I want to be a Game Maker

Experiences of digital game making with eleven year olds

Leonard Busuttil

A thesis submitted to the University of Sheffield
for the Degree of Doctor of Philosophy in the School of Education

October 2014

Acknowledgements

I would like to thank the students and teachers who volunteered to take part in this research and allowed me to share their enthusiasm towards digital games and digital game making. Thanks goes to the school who hosted this research and made the venue available after school hours throughout the workshop.

Thank you to my supervisor Dr Julia Davies for her wisdom, kindness and patience throughout this research project. She managed to keep a precise balance between readily offering her advice yet prompting me to think critically and independently.

Thanks to academics from the School of Education led by Professor Cathy Nutbrown for organising discussion sessions where we could present and discuss works in progress versions of our work. Thanks to fellow students for their support and friendship throughout the study weekends. The discussions that ensued helped me focus more on my research.

Thank you to my family Marilyn, Martina and Louisa for the inspiration and motivation to go through this journey and for being so patient with me when study had to come first.

Abstract

The purpose of this study was to empower a mixed group of eleven year old students with the skills to program digital games and then research the stages of development they go through whilst designing and authoring a game. This study also strived to identify the benefits children gained from a gaming literacy perspective.

This research used a qualitative case study approach. The analysis of this study was based on multiple data sources: statistics collected from the web portal used during the workshop; informal conversations with the children and teachers helping out with the workshop; participant observation and analysis of the games created.

Through the participation in this workshop the children were introduced to the Scratch programming language. The structure of the gaming workshop was influenced by pedagogic approaches to teach creatively for creativity and to introduce programming through a full system approach (Selby 2011).

The stages that children go through whilst creating a digital game are similar to a number of phases discussed in previous research (Robertson 2011, Resnick, Maloney et al. 2009). However the stages of development outlined in this research highlight the importance of the social aspect in game development. Group testing not only has an effect on the game being tested but also acts as a source of cross fertilisation of ideas between the students testing the game and the students developing the game.

The game making experience provided the students with a possibility to enact their systematic thinking when designing their games as a system made up of interrelated subsystems. Prior gaming experience contributed to the way the games were designed and allowed the students to engage with game making using a playful attitude. The students were competitive yet cooperative whilst making their games. All the games were complete and demonstrated that the children were savvy about multimodality. They created games that were well balanced from the difficulty point of view and that provided the game players with instructions on how to play as well as implemented appropriate feedback mechanisms.

Contents

Table of Figures

Table of Tables

1. Introduction

Playing digital games is an important leisure activity for a large number of us. Players tend to play for hours on end, at times losing track of time. Digital games range from simple two dimensional arcade games to virtual reality three-dimensional (3-D) multi user role playing games. The human – digital game interface is diverse too and varies from the traditional coin operated entertainment machines installed in public spaces such as video arcades to controller free consoles that track your body movement and recognise your face and voice through an array of sensors installed in one's living room.

According to research conducted by Pew 97% of teens aged between 12 and 17 in the US play computer, web, portable, or console games (Lenhart, Kahne et al. 2008). The statistics for the UK are similar with 78% of 16 to 19 year olds regularly playing digital games (Games 2008). Other reports for Europe (GameVision Europe 2010), Australia (Brand 2012) and Malta (Busuttil, Camilleri et al. 2014) also provide similar figures.

The increasing popularity of digital games has led governments around Europe to acknowledge the contribution digital game production brings to the economies of these countries. Malta too has launched initiatives in order to tap into the digital game creation market and to benefit from the positive effect this could have on the economy (Stagno-Navarra 2011).

In their influential report entitled "Next Gen", Livingstone and Hope (2011) outline ten recommendations which schools should adopt in order for UK to retain its current position in the digital game creation market.

Of special interest to me as an educator involved in the education of Computing pre-service educators is the emphasis on:

1. The use of game based learning to draw students towards science, technology, engineering, and mathematics (STEM) and computer science in particular
2. The importance given to an interdisciplinary approach including art and computer science in the English Baccalaureate
3. Work-based learning approaches through the use of school clubs

4. The introduction of a national video games development and animation schools competition

These recommendations point towards the inclusion of game based learning in schools and to the introduction of school clubs which offer a different environment to that found traditionally in classrooms. They also point to an interdisciplinary approach through the interweaving of subjects especially art and computer science. Livingstone and Hope also recommend the setup of a national video game development and animation schools competition to further increase the motivation and engagement of students.

Competitions to motivate student programmers have been used successfully in the past (White, Carter et al. 2007) with an international competition held on an annual basis since 1997 (IOI 2014). During the past years such competitions have also taken root in Malta through the *GameZing National Competition*, the *Robotics Challenge*, the *Scratch Competition*[1] and the *Only Girls Allowed* competition (Digital Games Malta 2011, MCST 2014, MITA 2012, MITA 2013).

The *GameZing* competition was launched in 2010 by the Digital Games Malta working group and has established itself as an annual event. *GameZing* aims at raising the profile of digital game development amongst students, teaching institutions, and parents, in order to promote the potential of game making for future employment. Teams of post-secondary and tertiary level students under the guidance of a teacher/lecturer team up to create digital games. Games submitted to the competition are judged by a panel of judges on the basis of innovation, fun factor, production quality and presentation.

The *Robotics Challenge* is an annual event organised by the Department of eLearning (DEL) and the Malta Council for Science and Technology (MCST) since 2011. In this challenge teams made up of up to three students and a teacher mentor are given pre-set tasks which they have to program a robot to perform. Students participating in this

[1] The Scratch competition adopts a new name according to the theme adopted. In 2012 the competition was called *Recreating our History from SCRATCH!*, in 2013 the competition was called *Scratch IT to see IT*, whilst in 2014 it was called *mScratch*.

competition are aged between nine and nineteen years and are placed into three categories: Elementary, Junior and Senior.

The *Scratch* competition and the *Only Girls Allowed* competition were launched in 2012 and 2013 respectively by the Malta Information Technology Agency (MITA) in collaboration with the Faculty of Education at the University of Malta and DEL. The Scratch competition is a yearly competition where groups of students between the ages of nine and sixteen are encouraged to create an entry based on a different topic each year. Throughout the years the topics ranged from history in 2012 to ecology in 2013. The Only Girls Allowed competition was aimed at shoring up interest in digital story telling for eleven and twelve year old girls. Girls who chose to participate had to create and animate a story using Microsoft Kodu or Carnegie Mellon University's Story Telling Alice.

Even though these competitions serve as a source of external motivation for the children they seem to have little to no effect on what happens in school during and after school time. The participation rate remains low with the number of schools participating in the competition decreasing slightly from nine schools in the first edition to eight schools in the third edition (MITA 2012, MITA 2014) .

1.1 Digital Games and Education

Playing games comes natural. Not only have humans been playing games since the beginning of our species, but intelligent animals have as well. Play should not be seen as the opposite of work and so "a waste of time". Reeves and Read (2009) envision a scenario where multiplayer games can be used to redesign the work environment by making it more challenging and making workers more productive. They see games as a possible solution to the challenges facing work imposed by broad economic changes and the dramatic rise of information work.

Crawford (1984) in his book "The Art of Computer Game Design" suggests that games are the most ancient and time honoured vehicle for education. As Van Eck (2006) points out 'Lions do not learn to hunt through direct instruction but through modelling and play'.

Various authors (Beavis, O'Mara et al. 2012, De Freitas, Maharg 2011, Gee 2003, Prensky 2003, Squire 2011, Steinkuehler, Squire et al. 2012, Willett, Robinson et al. 2009) have stressed the power of digital games based learning. Gee (2003) derives a set of thirty six learning principles from his study of the complex, self-directed learning, game players undertake as they encounter and master a new game. He suggests that these principles could transform learning in schools both for teachers and more importantly for students.

Irrespectively of whether digital games should be considered for their inherent learning potential, digital games form an integral part of the lives of a lot of students and the minimum schools can do is help students understand them and exploit the students' interest in digital games to support learning.

According to Van Eck (2006) educators have adopted three approaches for integrating digital games in the learning process:

1. Serious games

2. Commercial off the shelf games

3. Provide opportunity for children to author their own games

I briefly outline each approach below:

1.1.1 Serious Games

Serious games are games written with an explicit and carefully thought-out educational purpose. These games are not intended to be played primarily for amusement although they can be, and generally are entertaining. Serious games are defined as "a mental contest, played with a computer in accordance with specific rules that uses entertainment to further government or corporate training, education, health, public policy, and strategic communication objectives" (Zyda 2005). The following list, by no means exhaustive, includes some of the popular serious games located in literature and is intended as an illustration of serious games:

- Re-mission: a game from Hope lab aimed at helping gamers better understand the disease of cancer so as to develop a positive attitude toward defeating it (Kato 2010).
- River City: a serious game which allows teams of middle school students to collaboratively solve problems with health present in a simulated 19th century city. The children interact with each other's "avatars," digital artefacts, computer-based "agents" acting as mentors, visual and auditory clues and colleagues in a virtual community of practice (Ketelhut, Clarke et al. 2010).
- Food Force: a serious game that introduces children to the logistical challenges of delivering food aid in a major humanitarian crisis. Food Force is set on a fictitious island called Sheylan torn apart by drought and war. Food Force invites children to complete six virtual missions that reflect real-life obstacles faced by United Nations World Food Programme in its emergency responses to crises around the world (Simões, Redondo et al. 2013) .
- Quest Atlantis: a 3D multi-user learning environment that utilizes Active Worlds to immerse children, ages 9–15, in meaningful inquiry tasks in the science domain (Barab, Thomas et al. 2005, Barab, Dodge et al. 2007) .

1.1.2 Commercial off the shelf games

Commercial off-the-shelf games are designed purely for entertainment and not for a purposely thought out educational purpose. This does not mean that these games cannot be used effectively in class. Williamson (2009) points out that games are as much of a resource for learning as reference books and television. Various titles such as Sim City, Zoo Tycoon, Roller Coaster Tycoon, Civilization and Age of Empires are used for business and economics simulations as well as historical re-enactments.

1.1.3 Making games

The final approach for integrating digital games in the learning process is to have students' author games from scratch. Van Eck (2006) notes that through this approach students develop problem solving skills whilst learning a programming language. Van Eck elaborates that this approach is time intensive and requires specialist skills by teachers. He concludes that this approach is unlikely to be used widely due to these constraints.

There have been various advancements in technology which have resulted in game authoring tools becoming available which do not necessarily require a long period of time for students or teachers to master. These tools can be used by children to author games without having to engage with the intricacies of a traditional computer programming language.

In the following list I outline a number of studies found in literature which deal with game creation by children to attain one or more of the following objectives:

- Enhancing creativity
- Switching from reading to writing digital games
- Improving engagement and motivation
- Enhancing problem solving skills
- Teaching a subject by building an educational game about it

1.1.3.1. Digital Game development to enhance creativity

A study carried out by Eow Yee Leng, Wan Zah Wan Ali, Rosnaini Mahmud, Roselan Baki (2010) focused on using an appreciative learning approach to the teaching of computer games development in order to enhance the creative perception of secondary school children. The study involving sixty nine Malaysian form one students aged 13–14, adopted a control group experimental design. Students' creative perception was assessed using Khatena-Torrance Creative Perception Inventory (KTCPI). This study concluded that students in the treatment group gained a significantly higher mean score than that exhibited by the control group.

In another research project Navarrette (2013) interviews twelve students whilst they create thematic games on social and educational topics. She notes that the students experienced positive opportunities for engaging with the creative thinking process whilst synthesising information to be included in the games created. She concludes that the creative thinking process involved in game creation provides learners with rich and enjoyable learning.

1.1.3.2. Switching from reading to writing games

Literacy has traditionally been described as the ability to read and write. However as Burn (2009) accentuates when it comes to game literacy the balance is skewed towards the reading skills. The three projects that follow try to rectify the balance in game literacy by empower students with writing as well as reading game literacy skills.

Making Games: Developing games authoring software for educational and creative use was a research project conducted between 2002 and 2006 funded by PACCIT-Link programme in the UK. This project developed pedagogic approaches and created the software product *Mission Maker*, to enable young people create their own computer games (Pelletier, Burn 2005).

The *Adventure Author* project led by the University of Edinburgh explores how young authors can be supported to create nonlinear stories with believable and intriguing plotlines and characters. Through this project a toolkit based on the commercial game Neverwinter Nights 2 was built specifically to allow children to author their own 3D games. The aim of the project is summarised in the following paragraph taken from the Adventure Author portal:

> Encouraging creativity is a major aspect of the modern school curriculum. Kids read books, so we encourage them to write stories, and illustrate them. They watch movies, so we teach them drama. They also play video games, by far the most interactive and engaging of such forms of entertainment. So why not let our young writers, actors and artists become designers too? (Robertson, Nicholson et al. n.d)

Robertson and Howells (2008) conducted an eight week exploratory study with a class of ten year olds using the Adventure Author toolkit in order to assess successful learning during the game authoring process. They concluded that whilst authoring games children displayed:

- motivation and enthusiasm for learning
- determination to reach a high standard of achievement
- independent and group learning
- linking and applying learning in new situation

Similar results to the ones outlined by Robertson and Howells (2008) are put forward in another research project (Carbonaro, Cutumisu et al. 2008). Carbonaro et al, use a tool called ScriptEase based on the commercial game Neverwinter Nights 2 to enable two classes of 15 to 16 year olds to author interactive game stories. They conclude that very little training was required for the students to author their own interactive stories and that factors including gender, programming experience, amount of time spent playing computer games or participating in online activities had little bearing on the quality of interactive stories.

1.1.3.3. Improving engagement and motivation

Owston, Wideman, Sinitskaya Ronda, & Brown (2009) use the web resource Education Games Central to allow a group of students aged between 10 and 11 years to construct electronic versions of popular board games. The web resource provides a series of popular games such as TicTacToe and Snakes and Ladders to which students need to add a list of questions and answers. The students can specify appropriate responses players receive when providing a correct or incorrect answer to the question. Although the level of game authoring in this experiment was less elaborate that that found in projects outlines in section 1.1.3.2 Owston et al. (2009) conclude that game development helped improve:

- student content retention
- ability to compare and contrast information presented
- utilize more and different kinds of research materials including digital resources
- editing skills
- insights into questioning skills

Vos, van der Meijden and Denessen (2011) used a web resource similar to the one used by Owston et al. (2009) to enable a group of 10-12 year old Dutch students to construct a game to master a number of Dutch proverbs. The tool (http://www.memoryspelen.nl/index.php) allowed students to drag pictures with proverbs next to provided meanings of the proverbs. Vos et al. (2011) conclude that constructing the game rather than playing it had a positive effect on student motivation to learn the proverbs.

