Waikato Journal of Education

ISSN 2382-0373
Website: http://wje.org.nz

Wilf Malcolm Institute of Educational Research
Te Pūtahi Rangahau Mātauranga o Wilf Malcolm
THE UNIVERSITY OF WAIKATO

Title of Issue/section: Volume 23, Issue 2, 2018

Guest Editor: Dianne Forbes

Editor: Noeline Wright

To cite this article: Calder, N. (2018). Using Scratch to facilitate mathematical thinking'. *Waikato Journal of Education, 23*(2), 43-58. doi: 10.15663/wje.v23i2.654.

To link to this volume: 10.15663/wje.v23i2

Waikato Journal of Education
Te Hautaka Mātauranga o Waikato

Volume 23, Issue 2: 2018

Using Scratch to facilitate mathematical thinking

Nigel Calder
University of Waikato

Abstract

This article reports on a research project that examined the ways that 10-year-old students, who were using Scratch for coding, engaged with mathematical ideas. Interactive software is emerging that has cross-curricula implications and facilitates thinking in rich, problem-solving environments. Scratch, a free-to-use graphical programming environment provides opportunities for creative problem solving. When students process mathematics through digital technologies, the digital pedagogical media influences the learning process and students' understanding emerges in distinctive ways. The children used Scratch to create mathematical digital learning objects, including games. An interpretive approach was undertaken, with the data collected over a two-week research period. The students wrote daily blogs articulating their progress and reflections. Students and the teacher were interviewed, and classroom observations (both written and photographic) were recorded. The findings suggest that mathematical thinking, including geometry and problem-solving processes, was facilitated through this process. Together, these findings indicate that teachers should not only use Scratch in primary school classroom programmes to develop coding skills but also be aware of its potential to facilitate thinking in other related areas.

Keywords

Mathematics education; digital technologies; coding; primary school

Acknowledgements

I would like to acknowledge Wilf Malcolm Institute of Educational Research for their support of the project and Dr Mike Forret, Dr Kathrin Otrel-Cass, Merilyn Taylor and Sheena Saunders, who were part of the *Scratch* research team.

Introduction

This article reports on one aspect of a research project that examined the ways that students with laptops used *Scratch*, a graphical program for coding activities. The research project involved a case study of one class of 10-year-old children who used the software program to create mathematical digital learning objects, including games. In particular, it considers how this activity facilitated the

Wilf Malcolm Institute of Educational Research, Te Kura Toi Tangata Faculty of Education, University of Waikato, Hamilton, New Zealand
ISSN: 2382-0373
Contact details: Nigel Calder nigel.calder@waikato.ac.nz
(pp. 43–58)

students' engagement with mathematical ideas. Some mathematics educators contend that the use of digital technologies offers the opportunity to re-envisage aspects of mathematical education, including alternative ways to facilitate understanding (Borba & Villareal, 2005; Calder, 2011). For instance, the visual and dynamic elements of engaging mathematical thinking through digital technologies repositions the types of knowledge and understanding required and the ways the learning emerges. The affordances offered by learning through digital learning environments enable alternative ways to encounter, engage with, and explain mathematical ideas. Researchers have identified that the dynamic, interactive nature of learning with digital tools, facilitates students' experiences and understandings in particular ways. For instance, using dynamic geometry software can give a different sense of shape and movement, helping children to more easily explore what changes and what is invariant in geometry transformations (Laborde, 1998; Ng & Sinclair, 2015). In a similar way, the exploration and transformation of data with digital technologies offer interactive ways to analyse data and statistics (Forbes & Pfannkuch, 2009).

This article applies an interpretive lens to examine the manner in which mathematical thinking emerges when children work with the programming language Scratch. Scratch is a media-rich digital environment that uses a building block command structure to manipulate graphic, audio, and video (Peppler & Kafai, 2006). It incorporates elements of Logo including 'tinkerability' in the programming process (Resnick, 2007). This allows the user to combine the programming building blocks (at times incorporating measurements) and to immediately observe the outcome of that programming.

When learners investigate phenomena in a digital environment, some input is entered. This is based on the students' engagement with, or reflection on, the task. The subsequent output is produced visually, almost instantaneously (Calder, 2009). This output can initiate dialogue and reflection. Learners will reposition their perspective, even if only slightly, and re-engage with the task from the new perspective. They engage in an iterative process, alternatively attending to the task and their emerging understanding. This is a type of learning trajectory that can occur in various media (Gallagher, 1992), and is evident in many learning situations that involve a digital pedagogical medium (see also Borba & Villareal, 2005). However, there are affordances of the digital medium that influence the nature of the engagement (Calder, 2011). These affordances frame the nature of the problem-solving activity.

This article considers two forms of mathematical thinking that emerged when participants created mathematical games to facilitate understanding of number concepts with their younger, 'buddy' class. One is the evolution of logic and reasoning that developed through creative problem solving during the programming process, while the other involves the conceptual area of geometry. The first is examined through the iterative, interpretive process described above, to see how the children's mathematical thinking evolves as the group creates, and then refines their game. Geometric thinking is also examined in conjunction with measurement, as it is interspersed through the overall iterative process in which children transform their 'sprites' (animated figures), including moving them to specific locations.

First, the features and affordances of digital technologies will be considered, especially those that relate to working with Scratch. The ways these affordances might influence the learning process when mathematical phenomena are engaged through digital technologies is then outlined, including the potential that digital technologies have for reshaping the learning experience and the mathematical thinking. Next, coding and computational thinking are considered and how these ways of thinking might resonate with mathematical thinking. The ways that mathematical thinking might develop through coding with Scratch is the central purpose of this article. This leads to a description of Scratch and some of the features of the programme. Following this is an overview of the actual research project and the methods used to gain a better understanding of the learning situation when Scratch is used. Finally, the findings are reflected on to draw conclusions related to the students' mathematical thinking as they use Scratch to develop code for their project of designing a game.

Affordances of digital technologies

Affordance implies a complementarity of the learner and the environment. Affordances are not the abstract physical properties but the potential relationships between the user and the environment (Gibson, 1977). They are the potential for action, the capacity of an environment or object to enable the intentions of a user within a particular problem situation (Tanner & Jones, 2000). Allied to this notion is the symbiotic relationship between the digital media and the user. The digital medium affords opportunities for the student's approach and hence may influence the understanding, but existing knowledge simultaneously guides the way the technology is used, and in a sense shapes the technology. The student's interactions are shaped by the medium, but in turn shape the medium (Hoyles & Noss, 2003).

