questions

To reach more relevant studies, three explicit criteria were determined based on the following terms; age, content and publication type. As for age criteria, my study mainly targeted young children (aged 0-8). For content, merely studies including digital coding tools and children's engagement with them in the context of early childhood education were considered. Another criterion for content was that including only studies exploring skills or knowledge children developed within a coding/programming environment. My research for this review was limited to conference papers and articles in journals. The criterion for the inclusion of articles was that including only publications which have been published in a peer-reviewed journal. However, review papers, book chapters and books were excluded from this study. As a final criterion,

only publications in English that were available as full text were considered for inclusion in this research. There was no specific criterion for the publication year to reach further studies in this area.

Carrying out a comprehensive literature search in order to locate those studies

Based on the criteria provided in the previous step, between June 2019 and July 2019, the following two different platforms were selected and searched: Google Scholar and StarPlus (including EBSCOhost, Springer, ProQuest databases and more). These specific platforms were chosen due to the high level of regard achieved in their respective industries and rich content. I accessed these resources online and then searched for key phrases: 'Coding children', 'Computer programming children', 'Robot programming children' and 'Programmable Toys' in the keyword, title and abstract sections of studies.

Screening the results of that search (sifting the retrieved studies, deciding which ones will be included according to inclusion criteria.

After searching databases, I screened the first couple of hundred results from each search engine. Although I used specific criteria to search these platforms, I had some irrelevant or overlapping results. In this case, I have reviewed the abstracts and methods of the articles with a few screenings and excluded the following article types from this study; (1) studies not mentioning the outcomes of children's engagement with digital coding tools, (2) studies involving older children (over 8) and young children together as participants (due to the difficulty of extracting data specific to younger children), (3) studies targeting teachers or parents rather than children. Overall, 13 articles were selected to be reviewed in this study.

Synthesising the information and assessing the findings

In order to organise the data extracted from the reviewed studies, I created a descriptive table Table 2. in Chapter 4. Petticrew and Roberts (2006) noted that presenting the reviewed studies in a basic table with their overview might make it relatively easy to interpret. Therefore, I decided to create such a list by believing that it will allow me to organise my ideas as well as studies for analysis. This table presented the included studies' author, journal, year, aim, method and participants information as well as the name of digital coding tools investigated in research and outcomes of the studies in terms of the contribution of digital coding tools to children's learning. In order to reach this information, the included studies were read 2-3 times in detail and summarised to add to table. In summarising the outcomes of the included studies,

numerical results were not included, but their interpretations were addressed. In Chapter 4, this table was analysed with my interpretation and discussion on some of the key issues raised by the reviewed studies.

Drawing on Table 2, I created Table 3 where only digital coding tools and the skills promoted by them were matched (in Chapter 4). To decide which skills will be in this table, the method, results and conclusion chapters of included studies were reviewed again with thematic synthesis. A thematic synthesis focused on the essence of the findings and the recommendations arising from the studies in the review (Denscombe, 2017). By doing so, themes that emerged in the existing literature were added to table. For each digital coding tool, the skills promoted by them were marked.

Disseminating the findings of review and writing up a report

The main aim of this systematic literature review was to uncover what existing literature says about digital coding tools and their contribution to young children's learning and to make deductions by identifying some higher-level insights across multiple contexts. I wrote up a detailed analysis and discussion report that I reached at the end of the systematic review process in Chapter 4.

3.2.2. Semi-structured Interviews with Parents

As a second method of my research, semi-structured interviews were done with two parents to gather their thoughts and attitudes towards coding and digital coding tools. The interview method was chosen to gather the perspectives and knowledge of parents in depth (Bell, 2005). Based on the interpretivist paradigm, parents were the active contributors to this research in their context, rather than subjects being studied (Carter and Little, 2007). For semi-structured interviews, Opie (2004) note that there is no longer straightforward relationship between the questions asked and the conclusions drawn. In other words, although I thought how well my questions were, these questions may have a different meaning for my participants and may, therefore, be answered differently by them. Reducing these problems is not as easy as we might think but to do so requires a recognition of how significant it is to establish a harmony between the researcher and participants and consciousness of the complexity of social meanings (Opie, 2004). Besides, there was always a possibility of my assumptions and prejudice creeping into the interview that I conducted. Correspondingly, this process was somewhat unpredictable in

what the quality outcome of the findings could be due to the method involving asking people about a new topic although I followed the steps below, taken place in Opie (2004)'s book.

Sketching the Interview Schedule

Interview questions were prepared in light of the research questions of this study. Therefore, they aimed to elicit parents' existing knowledge, attitudes and perspectives towards coding and digital coding tools. As designing interview questions, less structured questions were preferred to more structured questions. By doing so, interviewees would become more at ease with the researcher, and the researcher would follow up with the interviewees' ideas (Opie, 2004). After writing down the questions, draft versions of them were discussed with my supervisor and decided on their final form.

Piloting

A pilot interview was conducted with a Master's student. It showed me that the approximate duration of the interview was around 15 mins, and there were no ambiguous or confusing questions.

Selecting Participants

Only parents with children aged from 0 to 8 and living in Sheffield were determined as eligible to take part. One of my local friends helped me to reach her two neighbours. The age of the children of the participants was 1 year and 10 months and 1 year and 11 months. The inclusion of two parents did not represent a great sample, but it was not a problem since the interview research was based on an interpretivist approach and did not aim to generalise any outcomes.

Collecting Data

Participants who agree to be interviewed deserve consideration, and the researcher should fit into participants' plans regardless of the fact that how convenient these plans for them (Bell, 2005). Before planning the interview, therefore, participants were asked in terms of their preferences for the day, time and place. According to their demands, two participants were interviewed for 15 minutes on the same day in home environment. The interviews were recorded to be transcribed later. It enabled to preserve the natural language and objective record but also led to reach too much data including irrelevance ones and consume time (Opie, 2004).

Analysing Data

To examine the answers of the participants, thematic analysis was used. This method enabled me to identify, analyse and report the emerging themes in the collected data (Gray, 2014). For thematic analysis of interviews, I followed the phases suggested by Braun and Clarke (2006). To familiarise with the transcribed data, it was read several times. Interesting points in the parents' answers were coded, and then the codes were collated into potential themes emerging from interviews. Each theme revealed a significant point within the data regarding the research question and represented similar answers in responses within the data set. The features of each theme were explained to learn what the overall story of analysis tells. The selected extract of parents' interview was analysed and discussed in terms of research questions and the literature. As a result, a report for this thesis was produced in Chapter 4.

3.3. Ethical dimensions

Since secondary sources were used in the systematic literature review method, it did not require ethical approval as per the guidelines of the University of Sheffield. However, there were some ethical issues in relation to the review of secondary sources. Regarding the inclusion criteria of the studies, I was as neutral as possible whether they were appropriate for analysis. The findings of this review largely depended on how the data from selected studies were manipulated. However, this was an inevitable handicap of doing secondary sources analysis. Besides, as a stand-alone method, analysing studies can bring a very specific and limited approach to experiences and processes (Flick, 2014). To reduce this risk, critical evaluation of all studies was given importance while reading. On the other hand, the awareness of the risk associated with subjective or misinterpretation and to adhere to ethical principles, the research tried to be written to accurately reflect the findings of the authors placed in the review to minimise this risk. However, since this study was conducted by only one researcher, there was no other researcher to discuss and evaluate the accuracy of the findings of reviewed studies.

As for the interview method, I applied for ethical approval from The University of Sheffield by providing my research information sheet, participant consent forms and other necessary information. I believed that it was important to ensure that all participants were fully aware of the aim of the research, their rights and how ethical and safe this process was. To provide this information, participants were provided information sheet and consent form before the interview and they signed it. Additionally, they were reminded that their personal information will not be used, and they were free not to answer any questions.

Summary

This chapter has attempted to explain the methodological approach and process of this research in as much detail as possible. The next chapter shows the findings and discussion of the systematic literature review as well as the parents' interview.



CHAPTER 4: DATA ANALYSIS and DISCUSSION

This chapter has attempted to derive meaning from the collected data from the systematic literature review and the interviews with parents, and that meaning-making process “involves consolidating, reducing, and interpreting what people have said and what the researcher has seen and read” (Merriam and Tisdell, 2016, p.202). Basically, this chapter has attempted to answer the questions of this research by analysing, interpreting and discussing the data. This meaning-making process was supported by existing literature. It began with the findings, interpretation and discussion of the systematic literature review, and ended with the findings, interpretation and discussion of the interview with parents.

4.1. Analysis and Interpretation of Systematic Literature Review

Overall, 13 studies were reviewed and listed in Table 2. This table provided an overview for further analysis by including studies’ author, journal, year, aim, method and participants information as well as the name of digital coding tools investigated in research and outcomes of the studies in terms of the contribution of digital coding tools to young children’s learning. It is important to note that as a weakness of using secondary resources as the data for the first method of this research, my analyses and discussions depended on the aim, findings and interpretations of the reviewed studies. Another important point is that while one study may mention that children practised their social skills via an X app, another study using the same app may not mention about enhancing social skills. This situation may stem from the different aims, design and the offered environment in the reviewed studies.

The reviewed studies were sorted according to their publication year from older (1984) to newer (2019) in Table 2. This time sequence supported the argument of Kölling (2016)’s on how old the history of educational coding or computer programming tools is. Besides, the accumulation of the reviewed studies in the last decade supported the literature that teaching coding to children has gained an interest in recent years (Gordon *et al.*, 2015).

Table 2. Descriptive Table for Reviewed Studies

Authors	Journal	Aim	Method	Participants	Digital coding tool	Outcomes
Clements and Gullo (1984)	Journal of Educational Psychology	To compare the influence of computer programming and computer-assisted instruction on children's cognitive style.	In this experimental study, children engaged in two different program environments with researchers in groups of 2-3 for 12 weeks and were pre-tested and subsequently tested to assess differences.	18 children with an average age of 6 years and 11 months	Computer assisted instruction (CAI) and Terrapin Logo computer programming	Children who engaged in Logo programming showed significant improvement for fluency (the number of children's original ideas), originality (how original the ideas they have are) and divergent thinking score between pre and post-test. They also outperformed on metacognitions tasks.
Levy and Mioduser (2008)	International Journal of Technology and Design Education	To investigate children's perspectives from psychological or technological explanatory frameworks by observing their descriptions for the behaviours of a robot as they learn how to program it.	Content analysis of the observations and records that examine each child's interaction with the robot during five 30-45 minutes sessions for one week.	6 children aged 5.6–6.3 years in Israel	A physical robot (made with Lego) and iconic-programming interface	Children developed a technological perspective (defining the technological relationship between commands and robot's behaviour) in deciphering more complicated sets of rules. Besides, they explored and constructed abstract concepts and schemas in the concrete robotic environment.

Fessakis et al. (2013)	Computers and Education	To explore the dimensions of 5-6 years old children's problem-solving ability in two programming environments	An exploratory case study where children tried to solve six path problems and one maze problem in the programming environments	10 kindergarten children in Greece	Ladybug leaf and Ladybug maze programming environments	Children had opportunity to improve their mathematical skills (counting, number comparison, angle turn concepts), fundamental computer programming skills (commanding, executing commands sequentially, testing), communication and collaboration skills and problem-solving strategies, in particular, moving step by step. They did not face any problem for detecting the position of the ladybug relative to the leaf, but they had trouble expressing the direction verbally.
Flannery and Bers (2013)	Journal of Research on Technology in Education	To examine how cognitive patterns of children are related to their programming approaches and their success in robotics context, what children can understand about programming and robotics and how children's learning can be supported through those tools	A retrospective analysis of a process that children attended one small group session for pre-assessment and introduction to programming and 3 individual sessions where they build a robot vehicle and learnt the concepts of programming. Post-assessments happened in the final session.	29 children aged between 4.4 years and 6.6 years in the USA	CHERP programming technology designed to create behaviours for robots that children build from LEGO robotics construction kits.	Children in the pre-operational stage of cognitive development performed only 1 or 2 actions based heavily on trial-and-error, and had difficulty in assessing whether the robot executed the expected action and if it did not what they were supposed to correct it. There was variety in the unique mixes of systematic and intuitive strategies followed by each child in the transitional phase (between pre-operations and concrete operations). The children in early concrete operations used cognitive strategies to create and adjust their programs and were quick to recognise the error and try to correct it.