1.1.3.4. Problem Solving

Problem solving is a thinking skill which is regarded as highly important. Notwithstanding children in schools do not get a lot of opportunities to solve complex problems with multiple possible solutions (Jonassen 2000, Mayer, Wittrock 2012). Instead problem solving is usually introduced in schools through problems which have one solution (Perkins 2013). This approach does not lead to students gaining skills to solve real life problems (Jonassen 2000).

There have been a series of studies that have looked at using game design as a context to teach higher order thinking skills (Akcaoglu, Koehler 2014, Denner, Werner et al. 2012, Ke 2008).

1.1.3.5. Teaching a subject by building an educational game

One of the approaches found in literature where game development is concerned is to make students engage with a subject by building a game (Ulicsak, Williamson 2010). This game is then used to teach the subject to fellow students. This approach was first used in the study by Kafai (1996) where fourth grade students in the US spent an hour a day for six months building a game to teach fractions. The students used the Logo programming language and engaged with a thorough understanding of fractions through the game creation exercise.

A similar approach was used in the study by Baytak (2009). In this study children aged between ten and eleven years learned about environmental issues by designing games that involved environmental concepts. These games, created using the Scratch programming language, where then presented to seven year old students.

Yang and Chang (2013) used a quasi-experimental setup where a group of students aged between thirteen and fourteen years were split into two groups. The teaching time for both groups was split in half with the first half dedicated to teaching topics from the biology syllabus using a traditional approach and the second half dedicated to teaching programming using Flash. The control group was taught Flash using a traditional approach whilst the experimental group was lead to create games about biology. Yang and Chang conclude that students participating in the experimental group demonstrated significant improvement in critical thinking skills and academic achievement.

1.2 An overview of this project

As discussed in the introduction of this chapter it is evident that gaming is an important activity for a large number of us including our students. Whilst playing and interacting with others through and about digital games, players build gaming capital. I share Carrington and Robinson's observation that an increasingly large number of students are obliged to leave an entire suite of competencies, practices and knowledge about digital technologies at the school gate (Carrington, Robinson 2009). Through this project student volunteers were given the possibility to learn how to create a digital game and express their creativity and knowledge about games. I also share Livingstone and Hope's opinion about the inclusion of programming in schools:

> Given that the new online world is being transformed by creative technology companies like Facebook, Twitter, Google and video games companies, it seems incredible that there is an absence of computer programming in schools. (Livingstone, Hope 2011, p.29)

I am not arguing that children in class will create the new Facebook or Twitter today but exposing them to programming is exposing them to a new means of expression that might help them shape their identity.

In this research project a group of fourteen eleven year old students volunteered to join an after school game authoring workshop. The group was composed of boys and girls attending a Maltese co-ed school. The pedagogy to introduce programming was based on existing practices found in literature that are used to foster creativity. Unlike previous studies (Baytak 2009, Kafai 1995, Kafai 1996) throughout the workshop the children were left at liberty to develop a game they wanted to create. They were not given any theme on which to base their game. This project is similar to other projects (Carbonaro, Cutumisu et al. 2008, Robertson, Howells 2008) found in literature since it empowered the students with writing skills to complement their existing game literacy skills. A major difference from these studies is that instead of using a toolkit based on a commercial game, the Scratch programming language was used. The use of a toolkit based on a commercial game limits the students to using only characters found in the original commercial game. Using a programming language allows the choice of characters to be decided upon by the game authors (Peppler, Kafai 2007).

The guiding research questions for this project are:

1. What processes do eleven year olds follow to create digital games? Through this research project I would like to identify the stages that children go through whilst creating their game. The game will be the creation of the children and I would like to compare the game making process undergone by the children with the process of creativity identified in literature.

2. What benefits does creating a digital game have from a gaming literacy perspective?
   Playing digital games forms an integral part of the lives of a number of students. By participating in this research children will create a game. Through this research question I would like to identify the benefits, if any, that the children will gain through their participation from a gaming literacy point of view.

1.2.1 My entry into research

In the introduction to this chapter I listed a series of digital game creation competitions organised locally. Since the start of these competitions I have actively participated in two ways. In 2010 I led two teams of pre-service educators in designing and submitting two games to the *GameZing* competition. One of these games *Math Planet World* was awarded the runners up price (MSTE 2011). In a bid to shore up the participation levels in other competitions aimed at school children I regularly volunteer to lead workshops for teachers participating in the competitions. These workshops delve into approaches that can be used to create games using the different software packages allowed in the various competitions such as Microsoft XNA, Kodu, Scratch and Story Telling Alice.

As a young child digital games introduced me to the world of programming. The personal computer fad hit Malta in the middle of the 1980s and most of my peers at secondary school had a Spectrum, an Atari or a Commodore 64 - popular personal computers of the time. Playing the popular games Space Invaders and Pacman on an Atari 800XL intrigued me to try making a game myself. The Atari 800XL was equipped with an integrated keyboard and a tape drive through which we used to load games from an audio tape. It used the television screen as a monitor. Once the machine was switched on the BASIC programming language was loaded automatically. In a bid to learn how to make games I used to try copying program listings from magazines. The first game I copied asked the player to take three tries in guessing a number between one and ten generated randomly by the machine. The game contained no graphics and was pretty simplistic however it was the start of a journey that led me into furthering my studies as a teacher of Computing and later to conduct training to people working in the ICT industry.

I hope that through this research I can introduce students to the exciting world of software development similarly to how authoring a very simple text based game introduced me to the world of software development.

1.3 A guiding learning theory

In this section I review learning theories which guide me throughout the project. I carried out this review for two main reasons. Firstly I plan to use these theories to inspire the pedagogic approach to use during the game authoring workshop with children. Secondly I share Ackermann's assertion that the beliefs about the way children learn and hence the way we shape our interactions with them are rooted in our convictions on what it means to be knowledgeable, intelligent, experienced, and what it takes to become so. "Whether implicit or explicitly stated, these convictions drive our attitudes and practices as educators, parents, teachers, and researchers" (Ackermann 2001). As an educator, parent and teacher I was always convinced that children learn best by exploring and doing rather than just by being told. This same innate believe was invariably present during this research too and so I consider it important to state this believe and to start this review by looking into constructivism.

1.3.1 Constructivist Epistemology

The constructivist epistemology looks at knowledge as constructed by the individuals. The main tenant of this process is that knowledge is constructed by the learner through the experience of interacting with the environment around him. The learning process is an individual matter such that if two individuals are exposed to the same learning experience they would build an individual version of reality based on their prior knowledge, understanding and experience.

Although the beginning of the constructivist approach is attributed to Jean Piaget (1896-1980) the central notion of constructivist theory, that the individuals construct the world in which they live and that thinking is based on what is observed and experienced, can be found in the works of Gautama Buddha (560-477 BC), Heraclitus (535-474BC) and Loa Tzu, a contemporary of Buddha (Pritchard, Woollard 2010).

Piaget is known for his genetic epistemology which looks at how knowledge is developed. For Piaget there are three central processes of development which are unconsciously put into action whenever a person encounters information through one of the senses. These processes are *assimilation*, *accommodation* and *equilibrium*. Assimilation occurs when a child perceives new objects or events. This process contests the internal cognitive structures which Piaget calls the schemas, destabilising them. The schemas are changed to accommodate the new experience bringing the schemas back into equilibrium.

The contribution by Piaget is important because it highlights that knowledge is not a commodity to be transmitted, something to be delivered from one end, encoded and restored in another. Instead it is something which must be engaged with and internalised. In his epistemology Piaget presents the learning journey as a personal journey without giving due credit to the influence of the social environment.

1.3.2 Social Learning

Although Vygotsky embraces constructivism, he considers social interaction as a fundamental aspect of successful cognitive and intellectual growth. Vygotsky introduced the notion of zone of proximal development (ZPD). The ZPD is the distance between what children can do by themselves and the next learning that they can be helped to achieve with competent assistance (Vygotsky 1978). By highlighting the help that the learners can be given to internalise the knowledge which is close to, but beyond, the learner's current level of understanding Vygotsky highlights the importance of social learning.

Interaction with more capable peers through social collaboration allows the child to progress and for learning to take place. The "more capable other" provides the scaffolds so that the learner can be assisted to accomplish the tasks that he or she could not accomplish otherwise, thus helping the learner through the ZPD (Bransford, Brown et al. 1999).

1.3.3 Exploratory learning

Bruner is another key figure who embraced constructivism and the notion that learning is journey of discovery best experienced by exploring and manipulating objects. As a result, students may be more likely to remember concepts and knowledge discovered on their own. "To instruct someone... is not a matter of getting him to commit results to mind. Rather, it is to teach him to participate in the process that makes possible the establishment of knowledge" (Bruner 1966, p.72). Bruner also accentuates the importance of social learning. In his influential book The Culture of Education he states that 'culture shapes the mind... it provides us with the toolkit by which we construct not only our worlds but our very conception of ourselves and our powers' (Bruner 1966, x).

1.3.4 Constructionism

The term Constructionism knows its origin to Seymour Papert. Papert worked closely with Jean Piaget at the University of Geneva in Switzerland during the late 1950s and early 1960s. Papert based his theory of Constructionism on the constructivist epistemology.

> "Constructionism—the N word as opposed to the V word— shares contructivism's view of learning as "building knowledge structures" through progressive internalization of actions… It then adds the idea

> that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe (Harel, Papert 1991, p.1)

Constructionism maintains that learning can happen most effectively when people are also active in making real world tangible objects. In the forward to the book "Mindstorms: Children, Computers, and Powerful Ideas" Papert (1980) illustrates how his fascination as a child for model cars and how they work, led him to experiment with gears. Gears turned out to be a transitional object (p. viii) which helped him internalise mathematics at a later stage in life. Papert went on to found the Life Long Kindergarden at MIT and to come up with Logo, a programming language for children.

Constructionism views young learners as mini-scientists and inventors who develop their knowledge of how the world works by building theories and experimenting. As Kafai elaborates learning happens best "when building external and shareable artefacts for the use of others" (Kafai 1996, p.72). During the creation process students are in constant dialogue with their ideas and the ideas of the intented users. The learning which is occuring whilst the artifact is being created is happening in-situ.

In this respect Constuctionism links well with situated learning theory as promoted by Lave and Wenger (1991) where learning occurs through the participation in communities of practice made of people who share an interest.

1.3.5 Learning in a world of constant change

The world we live in is constantly changing. One might argue that this has always been the case and that the world has always been wrapped in a flux of change; however this change has increased in momentum accelerated by the affordance and pervasiveness of technology. Up to some decades ago the telephone system in Malta was not widely spread. There were instances where the only phone available to a community was installed in the local police station and that people who needed to communicate with relatives overseas had to make use of the shared service. Today we have voice over IP capabilities which allow us to communicate with others using both voice and video for free as long as both parties have access to a mobile device with internet connectivity.

In a world which is constantly changing we need to focus more on the way we learn. Thomas and Seely Brown reflect on the work of Polanyi who suggests that knowledge is always made up of an explicit and a tacit dimension. Although explicit knowledge can be transferred by verbalising the knowledge or by writing it down, tacit knowledge is not easily transferable. One could teach the semantics of a language (explicit knowledge) but to truly learn a language the student needs to practice speaking and observe others as they speak the language.

As Thomas and Seely Brown (2011b) observe "In a world where things are constantly changing, focusing exclusively on the explicit dimension is no longer a viable mode of education" (p. 76). Thomas and Seely Brown suggest that learning in a world of constant change is based on three central elements: knowing, making and playing or what they call Homo Sapiens (human as knower), Homo Faber (human as maker) and Homo Ludens (human as player). For Thomas and Seely Brown it is the interaction of these three elements which is important for learning in an ever changing world.

1.3.5.1. Homo Sapiens

Homo Sapiens is about the acquiring of knowledge. In the context of rapid changes brought about by the networked world Thomas and Seely Brown outline three senses in which learning happens.

- **Learning about.** Learning about is the most basic sense and is used when we acquire information which is consistent and stable, unlikely to change over time.
- **Learning to be.** This sense requires engagement within a community of practice and allows one to participate and learn how to learn and shape practices within the community.
- **Learning as becoming.** This sense sees learning as a process which is always in flux and which changes according to the context. The focus for this sense is the context rather than the content as in the learning about sense. As Thomas and Seely Brown outline "The end result is not knowledge per se, but a new set of tools for looking at the world and engaging in inquiry, hopefully productive inquiry" (Thomas, Brown 2011a, p.7).

1.3.5.2. Homo Faber

Homo Faber stresses the ability of learns to learn by creating. As new media evolves, it is providing opportunities for learners to express themselves creatively in a context of peer feedback. Such peer feedback can be seen in popular sites such as YouTube. YouTube is full of videos created by remixing content produced by other members. Such sites are not just about video sharing. YouTube provides a community where viewers can comment on the videos posted by others. Comments posted by other viewers can land a video in the most discussed list which is featured in the YouTube section resulting in a dramatic increase in viewership. Registered members of YouTube can also rate videos. The user ratings are collated by YouTube which then features the highest rated videos on the videos page.

Thomas and Seely Brown link the concept of Homo Faber to the concept of “indwelling” as outlined by Polanyi. Indwelling is “the process of immersing oneself in the particulars of a subsidiary awareness by means of embodied activity until these particulars come together as a meaningful whole as an interactive act” (Gill 2000, p.52). To know something deeply, one must engage with the knowledge. Creating an artefact about a topic is a journey towards understanding the explicit knowledge about the topic by engaging with the tacit dimension.