An affordance commonly associated with digital environments is the potential to present multiple representations. The ability to link and explore visual, symbolic, aural and numerical representations simultaneously in a dynamic way has been frequently identified in research. For example, Ainsworth, Bibby, and Wood (1998) suggested that multiple representations promote learning by highlighting different features, with the information gained from combining representations more than that gained from a single representation. They also suggested that when learners connect representations with others, they have to engage in activities that promote understanding. *Scratch* is a platform with symbolic, numeric, visual and aural representations that directly interact with each other. In essence, the coder uses some representations, usually symbolic and numeric, to transform or manipulate other representations, often visual or aural ones. Benton, Hoyles, Kalas and Noss (2017) suggested that coding as another representation is likely to lead to deeper learning.

Moyer-Packenham and Westenskow (2013) identified affordances of focused constraint, creative variation, simultaneous linking, efficient precision, and motivation when students used apps in their mathematical learning. Three of these: focused restraint, where the app might focus students' attention on particular mathematical concepts or processes; creative variation, where the app might encourage creativity, hence evoking a range of student approaches and potential solutions; and simultaneous linking, where the app might link representations simultaneously and connect them to student activity, are applicable when using *Scratch*. Meanwhile, Sacristán and Noss (2008) discussed how interaction with computational tasks in a carefully designed micro-world might lead to different representational forms (such as visual, symbolic and numeric), a process that they called 'representational moderation'.

Digital technologies afford opportunities to engage dynamically, with learners gaining instantaneous feedback to input (Calder, 2011). Video recording and playing features enable the creation of, and reflection on, concepts and processes, while touchscreens open opportunity for direct haptic interaction with associated forms of cognitive embodiment (Sinclair & Heyd-Metzuyanim, 2014). They contend that this affordance makes touchscreen devices more accessible to young children. Hutchinson, Beshorner, and Schmidt-Crawford, (2012) suggested that when students encountered problems navigating various features, they worked collaboratively to address them. These affordances, when integrated appropriately by the teacher, can facilitate the exploration of powerful ideas in mathematics, with students posing problems, and creating personal explanations (e.g., Sandholtz, Ringstaff & Dwyer, 1997). Sandholtz et al (1997) also reported enhanced high-level reasoning and problem solving linked to learners' investigations in digital environments. The interactive and multi-representation affordances, coupled with appropriate teacher intervention, enable the learner to make links between areas that might otherwise have been engaged with discretely. The affordances, particularly gaining instantaneous feedback from input, can also foster risk-taking and experimentation (Calder, 2011), allowing space for students to explore.

Digital technologies, when used appropriately, enable opportunities to create, explore and organise data or mathematical phenomena in ways that might enhance mathematical thinking, and make sense of what is happening: to see patterns and trends more quickly in mathematical situations that might otherwise be too 'dense' or complex to do so. They allow the learner potential to look through the particular to the general (Mason, 2005). When coding, this offers the potential to learn through the iterative process of engagement and reflection, with incremental engagement with the code, and the output that the coding generates. The coder can try something and relatively quickly determine the effects of the new coding allowing them to generalize the particular coding attributes and modify their perspective. With a visual environment such as *Scratch,* where the coding and output screen sit side by side, these relationships are even more easily identified. For the purposes of this article, I am using digital pedagogical media as a term to identify digital media that are used in the teaching and learning process. Digital pedagogical media do not operate in isolation, however. Their influence is symbiotic with the pre-conceptions of the user, other societal and cultural discourses, and the nature of the learning process. In the next section we consider how these affordances and features might influence the learning process.

Learning mathematics informally through digital technologies

In this section, we consider the potential of digital technologies to influence the learning in areas of mathematics that were identified in the data: problem-solving, often including numerical elements and geometry. Geometry, with its visual and construction elements, lends itself to utilising digital technology as a pedagogical environment. Early proponents of using computer technology in mathematics education, such as Papert (1980), provided the catalyst for rich practical classroom experiences and the beginnings of associated mathematical research. Movement and time are incorporated into this process, with motion an integral part of its defining state (Stevenson, 2006). As such, it mediates forms of geometry that are not in the school curriculum. It also focuses the user's attention on the direct relationship between the code and movement, with the almost instantaneous response to input facilitating an exploratory and risk-taking disposition to emerge with users (Calder, 2011).

The Dynamic Geometry Software (DGS) forms most commonly adopted in school geometry, *Geogebra, Geometer's Sketchpad* and *Cabri-geometry,* utilise external dynamism in that the learner moves figures or their features on the screen. In both, the learner constructs diagrams, and examines the logical dependencies between figures and associated points, and the corresponding relationships. The learner can interact directly, in a dynamic manner, with the figures they have created, or that have been created for them. This facility, coupled with the ability to animate figures that have long been in static, two- dimensional form (Mackrell & Johnston-Wilder, 2005), set DGS apart from pencil-and-paper technology as a pedagogical medium and facilitates the re-organisation of thinking in geometry. A circle, for example, is understood differently according to whether it is constructed using a pencil and compass, a template, *Cabri-geometry* or *Logo* (Calder, 2011). Meanwhile, Balacheff and Sutherland (1993) discussed the epistemological domain of validity of a computer microworld by contrasting two microworlds, *Cabri-geometry* and *Logo* and illustrating that learning through these environments is likely to lead students to construct quite different meanings, even when the mathematics task is the same.

Other DGS, such as *Cabri 3D* have also enhanced students' ability to visualise when modelling physical constructions and motion (Mackrell, 2006). Mackrell suggested that using an integrated approach, that includes interactive demonstrations and pictures, helped in the emergence of new forms of mathematics. It is the dynamic visualisation of screen objects in these environments, perhaps by dragging directly on the screen, which most significantly differentiates them from engaging in geometric thinking through other media (Mackrell & Johnston-Wilder, 2005). Placing the emphasis on the visualisation dimension of geometry has opened opportunities for the design of software that enhances those qualities. While developing software that makes the construction and manipulation of geometric objects in three-dimensional space possible (*3DMath*), the key elements of visualisation were privileged. Jones, Christou, Pittalis and Mousoulides (2006) reported on this process, and how mental images, external representations, and the means and potentialities of visualisation were given significance. This is designing software to deliberately shape the learning process in a particular way, and as such recognises the influence that the affordances of digital pedagogical media have on the interpretation and organisation of meaning. Digital technologies may be designed to make an existing practice more efficient, but their complexity, and sometimes-contradictory character may reshape traditional processes into more innovative ones (Sutherland, Linstrom & Lahn, 2009). Conversely, pedagogical approaches can also evolve to reflect the affordances of the media.