Kazakoff et al. (2013)	Early Childhood Education Journal	To investigate the effect of programming robots on children's sequencing ability.	Participants participated in a 1-week intensive robotics workshop at an early childhood STEM magnet school, and their sequencing skills were evaluated before and after this workshop using a picture story sequencing task.	29 children (13 pre-kindergarteners and 16 kindergarteners) in the USA	CHERP	Children engaging in robotics and programming significantly increased their sequencing abilities which are applicable to several fields such as mathematics, reading, and even simple daily tasks.
Gordon et al. (2015)	Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction	To examine children's programming activities as they learn how to socially engage with a robot by using stickers that they create to command it.	Qualitative analysis of children's engagement with social robot toolkit by observing their skills and examining their understanding before, throughout and after the interaction process.	8 children aged 4-8 years	DragonBot is a cute furry dragon controlled by a smartphone or tablet	Children explored a variety of computational concepts and created their own basic rules by a single trigger and event. They perceived event-driven logic and rules such as the separation between commanding and programming and improved their explanations about the programming process.

Papadakis <i>et al.</i> (2016)	International Journal of Mobile Learning and Organisation	To examine the impact of using ScratchJr on children's basic programming skill acquisition and computational thinking development in pre-school education.	A case study where children had a chance to have 11-hours programming education and to create their own project in the last 2 hours	43 pre-school children in Greece	ScratchJr	ScratchJr enhanced children's problem-solving skills and collaboration with their peers as they work on their projects. Children advanced computational and digital skills (abstraction, compartmentalisation and synthesis capabilities) that facilitated their problem-solving process and helped them to create animations, collages, stories, and games. They used reverse logic in their commands to reconstruct necessary steps.
Newhouse <i>et al.</i> (2017)	Australian Educational Computing	To see whether children's interaction with digital programmable toys would address some requirements of the Digital Technologies topic taken place in the Australian Curriculum.	A two-stage exploratory study that collects data by observing children's interactions with two different programmable toys during free play and analysing their behaviours with a checklist in two early childhood classes over two years.	A total of 50 children aged 4-6 years in Australia	Sphero (a programmable Bluetooth ball controlled via iPad devices and Beebot (controlled via interface buttons)	Though most of the children did not need support to use these two devices, their uses were pointless such as chasing and racing games unless they were given an explicit scaffolding. For BeeBot; Children randomly pressed the buttons and struggled to understand directions, therefore a few of them demonstrated pre-programming skills. Some counted and measured the steps. More capable children helped less capable ones. For Sphero; Children randomly used and struggled to operate it, therefore did not display much improvement in pre-programming skills. Measuring the steps were hard for them because its movement looked like rolling rather than stepping.

Taylor <i>et al.</i> (2017)	Journal of Special Education Technology	To assess the feasibility of teaching basic computer programming skills to children with Down syndrome by using concrete manipulatives, robot and evidence-based practices such as explicit instruction	A single case study where participants with similar mathematics abilities expected to complete each level by identifying and sequencing programming symbols with a maximum of one error over two trials to move to the next level in the four steps treatment	3 children with Down syndrome aged 3.4-5.1 years in the USA	Blockly application to run Dash Robot	3 participants were able to learn and perform simple computer programming skills.
Gomes <i>et al.</i> (2018)	International Journal of Child- Computer Interaction	To explore how sequencing, loops and conditionals concepts are represented in experience-based digital games and how they contribute to children's comprehension of these concepts.	An exploratory study where children participated in activities in the Informatics class for 9 weeks with 40-minute sessions. Data were collected unstructured and semi-structured interviews, and observations.	42 children aged 5-7 years in Brazil	Code Baymax, Kodable, Code Studio, Lego Bits and Bricks, Lightbot and The Foos	Children perceived the proposed problem, constructed the solution in their mind and then inserted the instructions and executed. However, they had difficulties when they were required to complete long sequences (5 or more) and to combine various instructions due to their limited memory. Overall, children failed repetition loops in all games, they found it hard to understand and apply. They were not able to grasp using the repetition command for 2 or more instructions. For instance, if the character needed to walk and jump two times, children thought the character will walk twice and jump twice instead of walking-jumping and then walking-jumping when they inserted the instructions. For conditionals, only Kodable and The Foos were used. Though children could not use conditional in the Foos, they had satisfactory comprehension in Kodable. Even though there is no consensus about the order of presentation of programming concepts to the children, Kodable was the only game analysed that introduced conditionals before repetition (p.83).

Messer <i>et al.</i> (2018)	Education and Information Technologies	To compare educational programming based on a digital device with similar based on paper and pencil programming activities in terms of their effectiveness in children's mathematical abilities, spatial awareness and working memory.	Children were divided into three different groups: programming combined with iPad technology, programming combined with paper and pencils, and a comparison condition that included simple mathematical calculations. Two 10-minute intervention sessions were conducted each week for a six-week period with pre-intervention and post-intervention tests to assess children's mathematical abilities, spatial awareness, and working memory.	41 children aged 5-6 years in the UK	Bee-bot iPad app	There was a significant increase in children's mathematics abilities and spatial awareness, whereas there was no significant change in children's working memory from pre-test to post-test via iPad-based Bee-Bot programming. However, results showed that all children in three different groups showed improvement in terms of mathematics and spatial awareness, hence programming with Bee-Bot digital devices was not more effective than paper-based programming.
Pila <i>et al.</i> (2019)	Computers & Education	To examine the effectiveness of two different coding apps in children's learning.	A multi-method analysis of a one-week summer program designed to teach young children coding with two coding applications. The researchers conducted interviews, classroom observations, pre-test and post-test evaluations.	28 children with an average age of 5.15 years in the USA	Daisy the Dinosaur and Kodable coding apps	Children improved their command knowledge and explained move, jump, and grow sequencing commands except for the condition command 'when'. Children learned basic coding skills via these 2 apps. However, they have a shortage of vocabulary and verbal skills to define the concept.

Strawhacker and Bers (2019)	Educational Technology Research and Development	To assess young children's performance on a programming environment after participating in a 6-week program.	In this mixed case study, children's responses to post-intervention assessment were evaluated in terms of their programming comprehension and knowledge. The answers were numerically scored against a series of correct answers and tendencies in scores were reported. The wrong answers were also investigated for qualitative tendencies.	A total of 57 children aged 5-8 years (20 kindergarteners, 15 first graders, 19 second graders) in the USA	ScratchJr	Children developed causal reasoning skill which was key to giving meaning the programmatic relationship in ScratchJr environment. While younger children developed observations and inferences about the causes of visible cause-effect relationships, older children developed theories about invisible causal relationships. Children used problem-solving strategies. Meanwhile, they also developed their spatial knowledge while solving tasks. Throughout the process, children improved their verbal reasoning and social reasoning abilities.
-----------------------------	---	--	---	--	-----------	---

4.1.1. Answering the RQ1: The existing digital coding tools designed for young children

The systematic literature review identified 16 digital coding tools to teach young children coding. These are namely, Terrapin Logo, Iconic programming Interface, Ladybug leaf and maze, CHERP, DragonBot, ScratchJr, Sphero, BeeBot, Blockly, Code Baymax, Kodable, Code Studio, Lego Bits and Bricks, Lightbot, The Foos and Daisy the Dinosaur. The reviewed studies investigated these tools and children's engagement with them. However, it should be noted that this study included only research that met the explicit criteria, so digital coding tools designed for young children are not limited to these 16 tools. The review of the studies has revealed some main topics to be considered, they are discussed below.

4.1.1.1. *The importance of the design of digital coding tools*

In general children were enthusiastic and eager to take part in the coding process and highly engaged in digital coding environments, but, initially, they needed some time to familiarise with these tools in terms of understanding the direction buttons and the interface of games. To explore whether the appeal of apps influences children's engagement, Pila *et al.* (2019) investigated two different apps namely, Kodable and Daisy the Dinosaur. The study showed that although the appeal of both programming environments was quite high to children, only the appeal of Daisy app was found as a significant predictor of children's improvement. However, there was no further information in terms of what specifically makes these apps appealing to children. Mcmenamin and Linebarger (2010)'s research examined the appealing of the Island computer game to children at preschool age (average 58.63 months) and kindergarten age (average 73.08 months). They found that earning tickets and buying prizes are appealing features of the game. However, they noted that these features are more supportive and motivator for children aged between 6 and 11 since they are willing to achieve mastery. The characteristics of the game character were also important. Older children found the Island game character a bit childish. Therefore, the design of the apps was key to attract children's attention. Similarly, Fisch (2004) asserts that when a television program is not appealing to children, they are not willing to watch it, thus eliminating the potential educational benefits of the program. Although past research has mostly focused on the appealing of educational television (Fisch, 2004), their findings would be likely to be applied to other educational media tools. Therefore, it can be assumed that when children find coding apps more appealing, they

can show more attention to them and interact with them more often, thereby increasing their learning (Pila *et al.*, 2019).

Another reviewed study (Gomes *et al.*, 2018) marked the complexity of visual and textual programming languages for the children's age range as the main challenge. They thus pointed out the inappropriateness of the game interfaces as a major factor in children's failure. In the Code Baymax programming environment, for example, children were expected to develop their repetition skills. However, their learning was limited because the parameters could not be changed (only triple repetition parameters with different directions were available). In the same programming environment, however, children were able to develop their sequencing skills. Likewise, the Lightbot programming environment was one of the most complex environments for children to practice repetition skills due to the lack of visual specific commands representing the repetition concept. To find the source of these difficulties, researchers started an in-class activity where children should help their peers, who act like a robot, to cross the room. To complete this path, they needed to orient their peers in a way that avoid obstacles and use sequencing and repetition instructions. Similarly, researchers used on paper supplementary activities to support children's repetition skills in The Foos programming environment by using its syllabus. Although all children successfully completed on paper repetition activities of The Foos programming, then in the next trial, they were still not able to use repetition block on the digital environment. At the end of these activities, researchers found an interesting result that although children could easily identify and describe the repetition parameters in the physical activities, they were not capable of doing so in digital games. In short, if a child has difficulty in completing a certain level in the programming environment, this may not always be due to his/her incompetence. This inference replicates the findings of More and Travers (2013)'s study, where they stressed that so-called educational apps in the marketplace do not have the due design, direction, content and other significant qualities that should be in high-quality educational software that may result from uncontrolled increasing apps with a minimum content inspection (Chau, 2015). Therefore, teachers and parents have a responsibility to select and introduce to high-quality apps to children.

4.1.1.2. The differentiation of the promoted skills of children at different ages

In some games, however, children at different ages had different difficulties in implementing strategies and understanding how to manage the tools. Flannery and Bers (2013)'s study, for example, examined whether the age of children affects their CHERP programming approaches and achievement. They found that children with average age 5.2 years performed only 1 or 2 actions based heavily on trial-and-error and had difficulty in assessing whether the robot executed the expected action and if it did not what they were supposed to correct it, whereas children with average age 5.6 years showed variety in their strategies to solve problems. The children with average age 5.9 years used strategies to create and adjust their programs and were quick to recognise their error and could correct their errors. In sum, the age of children was a significant factor for their approaches, strategies and success in the programming environment. Therefore, Flannery and Bers (2013) assumed that children of different ages would take advantage of the learning goals and activities designed for their distinct cognitive features in programming environments. In Magnenat *et al.* (2014)'s research, for example, children from different age groups were taught robot programming and then were tested with the same task in the programming environment. Comparing the performance of groups with the same task, they found that most of the children were capable of understanding and solving simple tasks. Older children, however, showed better performance in complex programming tasks which required varied conditions or events.

4.1.1.3. The key role of adults in facilitating and enhancing children's learning

Adults played an important role in children's engagement with these tools. Levy and Mioduser (2008)'s research on Iconic Programming Interface noted that children were able to decipher the simpler robot's behaviour on their own, but their reasoning skills were enhanced by the adult's questions which draw children's attention to unnoticed behaviours of the robot. Newhouse *et al.* (2017) highlighted that children lost their interest in the Beebot app unless their teachers gave them support, detailed tasks or instructions. Similarly, Fessakis *et al.* (2013)'s study found that children who did not show a clear strategy to solve the problems in Ladybug leaf and maze app had lack of self-confidence, therefore, they asked for approval or support from their teachers.

These findings should remind us of Vygotsky's Zone of Proximal Development (ZPD). Vygotsky (1986) defines ZPD as a cognitive distance between what a child can do without help and what he can do his own after receiving help and guidance from a skilled peer or person as

solving a problem or executing a task. Indeed, literature has many studies on coding and robotics which implicitly address Vygotsky's ZPD as theoretical a framework. Jung and Won (2018), for example, systematically reviewed the research trends in robotics education for young children by examining included studies' theoretical traits. They found that 5 out of 47 reviewed studies used Vygotsky's ZPD as a theoretical framework, whereas Piaget's constructivism and Papert's constructionism (discussed further in Chapter 2.5) were two dominant theoretical frameworks. However, reviewed studies narrowed down these two dominant frameworks into a technological determinist paradigm which considers coding and robotic tools as shaping children's learning. Jung and Won (2018), therefore, drew attention to the need for alternative perspectives, particularly social and cultural perspectives, to better understand young children's engagement with coding tools. To understand the social and cultural relationship between these tools and children's learning, alongside Vygotsky's ZPD (focusing on the adult or peer mediated actions), Oliver (2011) suggests Cultural Historical Activity Theory (CHAT) (focusing on how children use coding tools to transform their world within materially and socially mediated process), communities of practice (focusing on a group of students who share a concern or passion for coding and interacts each other to learn how to code better (Wenger, 2009)) and Actor-Network-Theory (ANT)(understanding the interactions of teachers and students with coding tools). These frameworks may reveal how young children make personal, social, and cultural meanings from digital coding and robotic tools and in what ways they interact with these tools, their peers and adults beyond expected actions.