1.3.5.3. Homo Ludens

Thomas and Seely Brown look at play in digital games as ways of participating in complicated negotiations of meaning, interaction and competition for entertainment and meaning making. Play is seen as a powerful learning environment which allows the players to engage in a process of experimentation and is conducive to opening up the imagination. Thomas and Seely Brown idealise play as structures of learning which are ideally suited to the notion of flux and becoming since play does not provide a linear look at knowledge where we are presented with what we know and what we want to achieve, instead play presents an environment where one has to experiment, fail and continue playing in a bid to surmount the challenge created in the game. As Thomas and Seely Brown point out “In play, learning is not driven by a logical calculus but, instead, by a more lateral, imaginative thinking and feeling. In sum, playing, like making and knowing, derives its power from the tacit dimension" (Thomas, Brown 2011a, p. 9)

1.3.6 Designing the workshop

In this section I look at how the learning theories were used during the design of the game making workshop. Following on the mantra of constructivism that knowledge is not a commodity to be transmitted, something to be delivered from one end, encoded and restored in another but rather something to be experienced, whilst designing the workshop I ensured that the sessions took the form of guided explorations. The teacher acted as a guide and facilitator and at times a meddler in the middle rather than as a font of knowledge. This will not mean that the students will be left to their own devices since as Thomas and Seely Brown conclude the Homo Sapiens element is important ; however the students were given the time to explore the game authoring environment and try things out.

Taking a cue from scaffolding and exploratory learning I started the workshop with a working game which the children explore and tinker with. The game was deliberately incomplete and contained features which could be improved. Whilst drawing on their experience as gamers, the students came up with ways of extending the game. The teacher guided the students to add the missing functionality by exploring how the features present in the original game were constructed.

Another key feature of the game building workshop that was inspired from the learning theories was the interactivity between the children following the workshop. The workshop was designed to allow children to work in groups, if they wished to and to consult each other during the game building process.

1.4 Conclusion

This chapter provided a discussion of the motivation of this research project and a list of guiding research questions. I also discussed the guiding leaning theories which are used during the design of the game making workshop. The gaming workshop is discussed in more detail in section 2.5 (page 82).

The dissertation is structured as follows: Chapter 2 provides a synthesis of literature pertaining to play, creativity and the skills required to make a game. Chapter 3 presents the design of my research, the case study methodology adopted and how the data was analysed. Chapter 4 and 5 delve into the meaning derived from this study whilst Chapter 6 provides a synthesis of my interpretations. Chapter 7 provides a discussion of the implications for learning and effective integration of technology in education, and for further research.

2. Literature Review

2.1 Introduction

One of the most important developments in the local Maltese education scene for the past few years was the launch of the draft National Curriculum Framework (NCF) on the 18th of May 2011 by the Directorate for Quality and Standards in Education (DQSE) within the Ministry of Education, Employment and the Family. The NCF proposes a paradigm shift away from a prescriptive curriculum towards a framework which allows for some degree of internal flexibility and a break from independent standalone subjects to wider learning areas/ clusters that form the entitlement of all learners (DQSE 2011).

The NCF presents eight learning areas together with an outline of the essential knowledge, skills and attitudes that learners are expected to acquire in each learning area. The learning areas identified in the NCF are:

- Languages (Maltese and English; Foreign Languages)
- Mathematics
- Science and Technology
- Health and Physical Education
- Religious and Ethics Education
- Humanities Education (History, Geography)
- Education for Democracy
- Visual and Performing Arts

Of special interest for this research is the inclusion in this curriculum framework of Technology with Science. According to the NCF design and technology is about combining practical and technological skills with creative thinking to make useful products. The NCF stresses the importance of ‘design and make’ tasks in Design and Technology where learners work through “a creative process” (Ministry of Education and Employment 2012, p.35).

The NCF also presents six cross curricular themes that teachers should embed in the different learning areas and that will provide connecting strands across the learning areas.

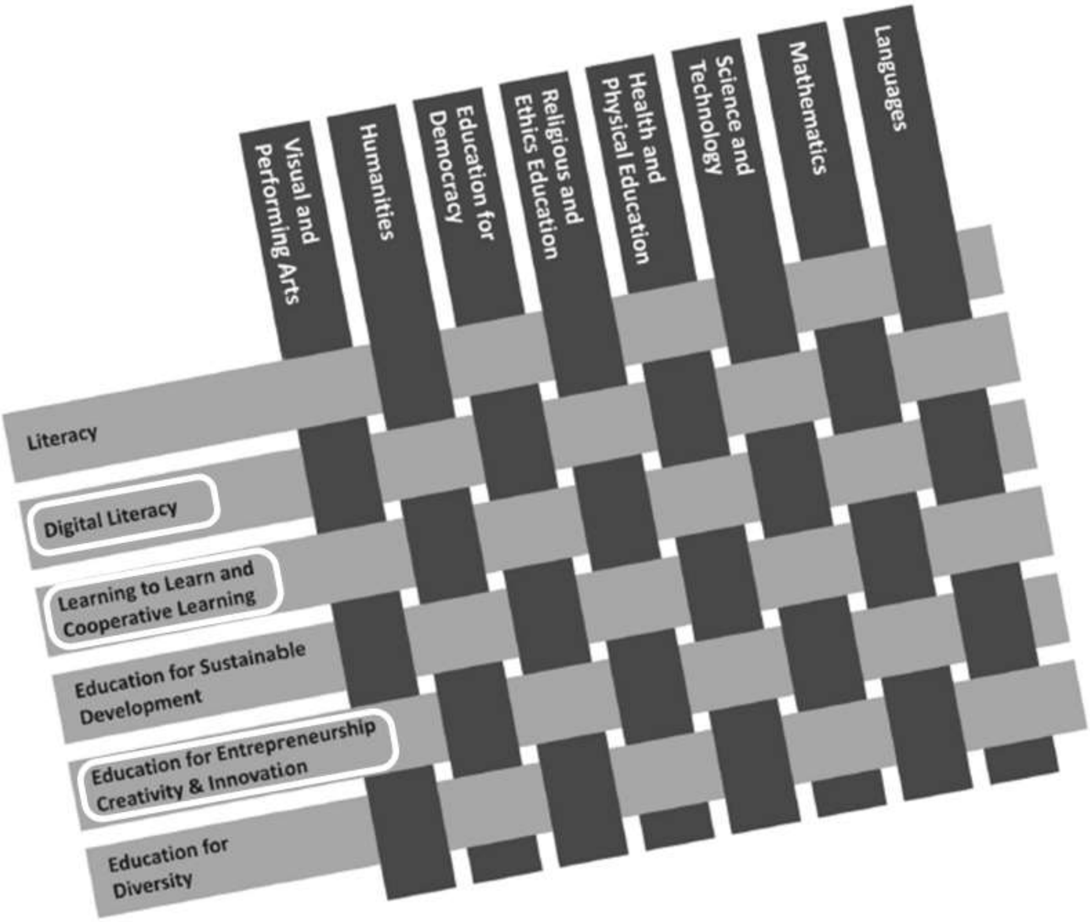

Visual and Performing Arts
Humanities
Education for Democracy
Religious and Ethics Education
Health and Physical Education
Science and Technology
Mathematics
Languages
Literacy
Digital Literacy
Learning to Learn and Cooperative Learning
Education for Sustainable Development
Education for Entrepreneurship Creativity & Innovation
Education for Diversity

Figure 2.1-1 Digital literacy, learning to learn and cooperative learning and Education for Entrepreneurship Creativity and Innovation as cross curricular themes. Adapted from Ministry of Education and Employment (2012, p.39)

The cross curricular themes include the themes of Digital Literacy, Learning to Learn and Cooperative Learning and Education for Entrepreneurship, Creativity and Innovation. The Digital Literacy theme outlined in the NCF is a skills oriented definition of digital literacy. Digital Literacy is set to be organised around four overlapping strands: data sources and manipulation; information communication and presentation; programmed control; and social, ethical and personal aspects. Creativity is seen as an agent for change contributing to economic prosperity of society and the well-being of the individual. The NCF promotes a whole-school approach that promotes a climate conducive to creativity. Creativity is seen as a source of flexibility, adaptability and provides the capacity to innovate. The importance of cooperative learning and learning to learn is also given a central role in the NCF by devoting a cross curricular theme to the subject. The NCF supports the fostering of a collaborative environment where learners work together in groups with and without teacher direction. This environment is idealised as providing possibilities for young people to actively engage with peers and in so doing learn several processes such as solving problems and creating products together. Such an approach also creates a context for social development amongst children as they become exposed to different viewpoints and personalities.

Given the focus on creativity and innovation and digital literacy as a cross curricular theme and the inclusion of science and technology as a learning area, I have focused on the creative aspect of game authoring by children in order to propose game development as a possible avenue for creativity and innovation by children in schools. The gaming workshop provided a collaborative environment where the children could collaborate with the whole group whilst building the games they designed. In so doing they put into practice co-operative learning.

Game authoring could be the medium we use to weave together the themes of creativity and innovation with science and technology. Depending on the game theme adopted by the children, digital game authoring could also weave creativity and innovation with the learning areas of languages, mathematics, health and physical education, religious and ethics education, education for democracy, humanities and visual and performing arts.

2.1.1 Gaming amongst the Maltese children

The pervasiveness of digital and video gaming among the Maltese population has been documented in a research project which I was involved in during the summer of 2012 (Busuttil, Camilleri et al. 2014). This quantitative research project targeted a representative sample of the Maltese population aged between three and fifty four and addressed the following goals:

- To establish the game playing patterns amongst the Maltese people aged between three and fifty four years
- To provide a measurement of tendencies of a representative sample of the Maltese population between seven and fifty four
- To explore age and gender influences on digital and video game play

In the review of this research I focus on the age group seven to twelve year olds since this was the age group of the students who participated in the game making workshop for my PhD project. As expected playing digital games turned out to be a popular activity for children in this age group. Almost nine out of ten respondents (88.7%) claimed that they play digital games. This pervasiveness of digital game play was more apparent in the boys with almost all boys (95.5%) claiming to play digital games, with eight out of ten girls (83.6%) claiming to play digital games.

The fact that the act of playing digital games forms an important aspect of the life of children in this age group can be seen in the frequency that the children play as well as in the amount of time the children spend playing. Nearly six out of ten children (59.7%) said they play at least once a day with 24% playing several times a day. Boys tend to play more frequently with 38% of the boys playing once a day and 26% playing several times a day compared to 33% and 22% of the girls respectively. Boys tend to spend more time playing digital games than girls. Seven out of ten boys (70.2%) admitted to playing at least an hour a day whilst six out of ten girls (60.8%) play for the same amount of time.

Boys	Girls
Racing (such as NASCAR, Mario Kart, Burnout) 16%,	Puzzle (such as Bejeweled, Tetris, Solitaire) 17.1%,
Sports (such as Maden, FIFA, Tony Hawk) 14.1%,	Racing(such as NASCAR, Mario Kart, Burnout) 15.5%
Adventure (such as Legend of Zelda, Tomb Raider) 13.5%,	Online Social Games (such as Farmville, Cityville) 13.3%
Action (such as Call of Duty, Grand Theft Auto, Devil May Cry) 13.2%	Adventure (such as Legend of Zelda, Tomb Raider) 10.4%

Table 1: The game children play: gender differences

Maltese children participating in this research project associated playing computer games with fun (33.5%) and relaxation (22.7%) whilst they also considered games as exciting (14.5%) and challenging (11.5%). They preferred playing alone (41%) or in a group (37.9%) with racing games being the most popular with boys, whilst puzzle games being the more popular with girls (see Table 1).

Children in this age group that participated in this research project tended to play on a desktop or laptop computer (26.5%) closely followed up by game consoles (22.9%), portable game devices (18.5%) and mobile phones (14.6%).

These statistics are important because not only do they show, that as suspected children consider playing digital games as important, but they also show that this importance is irrespective of gender differences. Both boys and girls enjoy playing games even if they enjoy playing different games. The research also shows that, when this research was being conducted, the computer rather than other devices was the most popular gaming device. It would follow that when conducting a project on game creation the same device would be used.

2.1.2 Conclusion

Although one cannot claim that all children play digital games, statistics quoted earlier indicate that games are integral to the lives of a lot of young people. Whilst digital games are played in leisure time, digital games do not normally feature in the schools. I am interested in trying to bridge the gap by researching the processes children go through whilst designing and authoring digital games. I am also interested in discussing whether the processes children go through whilst authoring digital games have learning benefits and whether such an approach towards learning can be integrated in the school curriculum.

Digital games are enacted through play. In the next section I review some of the literature about play focusing on concepts of gaming literacy, gaming capital and a framework that can be used to work with digital games in the school environment.

2.2 Play

Johan Huizinga, in his seminal book "*Homo Ludens: A Study of the Play Element in Culture*" (1932), defines play as:

> a voluntary activity or occupation executed within certain fixed limits of time and place, according to rules freely accepted as absolutely binding, having its aim in itself and accompanied by a feeling of tension, joy and the consciousness that it is 'different' from 'ordinary life' (Huizinga 1955, p.28).

According to Anchor (1978) whilst composing his theory of play Huizinga was influenced by other modern philosophers such as Hegel, Marx, Nietzsche, Heidegger and Schiller and their views of play. Notwithstanding, Huizinga was one of the first to attempt an exact definition of play and of the ways in which play infuses and manifests itself in culture.

In summary according to Huizinga play is an activity that:

- has boundaries of time and place
- is entered into willingly by the players
- is defined by rules
- allows actions which are not permitted in real life
- brings about feelings of tension and joy

Caillois (1958) builds on the work of Huizinga and adds that play activity is "unproductive" (p.5) and that the outcome of play is always "uncertain" (p.7). For Caillois play differs from work or art since no wealth or goods are created whilst playing and hence is unproductive. Even when considering games of chance such as gambling and lotteries, no wealth or goods are created since at best the winnings of a player will equate to the sum of loses of the other players. Since the actions of players are free as long as they are within the rules of the game the outcome of a play activity is always uncertain.