The use of mobile technologies, including iPads, and apps have become more prevalent in school mathematics programs, particularly in primary school and early years' settings. Using mathematical apps with pre-schoolers has indicated the potential to enhance the children's number sense, including improvements with the identification and writing of numbers and the associated quantities (Spencer, 2013). Likewise, Baccaglini and Maracci (2015) identified links between using iPad apps and the emergence of number sense. They noted that pre-schoolers' understanding of counting principles, approximation of quantities, subitising and recognising parts of a whole were aspects of number sense that might be associated with the use of apps (Baccaglini & Maracci, 2015). Increased engagement with mathematics concepts was evident when children used apps (Attard, 2015), with enhanced opportunities for repeated patterning and using mathematical language. She also reported that the strategies 7-year-olds' used in subtraction were supported when the children used *ShowMe* to articulate their strategies and thinking. Much of the discussion regarding the ways iPads and apps might influence the affective elements of the learning experience, is centered on the notion of student engagement; of students being actively engrossed and motivated, often by the visual and interactive characteristics (Carr, 2012; Li & Pow, 2011). The greater, more specific use of mathematical language in geometry has also been reported with 5 to 7-year-olds (Ng & Sinclair, 2015). In Ng and Sinclair's paper, the children used *Sketchpad* to explore symmetry and gained understanding in key properties of symmetry such as the image being equidistant from the line of symmetry as the object.

While the research is relatively cohesive that digital technologies have facilitated the reshaping of the learning experience and mathematical thinking, the specific situation of each particular research project, and the associated pedagogy that was used will have influenced the findings of each. Yet there are commonalities in the ways that the features of the digital pedagogical media afford learning opportunities. With *Scratch,* the affordances of multi-representation, dynamic interaction, and immediate feedback to input were particularly evident (Calder, 2011). Students can use numeric and symbolic representations in the coding blocks that link to these visual representations and movements. This linking is dynamic. When students write a new piece of code or change an existing piece, they might select and move blocks and input data on screen directly. Moreover, if they try the code, they would see its effect on the screen, and this feedback to their inputted code would be immediate.

Enhancing student engagement is also a consistent outcome, as is facilitating a more open problem-solving approach that frequently incorporates computational thinking. This was especially true when students worked in the *Scratch* environment. Using *Scratch* to code games, also suggests that computational thinking would be utilised and enhanced through the process. The next section considers coding and computational thinking; how the affordances influence this, and commonalities with mathematical thinking.

Coding and computational thinking

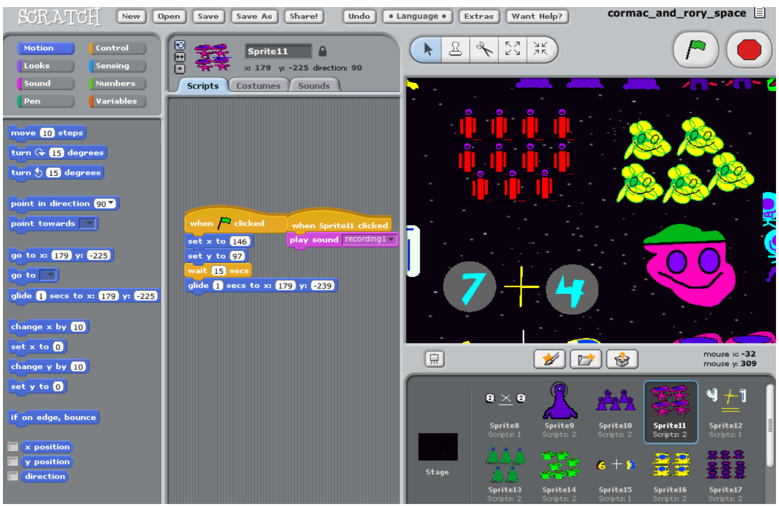
Coding has underpinned the development of digital technologies and the evolution of applications that support and facilitate mathematical thinking. Allied to this is the notion of computational thinking. Computational thinking, the thinking used in the coding process, is diverse in definition, with at times conflicting perspectives used to articulate it (Brennan & Resnick, 2012). While it is sometimes articulated as the thought processes involved with formulating solutions to coding problems, Bennan and Resnick (2012) also included the students' evolving perspectives of their ongoing relationships and the world around them: How these perspectives changed through coding activity. They suggested that a framework for computational thinking would include the dimensions of computational concepts, computational practices and computational perspectives. They also identified seven programming concepts that were utilised in a wide range of Scratch projects but were transferable to other programming contexts: sequences, loops, parallelism, events, conditionals, operators and data (Brennan & Resnick, 2012). The nature and application of these concepts can depend on the features and affordances of each coding language. However, computational thinking is generally considered a collection of problem-solving skills that relate to principles of computer science (Curzon, Black, Meagher & McOwan, 2009).

Computer science frequently is involved with creating applications to solve problems, usually in real-life contexts (Jaokar, 2013). Computational thinking is an analytical, computing concept approach for problem solving, modelling situations and designing systems (Wing, 2006). Wing (2006) also advocates that as well as drawing on concepts aligned with computer science to solve problems, that the problems might also be represented in ways that enable them to be evaluated through digital techniques and media. Abstraction is an essential component, with logical thinking, algorithmic thinking, innovation, and creativity all considered elements that are central to the constitution of computational thinking (Wing, 2008). These elements also resonate with mathematical thinking and problem solving in mathematics. As well, computational thinking enables the analysis of trends in vast data sets, while being influential in other fields of science and humanities (Wing, 2006). For instance, in biology, computational thinking has assisted with the sequencing and analysis of genes. While aspects of these uses might be considered as applications, in coding, there has usually been innovation and adaptation of existing processes to facilitate the desired outcomes. The proliferation of apps for mobile technologies, including in mathematics and literacy education (Calder, 2015) is also indicative of the growing importance of various forms of coding in a range of contexts. Students are not just using apps, but are increasingly creators of them, or collaborators in their development. *Scratch* is an app that can facilitate student engagement with coding (Otrel-Cass, Forret, & Taylor 2009) and the development of computational thinking (Brennan & Resnick, 2012).

Scratch

Scratch is a free-to-use graphical programming language (*http://Scratch.mit.edu/*) that is designed to facilitate and enhance technological fluency (Resnick & Silverman, 2005). It was created at the Massachusetts Institute of Technology's (MIT) Media Lab and has its developmental roots in the conceptual ideas of Seymour Papert (Papert, 1980) that led to the development of the programming environment *Logo*. Papert's ideas underpin the design of *Scratch* with one of its key aims being to provide 'tinkerability' so that child programmers can put together, take apart, and recombine coding building blocks to build whatever they wish (Resnick, 2007). The blocks can be deconstructed and recombined as users logically develop the desired movements and effects. In this way, it allows for accessible and creative ways to program. Moreover, *Scratch* has been designed particularly with young users in mind hence the features and media are highly appealing to children. The program interface has three distinct areas (see Figure 1).