However, to support children, teachers must first have the basic skills to use technology. Hence teacher training programs are needed. The study of Yurt and Cevher-Kalburan (2011) with 100 preschool teachers in Turkey showed that almost none of the teachers received in-service education for using the computer in children's learning. Furthermore, a survey of 1329 pre-school teachers found that teachers may have some barriers to the adoption and use of technologies in young children's learning that stem from their knowledge, skills and attitudes (Blackwell *et al.*, 2013). They also noted that although teachers knew how to use technology, it did not mean that they believed in the value of technology for their teaching practices. Teachers were willing to use technology more often when they believed in the benefits of technology in children's learning.

4.1.1.4. The gender stereotype in children's computer science learning

There was no gender difference in children's performance in programming environments in the reviewed studies. Underlying this similarity is important because it breaks existing gender stereotypes and encourages girls' participation in computer science. Papadakis *et al.* (2016), for example, found that there was no significant difference between the performance of girls and boys in the ScratchJr programming environment. Similarly, gender did not affect children's learning to code with apps, even girls' mean score was higher than boys in the Kodable gameplay test (Pila *et al.*, 2019). The literature has pointed out that girls and boys at the early childhood stage have a similar level of success in programming environments. A study, for example, specifically examined the influence of gender differences in kindergarten students' robotics and programming achievement (Sullivan and Bers, 2013). They found that boys' scores were significantly higher than girls in only two areas and that in general, all children were able to successfully complete the program. Likewise, Martinez *et al.* (2015) taught basic computer science concepts to children aged between 3-11 years old. Results showed that girls and boys had similar performance in simple concepts; however, girls' performance in complex concepts such as combining computer science and robot programming was better than boys. Researchers, therefore, suggested that the "gender gap" in computer science may arise later in the teenage years.

4.1.1.5. The significance of how we introduce coding tools to children

The ways that adults introduce coding tools to children were important. In other words, the range of abilities that can be promoted via digital coding tools depended on how open-ended tasks and opportunities were provided to children. Gordon *et al.* (2015) provided children with vinyl stickers to control the DragonBot in their study and children created their own sequences, rules and rulebooks to command the DragonBot. It enabled children to explore a wide range of possibilities in programming environments and to find new ways to express themselves creatively apart from the tasks given by adults. As noted in Gordon *et al.* (2015, p.68), when a child was asked "How does the robot know the order?", he answered "It can be any order". The child's answer showed that he was not worrying about taking correct or wrong actions, and he was aware of the open-ended actions that he could explore. In this study, researchers also provided opportunities for children to articulate the programming process, and findings showed that children improved their explanations at the end of the session. Before the interaction process, for example, an 8-year-old boy explained how robots decide what to do

next: “They are probably ... wires and everything that make them do things and everything, electric gadgets and everything” (Gordon *et al.*, 2015, p.68). After the interaction process with the social robot toolkit, his answer changed to:

That electric gadgets and everything make something inside of it get programmed to what you say and what you do, so that the robot does what you want.

Similarly, Strawhacker and Bers (2019) combined lesson contents, in-class games and activities, and open-ended programming play in their research. They found that making lesson contents diverse such as by combining ‘Simon Says’ game with programming instructions that require children to physically act out the motions with their bodies augmented the meaning of the block icons in children's mind. Likewise, Karadeniz *et al.* (2014) conducted a case study with 25 5-year-old children. They first performed unplugged activities where children play direction games in the classroom, and then children were introduced to mobile coding game Kodable on iPads. The findings showed that combining in-class and digital games were helpful for children.

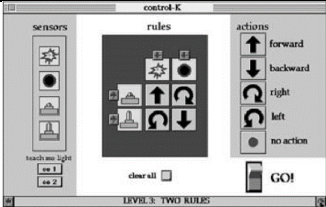
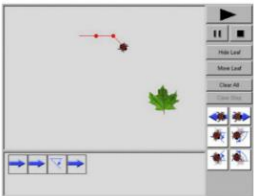
Overview

Overall, the reviewed studies showed the design of digital coding tools was important for children's engagement with them. Therefore, coding apps should be appealing to children (at different ages) in terms of the game character, level and prizes. The differentiation of the promoted skills of children at different ages was another finding. Studies have shown that children of different ages would benefit from learning objectives and activities in programming environments differently. Adults' key role in facilitating and enhancing children's learning in coding apps was another finding. Correspondingly, underlying theoretical frameworks for the design of research trends in coding and the need for social and cultural theoretical frameworks were also addressed. There was no gender difference in children's computer science learning in early childhood stage. Studies have also shown that the ways how we introduce coding tools to children were significant to provide meaningful experiences for them. Apart from the tasks required in coding apps, open-ended activities that combine in-class and digital coding games were proposed.

4.1.2. Answering the RQ2: The skills promoted by digital coding tools

Table 3. provides an overview of the digital coding tools and promoted skills via these tools.

Table 3. Matching table for digital coding tools and the skills promoted by them

The skills promoted by coding tools Digital Coding Tools	Basic Programming Skills				Cognitive Skills								Social Skills			Language Skills
	Sequencing	Loops	Conditionals	Commanding	Mathematics*	Creativity	Abstract Thinking	Spatial awareness	Metacognition	Reasoning	Computational thinking**	Deciphering Visual Cues	Communication	Pretend Play	Collaboration	Verbally Describing Process
 1.Terrapin Logo (Clements and Gullo, 1984)				✓		✓		✓	✓		✓					✓
 2.Iconic programming Interface (Levy and Mioduser, 2008)			✓	✓			✓			✓		✓				✓
 3.Ladybug leaf and maze (Fessakis <i>et al.</i>, 2013)	✓			✓	✓			✓			✓	✓	✓		✓	
 4.CHERP (Flannery and Bers, 2013) (Kazakoff <i>et al.</i>, 2013)	✓			✓						✓	✓					
 5.DragonBot (Gordon <i>et al.</i>, 2015)	✓					✓					✓	✓	✓		✓	✓

The skills promoted by coding tools	Basic Programming Skills				Cognitive Skills								Social Skills			Language Skills
	Sequencing	Loops	Conditionals	Commanding	Mathematics*	Creativity	Abstract Thinking	Spatial	Metacognition	Reasoning	Computational thinking**	Deciphering Visual Cues	Communication	Pretend Play	Collaboration	Verbally Describing Process
 6.ScratchJr (Papadakis <i>et al.</i>, 2016) (Strawhacker and Bers, 2019)	✓			✓		✓		✓		✓	✓	✓			✓	✓
 7.Sphero (Newhouse <i>et al.</i>, 2017)						✓					✓		✓	✓	✓	
 8.Beebot (Newhouse <i>et al.</i>, 2017) (Messer <i>et al.</i>, 2018)					✓	✓		✓			✓			✓	✓	
 9.Blockly works with Dash Robot (Taylor <i>et al.</i>, 2017)	✓															

The skills promoted by coding tools Digital Coding Tools	Basic Programming Skills				Cognitive Skills								Social Skills			Language Skills
	Sequencing	Loops	Conditionals	Commanding	Mathematics*	Creativity	Abstract Thinking	Spatial	Metacognition	Reasoning	Computational thinking**	Deciphering Visual Cues	Communication	Pretend Play	Collaboration	Verbally Describing Process
 10.Code Baymax (Gomes <i>et al.</i>, 2018)	✓															
 11.Kodable (Gomes <i>et al.</i>, 2018) (Pila <i>et al.</i>, 2019)	✓		✓							✓	✓					✓
 12.Code Studio (Gomes <i>et al.</i>, 2018)	✓															
 13.Lego Bits and Bricks (Gomes <i>et al.</i>, 2018)	✓															

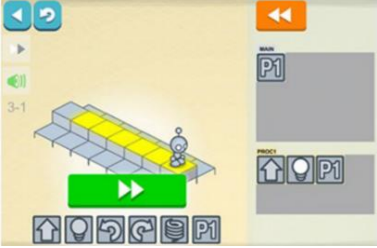
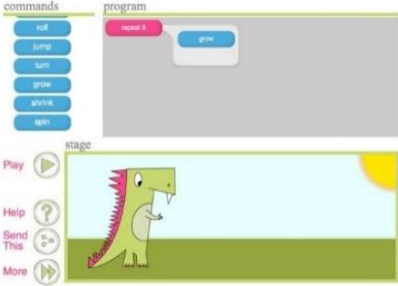
The skills promoted by coding tools Digital Coding Tools	Basic Programming Skills				Cognitive Skills								Social Skills			Language Skills
	Sequencing	Loops	Conditionals	Commanding	Mathematics*	Creativity	Abstract Thinking	Spatial awareness	Metacognition	Reasoning	Computational thinking**	Deciphering Visual Cues	Communication	Pretend Play	Collaboration	Verbally Describing Process
 14.Lightbot (Gomes <i>et al.</i>, 2018)	✓															
 15.The Foos (Gomes <i>et al.</i>, 2018)	✓															
 16.Daisy the Dinosaur (Pila <i>et al.</i>, 2019)				✓												

Table 3 presents 16 digital coding tools and the skills promoted by these tools in the reviewed studies. It was noticed that three of the coding tools were used in two different reviewed studies. These tools were CHERP, ScratchJr and Kodable. However, they were not given separately in the table above because they were complementary in terms of the promoted skills.

Drawing on the results of the reviewed studies, coding apps promoted potential 4 main skills (basic programming, cognitive, social and language) and 16 sub-skills (sequencing, loops, conditionals, commanding, mathematics, creativity, problem-solving, spatial awareness, metacognitive, reasoning, computational thinking, deciphering visual cues, communication, pretend play, collaboration and verbally describing programming process). However, it is important to note that these results depend on the objectives and frameworks of the reviewed studies. Other research with different objectives and frameworks may reach different results.

Similarly, Strawhacker and Bers (2019) has found that introducing coding to children in early years settings enhance their sequencing ability, design learning, executive functioning, mathematics and language development, and logical reasoning. Although children needed to use their fingers to press, tap and swipe icons on screens (Wohlwend, 2015) to command the characters or items in the apps, none of the reviewed research in this study mentioned the development of children's motor skills. Moreover, children should have fine motor skills and visual-motor integration to control the mouse or touchpad, and these skills are not highly developed in early childhood, thus, may hinder the effective use of the digital tools (*Dankert et al.*, 2003). Therefore, further research is needed in this area. As for the findings of Table 3, the promoted skills by digital coding tools are discussed below.

4.1.2.1. Basic Programming Skills

Overall, 14 out of the 16 coding tools promoted children's basic programming skills in the reviewed studies, except for Sphero and Beebot. Sequencing was the most common ability promoted among other basic programming skills that children enhanced with 11 out of 16 coding tools, it followed by commanding (6 out of 16), conditionals (2 out of 16) and loops (0 out of 16).

It is satisfying to see that most of the coding tools were found to enhance children's sequencing skills because it is a core skill in young children's learning not because sequencing is used not only in programming environment but also curricular frameworks and learning assessments where children are required putting items, actions, numbers in the correct order or retelling a story in a logical order (*Kazakoff et al.*, 2013). Similarly, research has suggested that children as young as age 4 are capable of learning how to programme a robot in a sequencing task correctly (*Pila et al.*, 2019). Commanding was the second most promoted skill, with studies

attributing this to 6 coding tools. These coding tools were noted as Terrapin Logo, Iconic programming Interface, Ladybug leaf and maze, CHERP, ScratchJr and Daisy the Dinosaur. Bers (2008) argued that commanding (giving instructions to coding tools) is a significant skill to support children to make their abstract ideas more concrete as they see the effect of their commands on the programming environment. The conditionals (2 out of 16) and loops (0 out of 16) were found as less developed basic programming skills by coding tools. Likewise, Bers *et al.* (2014) noted that loops and conditionals concepts are more abstract than other programming concepts, therefore, children in their study were not able to develop these concepts, and found them particularly challenging. Iconic programming interface and Kodable were the only two digital coding tools that children developed their conditional skills (the statements such as 'if, then, else' that lead the program what actions to take next). Nevertheless, one of the most striking findings was that none of the digital coding tools enabled children to enhance loops skills. Gomes *et al.* (2018) investigated the development of children's sequencing, loops and conditionals skills in 6 different coding environments, and found that loops tasks were one of the most difficult for children. In the study, children could not understand how to use the loops command for 2 or more instructions. For instance, if the character needed to walk and jump two times, children thought the character will walk twice and then jump twice instead of walking-jumping and walking-jumping. In the same study, however, children could use loops abilities in-class activities. Therefore, it can be suggested that loops concept should be introduced to children gradually shifting concrete approaches to abstract approaches as they are learning.