Huizinga's definition of play hints at the creation of a new reality created throughout the game play, what he calls the "consciousness that [the game activity] is 'different' from 'ordinary life'". Playing a game in this view means setting oneself in a different world. Games create a "magic circle" which separates the game from the outside world and whatever happens in the game has no effect on what happens outside the circle. The stance that whatever happens in the magic circle has no effect on the outside world might have been the effect of Huizinga's effort to protect play from what Huizinga saw as the destructive influence of the Protestant work ethic and a Western culture that valued seriousness over fun (Egenfeldt-Nielsen, Smith et al. 2013). This separation of the game world from the real world has been challenged in literature as I discuss shortly.

This feeling of alternate reality created by game play is also explored by Taylor (2006) in her book ‘Play between worlds’ where she explores the online/offline experience of players playing a massively multiplayer online game Everquest and how the boundaries of these worlds blur into each other. At a “Fan Faire”, a convention for the players of Everquest the game/non-game spaces, avatars/real world identities are blurred when the players meet each other face to face and introduce themselves using the game character name, guild and server. Taylor explains that players who never met each other face to face, virtually complete strangers, quickly identified with the shared server name and guild to form groups. “Quickly people are chanting server names as well as playfully taunting and teasing each other across table” (Taylor 2006, p.3). Attending the meeting also affected the online world since players who attended the faire and joined in the activities that ensued, formed playing groups in the virtual world as well. The experiences recounted by Taylor as well as the work of other authors such as Steinkuehler (2006), Consalvo (2009) and Malaby (2007) are at odds with Huizinga’s magic circle. Playing games has real world consequences. Games require time to play and tend to affect our moods. Games can be seen as communication media by branding certain products in our minds. Games have also real world financial implications since not only are games bought with real money but it is becoming common practice to find online game accounts on sale on popular online trading websites such as eBay.

2.2.1 The diversity of play

Sutton-Smith (1997) in his influential book ‘The Ambiguity of Play’ observes that there are different kinds of play. Sutton-Smith builds on research by Betcher (1987) , Caughey (1984), McCannell (1976), Spacks (1986) and Stephenson (1967) in order to identify play in most of our daily activities; outlining how play is pervasive in our lives. Although most of the activities are not called play but identified by other names such as entertainment, recreation, pastime and hobbies, in reality these activities all contain elements of play.

Private	Mind or subjective play	dreams, daydreams, fantasy, imagination, Dungeons and Dragons, playing with metaphors.
	Solitary play	hobbies, collections, listening to music, art projects, pets, reading, yoga, collecting and building cars, Civil War re-enactments, bird watching, crosswords.
	Playful behaviours	playing tricks, playing around, playing up to someone, playing a part, putting something into play, playing fair, playing by the rules.
	Informal social play	joking, parties, travel, leisure, dancing, getting laid, potlucks, malls, babysitting, creative anachronism, intimacy, bars and taverns, amusement parks.
	Vicarious audience	television, films, cartoons, spectator sports, theatre, jazz, rock music, parades, comic books, Renaissance festivals, museums.
	Performance play	playing the piano, playing music, being a play actor, playing the fishes, playing the horses, play voices, playhouses
	Celebrations and festivals	birthdays, Christmas, Easter, Mother's Day, Halloween, gifting, banquets, balls, weddings, carnivals, balls, Mardi Gras
	Contests (games and sports)	athletics, gambling, casinos, lotteries, pool, golf, parlor games, drinking, the Olympics, cockfights, poker, chance, board games, card games
Public	Risky or deep play	caving, hang gliding, kayaking, bungee jumping, skateboarding, windsurfing

Table 2: Play Categories adapted from Sutton-Smith (1997, p.4)

As an example the activity of watching television is equated to playing since the player chooses the station to view, identifies with a character on screen and decides when to stop playing by changing channel. Even watching the news is seen as having elements of playing since the "news" is not the real thing but an account from a studio with graphic backdrops.

Smith (1997, p.4) lists nine categories of play ranging from play which is mostly private to play which is mostly public. The list of categories together with example activities for each category is listed in Table 2.

2.2.2 Looking at play from different angles

Sutton-Smith follows up the different categories of play by looking into how different professions look at play. He points out that biologists, psychologists, educators and sociologists tend to view play as adaptive and contributing to growth, development and socialisation whilst anthropologists are more interested in links between play and rituals. In order to appease the different scholarly views of play, Sutton-Smith outlines seven different cultural ways of thought or what he calls rhetorics. The rhetorics are divided into two groups, the ancient rhetorics which advocate collectively held community values and the modern rhetorics which are rooted in individual experiences. In this research I only look at the 'play as progress' rhetoric since this rhetoric is important for the view of play in education. I discuss this rhetoric next.

2.2.3 The importance of play for education – play as progress

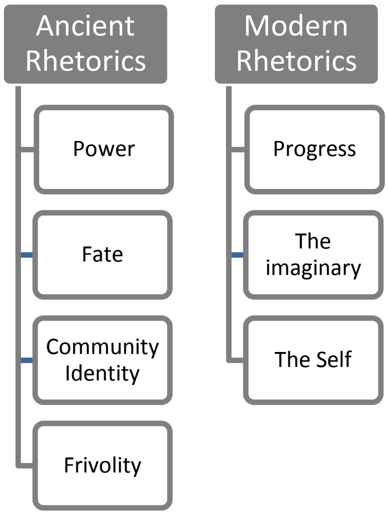


Figure 2.2-1 Rhetorics of play **Sutton-Smith (1997, p.10)**

Jean Piaget's work (1962) underlines the importance of play in children's development. His concept of play differs from that of Huizinga in that Piaget distinguishes between play with rules and play without rules. For Huizinga rules are central to any play.

In his book 'Play dreams and imitation in Childhood' Piaget (1962) argued that children's play evolved in three stages which could be linked to his four stages of intellectual development. Piaget distinguishes between practice or mastery play which occurs in the sensory motor stage of cognitive development, symbolic games which occur at the pre-operational stage of behaviour and finally games with rules. For Piaget children are only able to engage in games with rules when they reach the phase of concrete operations.

The importance of games for Piaget has to be seen in the light of Piaget's theory of learning. For Piaget knowledge is not information to be delivered at one end, and encoded, memorized, retrieved, and applied at the other end. Instead, knowledge is experience that is acquired through interaction with the world, people and things (Ackermann 2001).

Another influential figure in the area of play is Piaget's contemporary Les Vygotsky. For Vygotsky, play is a way of increasing a child's development and skill because it creates a Zone of Proximal Development (Vygotsky 1978). The Zone of Proximal development is the area between the level of performance a child can achieve when working independently and a higher level of performance that is possible when working under the guidance or direction of more skilled adults or peers (Wertsch, Tulviste 1992).

Both Piaget and Vygotsky subscribe to the constructivist epistemology, that argues that humans generate knowledge and meaning from an interaction between their experiences and their ideas.

Learners construct mental models to understand the world around them and alter these models based on their experiences. The learning process is viewed as a reconstruction rather than a transmission of knowledge.

The rhetoric of play as progress supports the notion that children adapt and develop through their play. This rhetoric does not apply to adults since it focuses on play as development rather than enjoyment. Sutton-Smith questions this rhetoric since according to him progress through play is often assumed rather than demonstrated. He states that "the evidence does not seem to show very clear causal relationships although it would be surprising if they did not share and transfer skills back and forth" (Sutton-Smith 1997, p.207)

Sutton-Smith also comments that seeing play as progress in children reflects educators' predisposition to perceive playful imitation as a form of children's socialisation rather than enjoyment.

Notwithstanding the criticism raised by Sutton-Smith on the view of play as progress in education one cannot claim that play hinders education in any way. I now move to discuss digital games as manifestations of play activity.

2.2.4 Defining traits of digital games

In the discussion up to now I have looked at play and games in general. The work of Huizinga and Caillois predate the computer and so when discussing play they were not referring to digital games. Even though, as outlined in section 2.2, the work of Huizinga and Caillois has come under criticism when applied to digital games, their work remains influential. In this section I focus on what the literature says about properties of digital games. Since as part of this project children were asked to design and create their own digital game I explore what, according to literature, are the defining traits of digital games in a bid to then use these characteristics to discuss the work of the children in my project.

One of the first writers to systematically address digital games was game designer Chris Crawford (Egenfeldt-Nielsen, Smith et al. 2013). Crawford (1984) in his pioneering work did not provide a one line definition of digital games but instead listed four features which he considered to be common to all video games: representation, interaction, conflict and safety.

Representation: Crawford states that “a game is a closed formal system that subjectively represents a subset of reality” (Crawford 1984, p. 3). A game is a representation of a deliberately simplified and subjective reality. Reality does not necessarily mean real-life situations but a game is specifically shaped to trigger the player’s fantasy and make the game physiologically real. Stressing that games should target subjective realities, Crawford distinguishes between a game and a simulation and stresses that when authoring a game a game designer deliberately simplifies representation of reality to focus the player’s attention on those factors the game designer judges to be important.

Crawford again makes reference to Huizinga’s magic circle when remarking that a game is a closed formal system. The game has formal explicit rules which are enforced by the game system – a series of parts which interact with each other to make the game.

Interaction: The second feature of digital games for Crawford is the ability of the game player to interact with the game world and in turn affect it. Interaction is important because it injects a social and interpersonal element into the game and transforms the game from a purely technical activity to a personal one and thus enhances the player's engagement.

Conflict: A game has a goal which the player must reach by overcoming obstacles. Hence for Crawford conflict tends to be an intrinsic element of games. Conflict can be direct or indirect, violent or nonviolent, but it is always present in every game.

Safety: Although conflict is present in every game playing games is a safe activity as playing games does not carry the same consequences as conflict in the real world. Crawford's notion of "safety" resemble Huizinga's "magic circle", however Crawford does acknowledge that playing digital games has consequences on the real world.

The four features outlined by Crawford are also present under different forms in more recent digital game definitions put forward by other researchers such as Salen and Zimmerman (2003), Juul (2003), Oxlan (2004) and Whitton (2010) .

Salen and Zimmerman define a digital game as being "a system in which players engage in an artificial conflict, defined by rules that result in a quantifiable outcome" (Salen, Zimmerman 2003, p. 80). Looking at Salen's and Zimmerman's definition the following terms stand out:

System: a game is a system made up of a series of parts which interact with each other to make the game.

Players: one or more players interact with the system to experience playing the game.

Artificial: The game is distinct from the real world and resemble Huizinga's magic circle.

Conflict: All games embody some sort of contest of powers. This conflict could be a solo conflict against the machine or it could be against other players as in multiplayer games.

Rules: provide the structure out of which play emerges by delimiting what players can and cannot do.

Quantifiable outcome: Every game has a goal to achieve. At the end of the game the player has either won or lost or achieved some sort of score.

Juul (2003) builds on the work of Avedon & Sutton (1981), Caillois (1958), Crawford (1984), Huizinga (1955), Kelley (1988), Salen & Zimmerman (2003) and Suits (1978) to define a game as follows:

> A game is a rule-based formal system with a variable and quantifiable outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels attached to the outcome, and the consequences of the activity are optional and negotiable (Juul 2003).

Juul's definition is more concerned with the player. Rather than stating that the player just takes part in the game, Juul focuses on the player by noting that the player exerts effort to play the game and win it. The player also feels attached to the outcome of the game, feeling sad when losing the game and happy when winning. Juul also gives more importance the outcome of the game by building on Salen and Zimmerman's "quantifiable outcome" to state that the outcome can be variable that is provide different possible outcomes and has negotiable consequences that is the game could have real-life consequences. These consequences depend on the kind of play, the location where play takes place and the character of the person playing.

Another influential figure in the study of games and literacy is James Paul Gee. Gee(2008) distinguishes between what he calls games, the in-game design of the software in the box , and Games (big G games) which is the social setting into which the game is placed and all the interactions that go on around the game. Gee (2012) defines a digital game as a play-based, well-designed, problem-solving experience meant to create motivation, engagement, and often creativity.

Whitton (2010) puts forward what she describes as a more open definition of a digital game made up of ten characteristics:

Competition: The goal is to achieve an outcome that is superior to others

Challenge: Tasks require effort and are non-trivial

Exploration: There is a context-sensitive environment that can be investigated

Fantasy: Existence of a make believe environment, characters or narrative

Goals: There are explicit aims and objectives

Interaction: An action will change the state of play and generate feedback

Outcomes: There are measurable results from game play (e.g. scoring)

People: Other individuals take part

Rules: The activity is bounded by artificial constraints

Safety: The activity has no consequence in the real world

McGonigal (2012) defines a game by using four defining traits: A goal, rules, feedback system and voluntary participation.

A goal: The goal is the quantifiable outcome the players aim to achieve. The goal gives the players a sense of purpose.

Rules: Rules place limitations on how the players can achieve the goal. Rules force the players to think creatively and use strategic thinking to achieve the goal.

Feedback system: The feedback system notifies the players how close they are at achieving the goal of the game. The feedback system can take various forms from scores and levels to progress bars. Feedback systems serve as a motivational force to compel the players to keep playing.

Voluntary participation: The players playing the game accept the rules of the game and are aware of the goal and the feedback systems that guide them to achieve the goal. Voluntary participation ensures that although playing the game might result in hard work, the experience is a safe and pleasurable experience.

McGonigal's definition of a digital game has a lot in common with the definition by Salen and Zimmerman with both definitions stressing the importance of a goal (quantifiable outcome in Salen and Zimmermann's definition), rules and participation. By stressing the importance of feedback systems in digital games, McGonigal gives more importance to the mechanics of the game rather than highlighting the properties of games such as conflict and the fact that games are enacted in a 'magic circle'.

The purpose of this review of definitions of games was to build a list of properties that describe the term ‘digital game’ in view of analysing the games created by the children based on these features. Although all features outlined in the various definitions discussed above are important, not all features deal with the structure of the game. The fact that games are meant to be well-designed, problem-solving experiences meant to create motivation, engagement, and often creativity is important but does not provide a tangible feature that can be used to describe a game created. These features describe the gaming process rather than the game itself. The same goes for other features listed above such as voluntary participation, safety, interaction and the fact that the player is attached to the outcome. Hence when describing the games created by the children I will look at the tangible characteristics present in most of the definitions analysed:

Goal: What goal does the player strive to achieve in this game?

Rules: What are the rules that govern this game?

Feedback System: What feedback mechanisms are employed in this game?