Figure 1. Scratch has three working areas on the screen: A programming block, a scripting area and a stage area



On the left the programming blocks are arranged, each with its own function. These can be dragged into the scripting area and snapped together to create programs. Scratch uses 'drag and drop' functionality that eliminates the need to remember codes or understand the syntax (Otrel-Cass, Forret, & Taylor 2009). Often in programming languages, it is possible to write a syntactically incorrect script. Within the Scratch environment, the command blocks will not fit together unless they can run together. This particular affordance of the environment makes it easier to learn and create programs using Scratch and allows the child-programmer to spend more time on the logic and creative elements of their program (Otrel-Cass, Forret, & Taylor 2009). In the middle is the scripting, or programming, area where programming blocks are combined to provide the instructions that control the 'sprites'. In the upper right corner is the 'stage', where graphic elements called sprites are placed. Programs that are used to create the desired effects control these sprites on this stage.

The visual building blocks in *Scratch* can interact with a variety of media, for example, to integrate video or music clips, and the building blocks allow users to create their own interactive stories, animations, games, or art. Users can share their projects online. The program is very versatile and can be used for teaching and learning in virtually any subject area, including mathematics, science, music and art. *Scratch* provides a playful learning environment that, although simple and easy to use, is capable of producing complex and sophisticated outcomes. It provides a context within which children can enjoy exploring and being creative with programming, while simultaneously developing their understanding of a variety of embedded programming and mathematical concepts, such as, position and orientation. In the first phase of a three-year project that used *Scratch,* Benton et al., (2017) reported that one focus was on problem solving, with the problem-solving process highlighted rather than the solution (. *Scratch* facilitates creative problem solving, logical reasoning, and encourages collaboration (Peppler & Kafai, 2006).

Research context

The aim of the research project was to gain some insights into a teacher and her Year 6 (9-10-year-old) students' perceptions of Scratch. The class, a digital learning class of 26 students, had already used software programs such as Keynote, Flash, and Hyperstudio. We were interested in finding out what they thought this software had to offer. This research considered the factors that supported learning with Scratch in this class, with the intention to provide suggestions for teachers using similar coding programs. Over a two-week period, students were involved in completing 'mini-projects' in their classroom to familiarise themselves with the program and ways to use it. The first week was an opportunity for the children to explore and become familiar with Scratch, while the second week was for the children to plan and implement their own projects. This article reports on how the mathematical thinking of the students was influenced through coding with Scratch. This research continues an ongoing examination of how digital pedagogical media influences the learning process in mathematics.

Methodology

An interpretive research methodology, contemporary hermeneutics, was used for this project both to analyse the data and to better understand the development of the students' mathematical thinking. This contemporary hermeneutics frame helps unpack the mathematical learning process when the mathematics is engaged through digital media (Calder & Brown, 2010). In this perspective of the learning process, the preconceptions that each learner brings to engaging with mathematical phenomena are derived from a specific cultural domain that the learner inhabits (Gallagher, 1992). From this contemporary hermeneutic viewpoint, mathematical thinking emerges from a process of interpretation, with understanding and concepts perceived as temporary fixes of ongoing development, rather than permanent realities. This resonates with the notion of the hermeneutic circle, with learning and understanding always in transition (Ricoeur, 1981). Our understanding of mathematical ideas emerges through cyclical engagements with the phenomena and the continual drawing forward of previous encounters and understandings. Hence, mathematical concepts are not considered as fixed realities, but more formative processes that are enhanced as the learner interprets the mathematical phenomena from new, ever-evolving perspectives. The mathematical task, the pedagogical medium, the preconceptions of the learners, and the dialogue that these encounters evoke are co-formative. Their relationship with the learner engenders understanding (Calder, Brown, Hanley & Darby, 2006). Here, understanding is the learner's interpretation of the mathematical situation through those various interdependent filters (Calder & Brown, 2010).

When learners engage in investigation, they interpret the task, their responses to it, and the output of their deliberations through the lens of their pre-conceptions; their emerging discourse in that perceived area. Social and cultural experiences always condition our situation (Gallagher, 1992), and thus the perspective from which our interpretations are made. Learners enter such engagement with pre-conceptions of both the phenomena and the pedagogical medium through which it is encountered. The engagement with the phenomena alters the learner's conceptualisation, which then allows the learner to re-engage with the task from a fresh perspective. This cyclical process of interpretation, engagement, reflection and reinterpretation continues until there is some perceived reconciled interpretation of the situation. Other researchers have perceived learning as an iterative process of re-engagement of collectives of learners, media, and other environmental aspects, with phenomena (e.g., Borba & Villareal, 2005). Likewise, the interpretation of research data emerges through a similar iterative process as the data are engaged through ever-evolving researcher perspectives.

Research design

This article continues an ongoing examination of how digital pedagogical media influence the learning process. Each student had access to their own computer and although this was their first experience with Scratch, they were confident and experienced with a range of software. Their teacher was the school's digital learning coordinator. The students worked in pairs, which were self-selected and single gender. Over a two-week research period, the students wrote daily blogs articulating their progress and reflections, students and the teacher were interviewed, and classroom observations (both written and photographic) were recorded. These data, along with informal observation and discussion, were then systematically analysed. Pseudonyms were used in the reporting of the project.

The first week involved the students doing a range of distinct, structured tasks to familiarise them with the *Scratch* environment. At the end of the first week all the pairs were then given the same design brief: To design and build a mathematics game suitable for facilitating the number understanding of their Year 1 (aged 5-6) 'buddies'. The students interviewed their Year 1 'buddy' class partners, two students with two buddies, and consulted the Year 1 teacher regarding appropriate mathematics concepts and activities that the children were familiar with. This helped determine the nature of games they would devise. After trialling the games several times part way through the development process, the younger children also gave formative feedback. The Year 6 student pairs selected their name (e.g., 'Jabadah', 'Jigsaw', 'Hemzie', 'Mats', 'XE2', 'Lisa' and 'Pig' are the pairs referred to in this article), which were then used throughout the rest of the project. While the pairs worked to the same brief, their solutions varied in both content and approach.

A feature of the approach taken by the teacher was the sharing of the work that had been done each day. Each project was loaded onto a USB stick near the end of the session and one student took responsibility to coordinate displaying the work on the data projector. Each group would explain what they were doing and any characteristics of their programming. The other students could ask questions and provide feedback and suggestions. The students' respect of each other and confidence with this process was a feature of the classroom culture and clearly had been engendered before the project took place. The feedback session also gave opportunities for the teacher to formatively assess, to identify aspects that might need individual or whole-class feedback. It invited students to identify other class members who could assist them with aspects of their design problems. The projects in these varying evolving stages, and the accompanying feedback, likewise became part of the data.