4.1.2.2. Cognitive Skills

Table 1. showed that the promoted cognitive skills of children in the reviewed studies were varied. Overall, children improved cognitive skills with 9 out of 16 digital coding tools. Blockly, Code Studio, Lego bits and Bricks, Lightbot, Code Baymax, The Foos and Daisy the Dinosaur were noted as 7 tools not mentioning children's cognitive development. Computational thinking skills were found the most common cognitive skill promoted by the 8 out of 16 digital coding tools. Creative thinking skills were found the second most common cognitive skills promoted by the 5 out of 16 digital coding tools. They were followed by spatial awareness, reasoning and deciphering visual cues skills with 4 out of 16 coding tools, mathematics skills with 2 out of 16 coding tools, and lastly metacognitive and abstract thinking skills enhanced with only 1 out of 16 coding tools. Cognitive skills were found to be the most

promoted field with coding tools. Likewise, previous research has emphasised that children participating in computer programming enhanced cognitive skills such as problem-solving, mathematics, and reasoning compared to children who did not experience similar environments (Clements *et al.*, 2001).

Metacognitive skills

The Terrapin Logo was the only coding tool enhancing children's metacognitive skills. Metacognitive skills refer to the ability of one's monitoring and evaluating his own cognitive process (Clements and Gullo, 1984), therefore it is an important skill in the development of children. Although many of the work on programming environments today do not mention their metacognitive contributions, Papert (1980 cited in Strawhacker and Bers, 2019) argues that programming environment enables children to evaluate their logic and thinking process from a bird's-eye view, and in this way, children are exposed to simple metacognition practices. Additionally, previous research has pointed out that there are many improvements in metacognitive skillfulness between the ages of 5-7 (Bryce and Whitebread, 2012), thus focusing more on promoting young children's metacognitive skills in programming environment might be considered in the future research.

Abstract thinking

Iconic Programming Interface was noted the only environment where children improved their abstract thinking. Differences in objectives and focus of other research may explain this result. Levy and Mioduser (2008) define a robot as a concrete system which embodies abstract ideas and concepts. According to them, children generate cognitive abstractions about this "abstractions-embedded concrete-agent" (p.353) and that is the realm of thinking processes, then they will call as the realm of "concrete-abstractions" (Mioduser *et al.*, 2009, p.20) in which children abstract schemas to understand the behaviour of the robot by intertwining the symbolic and the concrete.

Mathematics

A surprising finding was to see only Beebot and Ladybug leaf and maze coding tools promoted mathematical skills. It was surprising because educational coding and robotic tools are widely acknowledged as digital manipulatives which support children to develop a comprehension of basic mathematical concepts such as number, size, and shape in the similar way that traditional materials do (Resnick *et al.*, 1998). However, this result may be related to the research

objectives of the studies. While the mathematical skills that children enhanced on Ladybug leaf and maze tool were noted as one to one correspondence, counting, number comparison and angle turn concepts (Fessakis *et al.*, 2013), there was no detailed information about the skills promoted by Beebot (Messer *et al.*, 2018).

Creativity

Terrapin Logo enhanced children's fluency (the number of their original ideas), originality (how original the ideas they have) and divergent thinking score between pre and post-test (Clements and Gullo, 1984). In Dragonbot, children created their own vinyl stickers by using dry-erase markers that also enabled them to practise on their writing, drawing, and original ideas (Gordon *et al.*, 2015). Similarly, in ScratchJR, children created their own digital artefacts and characters, build animations, and stories (Strawhacker and Bers, 2019). Likewise, Resnick (2006) drew attention to the power of digital technologies to virtually represent unique and original projects that would be difficult or even impossible to perform in the real world. Contrary to these findings, however, Plowman *et al.* (2010) argued that one of the concerns of people who see the increasing technology activities in children's lives as threatening is the belief that technology promotes passivity and hence inhibits children's imagination. Meanwhile, it again reminds us of the importance of the quality of the tasks and opportunities are provided to children. Similarly, Marsh *et al.* (2018) investigated which apps foster children (aged 0-5)'s play and creativity in the UK. They found that children can develop creative thinking via apps, but the design of apps was key to enable children to produce original things. For the evaluation of the design of apps Robson (2014)'s ACCT (Analysing Children's Creative Thinking) Framework can be used. In ACCT Framework, they identified 3 particular behavioural areas (exploration, involvement and enjoyment, and persistence) that might be associated with children's creative thinking in digital environments. However, commercial interests have been found as a potential threat to limit children's play and creativity by using in-app advertisements (Marsh *et al.*, 2018).

Spatial awareness

The spatial awareness skills promoted by Terrapin Logo, Ladybug leaf and maze, Scratch Jr and Beebot environments. Clements and Gullo (1984) stated that children were able to grasp directions from their own and others' perspectives since Terrapin Logo programming required giving explicit spatial commands. Similarly, in the Ladybug leaf and maze environment children were required to move the ladybug towards the appropriate direction (Fessakis *et al.*,

2013). ScratchJr involved distances, locations, directions and rotations between objects required to be measured by children (Strawhacker and Bers, 2019). Since the educational programming tasks generally depend on finding the best route to reach a specific location, Baddeley (1996) noted that those environments may improve children's spatial awareness which involves the understanding of the relative position of an individual's to other objects in the environment. Moreover, Raghubar *et al.* (2010) stated that spatial awareness can be associated with mathematics achievement as children improve their measurement skills in distances or angles.

Reasoning

Reasoning skills of children were promoted by the Iconic Programming Interface, CHERP, ScratchJR and Kodable environments. Levy and Mioduser (2008)'s study noted that they challenged young children's reasoning with several problems in Iconic Programming Interface. It is known that children's reasoning dramatically changes throughout early childhood, and their reasoning competency might be an influential factor in children's achievement in programming (Flannery and Bers, 2013). Therefore, it is important to provide children with challenging environments, in which they need to think and evaluate what strategies to develop for the solution a problem, not only for programming but also for their daily lives. Children (at the age of around 6) rely increasingly on logical reasoning for causal relationships, empirical observations, and the distinction between appearances and reality that might be applied to such activities involving programming a robot (McDevitt and Ormrod, 2002 cited in Flannery and Bers, 2013).

Deciphering visual cues

Iconic Programming Interface, Ladybug leaf and maze, DragonBot and ScratchJR were noted as 4 coding tools where children deciphered the visual cues. Indeed, children need this skill in every coding environment, but this finding can be better explained with the differences in the aims of the reviewed studies. In simple words, if the authors of reviewed studies considered deciphering visual cues as a core ability that children should have already had, they then would not focus on the development of this ability in their studies. However, Gomes *et al.* (2018)'s study, for example, specifically aimed to understand how interaction elements influenced children's understanding of programming concepts. They found that the complexity of visual and textual language was crucial for children's engagement in these environments. Therefore, the selection of the apps that will be provided children is important because the complexity of

visual cues can be discouraging in terms of children's participation. What is more, Strawhacker and Bers (2019) drew attention to the need for more research to understand the engagement of children with specific symbols and concepts in the programming environment.

Computational thinking

Since computational thinking is widely associated with programming (Lye and Koh, 2014), nearly all coding tools were expected to promote children's computational thinking. Children's computational thinking skills were promoted by Terrapin Logo, Ladybug leaf and maze, CHERP, DragonBot, ScratchJr, Sphero, Beebot and Kodable. However, differences in objectives and focus of other research may explain this result. These tools challenged children to develop strategies to approach and solve problems. Papadakis *et al.* (2016) noted that children showed deep reflection and developed problem-solving skills while they were working on their own projects and were solving the problems in the ScratchJr environment. Wing (2006), the pioneer of computational thinking notion, claimed that computational thinking uses abstraction and decomposition when solving a large complex task. Similarly, the research found that children improved computational and digital skills such as abstraction, compartmentalisation and synthesise abilities in ScratchJr (Papadakis *et al.*, 2016). Therefore, researchers have noted that the ScratchJr has the potential to develop children's scientific and mathematical thinking. The age of children has found as an influential factor in their problem-solving strategies in CHERP programming environment (Flannery and Bers, 2013). In their study, they found that children with average age 5.6 had difficulty with recognising the problem, generating a hypothesis for the solution of problem, and attempting to solve the problem. However, older children were good at the recognition of whether an instruction was missing, not necessary, or out of order, since their cognitive flexibility was more developed than the others. There was only one evidence of promoted computational thinking with the Sphero (Newhouse *et al.*, 2017). A child tried to throw the Sphero off a ramp and fall it over the edge. It took him more than 15 minutes to try different strategies, but he eventually succeeded. This result re-emphasises the importance of persistence behaviour which is determined as one of the behavioural categories in Robson (2014)'s framework regarding creative thinking. To be able to reach the result, this child showed a high level of persistence, and so explored a different use of Sphero.

In an early work, DiSessa and Abelson (1986) defined programming environments as powerful reconfigurable tools that help students to analyse, organise, articulate, and evaluate their ideas

clearly and concisely in the problem-solving process. As developing strategies to solve a problem in a programming environment, 'solvers' express their thoughts, observe the results, clarify their thinking processes and receive immediate feedback, therefore not only develop problem-solving strategies but also metacognitive skills (Clements and Nastasi, 1999). Today all these thinking processes are called computational thinking (Wing, 2006), and it is seen necessary to teach every child growing up in the 21st century (Weintrop et al., 2016). Computational thinking is a type of analytical thought process (Bers and Horn, 2010) that requires children to use some approaches (Bocconi *et al.*, 2016) and to develop the proper methodology and models (Wing, 2006) as solving the problems. This thinking skill is not only applicable to a computer or digital environment but also daily life, therefore, García-Peñalvo *et al.* (2016) define it as an important transferable skill. In the past decade, therefore, coding, programming and computational thinking have been given increasing attention in the educational field (Bocconi *et al.*, 2016). Engaging in age-appropriate programming tools and curriculum enhances children's creative problem-solving skills and powerful transferrable skills (Bers and Horn, 2010). However, as cutting-edge tools reach ever-younger children, further research is needed to define the characteristics of age-appropriate technology, and expectations for them (Flannery and Bers, 2013).

4.1.2.3. Social skills

Overall, 5 out of 16 digital coding tools promoted social skills. However, this result may be related to different objectives of studies. Ladybug leaf and maze, DragonBot and Sphere were noted as 3 studies where children practised their communication skills. Children not only communicated with their peers, but also with DragonBot, they “made silly faces to the robot and even petted the robot and taught it to say ‘I love you’ (Gordon *et al.*, 2015, p.68). Ladybug leaf and maze, DragonBot, ScratchJr, Sphero and Beebot were noted as 5 studies where children worked collaboratively. Many of the children in Newhouse *et al.* (2017)’s research did not understand what to do with Sphero and Beebot tools, however, more capable children helped their less capable peers. It can be inferred from this finding that challenging learning environments may enable children to enhance their development to the next level (National Association for the Education of Young Children, 2009) and support their collaborative learning.

These findings provide evidence of positive social contributions of digital educational tools to children's development. Likewise, past research has noted that the incorporation of new technologies in young children's learning activities may foster positive peer-to-peer interactions and social development by encouraging teamwork and collaboration (Lee *et al.*, 2013). In contrast with the general view that the social development of children is at risk since they play alone in digital environments (Plowman *et al.*, 2010), the reviewed studies showed that children's collaboration and communication skills promoted.

According to Table 3, children benefited from Sphero and Beebot coding tools by using them in their pretend play. Regarding the traditional play of children emerge from digital environments, Plowman *et al.* (2010, p.72) noted that "there has been a technologisation of childhood". However, instead of separating their online and offline play, "we should view their experiences along a continuum in which children's online and offline experiences merge" (Marsh, 2010, p.25). Therefore, this change should not be seen as a negative thing for children. On the contrary, the ability to integrate children's everyday experiences on digital platforms into their games can be considered a successful orientation process.