2.2.5 Building capital about and through games

The term ‘gaming capital’ was introduced by Consalvo to describe how players interact with and relate to game, information about games, the game industry and other players (Consalvo 2007, p.4). This capital can be acquired by becoming knowledgeable about games and by exchanging this information with other players. Gaming capital is highly dependent on the so-called ‘paratexts’ that emerge on and about games. It is in this light that Gee’s observation about the importance of the social settings in which the game is played and the interaction that goes on around the game should be read. Gaming capital and the knowledge about the games is not only built by playing games but also through the interactions that occur in a social space, which could be physical or virtual. Gaming capital situates gaming in a game cultural framework by emphasising that gaming does not occur in a vacuum.

Consalvo’s gaming capital is inspired from Bourdieu’s cultural capital. However as Walsh and Apperly (2009) point out Bourdieu describes four kinds of capital: cultural capital, symbolic capital, social capital, and economic capital and categorise ways of identifying different types of capital in relation to game playing. The same approach is followed by Sotamaa (2010) who identifies different types of capital in relation to game achievements[2]. I will try to briefly explicate the different forms of capital in relation to game making.

Cultural capital consists of the knowledge, competencies and the dispositions of the individual. In gaming this capital is then traded with other game players through the exchange of experiences. In game making the knowledge about the games and the competencies gained by transacting in the cultural gaming capital is an important asset when designing games. The first-hand experience of gaming knowledge gained is crucial when designing a well-balanced game. The game making experience gained in turn is used to create more cultural capital for the game maker.

Symbolic Capital refers to the institutionally recognized authority that recognises the skill in this case to author games.

Social Capital is all about connections, about social relations and access to the cultural communities and networks. Having the ability to switch from playing digital games to making digital games should provide a boost for the status of the children in the eyes of fellow gamers.

Finally, ***Economic Capital*** consists- of the resources and commodities that can be translated into money; Switching from game players to game designer can introduce the children to new work roles which might eventually lead to careers.

2.2.6 Gaming Literacy

Various authors (Gee 2007, Prensky 2003, Squire 2011) argue that playing digital games embodies a new kind of literacy that blends significant elements of traditional reading and writing with new literacies that relate to accessing and evaluating information, decision making, navigating rich multimedia environments and constructing complex narratives. Beavis and O'Mara (2010) note that digital games push at the boundaries of literacy since they raise particular challenges when conceptualised as texts from a digital literacy perspective. This is due to:

[2] Achievement systems are reward structures providing additional goals for players, and thus extending the play time of videogames (Montola, Nummenmaa et al. 2009). Players can complete optional sub-goals to earn achievement rewards, such as badges, trophies and accolades, which are visible to other players

- the powerful interactive nature of games
- the ways games rely on action to proceed
- the ways in which gameplay and time (real time/game time) are difficult to replicate exactly between instances of games

The point that Beavis and O'Mara make is that games from a digital literacy perspective should be viewed from two viewpoints of games as text and games as action. I elaborate on this distinction in section 2.2.7 on page 43.

Drawing on experiences where children used a game authoring package to author a digital game, Burn (2009) proposes that a game literacy model would include the following elements:

1. Draws on cultural experience of games and other media texts.
2. Requires access to appropriate technological tools and the ability to use them.
3. Requires operational fluency: a fluency in the use of the tools for game design provided by the software.
4. Requires and develops an understanding of key concepts important to games including rules, narrative, protagonist and quest.
5. The whole process is multimodal and multiliterate since it involves visual design, writing in different genres, sound, music, speech and simple programming.
6. Involves interaction with other peripheral literacies mostly involving writing in genres such as proposals, walk through, fan fiction, narrative back stories. (Burn 2009, p.131).

Zimmerman (2007) defines Game Literacy as an approach to literacy based on game design. He argues that game literacy is based on three concepts: systems, play, design. These main concepts systems, play and design are an emergent set of skills and competencies that are increasingly part of being literate in today's world.

2.2.6.1. Systems

Every system is made from a collection of smaller systems which share complex and constantly changing interrelationships. There are multiple ways of defining the term system. One could focus on the biological or natural term of the word system, or mechanical systems or even still systems of transportation and communication. Games are systems and to play and understand them is to understand the characters in the system and the rules that drive them. Salen and Zimmermann (2003) identify four elements shared in these definitions of the word system: Objects; Attributes; Interrelationships between objects, and Environment. I use the traditional board game *Monopoly* to describe these elements.

Objects: The elements that make up the system. These objects can be physical, abstract or both. In the traditional board game *Monopoly* the tokens the players use, the *Chance* and *Community Chance* cards that are used throughout the game and the dies that are thrown by the players to start the game are all Objects. The squares themselves on the board are objects since they can be bought by the player once the token is moved to a free square.

Attributes: The qualities and properties that describe objects in a system. The location on the Monopoly board is an attribute of the token, whilst which number is drawn by the die is a property of the die. As the game progresses the attributes of the objects are updated. A player throws the dice and the token is moved. The action in the game updates the attributes of the object.

Internal relationships among objects: Objects making up a system are interrelated since an action on an object effects and changes other objects in the system. The interrelationship between objects in *Monopoly* can be seen in the relationships between the tokens and the squares. When a token is placed on a square the square can be acquired by the player. These relationships change throughout the game. If a token is placed on a square which has been purchased before by another player then the player has to pay ground rent on the property rather than acquire the square.

Environment: Systems do not exist in a vacuum but exist in an environment and are effected by the environment. The environment in which the Monopoly game is played adds to the context of play. The environment also includes the players taking playing the game directly as in throwing the dice and moving the tokens and indirectly such as managing the bank.

People are best set to learn skills, strategies, ideas when they see how these fit into a larger system to which they give meaning (Gee 2007). Good games are ideal to immerse the players into systems thinking. Whilst playing players experience how the elements of the game fit together into the overall system of the game. In addition to playing the game, players have to configure the machine or device that the digital game is played on. This too contributes to what Walsh (2010) calls “systems-based literacy”. Games are enacted through the narrative and the rules of the game which work together to create the system. Systems literacy is about immersing oneself in a system to understand how the objects making up the system interact with each other to produce this system. I contend that making a game is a great task to promote the systematic knowledge gained whilst playing games by designing one’s own game system.

2.2.6.2. Play

A game is a system where objects interact based on a set of rules in an environment. But games are much more than that. The rules making up the game are closed and fixed whilst playing involves improvisation and uncertainty.

A literacy based on play is based on innovation and invention. As Zimmerman stresses “just as systems literacy is about engendering a systems based attitude being literate in play means being playful having a ludic attitude that sees the world’s structures as opportunity for playful engagement” (2007, p. 27). I consider the ludic aspect presented here to be quite important especially for education and schools in particular. In schools we are taught to take everything seriously, to always aim at succeeding and make a big deal out of failing. This fear of failure inhibits us from experimenting and trying things out and so arguably limits us from building up our creativity. In games it is fine to fail. I am not arguing that one does not feel bad for losing, but it is quite easy to play the game all over again in a bid to succeed. I wonder whether the children will take this ludic element with them on their journey to build a game.

Gameplay occurs when the game rules are played by a player. The verb play can take two connotations. A player can play within the system and a player can also play the system. Playing within the system is when the player plays the game by following the rules. However there is another kind of playing that happens when players decide to alter the game system by altering the rules. In *Monopoly* it is quite usual for the players to decide to alter the rules of the game and so altering the game system. For example one might decide that whenever a token falls on the income tax square the money is placed on the middle of the board rather than in the bank as the game rules specify. The funds are then given to the player whose token falls on the free parking square.

Such an altering of the game rules is quite common and typically modified rule sets are published on the internet within what Gee (2004) terms as ‘affinity spaces’. I discuss this term on page 43.

Although bending the rules is easier to do with a board game, playing with the rules is not reserved to board games. A number of digital games allow the players to engage in modding or creating new levels in the game by modifying the rules of the game. Modifying the rules is not reserved for games that allow modding only. Players still tent to find ways to somehow bend the rules in games that do not allow modding too.

2.2.6.2.1. Bending the rules

Consalvo (2007) has written extensively about the practice of cheating in games. Consalvo defines cheating as breaking unseen rules or violating the spirit of the game (Consalvo 2007, p.84). Cheating might include:

- Going beyond the game manual to consult strategy guides, gaming magazines, friends, videos on video sharing sites that outline how a particular stage of a game is best played.
- Using cheat codes entered through the game controller or keyboard. These cheat codes are implanted in the game by the game developers and provide the players with benefits such as full health or unlimited ammunition. The cheat codes would normally be publicised and are generally obtained from gaming websites, friends or magazines.
- Using a video game cheating cartridge such as Gameshark to load cheats onto a gaming console. Players load cheat codes from Gameshark cartridges onto the gaming console's internal memory. When the game is loaded the selected cheats are automatically applied.
- Hacking the game itself. This would involve specialist knowledge to alter the game code and gain in-game benefits. At times the procedure might involve changing the file created when a game session is saved rather than changing the game code itself.
- Paying real money for in-game items and characters using servers such as the Sony Station Exchange.
- Game achievement auctions (Sotamaa 2010). Typically, once an auction is completed on an online auction site such as eBay, the winner would provide

account details to the service provider so that the service provider can login, play the game and accumulate achievements. The winner of the auction would then be able to use the account and claim the score and achievements achieved as his/her own.

In the case of cheat codes and video game cheating cartridge cheating can be seen as “soft programming” (Surman 2010) as this mode of cheating does not change the code of the game but is an expression of code that was already deliberately put there by the game designer.

Interesting in Consalvo’s discussion of cheating is the interviewed players’ perspective of rules. Instead of seeing rules as rigid, they see them as soft rules that can be negotiated. In multiplayer games these rules have to be negotiated between the players in order to establish acceptable game play.

2.2.6.3. Design

Design is central to game literacy as defined by Zimmerman (2007). Salen and Zimmerman (2003) define design as a process through which a game designer creates a context. A game participant experiences this context to make meaning. The process of meaning making in games is significantly complex. The game designer is responsible for creating the rules and the narratives that embody the game but it is the player that enacts the rules and creates meaning. A game designer does not create a fixed object but a multitude of possibilities.

The benefits of learning by design are highlighted by Resnick and Rusk (1996):

- Design activities engage youth as active participants, giving them a greater sense of control (and responsibility) over the learning process, in contrast to traditional school activities in which teachers aim to "transmit" new information to the students.
- Design activities encourage creative problem-solving avoiding the right/wrong dichotomy prevalent in most school math and science activities, suggesting instead that multiple strategies and solutions are possible.
- Design activities can facilitate personal connections to knowledge, since designers often develop a special sense of ownership (and caring) for the products (and ideas) that they design.

- Design activities are often interdisciplinary, bringing together concepts from the arts, math, and sciences.
- Design activities promote a sense of audience, encouraging youth to consider how other people will use and react to the products they create.
- Design activities provide a context for reflection and discussion, enabling youth to gain a deeper understanding of the ideas underlying hands-on activities.

2.2.7 It’s not just about the game

Gee and Hayes (2012) stress that the game software is only part of the equation that makes up a good video game – what they call a big G game (Game). A Game is the combination of the game (the game software) and the meta-game. The meta-game refers to “aspects of the game play that derive from the interplay with surrounding contexts” (Salen, Zimmerman 2003, p.481). Gee (2004) contends that an affinity space is an important component of the meta-game.

An affinity space is a location, mostly virtual but not necessarily, where groups of people are drawn together because of a strong interest or engagement in a common activity. An affinity space is about content. This content is created by what Gee (2004, p. 85) calls a generator. A generator, such as a game or in the case of a cooking club cooking recipes, creates a set of multimodal signs to which people attribute meaning. One of the major issued faced in this research project was that the workshop sessions were spaced in weekly intervals. In the case of this research project the game making activity and the games created are generators. An affinity space, that spans the game making workshop and a portal, provided continuity between the sessions and provided the students access to an environment which is associated with the design of a big G game.

Gee and Hayes outline a set of fifteen features defining an affinity space (Gee, Hayes 2012, p.134). Gee stresses that an affinity space is “not an all-or-nothing thing” (Gee 2004, p.85). The more features implemented the closer a space is to the model affinity space or what he calls a nurturing affinity space. I briefly explore the fifteen features outlined by Gee and Hayes below. I use the affinity space built around Scratch[3] to provide examples for features defining affinity spaces:

1. **Common endeavour not race, class, gender or disability is primary**: People join the Scratch affinity space because they are interested in Scratch and not for any other reason. Even the identity used to mark the participant contribution to the space is marked using a name of their own making which does not usually foreground the race, gender, age, disability or social class.
2. **Affinity spaces are not segregated by age**: In an affinity space there is no assumption that older people know more than younger ones. Participants in the affinity space judge each other on their passion, desire to learn and growing skills. In the Scratch affinity space each member is only judged on the contributions to the space. Seeing that a participant in the space has contributed more than 500 posts to the site gives me the message that the contributor has been actively involved in this space.
3. **Newbies and masters and anyone else share same space:** The Scratch portal does not segregate members on the basis of their expertise or contribution to the online space. Whenever I post a message on the Scratch space I get replies from newbies and masters alike. As a newbie I can also post in the fora dedicated to suggestions on Scratch. There is no segregation based on expertise.
4. **Everyone, if they wish, can produce and not just consume:** Initially, upon joining an affinity space, one tends to lurk and read the posts loaded by others. Production is encouraged but not mandated. Tutorials are provided by other members of the space and any member can download games and simulations uploaded by other members to see how a feature was implemented, make changes to the artefact and uploaded the new artefact to the site if the user so wishes. This process is known as remixing a project in the Scratch world.
5. **Content is transformed by interaction:** The content available in the affinity space is transformed continuously by the social interaction of the members of the space. Members can provide tutorials, post questions and provide answers/suggestions to other questions posted. Members can also provide

[3] The online affinity space built around Scratch is found at http://scratch.mit.edu/

suggestions on how to improve Scratch and other members can voice their opinions about the suggestions.