The research project involved a case study approach with one digital classroom. The interpretive lens used to interpret the data reflects the socio-cultural discourses that influence learners as they move through cycles of interpretation, action and reflection in the learning process. The project considered that learning is mediated by language and the use of tools. Not only does the dialogue of the teacher and the learners in the classroom act as a mediator, the app itself acts as a mediating tool. The learner's preconceptions of the pedagogical media, in conjunction with the opportunities and constraints offered by the media itself, promote distinct pathways in the learning process. Hence, the device will inevitably influence the mathematical ideas developed. This pedagogical device is more than an environment. It is imbued with a complexity of relationships evoked by the users, and the influence of underlying discourses.

As well as the learning being envisaged as a contemporary hermeneutic process, this approach also underpinned the data analysis. The data from the first week were initially analysed by identifying themes that were evident in the data, similar to a grounded theory approach. As subsequent data was generated and considered, the interpretation of the findings went through transitions. The researchers' perspectives were likewise modified. This occurred through iterations of engaging with data, reflecting on interpretations and modifying underlying perspectives. This process was ongoing as data such as further student blogs and observation were recorded, and also continued beyond the end of the data collection period. The aim of the overall research project was to gain insights into the ways that *Scratch* might facilitate learning opportunities for the class and teacher. However, this article is concerned with two aspects of mathematical thinking that were facilitated through the coding process and hence are reported in the findings below.

Results and discussion

There were two aspects of the data that were particular to the development of mathematical thinking. The first one was the emergence of a general problem-solving approach that resonates with computational thinking and problem-solving in mathematics, while the second was the use and development of measurement and geometry concepts: space, location, distance, and time, in particular. This was consistent with the development of mathematical thinking in those areas. The excerpts used are illustrative and typical, although each pair created a unique solution to the brief given.

Problem-solving in *Scratch*

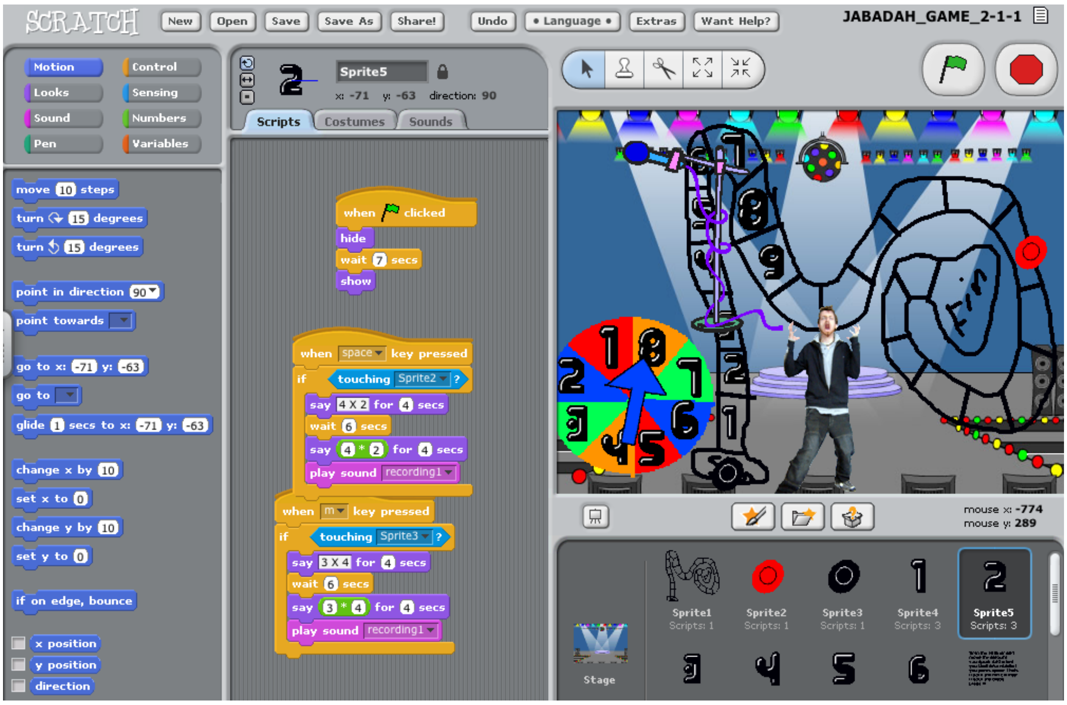
The students' design brief was set within a mathematics context. While this indicated a mathematics learning situation, a central element of the thinking that took place was problem-solving. The students became familiar with the task, and the associated coding processes, and then through iterations of action and reflection, modified their game. At each juncture, the response to their engagement with the task modified their approach and enabled them to re-engage from their modified perspective. Thus, their thinking evolved, and the games became more refined, as they reset their investigative sub-goals based on the feedback and subsequent reflection. The feedback was in various forms: immediate visual feedback within the program as they changed their programming script; fellow student and teacher feedback and suggestions, feedback from the intended users, and feedback involving other groups that unfolded in the public domain. Each of these varying forms of feedback led to reflection, followed by re-engagement with the task from a modified perspective. Hence, their learning about coding in *Scratch* developed over iterations of engagement, reflection and re-engagement. Frequently, they used a guess and modify approach. Sometimes that moved them towards a generalisation.

For instance, the 'Jabadah' group began with the 'stage' window of the workspace and explored changing the colour of it, how to move the sprites, and some of the pre-programmed effects. They settled on a stage colour and then experimented with moving the sprites that made up the letters of their group name. They wanted to make the J sprite hit the A sprite, to then set the A sprite off spinning. However, the J moved in a continuous loop, without hitting the A. The following observational data, recorded their discussion:

> James: We can't get it to go forever - we'll need to explore different loops.
>
> Don: What if we glide until it points to the direction?
>
> James: We can point towards.
>
> Don: What about exploring the use of 'sense'?

They tried some options and considered the visual feedback resulting from each change in the coding. Through this, they were developing a sense of the relationship between the programming script they had selected and modified the measurements of, and the associated movement of the sprite on the screen. The next day they continued this relational experimentation by 'using existing scripts to see how to manipulate things differently'. During this experimentation, they worked out how to design and operate a spinner (see Figure 2). When they reached a point of uncertainty they used a 'predict and check' approach. They reflected on the outcome, before refining their evolving script. This involved further relational thinking, as recorded in the written observations, they 'looked at how the different scripts affected the action of the sprites' and 'experimented with the number scripts in their own project, by putting in variables and then running the script to see what would happen.' They could see the relationship between the modification of the script and the movement of the sprite. They were able to generalise this relationship and predict the likely movement of a sprite given the change to the programming script. This illustrated a development in their relational thinking as they became more effective at predicting the outcome of their changes to their script.