4.1.2.4. Language skills

Children enhanced their language skills by verbally describing the programming process on the Terrapin Logo, Iconic Programming Interface, DragonBot, ScratchJr and Kodable. In these environments, children articulated the reasons for the robot's actions (Levy and Mioduser, 2008), their ideas about programming process (Gordon *et al.*, 2015). Children encoded the visual displays on Terrapin Logo environment into verbal directions from the turtle's perspective (Clements and Gullo, 1984). They increased their vocabularies by using directional words such as move forward, turn left and turn right. While working on conditionals in Kodable environment, for example, one child described the functioning of the interaction element (Gomes *et al.*, 2018, p.82):

... I just need to put this choice square, and then we put the arrow up or down to say where the fur ball [as the children named the character] should go. Otherwise it'll get hurt and die.

Similarly, a child described robot's behaviour in Iconic Programming Interface by saying: "He all the time wants to walk on the blocks [black squares] and he never wants to walk on the

white...’’ (Levy and Mioduser, 2008, p.346). These findings show that children perceive an icon in programming language both as a creature and as a computational object which can be controlled. Thus, children bring these creatures closer to their world and construct personal meanings regarding the robot’s curious actions (Levy and Mioduser, 2008).

Overview

Coding apps promoted a range of skills and addressed the different developmental areas of children. These differences may stem from the differences in focus and frameworks of the reviewed studies.

4.2. Analysis and Interpretation of Interviews with Parents

Before I start writing this chapter, I would like to note that the interview process did not go as planned. One of the reasons was that one participant’s English level was limited and thus she gave short answers that have affected the amount of data to be collected from this such a small number of participants. Another reason was that both children of participants do not play digital games let alone coding games. Therefore, some of the questions based on playing digital games have remained unanswered. This may be due to the design of my questions or the selection of participants. Trying to reach families with older children could be better. Nevertheless, the interviews were conducted with 2 parents of children. However, a child's mother and father participated in the interview together. Although the main interviewee was the father, the mother attended our conversation sometimes. Their features were presented below.

Table 4. The Features of the Participants and Their Children

Participants Children	Age	Gender
Mother*	1 year, 10 months	Male
Mother and Father	1 year, 11 months	Female

The mother of the boy was marked as Mother*, and the mother and father of the girl were written as Mother and Father during the following quotes. The demographic features of parents were not determined as a variable in this study. Therefore, their age, ethnicity, education or socio-economic levels were not asked.

As for the analysis process, the collected data was transcribed and read repeatedly. This process enabled me to start making sense of the data. Having familiarised with the data, interesting features of the parents' responses were highlighted, and initial codes were generated. Generating the codes, participants' answers and the aim of this research were considered. As the next step, the codes were sorted into themes, and then all answers were collated with relevant themes. These themes were discussed below.

4.2.1. Type of digital devices used by children

The answers of the parents have revealed that their children have access to their iPad, smartphone and laptop. These are some of the popular digital media tools in children's lives today. Likewise, a survey undertaken by The National Literacy Trust (Formby, 2014) with 1,028 children aged 3-5 in 2013 found that 72.9% of the children have access to a touch-screen device in their home.

4.2.2. The features of children's engagement with digital devices

When parents were asked the type of games that their children play. They both answered that their children do not play screen games, they just watch videos on YouTube.

4.2.3. The selection priorities for apps

Parents were asked who choose the apps that their children use. They both said that they choose one channel on YouTube, and then their children skip from one video to another.

The answers of the participants in this study corroborate the findings of Marsh et al. (2015)'s research focusing on very young children just as the current study. Likewise, they found that YouTube was the most popular app downloaded by parents or caregivers of pre-school children in the UK, and was the main app used by children aged 0-2 years on a typical day. Considering what makes YouTube so attractive in the selection of parents and children, some reasons might be that children and their parents can create their favourite lists of age-appropriate videos (Marsh et al., 2015), and YouTube has an easy-to-use interface and short videos, so even children aged 2-3 years can activate the next video from the playlist (Buzzi, 2012).

Parents have also noted that the videos that their children watch have some educational aspects. One parent explained that she prefers some particular singing channels on YouTube because her son has a high level of interest in watching them. Their answers again corroborate the findings of the large-scale study of Marsh *et al.* (2015) where they found that parents' priorities for app searching process were the apps were perceived should be fun, educational, and easy for their child to use.

When they were asked whether they want to encourage their children's use coding tools after the interview, one mother's answer was remarkable.

Mother: We have not heard much about that games before. So, after that, maybe we research more. If they are useful for my daughter, then we will download for her.

This mother's decision as to whether she wants her daughter to play with the coding tools depends on seeing how useful these tools for her daughter. As discussed before in the literature review, parents' willingness to allow their children to access the media and technology depended on how valuable they consider these technologies (Vittrup *et al.*, 2016).

4.2.4. Answering the RQ3: The perspectives of parents towards coding and coding tools

The following themes revealed that parents' knowledge level and attitudes about coding.

4.2.4.1. Awareness of what coding is.

When they were asked to give an idea of what coding means to them, one parent had no idea. It was not surprising for me because I knew coding is a quite new topic even for educators. However, one parent was more knowledgeable than the other. When he was asked to give a definition for coding, he explained it with a few superficial words, by saying:

Father: I heard about coding, but I do not know how to explain it. I just know that it is the way they make like software or things like that.... but I do not know exactly what it is.

Although he did not give a depth definition, his explanation was enough to see he knew something about coding.

4.2.4.2. Too young to learn coding

After receiving their opinions, parents were shown an example coding game interface and game process shortly explained to them. Then, they were asked about their opinions about teaching coding to children and their willingness for their children to play with similar coding tools. Parents emphasised that their children are too small to play with coding games and learn how to code. However, their attitudes towards coding were not negative. They all noted that they want their children to learn how to code but not at this age. Similarly, the age range of the reviewed studies was between 3.7 and 8 years, so older than the participants' children.

One parent, who said she has never heard coding, noted that her son is too young to play with coding games, and defined them as hard:

Mother: I want him to play. But he does not play with them yet. He is really little. He is into stuff that is gonna catch his attention such as singing or video or cartoon. But game is really hard.*

The parent who gave a definition for coding said:

Father: I think coding games will help children to develop their brain and mind. So, it is good for the children.

Then his wife attended to conversation first time and said:

Mother: I think just useful for her if she is over 5-year-olds. Because, now, my daughter, she is too small. So, I think, I want her to play with something not on screen that she will just touch.

Father: Yes. Just normal toys instead of games on tablet.

Interviewer: So, you do not prefer screen games on smartphone or tablet.

Both: Not for now. After

Then, they were asked whether they believe that their child can learn how to code with this kind of tools;

Father: I think, it is a bit hard for her. At least, she should be 7 or 8. So, then she can start learning about coding.

These parents might be right in thinking that their children are too young to learn how to code because there is no consensus on children at what age can begin to learn how to code. As

discussed in 4.1.2, sequencing is the simplest skill that children are required to use in programming environments, however, but various sources discussed whether children can learn sequencing skills at a very young age. An early study has found that kindergarteners (aged 5.4-6.6) can make sequences, but not comprehend the logic or cause and effect regarding the sequenced stories (Brown and French, 1976). Another study has found that children as young as 24 months can understand and imitate some familiar short sequences (involving 2-3 actions) (O'Connell and Gerard, 1985). In recent years, however, research on coding apps has suggested that children as young as age 4 are capable of learning how to programme a robot in a sequencing task correctly (Pila *et al.*, 2019).

On the other hand, Marsh *et al.* (2015)'s study with pre-schoolers found that children's competencies on tablet use were age-related, the abilities of children under the age of 3 were limited to only some of the actions that older children (aged 3-5) can do, such as unlocking the screen and dragging a character or item. They also stated that children between the ages of 3-5 are more likely to use educational and certain types of apps, such as styling, overcoming the obstacles, basic strategy, creating virtual world apps compared to their younger counterparts. However, we cannot know what children are capable of unless we provide children with environments where they can try new skills. Therefore, introducing them to these new environments is important.

4.2.4.3. Coding as brainstorming

Parents were asked in which way this kind of games can be helpful for their children. The father who gave the coding explanation likened children's thinking in coding games to brainstorm by saying:

She has to think how to do it. Like make her brainstorm.

He considered coding games as environments where children required to think and thus likened coding to brainstorm. His analogy can remind computational thinking skills that were widely promoted by digital coding tools in the systematic literature review. As discussed in the 4.1.2. section where computational thinking was defined as a thought process (Bers and Horn, 2010) that children need to use some approaches (Bocconi *et al.*, 2016) as solving the problems. Likewise, brainstorming refers to generate an idea or way of solving a problem, but not as an individual as a group. It is remarkable to see that computational thinking which was found as the most promoted skill by coding tools in the systematic literature review was the first thing that comes to this parent's mind.

Summary

In this chapter, I have attempted to analyse and derived meanings from the findings of two different methods in detailed. The findings and interpretations were discussed and supported by the existing literature. In the next and last chapter, I will draw some broad conclusions based on the discussions of the previous four chapters and bring the thesis to a close.



CHAPTER 5: CONCLUSION

This thesis emerged from my personal interest and the recognised need for more research on digital coding tools designed for young children and how these tools can promote children's learning. Although teaching coding to young children has gained an interest (Gordon *et al.*, 2015) and so a range of digital apps and tools have been designed for children (García-Peñalvo *et al.*, 2016), we know little about how effective they are in children's learning. In line with this awareness, a systematic review of the literature allowed me to discover more information about coding tools and their contribution to children's learning than those would be discovered in a single experimental study involving only 1 or 2 coding tools. There was also a gap in the literature in terms of the parents' perspectives about coding; therefore, interviewing parents enabled me to learn their views and awareness of coding. I have addressed these gaps by conducting mixed methods research. Using systematic literature review and parent interviews together to collect data on such a current topic has provided a wider perspective to see the place of coding in children's both school lives and home lives. This thesis, therefore, has made some important contributions to knowledge.

Summary of Findings and Recommendations for Further Research

This thesis has shown that digital coding tools can promote children's several skills. These skills are placed in Chapter 4 as 4 main skills (basic programming, cognitive, social and language) and 16 sub-skills (sequencing, loops, conditionals, commanding, mathematics, creativity, problem-solving, spatial awareness, metacognitive, reasoning, computational thinking, deciphering visual cues, communication, pretend play, collaboration and verbally describing programming process). However, it is important to note that these results depended on the objectives and frameworks of the reviewed studies.

Preparing a matching table (Table 3) was one of the most important contributions of this research to the literature. This table demonstrated what skills of children were promoted by each coding tool in the reviewed studies and thus enabling readers to see the potential skills of children that can be enhanced through coding tools. Computational thinking skill was promoted by many of the digital coding tools. This finding is significant because computational thinking is widely acknowledged as an important transferable 21st century skill such as how to solve problems or use logic and creativity that can be applied not only to programming process but

also to real life (García-Peñalvo *et al.*, 2016). Therefore, coding tools can be used more to promote computational thinking of children. However, further empirical research is needed to explore the depth relationship between computational thinking and engaging in digital coding tools. Surprisingly, although having fine motor skills and visual-motor integration is essential for children's efficient use of the tablets or smartphones, none of the reviewed research in this study mentioned the development of children's motor skills. Further research examining the motor skills of children as they are engaging in programming environments is needed.

Table 3. might also be a good starting point for demonstrating the value of coding tools to readers, educators, parents and others who else also might be interested. It is important because when parents were asked whether they would like to encourage their children to play with coding apps after the interview, a mother answered “If they are useful for my daughter, then we will download for her”. Similarly, Vittrup *et al.* (2016)'s research showed that when parents believed in the value of tools, toys or apps, they allow their children to have access to them. It is also known that when teachers believed in the value and benefits of technology in children's learning, they were tended to use it more often in their practices (Blackwell *et al.*, 2013). Furthermore, teachers can use such a table (Table 3) as a guide when they want to design technology-assisted activities to enhance children's particular skills. Further research can be considered for such a contribution to the practice of teachers.

Another significant contribution of this study was to reveal how important the ways of introducing coding tools to children are. These ways play an important role in children's potential benefits from coding environments. The reviewed studies have shown that coding tools not only promoted children's programming skills and cognitive skills but also social skills and language skills (Gordon *et al.*, 2015). However, the reviewed studies showed that promoting these skills depended on the tasks given and opportunities provided to children. It has been also found that providing meaningful coding experiences for children require well-designed activities with smooth transitions from concrete thinking to abstract thinking and a combination of in-class and digital coding tools. So, the creativity of adults who provide these tools to children becomes importance. Additionally, the facilitator role of adults in children's engagement with digital coding tools was important.