6. **The development of both specialist and broad, general knowledge is encouraged and specialist knowledge is pooled:** The portal encourages participants to develop specialist knowledge by discussing advanced features such as connecting Scratch to the physical sensors or creating mods of Scratch. No individual usually has all the knowledge and so specialist knowledge is pooled. The space is designed so that people can also gain general knowledge. The portal allows Scratchers to discuss things they are making which might not be directly related to Scratch. By reading about these projects a member of the space can gain general knowledge about other topics.
7. **Both individual and distributed knowledge are encouraged:** .An affinity space encourages an individual to gain individual knowledge as well as use the knowledge pool possessed by other members of the affinity space. When learning how to use a new construct in Scratch or how to solve a problem with a game created one can post on the discussion fora or search for previous threads which dealt with similar issues. Once the problem is solved the member initially trying to solve the problem or learning about the new construct might create a tutorial about the method used to solve the problem and contribute to the knowledge pool in the affinity space.
8. **The use of dispersed knowledge is facilitated:** An affinity space allows participants in the space to link to knowledge which might not reside in the space itself. When answering a question posted to the discussion board a member might link to an external website which contains information useful for the topic under discussion.
9. **Tacit knowledge is used and honoured; explicit knowledge is encouraged:** An affinity space encourages tacit knowledge; knowledge which has been built up through practice and which the members might not be able to articulate fully in words. Explicit knowledge in the form of tutorials is encouraged on the portal. However members posting these tutorials are expected to answer to questions posted by other members of the space, thereby supplementing the explicit knowledge created with the tacit knowledge.
10. **There are many different forms and routes to participation:** Membership in the affinity space is fluid. At times one might lurk and gain knowledge by

reading the posts of others whilst at other times a member might take a more active role contributing to the discussion.

11. **There are a lot of different routes to status:** Different people might be good at different things and might build a reputation for different things. One might be good at writing tutorials or at providing answers to questions posted by others and builds a reputation through the knowledge contributed to the affinity space. There is no one defined way on how to become a master.
12. **Leadership is porous and leaders are resources:** Roles are not fixed since leaders can be followers in different situations. Leadership is seen as a means of contributing resources and mentoring rather than instructing.
13. **Roles are reciprocal:** In an affinity space people sometimes lead whilst at other times follow; they teach and learn; ask questions and answer them; provide encouragement to others and get encouraged.
14. **A view of learning that is individually proactive but does not exclude help is encouraged:** The onus in an affinity space is on the individual. It is up to the individual to research information although posing questions and asking for help is greeted with
15. **People get encouragement from an audience and feedback from peers, although everyone plays both roles at different times:** Affinity spaces tend to be supportive environments. Tutorials and ideas for new projects tend to be met by comments from fellow members of the Scratch website who form an audience.

Mapping out the features of an affinity space is important for this research project. Gamers are familiar with affinity spaces be they online or otherwise. Since the game workshop in this project was held once a week I designed an online space where the students and teachers could interact throughout the week. The online space was modelled on the features of affinity spaces (see section 3.5.5 page 97).

2.2.8 A model for working with digital games in the classroom

Games as Text, Games as Action is model that knows its origin in the research project *Literacy in the Digital Age: Learning from Computer Games* funded by the Australian Research council. In this project a research team from Deakin University worked in partnership with the Victorian Department of Education and Early Childhood

Development, the Australian Centre for the Moving Image (ACMI) and the Victorian Association for the Teaching of English to research what might be learnt about literacy in the digital age and the implications for English and the literacy curriculum. This project looked at computer games and the social and literate practices entailed in playing them (Beavis 2012). Important outcomes of the project were the development of a model for computer games literacies and resources to support curriculum planning and pedagogy in this area (Apperley, Beavis 2011).

The model was built with the premise that looking at games as simply multimodal texts is an incomplete way of looking at games as it omits a major component in games - action. This gaming literacy model takes account of games' double sided nature as both text and action and can be used as a basis for planning curriculum pedagogy and assessment. This model is represented as a pinwheel with two related yet independent layers. Depending on the context where this model is applied specific sectors of each layer are placed in the foreground (Beavis, 2012).

2.2.8.1. Games as Action

Games as Action draws on research by Apperley (2010), Aarseth (1997), Bogost (2007), Galloway (2006) and Stevens, Satwicz, and McCarthy (2008). This layer acknowledges the active changing situated nature of gameplay by including sectors on situation, action, and design.

The situations sector draws on the interactions between players, non-players and technologies. It brings to the forefront the ways in which physical contexts, contexts of time, space and colour have an effect on the gameplay experience with a result that they affect the way a game is played and the number of times it is played. The status of the in-world game effecting the offline world is also acknowledged in this sector

The Action sector incorporates the interactions between players and other players and the players and the machine. It looks at the consequences of actions taken in the game world and on knowledge acquired through previous play on the way the game play is developed.

The player's active agency in design is included in the design sector with the inclusion of tailored choices within the game and the extension of games beyond the boundaries of the game.

2.2.8.2. Games as Text

Games as Text, draws on work in the areas of new literacies, digital media, and contemporary childhood (Alvermann 2010, Willett, Robinson et al. 2009), multiliteracies (New London Group 1996), and literacy and computer (Buckingham, Burn 2007, Carr, Buckingham et al. 2006, Gee 2003, Pelletier, Burn 2005, Steinkuehler 2007).This layer of the model is organized into four sectors: knowledge about the game, learning through games, the world around the game and me as game player.

The ‘Knowledge about games’ sector encompasses the player’s knowledge of related games and characteristic features of the games they are playing. This is not limited to how to play these games but also includes related texts and how games structure knowledge and participation and developing critical perspectives on games.

‘Learning through games’ explores ways in which games are used to teach explicitly through serious games and through the use of commercial, off the shelf games in curriculum areas. This sector also looks at developing critical perspectives through and about games.

‘The World around the game’ sector - literacy practices surrounding games : reading / playing games, discussion and problem solving around games, reading and analysing the wealth of texts of different kinds and literate forms that surrounds the game

‘Me as games player’ is the final sector in the Games as Text layer. This sector draws attention to the player’s involvement as a player and a reader and includes issues of engagement and reflection.

2.2.8.3. The model applied to game making

As part of a project Literacy Learning in the 21st century: Learning from Computer Games, a group of boys aged between twelve and fourteen years were taught how to make their own games using the tool GameMaker (O'Mara, Richards 2012). In their application of the model *Games as Text, Games as Action* to the game creation project, O’Mara and Richards conclude that the most prevailing aspect of the model that featured in this project was the design aspect from the *Games as Action* layer. The iterative approach adopted by the students whilst they were designing the game allowed them to reflect on the formal aspects of the game including actions and narratives. They explored the aspects of actions and narratives whilst comparing what worked and what did not in the games they designed as part of the class project. Even though the *Games as Action* dimension was the predominant dimension in this project the students also called upon the *Games as Text* dimension as they drew on their knowledge of games in order to design games which are effective. The students had to also draw on their knowledge of games they had played through the *me as game player* theme in order to design games which were attractive for themselves and their peers.

2.2.9 Conclusion

In this section I discussed play and digital gaming. I reviewed the concept of gaming capital generated whilst playing and discussing games and the notion of gaming literacy. Finally I explored a model for the classroom study of digital games that brings together two related perspectives of games: games as action and games as text. This model will be used in section 5.3 (page 160) to analyse a selection of games created by the children. This analysis is carried out to identify which sector of the model resonates most with the game creation activity conducted by the children.

Since this research deals with children designing and making their own digital games, in the next section I turn my focus onto the concept of making and creating limiting my discussion to making that involves digital technology. I then link the concept of making with creativity.

2.3 Don’t be bored – Make something

Making is linked to fostering everyday creativity (Gauntlett 2013). Given the importance devoted to creativity and innovation in the Maltese national curriculum framework and the popularity of digital gaming amongst the Maltese young generation (Busuttil, Camilleri et al. 2014), one of the main objectives of this research is to seek how to foster creativity through digital game making.

In this section I explore the concept of creativity and how teachers can teach for creativity. I then look at the stages people go through during the creative process and compare them to the digital game development stages as outlined in literature. In this way I hope to strengthen the proposal that digital game development is a valid avenue for creativity and innovation.

There is currently a re-emergence of interest in making and tinkering sparked by the publishing of the Maker magazine in 2005 and the first Maker Faire in 2006. In the Maker Faire, makers are encouraged to demonstrate and answer questions about the objects they create (Dougherty 2012). "Don't be bored make something" is the credo of Joe Hudy, a fifteen year old student who designed and constructed an *extreme marshmallow cannon* that can shoot a marshmallow a distance of 175 feet. Joe was one of a hundred students who attended the 2012 White House Science Fair to demonstrate the shooting of his *extreme marshmallow cannon* to US president Obama (Slack 2012). These students are part of a growing community of people who design and make things on their own time because they find it intrinsically rewarding to make, tinker, problem-solve, discover and share what they have learned (Kalil 2013). Makers act as amateur interaction designers, crafters and engineers creating their own meaningful project and sharing and supporting each other in Web 2.0 communities (Katterfeldt, Zeising et al. 2013). Previously DIY was mostly motivated by lack of capital or material resources. The Maker movement often appears to be a life style choice with self-expression in a mass culture playing a very important role. Projects which in the past were mostly privately shared with family and friends are now projected over the web with an extended audience.

In Education, Making is seen as a means to and a mode of participation in science, technology, engineering and math (STEM). In the US, Making is being positioned by educators and policymakers as a new and promising program of national education reform and the pathway towards future economic success (Brahms 2014). Making is not a new phenomenon, Rousseau, Pestalozzi, Froebel, Dewey and Piaget all value making and creating in their theories about learning (Martinez, Stager 2013).

Gauntlett (2013) stresses that making proposes a new take at creativity what he calls everyday creativity. He defines everyday creativity as a process that brings together at least one human mind and the material or digital world in the activity of making something which is novel in that context and which evokes a feeling of joy (p.76). I will discuss more this reframing of creativity in section 2.3.1 on page 52.

It may not be the case that all people have a drive to make and share. This is partly because modern life has sought to render personal creativity unnecessary. However there is a significant number of people who enjoy making and sharing without the needs of external rewards such as money but with low level recognition acting as a motivational force. They enjoy making for its own sake, enjoying the process of seeing a project from start to end. The process provides a space for thought and reflection which cultivates the sense of self as an “active creative agent” (Gauntlett 2013, p.222). This sensation is coupled by a desire to connect and share with others and it is this coupling of desires that websites such as www.instructables.com and magazines such as Maker magazine tend to harness so successfully.

In her analysis of the learning practices of the making community Brahms (2014) analysed three volumes of the Make magazine to identify learning practices that are associated with recognizable participation in the maker community. Brahms (2014) identified seven core learning practices:

- **Explore and Question:** Interrogation of the material properties of the context in order to find inspiration or to determine intention for a process or project
- **Tinker, Test and Iterate:** Purposeful play, experimentation, evaluation and refinement of the context
- **Hack and Repurpose:** Harnessing and salvaging component parts of the made world to modify, enhance, or create a product or process
- **Combine and Complexify:** Developing skilled fluency with diverse tools and materials in order to reconfigure existing pieces and processes and make new meaning
- **Seek out Resources:** Identifying and pursuing the distributed expertise of others, includes a recognition of one’s own not-knowing and desire to learn
- **Customize**: Tailoring the features and functions of a technology to better suit one’s personal interests and express identity
- **Share:** Making information, methods and modes of participation accessible and usable by members of the community

In the 1970s making and tinkering played a central role in the introduction of computers into the mainstream use. Apple II, one of the first highly successful mass-produced computers was initially put to market in two versions as a computer with its own keyboard monitor case and power supply as well as a circuit-board only for the do-it-yourself hobbyist. In an advert to market the Apple II computer, Steve Jobs and Steve Wozniak stated that anyone can take Apple II as far as imagination can take it by programming it in Apple Basic. The *generativity* of this machine was quite high and this was one of the main factors that lead to its huge success. *Generativity* relates to the system's capacity to produce unanticipated change through contributions from broad and varied audience (Zittrain 2008). Bricklin and Franston worked on the first spreadsheet program they called VisiCalc which ran on Apple II. VisiCalc helped make Apple II popular with businesses that saw the relevance of using a computerised spreadsheet to help run their businesses. Over time the *generativity* offered by systems was abused to create malicious systems such as viruses and facilitate identity theft. This led to the *generativity* potential diminishing in a number of systems, to the point that when the first version of iPad was released, by the same company that released Apple II, it was advertised as a safe device that only allowed Apple approved software to be installed on the machine. With the release of iPad the *generativity* had hit a low as the device did not allow the audience to write content for the device. This stance was later revised with the inclusion of the apple developer programme which allowed anyone rather than just Apple themselves to write apps for iPad.

In recent years we have seen a number of initiatives that are aimed at allowing the user to have total control on the machine. Google has released the App Inventor which allows users to create applications that can run on Android powered phones and tablets by using a very simple programming language. A number of credit-card sized, relatively inexpensive computers such as the Arduino, Raspberry Pi and BeagleBone have been released with the purpose of providing devices which adults and children can use to create projects. Hence in a way the industry is doing a full circle and providing building blocks which allow children and adults to practice "everyday creativity".

2.3.1 Creativity

In education, the term creativity is often used but seldom defined. As Beghetto (2005) points out, teachers might ask students to use their creativity in the design of a project, or might refer to a student's response as creative, without explaining what they mean.

The NCF defines creativity and innovation as "agents for change which contribute to the economic prosperity of society in general" (DQSE 2011, p.47). This view echoes the general assumption about the value of creativity and innovation in Britain as identified in critical discourse analysis of governmental policy documents, academic and think-tank publications, and consultancy reports conducted by Böhm and Land (2009). Böhm and Land state that the prevailing view about creativity and innovation is that in a knowledge economy creativity and innovation are the engines of productivity and economic growth. Craft (2005) maintains that creativity is emerging as part of a universalised discourse in the western world. The globalisation of economic activity and the increased competition has introduced a fear of obsolescence. Creativity is seen as a response to this fear since innovation is seen as necessary for economic reality (Choe 2006, Shaheen 2010). Craft (2005) also points to Maslow's belief that ties creativity to personal fulfilment when noting that a creative individual is a fulfilled one.