Figure 2. The group 'Jabadah's' workspace as their game is in development.



While the spinner was now operating successfully, they had encountered another problem. Although they were able to move the blue and red counters on their board game whilst *Scratch* was in design mode, they had not been able to move the counters in full screen-mode. They experimented with other scripts line by line. Eventually, through evaluation of the feedback to their input, they were able to see the pattern in the script and achieve this aspect. With each engagement, they reflected on the digital feedback, modified their interpretation of the situation, and re-engaged with the task from this modified perspective. Their thinking evolved through the problem-solving process. As well as the relational thinking, they also used logic and reasoning to evaluate and interpret the situation, before resetting their sub-goal in the investigative process. They generalised from a range of actions, and after reflection, determined the type of command that produced the desired effect. They also responded to other feedback. One of the researchers, MF said, 'How does the code work, tell me what that code means?' and James responded, saying 'It just spins randomly and lands on a random place'.

Although the question was not answered with the detail intended, the student nevertheless has reflected on the question and articulated a response in terms of both the language of movement (spins, lands) and chance (random). The children also articulated the movement of the spin in mathematical language, which they understood, even though the script was modified rather than created. Don, one of the children said, 'All we did was go: "When sprite 15 clicked repeat random 3 to 100 and turn 45 degrees."'

This suggested an understanding of rotation and the link between the numerical size and the movement of the turn (45 degrees). In order to make the spinner (sprite 15) rotate, they stipulated the 45-degree angle size, after trialling several different angles. This helped their understanding of the link between the numerical value of the angle and the size of the corresponding angle or turn. It also indicated a practical understanding of the coding functions, 'repeat' and 'random', with random in particular, having mathematical meanings. In the interviews, they articulated the value of the buddy feedback and how they responded to it by adapting the game. James observed that 'They said it was fun. They thought the spinner was cool', while Don said, 'We changed the questions from multiplication to addition because it was too hard for them'.

This was a further refinement, enacted within the ongoing problem-solving process. Eventually, they were satisfied with the game and the way it operated. Data from other groups also highlighted the way *Scratch* facilitated problem solving. For instance, when Geoff had run into a problem with the logic of the scoreboard of their game, he said, 'I'll need to problem solve that'.

He investigated spacing, proportion, colour, and size aspects of the scoreboard, and then trialled and evaluated various options. The challenge of the problem-solving process was evident in the blog from another group when they admitted that 'We are trying to figure out how to use a gravity effect and how to use the variables. We are finding it challenging to make our character Jetman jump in the air without spinning 15 degrees.'

The teacher also discussed problem solving in her final interview. She indicated that the children engaged in problem-solving processes through their work with *Scratch* and discussed some of the benefits, saying that 'The communication and competencies coming through with the use of it. That whole problem solving and questioning (aspects). So the whole thing of exploratory learning was where it was a very valuable bit of software.'

The competencies that she refers to are the key competencies in the New Zealand Curriculum (Ministry of Education, 2007). These are: Thinking; using language, symbols and text; managing self; relating to others and participating and contributing. While all these were evident to some extent, the data indicated that thinking, and using language, symbols and text, were most noticeable (see Figure 2). She further stated that two of the benefits for the children using the program were in problem solving and mathematics.

Further thinking in geometry and measurement

Instances of thinking in geometry and measurement emerged during the iterative process through which 'Jabadah's' game evolved. The trialling of variations of movement, angle size and coordinates, and linking these to the instantaneous effects would have enriched their understanding of these aspects. There was also evidence of geometric thinking from other groups. These will be reported as different snippets rather than endeavouring to situate them within each group's overall process.

The 'Jigsaw' group explored changing the length of time for repeat movements and varying angle sizes. They later articulated their attempt to make the letters glide into place, eventually figuring out how to use coordinates to specify where the 'sprite' was to glide to and how to keep the characters in place. The 'Mats' group likewise aimed to explore animation and movement. They worked out how to use the glide command and x- and y- coordinates to move to different positions on their stage. This activity gave them a sense of specific coordinates on the 'stage'. Likewise, it helped develop their sense of where the coordinates were relative to each other. As they changed the coding to move a 'sprite' from, for instance, coordinate (1, 4) to (13, 17), they immediately saw the change in location of the sprite that was associated with the change of coordinates. Their understanding of coordinates and their relative position on a Cartesian plane seemed to be enhanced through this activity. They used this approach to position the numbers for their game in different positions on the stage. From the interview data:

> Stan: We have to remember where the numbers (their sprites for the game they were devising) go, so they all move to the middle and then they mix around to different places.
>
> Matiu: We want to put the numbers in position.

Later, they applied this learnt skill to their game, when moving asteroids through space. They were observed manoeuvring a spaceship and dodging spinning asteroids. Interestingly, at another point, when writing another piece of script for their sprite to move, they initially recorded: "turn 90 degrees, wait one second" 10 times, rather than using a more efficient: "repeat 10 times command."

After choosing a stage and sprite, another group, XE2, were observed immediately engaging with movement and the positioning of their sprite. This involved the use of coordinates to indicate the position they wanted the sprite to glide to. They were not concerned with the exact position of the coordinates, but more the general position associated with them. They spent time exploring different coordinates and how this affected the position of the sprite, gaining a sense of the relationship between the values of the coordinate and the position on the screen. They also programmed 'wait time' of 5 seconds and 'hide time' of 6 seconds. 'PC' also experimented with time, and what the interval signified when creating their game. They formulated a program that offered simple addition equations such as ' 7 + 4 = ' and the 'buddy' children needed to match the solution to the appropriate number of aliens. They built in a feedback function to the game if the addition question was answered correctly. Peter, for example, says 'If you get this right, it tells you, and then it changes to the next question in fifteen seconds.'

In the 'Hemzie' group blog, the data indicate that they had marked plots on a pencil-and-paper map they had made, to help them work out how to move the sprite from one place to another. Their aim was to explore sprites and how to change from one sprite to the next. They thought this aspect was challenging. Later they worked out the movement effects they required and were incorporating sounds. They recorded and linked an appropriate sound for each movement. They eventually enabled a car sprite to be moved by inputted commands. Brian noted that 'The reward is that you get to steer the car around for 20 seconds'.

The 'Lissa' group also had initial difficulty with the movement but learnt from the feedback sessions. They were eventually able to have a beach ball move around the screen through a maze, controlled by the keyboard arrows. The 'Pig' group explored similar areas but with an additional transformation. "They wrote in their blog: We have learnt how to move letters and characters by programming a key on the keyboard to move an object. We learnt that if you use a text box you can't make an animation with effects, it will just enlarge your sprite." They articulated that their initial aim was to find out about position and effects. They were exploring movement and angles.