The design of apps and coding tools had a significant effect on children's understanding of interfaces, interaction elements and what to do in the programming environments. Having good experience and knowledge of coding at an early age can also encourage children to play more similar games and foster their willingness to learn STEM in their later lives. Children's positive early experiences with these coding and programming tools may play a facilitator role in the training of future computer scientists and eventually, may help to meet the demand for skilled STEM workers (Pila *et al.*, 2019) and to make today's children more technology-savvy citizens of the future (Weintrop *et al.*, 2016). Therefore, if we are to develop high-quality learning environments, we should design curricula, tools, and activities by taking into account targeted children's age, interest and development. However, as cutting-edge tools reach ever-younger children, further research is needed to define the characteristics of age-appropriate technology, and expectations for them (Flannery and Bers, 2013).

This study also showed that children of different ages (Flannery and Bers, 2013) and different gender (Papadakis *et al.*, 2016) and children with learning difficulties or disabilities (Taylor *et al.*, 2017) could benefit from coding environments. These findings replicate the claim "All pupils, regardless of their ability, will benefit from coding, largely because they can work at their own pace and level according to their individual needs" (Bocconi *et al.*, 2016, p.38). The gender finding also stresses the importance of including computer science in the learning of pre-schoolers because there is a view that the "gender gap" in computer science may arise later in the teenage years (Martinez *et al.*, 2015). As previously argued in the literature, children who exposed to STEM curriculum in early childhood, including programming languages, demonstrate a less gender-based stereotype related to STEM careers (Steele, 1997) and have less difficulty entering these fields in the future (Markert, 1996). However, to examine this relationship, a longitudinal study is needed that examines the computer science competency of the same group of children with different genders over the years.

The interview with parents showed that the parents did not know much about coding, but they were willing to introduce coding tools to their children if they believed it would benefit. The most important finding of the interviews was that a parent compared coding to brainstorm. This drawn analogy between the coding and brainstorm shows that this parent considers coding as something which encourages children to think and develop strategies to solve problems. In addition to this finding, parents in this study noted that their children (under 2) are too young to play with digital coding games and to learn how to code. To play with a digital coding game,

a parent said that her child should be a minimum of 5 years old, additionally, another one stated that his child should be 7 or 8 years old to learn how to code with digital coding tools. Similarly, in literature, there is no consensus on children at what age can begin to learn how to code. Although, recent research on coding and programming tools has suggested that children as young as age 4 can learn how to programme a robot in a sequencing task correctly (Pila *et al.*, 2019), further research into the abilities of children of different ages on digital coding tools may be beneficial. Additionally, during the review of the literature in this research, no specific study has been found on parents' perspectives or attitudes about coding. Therefore, future research with a larger group of parents, particularly parents of older children, such as a survey can make an important contribution to literature.

Limitations

There were some limitations to this study. Regarding the systematic literature review method, my interpretations were limited to the findings of the reviewed studies. Although I was as neutral as possible to evaluate critically the included studies, there was no other researcher to discuss and evaluate the accuracy of the findings of reviewed studies. As for the interview method, I want to note that I feel personally dissatisfied with the collected data. Due to the problems of finding participants, limited parents were found, and one parent's English level affected the potential amount of data that could be collected. Additionally, the children of participants were quite young (under 2), and not even playing with digital games. Therefore, their parents could not make lots of interpretation of coding and children's engagement with coding tools. Interviewing with the parents of children 4 years and older could be better in terms of providing more data and more perspectives.

Summary

This thesis has identified some gaps in the existing literature regarding digital coding tools designed for young children, the skills promoted by these tools and the parents' perspectives about these tools. Drawing on the work of the previous four chapters, the final chapter brings this study to a close by emphasising some of its significant contributions to the literature, making recommendations for further research and mentioning its limitations.

BIBLIOGRAPHY

- Ackermann, E. (2001). Piaget's Constructivism, Papert's Constructionism: What's the difference? *Constructivism: Uses and Perspectives in Education*, 1, 1–11.
- Baddeley, A. (1996). Exploring the central executive. *The Quarterly Journal of Experimental Psychology*, 49A(1), 5–28.
- Balanskat, A., & Engelhardt, K. (2014). Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe. In *European Schoolnet (EUN Partnership AISBL)*.
- Bateman, A., & Church, A. (2017). Children's use of objects in an early years playground. *European Early Childhood Education Research Journal*, 25(1), 55–71.
- Bell, J. (2005). *Doing your Research Project* (Fourth). Berkshire: Open University Press.
- Bers, M U, & Horn, M. S. (2010). Tangible Programming in Early Childhood: Revisiting Developmental Assumptions Through New Technologies: Childhood in a Digital World. In & I. R. B. M. J. Berson (Ed.), *High-Tech Tots: Childhood in a Digital World* (pp. 49–70). Greenwich: Information Age Publishing.
- Bers, Marina Umaschi. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York: Teachers College Press.
- Bers, Marina Umaschi. (2017). The Seymour test : Powerful ideas in early childhood education. *International Journal of Child-Computer Interaction*, 14, 10–14.
- Bers, Marina Umaschi. (2018a). Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe. *European Journal of STEM Education*, 3(3), 1–13.
- Bers, Marina Umaschi. (2018b). Coding as a literacy for the 21st Century. Retrieved 16 June 2019, from https://blogs.edweek.org/edweek/education_futures/2018/01/coding_as_a_literacy_for_the_21st_century.html
- Bers, Marina Umaschi, Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering : Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145–157.
- Bettany-Saltikov, J. (2012). *How to do a systematic literature review in nursing: A step-by-step guide*. New York: Open University Press.
- Blackwell, C. K., Lauricella, A. R., Wartella, E., Robb, M., & Schomburg, R. (2013). Adoption and use of technology in early education: The interplay of extrinsic barriers and teacher attitudes. *Computers & Education*, 69, 310–319.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). *Developing Computational Thinking in Compulsory Education: Implications*

for policy and practice. European Union.

- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101.
- Brown, A. L., & French, L. A. (1976). Construction and Regeneration of Logical Sequences Using Causes or Consequences as the Point of Departure. *Child Development*, 47(4), 930–940.
- Brown, C. P. (2015). Conforming to reform: Teaching pre-kindergarten in a neoliberal early education system. *Journal of Early Childhood Research*, 13(3), 236–251.
- Brush, T., Glazewski, K., Rutowski, K., Berg, K., Van-nest, M. H., Stock, L., ... Van-nest, M. H. (2003). Integrating Technology in a Field-Based Teacher Training Program : The PT3 @ ASU Project. *Educational Technology Research & Development*, 51(1), 57–72.
- Bryce, D., & Whitebread, D. (2012). The development of metacognitive skills: Evidence from observational analysis of young children's behavior during problem-solving. *Metacognition and Learning*, 7, 197–217.
- Buzzi, M. (2012). What Are Your Children Watching on YouTube? In K. F. (eds) Cipolla-Ficarra F., Veltman K., Verber D., Cipolla-Ficarra M. (Ed.), *Advances in New Technologies, Interactive Interfaces and Communicability*. (pp. 243–252). Berlin: Springer.
- Carter, S. M., & Little, M. (2007). Justifying knowledge, justifying method, taking action: Epistemologies, methodologies, and methods in qualitative research. *Qualitative Health Research*, 17(10), 1316–1328.
- Chau, C. L. (2015). *Positive Technological Development for Young Children in the Context of Children's Mobile Apps*. Tufts University, USA.
- Chaudron, S. (2014). *Young Children (0-8) and Digital Technology*. Joint Research Centre.
- Clements, D. H. (1999). The Future of Educational Computing Research: The Case of Computer Programming. *Information Technology in Childhood Education*, 1999(1), 147–179.
- Clements, D. H., Battista, M. T., & Sarama, J. (2001). Logo and Geometry. *Journal of Research in Mathematics Education Monograph Series*, 10.
- Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051–1058.
- Clements, D. H., & Nastasi, B. K. (1999). Metacognition , Learning , and Educational Computer Environments. *Information Technology in Childhood Education*, 1, 3–36.
- Coravu, L., Marian, M., & Ganea, E. (2015). Scratch and recreational coding for kids. In *14th RoEduNet International Conference - Networking in Education and Research, RoEduNet NER 2015 - Proceedings* (Vol. 217, pp. 85–89). Craiova, Romania.
- Cordes, C., & Miller, E. (2000). *Fool's Gold: A Critical Look at Computers in Childhood*.

Alliance for Childhood.

- Corsaro, W. A. (1979). 'We're friends, right?': Children's use of access rituals in a nursery school. *Language in Society*, 8(3), 315–336.
- Corsaro, W. A. (2015). *The Sociology of Childhood* (4th ed.). Los Angeles: Pine Forge Press.
- Crow, D. (2014). Why every child should learn to code. *The Guardian*. Retrieved from <https://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick>
- Cunha, F., & Heckman, J. (2007). The technology of skill formation. *American Economic Review*, 92(2), 31–47.
- Dankert, H. L., Davies, P. L., & Gavin, W. J. (2003). Occupational therapy effects on visual-motor skills in preschool children. *American Journal of Occupational Therapy*, 57, 542–549.
- Denscombe, M. (2017). *The Good Research Guide For Small Scale Research Projects* (6th ed.). London: Open University Press.
- Department for Education. (2013). Computing Programmes of Study: Key Stages 1 and 2. Retrieved 7 June 2019, from https://www.computingschool.org.uk/data/uploads/primary_national_curriculum_-_computing.pdf/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf
- DiSessa, A. A., & Abelson, H. (1986). Boxer: A reconstructible computational medium. *Communications of the ACM*, 29(9), 859–868.
- Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? In *WiPSCE '14, November 05 - 07* (pp. 60–69). Berlin, Germany.
- Dýrfjörð, K., & Magnúsdóttir, B. R. (2016). Privatization of early childhood education in Iceland. *Research in Comparative and International Education*, 11(1), 80–97.
- Edwards, S., & Bird, J. (2017). Observing and assessing young children's digital play in the early years: Using the Digital Play Framework. *Journal of Early Childhood Research*, 15(2), 158–173.
- Eisenberg, M., Hsi, S., & Oh, H. (2017). Machines and minds: The new cognitive science, and the potential evolution of children's intuitions about thinking. *International Journal of Child-Computer Interaction*, 14, 1–4.
- EU Digital Single Market. (2018). Coding - the 21st century skill. Retrieved 9 June 2019, from <https://ec.europa.eu/digital-single-market/en/coding-21st-century-skill>
- European Schoolnet. (2016). *The e-Skills Manifesto*. (G. S. C. Bluck, E. and Carter, A., Ed.). Brussels: European Schoolnet.
- Farrell, A., Kagan, S., Tisdall, E., & De Graeve, K. (2015). *Theorizing Identities in Early Childhood*. London: SAGE Publications.

- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education*, 63, 87–97.
- Fisch, S. M. (2004). *Children's learning from educational television: Sesame Street and beyond. Children's Learning from Educational Television: Sesame Street and Beyond*. New York: Routledge.
- Flannery, L. P., & Bers, M. U. (2013). Let's dance the 'Robot Hokey-Pokey!': Children's programming approaches and achievement throughout early cognitive development. *Journal of Research on Technology in Education*, 46(1), 81–101.
- Flick, U. (2014). *An Introduction To Qualitative Research*. London: SAGE.
- Formby, S. (2014). *Parents' perspectives: Children's use of technology in the Early Years*. National Literacy Trust. London.
- García-Peñalvo, F. J., Reimann, D., Tuul, M., Rees, A., & Jormanainen, I. (2016). An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers. In *TACCLE3 Consortium*. Belgium.
- García-Peñalvo, F. J., Rees, A. M., Hughes, J., Jormanainen, I., Toivonen, T., & Vermeersch, J. (2016). A survey of resources for introducing coding into schools (pp. 19–26). Salamanca, Spain.
- Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. de A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction*, 16, 77–84.
- Gordon, M., Ackermann, E., & Breazeal, C. (2015). Social Robot Toolkit: Tangible Programming for Young Children. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (pp. 67–68). Oregon, USA.
- Gray, D. E. (2014). *Doing Research in the Real World* (3rd ed.). London: SAGE.
- Greenbank, P. (2003). The Role of Values in Educational Research: The case for reflexivity. *British Educational Research Journal*, 29(6), 791–801.
- Greene, J. C., Carcelli, V. J., & Graham, W. F. (1989). Toward a Conceptual Framework for Mixed-Method Evaluation Designs. *Educational Evaluation and Policy Analysis*, 11(3), 255–274.
- HITSA. (2015). *ProgeTiger Programme 2015–2017*. Retrieved from <http://www.hitsa.ee/it-education/educational-programmes/progetiger>
- Holloway, D., Green, L. and Livingstone, S. (2013). *Zero to eight. Young children and their internet use*. LSE, London: EU Kids Online.
- Ihmeideh, F., & Alkhawaldeh, M. (2017). Teachers' and parents' perceptions of the role of technology and digital media in developing child culture in the early years. *Children and Youth Services Review*, 77, 139–146.