2.3.1.1. Drivers for creativity in Education

Despite a backdrop in education where everything is measured in scores attained in exams of the so called core curriculum subjects, there has been a significant push to include creativity in education since the end of the 20th century. According to Craft (2011) there are three main drivers for including creativity in education: the economic, social and technological drivers.

2.3.1.1.1. Economic

One of the main drivers towards including creativity in education is the economic driver. It is fair to say that a large number of us are engaged in work and employment which did not exist when we were in schools. This trend will increase in the future. With the increase of globalisation, economies are becoming more interdependent. Creativity is seen as a driver required to keep the economy changing fast to keep up with consumerism. Changes in employment and the speed of economic development and re-development imply that both knowledge and creativity are seen as a feature of business success and intrinsically tied to education.

2.3.1.1.2. Social

Society too is changing a lot. Geographical, social and emotional mobility is increasing and more value is placed on the personal choice of the individual. Education is seen as needing to gear up towards helping children and young people make sense of an array of choices and exercise creativity in imagining potential.

2.3.1.1.3. Technological

A major driver towards creativity in education has been technological change. Technological change both offers and demands opportunities for creativity. A vast amount of what we do involves digital technology. This human technology interaction demands a certain level of creativity in envisioning what we can do. However a bigger driver towards creativity will emerge once we start debating what it means to be human when reality is supported or augmented by technology.

2.3.2 Defining creativity

Although traditionally people associate creativity with the creative arts of music, drama, art, dance and literature, creativity is not unique to the arts (NACCCE 1999) Creativity is equally fundamental to advances in the sciences, in mathematics, technology, in politics, business and in all areas of everyday life (NACCCE 1999) .

As Sharp (2004) points out, definitions of creativity are not straight forward and many writers hotly contest different views. She points out that most theorists agree that the creative process involves a number of components including:

- imagination
- originality (the ability to come up with ideas and products that are new and unusual)
- productivity (the ability to generate a variety of different ideas through divergent thinking)
- problem solving (application of knowledge and imagination to a given situation)
- the ability to produce an outcome of value and worth.

The NCF too proposes a series of characteristics and maintains that creativity and innovation is about developing

- affective communication
- lateral thinking
- originality
- emotional development
- problem-solving

- leadership
- questioning
- intuition
- fostering entrepreneurial mindsets
- openness to cultural diversity
- self-expression

Some of the characteristics listed by the NCF stand out since they are not usually associated with creativity and innovation. These characteristics include affective communication, leadership, fostering entrepreneurship and openness to cultural diversity. This might be the reason why the "creativity and innovation" theme was renamed to "Education for entrepreneurship, creativity and innovation" in the launch of the national curriculum framework (Ministry of Education and Employment 2012) further reinforcing the view that the reasons Creativity and Innovation has been included in the national curriculum framework is the prevailing assumption that creativity can foster economic growth. Although I agree with this standpoint I favour more the views that creativity is an avenue for self-expression and that whilst creating every creator leaves his or her digital fingerprint in the work created.

2.3.2.1. Big C and Little c views of creativity

There are two predominant views which emerge from the discussion on defining the term creativity. The first view of creativity focuses on exceptional creative individuals who made a major impact on the world. This view reserves creativity for the very few individuals and is termed as Big C creativity. The national advisory committee on creative and cultural education (NACCCE) (1999) calls this formation of creativity "The elite conception of Creativity". NACCCE state that the Big C creativity is important because it focuses attention on creative achievements which are of historic originality, which push back the frontiers of human knowledge and understanding.

The second view of creativity was first voiced by Maslow (1970) who put forward the notion that creativity is not for the few. Maslow stressed that creativity is found in everyday activities. This approach to creativity is more focused on creative activities conducted every day by laypersons or individuals who would not necessarily be considered experts or luminaries (Kaufman, Beghetto 2009). Various terms have been used to describe the little c creativity such as democratic creativity (NACCCE 1999), everyday creativity (Gauntlett 2013) and personal creativity (Robinson 2011). For NACCCE a democratic society should provide opportunities for everyone to succeed according to their own strengths and abilities. They maintain that the democratic conception of creativity recognises the potential for creative achievement in all fields of human activity; and the capacity for such achievements in the many and not the few.

The application of little c creativity in the day to day setting has been discussed by a number of researchers (Amabile 1996, Craft 2001, Craft 2003b, Gauntlett 2013, Runco, Richards 1997) with Craft using the term "lifewide creativity" to describe the application of creativity in the everyday life.

These two views of creativity, that is the little c and Big C views, should not be seen as two distinct views but as two ends of a continuum with a spectrum of creativity levels in between (Craft 2005). Indeed the NACCCE (1999) suggests that fostering creative education in schools will promote the growth of creativity. As Robinson (2011) concludes exceptional individual achievement - that is historic originality (Big C) is more likely to emerge from a system of education which encourages the creative capacities of everyone.

2.3.2.2. Adopting a definition of creativity

For the purpose of this study I adopt the definition of creativity as outlined in NACCCE (1999) Creativity is defined as:

> Imaginative activity fashioned so as to produce outcomes that are both original and of value. (NACCCE 1999, p.30)

This definition focuses on the little c creativity and looks at creativity as a process – a fashioning activity. It includes the features of creative processes that need to be encouraged for educational purposes. It also fits neatly with the concept of game authoring by children as a creative activity.

2.3.2.3. Imaginative activity

Robinson (2011) emphasizes that imagination is the source of creativity and that creativity is about putting imagination to work. The term imaginative activity as used in the definition of creativity is not simply producing mental representations of things that are not present or have not been experienced before. NACCCE (1999) defines Imaginative activity as "the process of generating something original: providing an alternative to the expected, the conventional, or the routine" (p 31). Imaginative activity is thinking "outside the box" where a person uses mental play to look at a situation from different perspectives envisioning alternatives.

2.3.2.4. Fashioning imagination to produce outcome

The process of imaginative activity is focused to achieve the final goal. Creativity is not seen as magic, but more of a process where the imaginative activity is shaped and reshaped to arrive to a creative act. Creative insights or breakthroughs may occur unexpectedly along the way, however the insights or breakthroughs are part of a dynamic journey towards the end result, the end result which might turn out to be quite different from what was being anticipated initially. As Robinson (2011) states creativity is about doing something. To describe someone as being creative suggests that the person was actively producing something in a deliberate way. Creativity as a process is emphasised by Gauntlett (2013) who states that creativity is a process that brings together the creator and the material or digital world.

2.3.2.5. Originality

Creativity and originality have always been linked. However one can look at originality from different perspectives. NACCCE (1999) outlines three perspectives:

- **Individual**: The creative outcome might be original in relation to the previous outcomes by the individual. In this respect what matters is that the outcome has not been achieved by the person before (Gauntlett 2013).
- **Relative**: The outcome might be original in relation to the peer group.
- **Historic**: The outcome might be original in terms of any other person's previous outcome.

The historic originality is the ultimate creative outcome, creativity which is in line with the Big C creativity. However as Beghetto (2005) argues social context is very important when looking at creativity and originality. An eighth-grader's poem, though not demonstrating the same level of creativity as Emily Dickinson's poems, certainly can be considered creative, i.e. novel and appropriate within the context of her language arts class, her school, state and even beyond (Beghetto 2005, p.255).

The judgement of how original a contribution is depends on the context where the creativity is being assessed. In the scenario depicted by Beghetto whether the poem is judged as creative would depend on whether the eighth grader's poem is being judged in a language classroom, at an after-class poetry club or at an international contest. It would also depend on the stakeholders in that context, the classroom teachers, fellow students or a panel of international poetry experts. As Sharp (2004) argues only a child prodigy would come up with something which is new for society.

In this light the individual and relative perspectives of originality are both important for fostering creativity in the classroom.

2.3.2.6. Value

The originality of creativity is very important but is not enough. An original contribution could be irrelevant to the purpose in hand, bizarre or even faulty. The outcome of imaginative activity can only be called creative if it is of value in relation to the task at hand. This calls for a judgment call to take place to evaluate the value of the contribution according to the area of activity. Since the creative process involves mental play and envisioning alternatives, the evaluative mode of thinking needs to be present throughout the creative process. The evaluation mode of thinking needs to focus on what works and what does not, as well as on the originality of the contribution.

NACCCE (1999) states that the evaluative process can be shared with others or involve periods of quiet reflection. It could involve instant judgements or long term testing.

2.3.3 Teaching for creativity

With the importance given to creativity and innovation it is important to explore how creativity can be instilled through the teaching process. This argument has existed for long and there seems to be a consensus that creativity is amenable to teaching (Amabile 1996, Craft 2003b, Jeffrey, Craft 2004, Craft 2005, Kaufman, Beghetto 2009, Philip 2013). The review of literature on teaching for creativity is especially important for this project since the suggestions found in literature will be used to shape the structure of the game making workshop.

The NACCCE report makes a distinction between teaching creatively and teaching for creativity. Teaching creatively is defined as "using imaginative approaches to making learning more interesting and effective" (NACCCE 1999, p.89) whilst teaching for creativity is defined as forms of teaching that develop student's own creative thinking or behaviour. NACCCE acknowledges that these terms are closely related whilst Jeffrey and Craft (2004) note that teaching for creativity denotes teachers teaching in a creative way. Whilst teaching creatively involves using imaginative approaches in teaching to make learning more interesting and effective for the students, Craft (2005, p.42) suggests that teaching for creativity involves:

- The passing of control to the learner and the encouraging of innovative contributions.
- Teachers placing value on learners' ownership and control when innovation often follows.
- Encouraging children to pose questions, identify questions and issues.
- Offering the children the opportunity to debate and discuss their thinking.
- Encouraging children to be co-operative in learning, resulting in further control for learning over appropriate strategies for their learning.
- Being at the least learner considerate but ideally learner inclusive, thus prioritizing learner agency.

2.3.3.1. The role of the teacher

McWilliam (2008) notes that teachers have 'un-learned' the role of "Sage on the stage" as the dominant model of teaching, and the shift to "Guide on the side" has served an important function in changing the focus of pedagogy from the teacher to the learner. However McWilliam concludes "Guide on the side" is no longer sufficient for our times. Instead she proposes the "Meddler in the middle approach". The motivation for the "Meddler in the middle" approach comes from the fact that it is nearly impossible to know everything about a subject. As McWilliam (2008) points out "we have never been more ignorant". This might sound strange but it is quite true especially in areas where technology is concerned. Technology changes so quickly and is so vast that no one can claim to possess all answers. Adopting the "Meddler in the middle" approach involves:

- less time giving instructions and more time spent being a usefully ignorant co-worker in the thick of the action
- less time spent being a custodial risk minimiser and more time spent being an experimenter and risk-taker

- less time spent being a forensic classroom auditor and more time spent being a designer, editor and assembler
- less time spent being a counsellor and more time spent being a collaborative critic and authentic evaluator

She reminds us that we have a much less intimate knowledge of the technologies that we use every day than our forebears had, and will continue to experience a growing gap between what we know and what knowledge is embedded in our manufactured environment. Reflecting on the points raised by McWilliam(2008) I come to the conclusion that being effective in an ever changing technological world is not about knowing all the answers but is about fostering the ability to finding the answers and to be critical of the answers found. In this way we can put this *ignorance* to work. We can make it useful by providing opportunities for ourselves and others to live innovative and creative lives. In this light teaching is seen as a form of value creation rather than knowledge transmission.

Various researchers concur that teachers play a fundamental role in fostering creativity in young children (Lin 2011, Daws 2009, Jeffrey, Craft 2004, Craft 2003a, NACCCE 1999, Mellou 1996, Runco 1992, Tegano, Moran et al. 1991). The following is a list of stances that a teacher can take in order to play an important role in fostering creativity in young children as identified in the literature surveyed.

- Ask open ended questions.
- Tolerate ambiguity.
- Model creative thinking and behaviour.
- Encourage experimentation and persistence.
- Praise children who provide unexpected answers.
- Encouraging young people to believe in their creative identity.
- Identify young people’s creative identities.
- Encourage curiosity.
- Provide opportunities to be creative in a hands on approach.
- Adopting an inclusive pedagogy in which teachers and learners enter a co-participating process around activities and explorations, posing questions, identifying problems and issues together with debating and discussing their thinking.

2.3.3.2. Fostering possibility thinking

In the previous section I discussed the role that a teacher should take to play an important role in fostering creativity in children. But what role must children take on to be part of this creative learning experience? Jeffrey (2004) suggests that being creative involves being innovative, experimental and inventive and this implies that the students engage in aspects of intellectual inquiry. Craft, Cremin, Burnard and Chappell (2008) suggest that at the heart of intellectual inquiry lies the aspect of possibility thinking and engaging with problems.

A number of studies (Burnard, Craft et al. 2006, Craft, McConnon et al. 2012, Craft, Cremin et al. 2013, Craft 2013, Cremin, Burnard et al. 2006) have explored the notion of possibility thinking. According to these studies possibility thinking is about finding problems and providing possible solutions to them. Possibility thinking can be practiced by individuals working on their own or in collaboration with others.

Craft (2013) outlines the core features in children's possibility thinking. These include:

- question posing (investigative behaviour).
- question-responding (investigation response behaviour).
- self-determination (self-directed actions, self-chosen).
- intentional action (activity/behaviour with a goal).
- development (thinking moving forwards).
- being imaginative ('as if' thinking and going beyond the expected).
- play/playness (being in an 'as if' space, improvising).
- immersion (concentration, absorption, orientation).
- innovation (original/unique outcome/behaviour).
- risk-taking (danger, failure, fear, 'going to the edge').

Craft (2013) concludes that there is greater potential for children to engage in possibility thinking when children interact with digital media since children have greater control over their creative endeavours and fewer adult framings of these endeavours occur.

2.3.4 The creative process

A suggestion frequently found in literature (Craft 2003a, Sharp 2004, Tegano, Moran et al. 1991) is to put the emphasis of education on the creative process rather than judging the quality of the products produced. I knew the games produced by the children in the games workshop that I planned to set up would not be able to compete in the commercial or indy games arena. However what was important was that the children would have gone through the creative process whilst making a game they designed. This view echoes that of Malaguzzi, one of the driving forces behind the Reggio Emilia approach who in an interview had stated that “Creativity becomes more visible when adults try to be more attentive to the cognitive processes of children than to the results they achieve in various fields of doing and understanding” (Gandini 2011, p.52).