While engaged in the programming experiences, *Scratch* appeared to facilitate the children's understanding of angles and measurement, with experimentation enabling them to find what was appropriate to use in their particular context. Errors with programming appeared to have a positive effect in that they prompted the children to willingly experiment with commands to achieve the desired appearance and effects. The tinkerability of *Scratch* facilitated exploration with angles and the measurement of time and length. Students could actively experiment with angle size, for example, in ways that would not be possible without the digital medium. Likewise, the understanding that emerged regarding coordinates was inherent in the process of exploring the movement and position of the sprites. Game contexts and practice can significantly improve spatial performance (Clements, Sarama, Yelland, & Glass, 2008). The study not only involved participants with spatial movement and location while designing the games, the trialling and modification process would also have influenced the children's spatial awareness.

Conclusions

Scratch software appeared to be an engaging and relatively easy to use space for problem solving. Additionally, the findings indicated that it was an effective medium for encouraging communication and collaboration (Otrel-Cass, Forret, & Taylor, 2009). Each of the above episodes illustrated how *Scratch* provided a worthwhile and motivating programming environment to explore some mathematical ideas. The challenge of creating a mathematical activity or game for younger students overtly positioned the program in mathematics, while the participants drew on mathematical concepts to develop their game. What is not quite so certain is the extent to which new mathematical learning occurred during this process. The students in this digital class were able to access and understand the programming capabilities and used mathematical thinking in their approach to problem-solving. In the classroom, where electronic media and an environment where discussion and sharing were the norm, the students were able to transform their ideas into workable programs. It proved to be a medium whereby programs were easily composed and decomposed, thus encouraging the use of critical, meta-cognitive and reflective skills. The sharing sessions were pivotal in that they provided a forum for displaying work and as a way of collectively helping each other to solve programming problems. As well, to some extent, the themes that emerged from the study were dependent on the nature of this task, the teacher role, the classroom culture, and the way the teacher set up and interacted with the students through the task.

Scratch, therefore, provided an opportunity for students to develop their computational thinking. They used logical thinking and engaged an analytical approach to the problem-solving, as they tested, reflected on, and then modified their coding. They used abstraction in determining the processes that enacted their desired outcomes and generalized from the responses they received to specific coding. They used creative and innovative approaches to address issues that they encountered with the coding. The whole-class formative feedback process demonstrated that their emerging solutions to the design problems were represented in ways that enabled them to be evaluated through digital media (Wing, 2006). The overall process facilitated logical and geometric thinking as the students tested ideas in response to feedback. The influence of program feedback in the evolution of students' geometric ideas has been reported elsewhere (e.g., Clements, Sarama, Yelland & Glass, 2008).

While not specifically designed to facilitate conceptual thinking in a particular mathematical area, there were clear indications of the children engaging with mathematical ideas. To some extent, their mathematical thinking was enhanced through the use of *Scratch* in the development of the digital learning objects. Their spatial awareness, understanding of angles, and positioning sense through the use of coordinates, were all engaged to varying degrees. There was also evidence of relational thinking as the children made links between their input, the actions that occurred on screen, and the effect of specific variations of size in coding procedures. However, the process the participants undertook more directly facilitated mathematical thinking through the creative problem-solving process it evoked, and the development of logic and reasoning as they responded to the various forms of feedback. These mathematical conclusions can nevertheless only be tentative. While consideration of mathematical thinking was one intention of the research study, it was predominantly set up as an open investigation into the potential of the software across a range of learning areas. A more focused study on the mathematical learning implications may have been more productive in the revealing of mathematical thinking, and may have reached less tentative conclusions.

References

Ainsworth, S. E., Bibby, P. A., & Wood, D. J. (1998). Analyzing the costs and benefits of multi-representational learning environment. In M. W. van Someren, P. Reimann, H. P. A. Boshuizen, and T. de Jong (Eds.), *Learning with Multiple Representations* (pp. 120-134). Oxford, U.K.: Elsevier Science.

Attard, C. (2015). Introducing iPads into primary mathematics classrooms: Teachers' experiences and pedagogies. In M. Meletiou-Mavrotheris, K. Mavrou & E. Paparistodemou (Eds.). *Integrating touch-enabled and mobile devices into contemporary mathematics education*. DOI: 10.4018/978-1-4666-8714-1.ch009.

Baccaglini, A. & Maracci, M. (2015). Multi-touch technology and pre-schoolers' development of number sense. *Digital Experiences in Mathematics Education 2015:2*. DOI: 10.1007/s40751-015-0002-4.

Balacheff, N. and Sutherland, R. (1993). Epistemological domain of validity of microworlds: the case of LOGO and Cabri-Géomètre. In R. Lewis and P. Mendelsohn (Eds.). Proceedings of the IFIP TC3/WG3.3 *Working Conference on Lessons from Learning,* 137-150. Amsterdam, The Netherlands: North-Holland Publishing Co.

Benton, L., Hoyles, C., Kalas, I. & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education 3,* 115-138. DOI 10.1007/s40751-017-0028-x.

Borba, M. C., & Villarreal, M. E. (2005). *Humans-with-media and the reorganization of mathematical thinking: Information and communication Technologies, modeling, experimentation and visualisation*. New York, NY: Springer.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

Calder, N. S. (2009). Visual tensions when mathematical tasks are encountered in a digital learning environment. In Tzekaki, M., Kaldrimidou, M. & Sakonidis, C. (Eds.), *In search of theories in Mathematics Education*, Proceedings of the 33rd annual conference of the International Group for the Psychology of Mathematics Education. Athens: PME.

Calder, N. S. (2011). *Processing mathematics through digital technologies: The primary years*. Rotterdam, The Netherlands: Sense.

Calder, N. S. (2015). Apps: Appropriate, applicable and appealing? In T. Lowrie & R. Jorgensen (Eds.). *Digital games and mathematics learning: Potential, promises and pitfalls*. The Netherlands: Springer.

Calder, N.S. & Brown, T. (2010). Learning through digital technologies. In M. Walshaw (Ed) *Unpacking pedagogy: New perspectives for mathematics classrooms* (pp 223 – 243). Charlotte, USA: Information Age Publishing.

Calder, N.S., Brown, T., Hanley, U., & Darby, S. (2006). Forming conjectures within a spreadsheet environment. *Mathematics Education Research Journal, 18*, no. 3, 100-116.

Carr, J. (2012). Does math achievement h'APP'en when iPads and game-based learning are incorporated into fifth-grade mathematics instruction? *Journal of Information Technology Education, 11*, 269–286.