- Jung, S. E., & Won, E. S. (2018). Systematic review of research trends in robotics education for young children. *Sustainability*, 10(4), 1–24.
- Kachouie, R., Sedighadeli, S., Khosla, R., & Chu, M. T. (2014). Socially Assistive Robots in Elderly Care: A Mixed-Method Systematic Literature Review. *International Journal of Human-Computer Interaction*, 30, 369–393.
- Kanbul, S., & Uzunboyulu, H. (2017). Importance of coding education and robotic applications for achieving 21st-century skills in north cyprus. *International Journal of Emerging Technologies in Learning*, 12(1), 130–140.
- Karadeniz, Ş., Samur, Y., & Özden, M. Y. (2014). Playing with Algorithms to Learn Programming : A Case Study on 5 Years Old Children. In *Proceedings of the International Conference on Information Technology and Applications*.
- Kazakoff, E. R. (2015). Technology-based Literacies for Young Children: Digital Literacy through Learning to Code. In K. L. Heider & M. R. Jalongo (Eds.), *Young Children and Families in the Information Age: Applications of Technology in Early Childhood* (pp. 1–292). Springer.
- Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, 41, 245–255.
- Kölling, M. (2016). Lessons from the Design of Three Educational Programming Environments: Blue, BlueJ and Greenfoot. *International Journal of People-Oriented Programming*, 4(1), 5–32.
- Kress, G. (2003). *Literacy in the new media age. Literacy in the New Media Age*. Abingdon: Routledge.
- Lee, K. T. H., Sullivan, A., & Bers, M. U. (2013). Collaboration by Design : Using Robotics to Foster Social Interaction in Kindergarten, 30(3), 271–281.
- Levy, S. T., & Mioduser, D. (2008). Does it ‘want’ or ‘was it programmed to...’? Kindergarten children’s explanations of an autonomous robot’s adaptive functioning. *International Journal of Technology and Design Education*, 18(4), 337–359.
- Liao, Y.-K. C., & Bright, G. W. (2503). Effects of Computer Programming on Cognitive Outcomes: A Meta-Analysis. *Journal of Educational Computing Research*, 7(3), 251–268.
- 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. Retrieved from <http://dx.doi.org/10.1016/j.chb.2014.09.012>
- Magnenat, S., Shin, J., Riedo, F., Siegwart, R., & Ben-Ari, M. (2014). Teaching a core CS concept through robotics. In *Conference on Innovation and Technology in Computer Science Education* (pp. 315–320). New York.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*,

10(4), 1–15.

- Maloney, J., Resnick, M., Rusk, N., Peppler, K. A., & Kafai, Y. B. (2008). Media Designs with Scratch: What Urban Youth Can Learn About Programming in a Computer Clubhouse. In *8th International Conference on International Conference for the Learning Sciences* (Vol. 3, pp. 81–82). Utrecht, Netherlands.
- Markert, L. R. (1996). Gender Related to Success in Science and Technology. *The Journal of Technology Studies*, 22(2), 21–29.
- Marsh, J., Plowman, L., Yamada-Rice, D., Bishop, J.C., Lahmar, J., Scott, F., Davenport, A., Davis, S., French, K., Piras, M., Thornhill, S. (2015). *Exploring Play and Creativity in Pre- Schoolers' Use of Apps: Final Project Report. Technology and Play*. Retrieved from http://www.techandplay.org/reports/TAP_Final_Report.pdf
- Marsh, J. (2010). Young children's play in online virtual worlds. *Journal of Early Childhood Research*, 8(1), 23–39.
- Marsh, J., Plowman, L., Yamada-rice, D., Bishop, J., Lahmar, J., & Scott, F. (2018). Play and creativity in young children ' s use of apps. *British Journal of Educational Technology*, 49(5), 870–882.
- Martinez, C., Gomez, M. J., & Benotti, L. (2015). A Comparison of Preschool and Elementary School Children Learning Computer Science Concepts through a Multilanguage Robot Programming Platform. In *Innovation and Technology in Computer Science Education* (pp. 159–164).
- McLennan, D. P. (2017). *What is coding ? What do children need to know ? Why introduce coding to preschoolers ? Teaching Young Children*. Retrieved from <https://www.oecd.org/site/schoolingfortomorrowknowledgebase/futuresthinking/trends/41286179.pdf>
- Mcmenamin, K., & Linebarger, D. L. (2010). *Summative Evaluation of PBS Kids Island: Usability, Outcomes, and Appeal*.
- Merriam, S. B., & Tisdell, E. J. (2016). *Qualitative research: A Guide to Design and Implementation* (Fourth). San Francisco: Jossey Bass.
- Merrill, S. (2017). The Future of Coding in Schools. Retrieved 18 June 2019, from <https://www.edutopia.org/article/future-coding-schools>
- Messer, D., Thomas, L., Holliman, A., & Kucirkova, N. (2018). Evaluating the effectiveness of an educational programming intervention on children's mathematics skills, spatial awareness and working memory. *Education and Information Technologies*, 23, 2879–2888.
- Mikelic Preradović, N., Lešin, G., & Šagud, M. (2016). Investigating parents' attitudes towards digital technology use in early childhood: A case study from Croatia. *Informatics in Education*, 15(1), 127–146.
- Miller, R. (2001). 21st Century Transitions : Opportunities , Risks.

- Mioduser, D., Levy, S. T., & Talis, V. (2009). Episodes to scripts to rules: Concrete- abstractions in kindergarten children's explanations of a robot's behavior. *International Journal of Technology and Design Education*, 19, 15–36.
- Moon, K., & Blackman, D. (2014). A Guide to Understanding Social Science Research for Natural Scientists. *Conservation Biology*, 28(5), 1167–1177.
- More, C. M., & Travers, J. C. (2013). What's App With That? Selecting Educational Apps for Young Children With Disabilities. *Young Exceptional Children*, 16(2), 15–32.
- Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting Plans to Action: The Effects of a Card-Coded Robotics Curriculum and Activities on Korean Kindergartners. *The Asia-Pacific Education Researcher*. <https://doi.org/10.1007/s40299-019-00438-4>
- National Association for the Education of Young Children. (2009). *Key messages of the position statement*. Retrieved from <https://www.naeyc.org/files/naeyc/file/positions/KeyMessages.pdf?>
- Neumann, M. M. (2014). An examination of touch screen tablets and emergent literacy in Australian pre-school children. *Australian Journal of Education*, 58(2), 109–122.
- Neumann, M. M., & Neumann, D. L. (2017). The use of touch-screen tablets at home and pre-school to foster emergent literacy. *Journal of Early Childhood Literacy*, 17(2), 203–220.
- Newhouse, C. P., Cooper, M., & Cordery, Z. (2017). Programmable toys and free play in early childhood classrooms. *Australian Educational Computing*, 32(1).
- Nguyen, M. (2016). Turning Ghana Upside Down with Coding. Retrieved 9 June 2019, from <https://medium.com/scratchteam-blog/turning-ghana-upside-down-with-coding-3811c4777e4b>
- Nikken, P., & Jansz, J. (2014). Developing scales to measure parental mediation of young children's internet use. *Learning, Media and Technology*, 39(2), 250–266.
- O'Brien, A. M., & Mc Guckin, C. (2015). *The Systematic Literature Review Method: Trials and Tribulations of Electronic Database Searching at Doctoral Level*. London: SAGE.
- O'Connell, B. G., & Gerard, A. B. (1985). Scripts and Scraps : The Development of Sequential Understanding. *Child Development*, 56(3), 671–681.
- Ofcom. (2018). *Adults' media use and attitudes report*. Retrieved from https://www.ofcom.org.uk/__data/assets/pdf_file/0011/113222/Adults-Media-Use-and-Attitudes-Report-2018.pdf
- Oliver, M. (2011). Technological determinism in educational technology research: Some alternative ways of thinking about the relationship between learning and technology. *Journal of Computer Assisted Learning*, 27, 373–384.
- Onwuegbuzie, A., & Leech, N. (2005). On becoming a pragmatic researcher: The importance of combining quantitative and qualitative research methodologies. *International Journal of Social Research Methodology*, 8(5), 375–387.

- Opie, C. (2004). *Doing Educational Research: A Guide to First-Time Researchers*. London: SAGE.
- Palaiologou, I. (2016). Children under five and digital technologies: implications for early years pedagogy. *European Early Childhood Education Research Journal*, 24(1), 5–24.
- Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. *International Journal of Mobile Learning and Organisation*, 10(3), 187.
- Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2018). How do you feel about learning to code ? Investigating the effect of children ' s attitudes towards coding using eye-tracking. *International Journal of Child-Computer Interaction*, 17, 50–60.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Parette, H. P., Quesenberry, A. C., & Blum, C. (2010). Missing the boat with technology usage in early childhood settings: A 21st century view of developmentally appropriate practice. *Early Childhood Education Journal*, 37, 335–343.
- Petticrew, M., & Roberts, H. (2006). *Systematic Reviews in the Social Sciences: A Practical Guide*. *Systematic Reviews in the Social Sciences: A Practical Guide*. Oxford: Blackwell.
- Piaget, J. (1997). *The Principles of Genetic Epistemology*. London: Routledge.
- Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications : An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. *Computers & Education*, 128, 52–62.
- Plowman, L., McPake, J., & Stephen, C. (2010). The technologisation of childhood? Young children and technology in the home. *Children & Society*, 24, 63–74.
- Prensky, M. (2001). Digital Natives, Digital Immigrants Part 1. *On the Horizon*, 9(5), 1–6.
- Raghubar, K. P., Barnes, M. A., & Hecht, S. A. (2010). Working memory and mathematics: A review of developmental, individual difference, and cognitive approaches. *Learning and Individual Differences*, 20, 110–122.
- Resnick, M. (2006). Computer as Paintbrush: Technology, Play, and the Creative Society. In D. Singer, R. M. Golikoff, & K. Hirsh-Pasek (Eds.), *Play = Learning: How Play Motivates and Enhances Children's Cognitive and Social-Emotional Growth* (pp. 192–206). Oxford: Oxford University Press.
- Resnick, M. (2015). Coding as the New Literacy. Retrieved 18 June 2019, from <http://serious-science.org/coding-as-new-literacy-1546>
- Resnick, M., Maloney, H., Andrés, M.-H., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67.

- Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., & Silverman, B. (1998). Digital manipulatives: new toys to think with. In *Conference on Human Factors in Computing Systems: Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 281–287).
- Robson, S. (2014). The Analysing Children’s creative thinking framework: Development of an observation-led approach to identifying and Analysing young children’s creative thinking. *British Educational Research Journal*, 40(1), 121–134.
- Rothwell, J. (2014). Short on STEM talent. Retrieved from <https://www.usnews.com/opinion/articles/2014/09/15/the-stem-worker-shortage-is-real>
- Rushkoff, D. (2010). *Program or Be Programmed: Ten Commands for a Digital Age*. USA: BookMobile.
- Scott, F. L. (2018). *Young children’s engagement with television and related media in the digital age*. PhD Thesis, The University of Sheffield.
- Shilling, R. D. (2016). *New website highlights progress in early STEM education*. Retrieved from <https://innovation.ed.gov/2016/09/15/new-website-highlights-progress-early-stem-education/>
- Smith, M. (2016). Computer science for all. *The White House Blog*. <https://doi.org/10.1145/2524713.2524716>
- Sonu, D., & Benson, J. (2016). The quasi-human child: How normative conceptions of childhood enabled neoliberal school reform in the United States. *Curriculum Inquiry*, 46(3), 230–247.
- Steele, C. M. (1997). A Threat in the Air. *American Psychologist*, 52(6), 613–629.
- Strawhacker, A., & Bers, M. U. (2015). “I want my robot to look for food”: Comparing Kindergartner’s programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education*, 25, 293–319.
- Strawhacker, A., & Bers, M. U. (2019). *What they learn when they learn coding: investigating cognitive domains and computer programming knowledge in young children*. *Educational Technology Research and Development* (Vol. 67). Springer US.
- Sullivan, A., & Bers, M. U. (2013). Gender differences in kindergarteners’ robotics and programming achievement. *International Journal of Technology and Design Education*, 23, 691–702.
- Svensson, A.-K. (2000). Computers in School: Socially Isolating or a Tool to Promote Collaboration? *Journal of Educational Computing Research*, 22(4), 437–453.
- Tacconelli, E. (2010). Systematic reviews: CRD’s guidance for undertaking reviews in health care. *The Lancet Infectious Diseases*, 10(4), 226–294.
- Taylor, M., Harlow, A., & Forret, M. (2010). Using a Computer Programming Environment and an Interactive Whiteboard to Investigate Some Mathematical Thinking. In *Procedia - Social and Behavioral Sciences Social and Behavioral Sciences* (Vol. 8, pp. 561–570).