Clements, D.H., Sarama, J., Yelland, N.J., & Glass, B. (2008). Learning and teaching geometry with computers in the elementary and middle school. In M.K. Heid & G.W. Blume (Eds.). *Research on Technology and the Teaching and Learning of Mathematics: Vol 1. Research Synthesis,* pp 109-154.

Curzon, P., Black, J., Meagher, L. R., & McOwan, P. (2009). cs4fn. org: Enthusing students about Computer Science. In C. Hermann, T. Lauer, T. Ottmann, and M. Welte. (Eds). *Proceedings of Informatics Education Europe IV*, 73-80.

Forbes, S. & Pfannkuck, M. (2009). Developing statistical thinking: Teaching and learning. In R. Averill & R. Harvey (Eds.), *Teaching secondary school mathematics and statistics: Evidence-based practice* (pp. 93–128). Wellington, New Zealand: NZCER Press.

Gallagher, S. (1992). *Hermeneutics and education*. New York: State University of New York Press.

Gibson, J. J. (1977). The theory of affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving, acting, and knowing: Toward an ecological psychology* (pp. 67-82). Hillsdale, NJ: Lawrence Erlbaum.

Hoyles, C., & Noss, R. (2003). What can digital technologies take from and bring to research in mathematics education? In A. J. Bishop, M.A. Clements, C. Keitel, J. Kirkpatrick and F. Leung (Eds.), *Second International Handbook of Mathematics Education* (Vol. 1, pp. 323-349). Dordrecht, the Netherlands: Kluwer Academic.

Hutchison, A., Beschorner, B., & Schmidt-Crawford, D. (2012). Exploring the use of the iPad for literacy learning. *The Reading Teacher*, *66*(1), 15–23.

Jaokar, A. (2013). Evolving the definition of Computational thinking, retrieved November 29 from http://www.opengardensblog.futuretext.com/archives/2013/07/evolving-the-definition-of-computational-thinking.html

Jones, K., Christou, C., Pittalis, M., & Mousoulides, N. (2006). Theoretical perspectives on the design of dynamic visualization software. In C. Hoyles, J-B Lagrange, L.H. Son, and N. Sinclair (Eds.), *Proceedings of 17th ICMI Study conference, Technology Revisited*. Hanoi: Hanoi University of Technology.

Laborde, C. (1998). Visual phenomena in the teaching/learning of geometry in a computer-based environment. In C. Mammana & V. Villani (Eds.), *Perspectives on the Teaching of Geometry for the 21st Century* (pp. 121-128). Dordrecht: Kluwer.

Li, S. C., & Pow, J. C. (2011). Affordance of deep infusion of one-to-one tablet-PCs into and beyond classroom. *International Journal of Instructional Media, 38* (4), 319–326.

Mackrell, K. (2006). Cabri 3D: potential, problems and a web-based approach to instrumental genesis. In C. Hoyles, J-B Lagrange, L.H. Son, and N. Sinclair (Eds.), *Proceedings of 17th ICMI Study conference, Technology Revisited*. Hanoi: Hanoi University of Technology.

Mackrell, K., & Johnston-Wilder, P. (2005). Thinking geometrically: dynamic imagery. In S. Johnston-Wilder & D. Pimm (Eds.), *Teaching Secondary Mathematics with ICT*. (pp. 81-100). Berkshire, UK: Open University Press.

Mason, J. (2005). Mediating mathematical thinking with e-screens. In S. Johnston-Wilder & D. Pimm (Eds.), *Teaching Secondary Mathematics with ICT*. (pp. 81-100). Berkshire, UK: Open University Press.

Moyer-Packenham, P. S., & Westenskow, A. (2013). Effects of virtual manipulatives on student achievement and mathematics learning. *International Journal of Virtual and Personal Learning Environments*, *4*(3), 35–50.

Ng, O., & Sinclair, N. (2015). Young children reasoning about symmetry in a dynamic geometry environment. *ZDM, 47*(3), 421-434.

Otrel-Cass, K., Forret, M., & Taylor, M. (2009). Opportunities and challenges in technology-rich classrooms: Using the Scratch software. *Set: Research Information for Teachers*, *1*, 49-54.

Papert, S. (1980). *Mindstorms. Children, computers and powerful ideas*. Brighton: Harvester Press.

Peppler, A.P. & Kafai Y.B. (2006). Creative codings: Personal, epistemological, and cultural connections to digital art production. *Learning Sciences,* Proceedings of the 2006 International Conference of the Learning Sciences, Bloomington, IN.

Resnick, M. (2007). Sowing the seeds for a more creative society. *Learning & Leading with Technology*, *35*(4).

Resnick, M. & Silverman, B. (2005). *Interaction Design and Children*. Proceedings of the 2005 Conference on Interaction Design and Children. Boulder, Colorado.

Ricoeur, P. (1981). *Hermeneutics and the human sciences*. Cambridge: Cambridge University Press.

Sacristan, A.I., & Noss, R. (2008). Computational construction as a means to coordinate representations of infinity. *International Journal of Computers for Mathematical Learning*. DOI: 10.1007/s10758-008-9127-5.

Sandholtz, J.H., Ringstaff, C., & Dwyer, D.C. (1997). *Teaching with technology: Creating a student centred classroom*. New York: Teachers' College Press.

Sinclair, N. & Heyd-Metzuyanim, E. (2014). Learning number with *TouchCounts:* The role of emotions and the body in mathematical communication. *Technology, Knowledge and Learning 19*(1), 81-99.

Spencer, P. (2013). iPads: Improving numeracy learning in the early years. In V. Steinle, L. Ball & C. Bardini (Eds.), *Mathematics education: Yesterday, today and tomorrow* (Proceedings of the 36th annual conference of the Mathematics Education Research Group of Australasia). Melbourne, VIC: MERGA.

Stevenson, I. (2006). Geometry through the lens of digital technology: Design and the case of the non-Euclidean turtle. In C. Hoyles, J-B Lagrange, L.H. Son, and N. Sinclair (Eds.), *Proceedings of 17th ICMI Study conference, Technology Revisited*. Hanoi: Hanoi University of Technology.

Sutherland, R., Linstrom, B. & Lahn, L. (2009). Socio-cultural perspectives in technology-enhanced learning and knowing. In N. Balacheff et al. (Eds.). *Technology-Enhanced Learning*: *Principles and products*. Dordrecht, Netherlands: Springer. DOI 10.1007/978-1-4020-9827-7 3.

Tanner, H., & Jones, S. (2000). Using ICT to support interactive teaching and learning on a secondary mathematics PGCE course. *Proceedings of the 2000 annual conference of the Australian Association for Research in Education,* Sydney. Retrieved 26 March 2008 from http://www.aare.edu.au/00pap/00226.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society. *Mathematical, Physical and Engineering Sciences*, *366* (1881), 3717-3725.