- Taylor, M. S., Vasquez, E., & Donehower, C. (2017). Computer Programming with Early Elementary Students with Down Syndrome. *Journal of Special Education Technology*, 32(3), 149–159.
- The European Coding Initiative. (2014). Retrieved from http://www.allyouneediscode.eu/documents/12411/14644/EU+Coding+Initiatives_Web.pdf/7c7c01f2-9e5a-498e-a0fb-aab128eec8d3
- The Logo Foundation. (2015). Logo History. Retrieved 17 May 2019, from https://el.media.mit.edu/logo-foundation/what_is_logo/history.html
- Turkle, S., & Papert, S. (1990). Epistemological Pluralism : Styles and Voices within the Computer Culture. *Signs*, 16(1), 128–157.
- UNESCO Institute for Statistics. (2016). *Literacy Rates Continue to Rise from One Generation to the Next*.
- Vittrup, B., Snider, S., Rose, K. K., & Rippy, J. (2016). Parental perceptions of the role of media and technology in their young children’s lives. *Journal of Early Childhood Research*, 14(1), 43–54.
- Vygotsky, L. (1986). *Thought and Language*. Cambridge: The MIT Press.
- 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, 127–147.
- Wellington, J. (2015). *Educational research: Contemporary issues and practical approaches*. London: Bloomsbury.
- Wenger, E. (2009). Communities of Practice: A brief introduction (pp. 1–7). Retrieved from <http://f-a-c-e.org.uk/conference-and-events/face2010/grainger.pdf>
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35.
- Wohlwend, K. E. (2015). One Screen, Many Fingers: Young Children’s Collaborative Literacy Play With Digital Puppetry Apps and Touchscreen Technologies. *Theory Into Practice*, 54(2), 154–162.
- Yurt, Ö., & Cevher-Kalburan, N. (2011). Early childhood teachers’ thoughts and practices about the use of computers in early childhood education. *Procedia Computer Science*, 3, 1562–1570.
- Zaman, B., Nouwen, M., Vanattenhoven, J., de Ferrerre, E., & Looy, J. Van. (2016). A Qualitative Inquiry into the Contextualized Parental Mediation Practices of Young Children’s Digital Media Use at Home. *Journal of Broadcasting and Electronic Media*, 60(1), 1–22.

APPENDICES

Appendix 1. Example participant information document



Research Project Information Document

For MA student research

Exploring Coding Tools and Their Contribution To

Children's Learning In Early Childhood Education

A Systematic Literature Review With Parental Perspective

Name and email of student responsible for the research: Miss Aslan maslan1@shef.ac.uk
University tutor supervising the work (name and email): Fiona Scott f.scott@sheffield.ac.uk

A student at the University of Sheffield, School of Education is required to complete research-based coursework as part of her programme of study. Approval of the project identified above has been obtained from the university tutor who is supervising the work.

1. **What is the project's purpose?**

The purpose of this project is to explore coding tools and their contribution to children's learning. Coding tools include several types of tangible toys and digital games and apps that require children's participation to direct a real item on the floor or solve a problem in digital games by using arrows or typing commands. In addition, the perspectives of parents about coding will be investigated. I am undertaking this research project for my studies on the MA: Early Childhood Education at the University of Sheffield.

2. **Why have these participants been chosen?**

These participants have been chosen for the following reason:

a) You are a parent of a child between 1,5 and 6-year-old.

3. **Do the participants have to take part?**

Participation in the research project is entirely voluntary and there will be no negative consequences if you decide not to take part. Participants can withdraw consent for their information to be used up to one week after the interview. At this time, the data will be completely anonymised and collated into a large dataset and your contribution will no longer be identifiable.

4. **What will happen to me if I take part? What do I have to do?**

*This research project consists of one activity, **you may be asked to participate in the following:***

Interview: *You may be requested to be interviewed to discuss experiences, attitudes and beliefs about coding tools to use children's learning. This is intended to take no longer than 30 minutes of your time. **Your interview will be audio-recorded,** and you may request further details of the questions to be asked in advance. You will be expected to answer questions honestly, but you may refrain*

Research Project Information Document

from answering any or all questions as you see fit. You may also ask the interviewer questions if desired. Excerpts from your interview may be quoted (anonymously).

5. **What are the possible disadvantages and risks of taking part?**

It is not expected that there will be any risks or disadvantages of taking part. You may however be asked to discuss your views, observation and attitudes towards aspects of coding tools. At no point is this research intending to be judgemental or infer any preference in parental or individual attitudes; it is hoped that participation will not lead to any offense, feelings of insecurity or questioning of professionalism of a particular individual's or institution's practice.

6. **What are the possible benefits of taking part?**

Whilst there is may be no immediate benefits for those people participating in the project, it is hoped that this project will help inform my future professional life as well as the work of other researchers in this field.

It is hoped that participants in interview for this study may find the content of interest.

7. **What happens if the research study stops earlier than expected?**

If the research study stops earlier than expected you will be informed of when and why this has occurred. You will still have the option to withdraw consent for your data to be used.

8. **What if something goes wrong?**

If you wish to make a complaint or withdraw your consent for participation in the study you can contact the researcher (Miss Aslan, maslan1@shef.ac.uk) in the first instance.

If you have any further concerns arising, such as your treatment by the principal researcher or something serious occurring during or following your participation in the project, please see the contact details at the end of this document.

9. **Will my taking part in this project be kept confidential?**

All the information that we collect about you during the course of the research will be kept strictly confidential except in cases where a disclosure is made about criminal activity or risks to public safety, in which case the researcher is legally obliged to report such information to the relevant authorities. Participants will not be able to be identified in any reports or publications.

10. **What is the legal basis for processing my personal data?**

This data will be stored and processed in the United Kingdom by the researcher conducting this study. Therefore, according to data protection legislation, we are required to inform you that the legal basis we are applying in order to process



Research Project Information Document

your personal data is that 'processing is necessary for the performance of a task carried out in the public interest' (Article 6(1)(e)). Further information can be found in the University's Privacy Notice <https://www.sheffield.ac.uk/govern/data-protection/privacy/general>. No data will be transferred outside of the EU.

11. **What will happen to the results of the research project?**

These results will form part of my research project and some of my findings may be published in academic journals.

12. **Who has ethically reviewed the project?**

This project has been ethically approved by a panel of reviewers at the University of Sheffield School of Education. The University's Research Ethics Committee monitors the application and delivery of the University's Ethics Review Procedure across the University.

13. **Will I be audio or video-recorded, and how will the recorded media be used?**

The only audio-recordings conducted will be of interview, **you will be informed if you are to be audio/video-recorded.**

Any recordings of your activities made during this research will be used only for analysis and for illustration in my project. All contributions will be anonymous. No other use will be made of them without your written permission, and no one outside the project will be allowed access to the original recordings.

14. **Who is the Data Controller?**

The University of Sheffield will act as the data controller for this study. This means that The University is responsible for looking after your information and using it properly.

If there is concern about any aspect of this research project it should be addressed in the first instance to me via my email address maslan1@sheffield.ac.uk and to my supervisor Dr. Fiona Scott via f.scott@sheffield.ac.uk.

In addition to these avenues the University also has a complaints procedure, details of which may be found at: www.shef.ac.uk/ssid/procedures/grid.html#complaints

Thank you for considering participating in this study.

Appendix 2. Example consent form



Adult-participant consent form

Exploring Coding Tools and Their Contribution To Children's Learning In Early Childhood Education

A Systematic Literature Review With Parental Perspective

You are being invited to take part in a research project. Before you decide whether or not to participate, it is important for you to understand why the research is being done and what it will involve.

Please take time to read the information provided carefully and feel free to speak to the researcher or contact them at maslan1@sheffield.ac.uk if you would like any more information. Take time to decide whether or not you are happy to take part. **Thank you for reading this.**

Taking Part in the Project

Please tick the appropriate boxes ☒

	Yes	No
I have read and understood the project information sheet dated . . . / . . . / . . . or the project has been fully explained to me. (If you will answer No to this question please do not proceed with this consent form until you are fully aware of what your participation in the project will mean.)	☐	☐
I have been given the opportunity to ask questions about the project.	☐	☐
I agree to take part in being interviewed.	☐	☐
I agree to take part in being recorded (audio and / or video).	☐	☐
I understand that my taking part is voluntary and that I can withdraw from the study before 29.08.2019; I do not have to give any reasons for why I no longer want to take part and there will be no adverse consequences if I choose to withdraw.	☐	☐

How my information will be used during and after the project

I understand my personal details such as name, phone number, address and email address etc. will not be revealed to people outside the project.	☐	☐
I understand and agree that my words may be quoted in publications, reports, web pages, and other research outputs. I understand that I will not be named in these outputs unless I specifically request this.	☐	☐
I understand and agree that other authorised persons (e.g. supervising tutors) will only have access to this data under the conditions of confidentiality specified in the information documents.	☐	☐
I understand and agree that other authorised researchers may use my data in publications, reports, web pages, and other research outputs, only if they agree to preserve the confidentiality of the information as requested in this form.	☐	☐
I understand that raw data from the project will be destroyed on completion of the research study, no later than study before 29.08.2019;	☐	☐

So that the information you provide can be used legally by the researchers

I agree to assign the copyright I hold in any materials generated as part of this project to The University of Sheffield.	☐	☐
---	--------------------------	--------------------------

Name of participant [printed]

Signature

Date

Melike Aslan [printed]

Signature

Date

Project contact details for further information:

Researcher: Melike Aslan

E-mail: Maslan1@sheffield.ac.uk

Address: 241 Glossop Rd, Sheffield S10 2GW

Supervisor of This Research Project: Fiona Scott

Tel: +44 (0)114 222 8159

Email: f.scott@sheffield.ac.uk

Appendix 3. Ethics approval letter



Downloaded: 29/05/2019
Approved: 29/05/2019

Melike Aslan
Registration number: 180132498
School of Education

Dear Melike

PROJECT TITLE: Exploring Coding Tools and Their Contribution To Children's Learning In Early Childhood Education: A Systematic Literature Review With Parental Perspective

APPLICATION: Reference Number 027210

ORIGINAL APPLICATION: Reference Number 022770

On behalf of the University ethics reviewers who reviewed your project, I am pleased to inform you that on 29/05/2019 the above-named project was **approved** on ethics grounds, on the basis that you will adhere to the following documentation that you submitted for ethics review:

- University research ethics application form 027210 (dated 28/05/2019). This is an en bloc application based on University research ethics application form 022770
- Participant information sheet 1062781 version 1 (28/05/2019).
- Participant consent form 1062782 version 1 (28/05/2019).

If during the course of the project you need to [deviate significantly from the above-approved documentation](#) please inform me since written approval will be required.

Yours sincerely

David Hyatt
Ethics Administrator
School of Education

Appendix 4. Interview Questions

Introduction

Hello. First of all, thank you very much for your participation. Today, I would like to talk to you about the latest popular digital games. I am very curious about your ideas. So, I would like to ask you a few questions. Please remember that you are completely free not to answer any question.

Questions

1. Does your child use tablet or similar digital devices? (If so, can you give me more information about the frequency of use? Such as how often? how long in a day?)
2. What kind of games does she/he like to play? Does she/he have any favourite?
3. How did she/he find it? Do you remember?
4. Who choose the apps she/he plays?
(If answer is parent: How do you decide which application to download? How do you judge whether it is a good app for your child? I mean... Are you looking for apps by looking at how playful, attractive or educational they are?)

(If answer is child: Do you have any idea what her/his preference is to choose apps? Is she/he looking for apps by looking at how playful, attractive or educational they are?)
5. Well. What do you think about educational apps that aim to teach numbers, shapes or new words to children? Are you keen on downloading such educational tools for your child?
6. You probably heard coding or programming as something special for computer science or computer engineers. I wondered if you could give me an idea of what coding means to you? How would you define 'coding' in terms?

7. Have you ever heard of coding or programming tools for children?
(If answer is no <Explain more: Coding tools include both digital games on iPads or computers and real objects such as robot or toy animal. Through coding apps or tools children enable to use arrows to move an item to reach a specific place.>)
8. Does your child play any games related to coding or programming? or Have you ever observed your child playing with similar games?
9. Lately, there have been many attempts to teach coding and programming to children. What do you think about that? Would you like your child to learn coding? If so, why?
10. What do you think about using coding and programming tools (real toys or digital games/apps) for the education of children? How can the coding/programming tools be useful/useless for your child's development?
11. Do you believe that your child can learn how to code or programme through these tools at a young age?
(If answer is no: Why do you think like that?)
(If answer is yes: Do you do anything to support your child for it?)
12. Would you like to encourage your child to use coding tools after this interview?