According to Lubart (2001) the creative process has been one of the key topics of creativity research for a long time. Lubart defines the creative process as the sequence of thoughts and actions that leads to a novel, adaptive production.

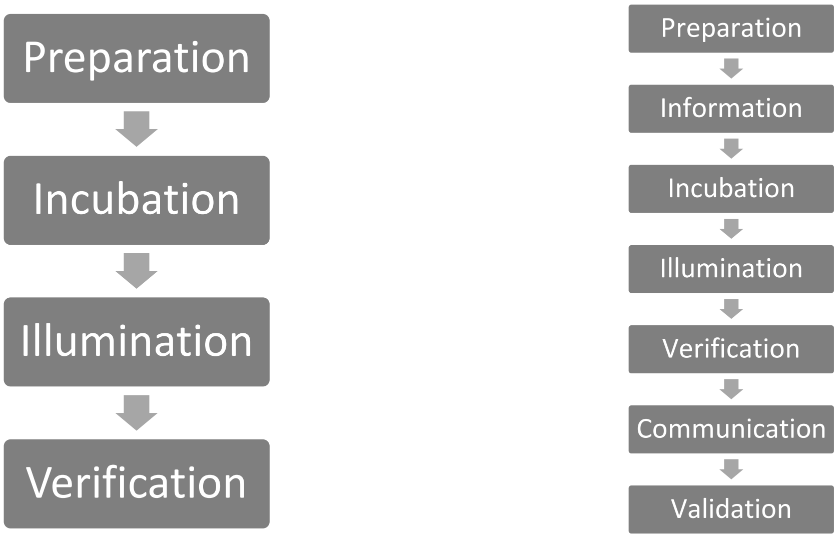


Figure 2.3-1 Wallas (1926) vs Lubart (2001) Creativity proocess models

One of the earliest models of the creative process is attributed to Graham Wallas. Wallas (1926) proposed that creative thinking proceeds through four phases. During the **Preparation** phase the issue under study, the problem, is defined and studied. This phase is followed by the **Incubation** stage where the problem is reflected upon unconsciously. During this stage the individual would carry out various activities whilst unconsciously mulling over various approaches to the problem at hand. When a solution is identified the process shifts to the **Illumination** phase where the solution is detailed and checked out at the **Verification** phase.

Although the model presented by Wallas (1926) is a linear model, Lubart (2001) notes that during creative problem solving a person could return to earlier phases in the process. For example, if an idea proves to be flawed during verification, one may revert back to the incubation stage and ponder on how to resolve the flaw identified.

Cropley (2001) asserts that the role of society in creativity and Csikszentmihayi's (1996) emphasis on the importance of socio-cultural validation necessitate the addition of two phases to the follow the model presented by Wallas. Cropley also stresses the importance of human agent acting with intention prior to the Preparation phase in Wallas' model and adds a further phase to acknowledge this importance.

The new seven layer extended model of the creative process consists of:

- **Preparation** phase where the problem at hand is identified and convergent thinking is used to identify the goals
- **Information** stage where the person becomes familiar with the content area
- **Incubation** stage where the person mulls over the information obtained in the information stage in what Wallas calls the "unconscious state" whilst carrying out other tasks not necessarily related to the creative process
- **Illumination** stage where the solution emerges seeming to the person involved to come like a bolt from the blue
- **Verification** stage where the person tests the solution
- **Communication** Following verification of the solution the solution is presented to the community
- **Validation** The community validates the solution resulting in acceptance or further iteration in the previous states

2.3.5 Creative process in game development

In this section I delve into two game development models proposed by Resnick (2008) and Robertson (2011) and compare them to the extended creative process model outlined by Cropley (2001).

In his article "Sowing the seeds for a more creative society" Resnick (2008) discusses Crickets and Scratch, digital technologies developed at MIT Media Lab to help learners develop into creative thinkers.

Crickets are small programmable devices that can make things spin, light up, and play music. One can plug lights, motors, and sensors into a Cricket and then write computer programs to tell them how to react and behave. Through Crickets one can create musical sculptures, interactive jewellery, dancing creatures, and other artistic inventions.

Scratch is an iconic programming language that makes it easy to create interactive stories, animations, games, music, and art. The projects created can be shared on the Scratch website. Crickets and Scratch support what Resnick (2008) calls the *creative thinking spiral*.

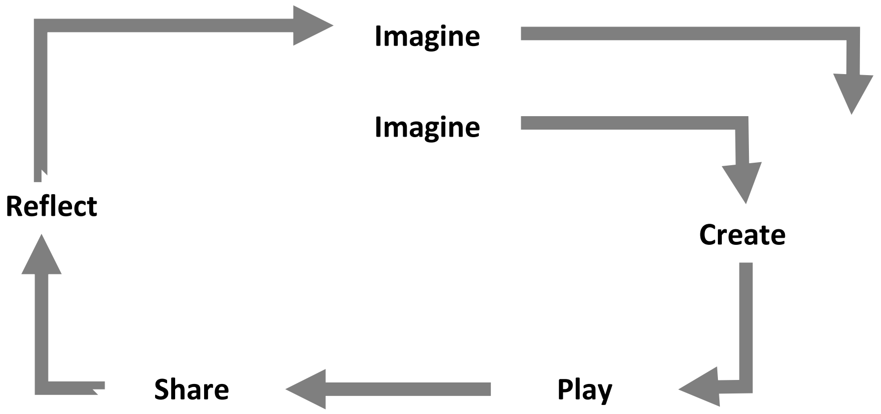


Figure 2.3-2 The Creative Thinking Spiral

During the creative thinking spiral process the creator goes through five stages starting off with the **Imagine** stage. During the imagine stage the author of the digital artefact comes up with the initial project idea. The project is created in the **Create** stage, closely followed by the **Play** stage where the creator plays his creation before sharing it with peers in the **Share** stage. The final stage of the creative thinking spiral process is the **Reflect** stage where the author of the digital artefact reflects on the comments posted by peers and starts on a journey to refine the project through the next iterative pass of the spiral process. "As students go through this process, over and over, they learn to develop their own ideas, try them out, test the boundaries, experiment with alternatives, get input from others, and generate new ideas based on their experiences" (Resnick 2008, p.18).

Robertson (2011) proposes a five stage creativity model students go through whilst creating computer games using Adventure Author. Adventure Author is a game creation tool for children aged 10-14 based on Atari’s Neverwinter Nights 2 game-making toolset. Adventure Author enables children to design and build interactive stories for anybody to play.

The model proposed by Robertson was based on field observations of children making games. Students are expected to progress through the stages in the model in a non-linear way. The first stage in the model is the **Exploration** stage. During exploration the student playfully tries out the features of the software and tests the boundaries to identify the possibilities allowed by the environment. The Exploration stage normally leads to the **Problem Finding** stage. In this stage the student generates and selects ideas to be used in the game. Usually the problem finding stage leads to **Problem Solving** stage where the scenario picked up in problem finding stage is turned into a game. Robertson notes that there are instances when the Problem finding stage is skipped and students move from the exploration stage directly to the problem solving stage. This usually happens when students are exploring a new feature in the software and create a prototype game to exploit this feature. The problem solving stage is followed by the **Internal Validation** stage where the game is tested by the user. Usually any flaws which are identified send the student back to the problem solving stage where the flaws identified are fixed. Once the student is satisfied with the game created the game is presented to peers in the **External Validation** stage. External

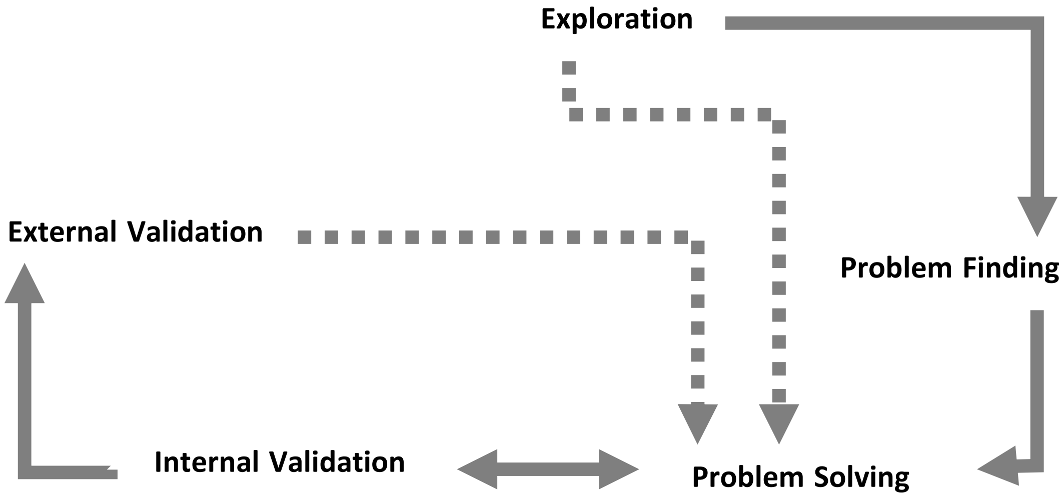


Figure 2.3-3 The creative process of game making during Adventure Author

validation can lead the student back to the problem solving stage to iron out any deficiencies identified by the peers.

2.3.5.1. Links between the models

The models outlined by Resnick (2008, p.18) and Robertson (2011) contain a lot of similarities, with most stages in a model mapping to stages in the other model.

The exploration stage is missing from Resnick's model. In this stage the student is trying out the software to identify the potential and limitations. At this point the student is not thinking about the outcome, that is the game to be produced, but focusing more on the tool to achieve the outcome. During field studies Robertson observed that "young people need to be immersed in the technological environment for some time to explore the possibilities it affords before they commit to an idea about a project" (Robertson 2011, p.6). I would argue that the exploration stage is important during the initial period, when the students are not yet familiar with the authoring environment and that its importance will diminish once the students become more acquainted with the software development kit. As Robertson points out learners might get frustrated if they commit to a creative idea which turns out not to be supported by the software.

The identification of the problem to tackle occurs during the imagine phase in the creative spiral and in the problem finding stage of the Adventure Author creative process. The create stage maps to the problem solving stage since in both cases the game is authored in this stage. Once the game is created in the creative spiral, the process proceeds to the play stage.

Although the play stage in the creative spiral is simply described by Resnick (2008, p.18) as the stage where students "play with their creations", one would presume that during this stage, testing of the game is occurring through the playing of the game by the game author since the game cannot be effectively played if it still contains problems. If the play stage is interpreted in this light, it would map to the internal validation stage of the Adventure Author creative process.

In both game development processes the game is shared with peers for validation. In the Adventure Author creative process this is done through the external validation phase whilst in the case of the creative spiral this is done in the share stage.

The share stage in the creative spiral leads the students to the reflect stage where students reflect on their experiences in view of starting the creative spiral journey again. This stage can be seen as an evaluation of the game taking into consideration the feedback obtained from peers and the game experience whilst playing the game in the play stage. Although the *Adventure Author* creative process does not contain a formal evaluation stage, the reflection is carried out in the internal and external validation stage.

2.3.6 Conclusion

The NCF stresses the importance of design and make tasks where students work through a creative process task (see section 2.1, page 22). In this section I explored literature pertaining to creativity and creative processes. Two different game making processes where analysed and compared to creative processes in a bid to identify a model which can be compared with what happens during the game making workshop.

In this section I also reviewed the role of a teacher in teaching creatively for creativity in order to design a workshop aimed at helping the students exploit their creative potential. The next section will deal with the skills required to build the game.

2.4 The skills to build a game

A number of studies (Baytak 2009, Carbonaro, Cutumisu et al. 2008, O'Mara, Richards 2012, Owston, Wideman et al. 2009, Pelletier, Burn 2005, Robertson, Howells 2008, Vos, van der Meijden et al. 2011, Yee Leng, Zah bte Wan Ali, Wan et al. 2010) deal with building games with children however almost all of them shy away from using a programming language with children to make games. Programming is seen as a traditionally difficult skill to master (Caspersen, Bennedsen et al. 2008) even if Papert has been advocating the use of computer programming as an educational tool since the 1970s (Papert 1980, Papert 1994). Over the recent past a number of tools have been released aimed at introducing programming to children of various ages through an interface which makes programming an easier task. Kafai has used Scratch, a multimedia programing language, through the computer clubhouse project to introduce programming to children and youths (Kafai, Peppler 2012, Kafai, Peppler et al. 2009). A new version of Scratch called Scratch Jr has also been released aimed at introducing programming to children in early years classes (Flannery, Silverman et al. 2013). There has been a recent push towards the inclusion of coding into schools (Livingstone, Hope 2011, European Commission 2014). Similarly to the motivation for the inclusion of creativity in schools (see section 2.3.1.1.1 page 53), the main driver used is the economic driver. Even though I can see the economic benefit of more children being exposed to programming from a young age, I still believe that the main benefit is the affordance of self-expression offered by the ability to create your own digital media, in this case a digital game. As Lange and Ito (2010) point out media creation is of central importance in the everyday social communication of youth. Youths use media as a means of self-expression (Lange, Ito 2010). This view is also supported by Kafai and Burke (2013) who attribute the renewed interest in teaching programming from a young age to the philosophy of digitally based youth cultures and the re-emergence in making and tinkering (see section 2.3 page 49). Building a digital game can expose the children to computational thinking skills with programming being just part of the whole picture. I discuss computational thinking in further detail in section 2.4.2 (page 72). I now focus on different approaches that can be used to introduce students to programming. I will use this review of literature to reflect on an approach that I will use to introduce the students to game making using the Scratch programming language.

2.4.1 Introducing students to programming

Selby (2011) outlines four approaches to the teaching of programming found in a survey of literature: The code analysis approach, building blocks approach, simple units approach and the full systems approach. Each approach has its advantages and disadvantages for the teacher and the students. I will briefly outline these four approaches below:

2.4.1.1. Code Analysis approach

In this approach to the teaching of programming students learn how to read and understand programming logic before writing their own. This approach is based on providing the students with practice exercises using structured English rather than a programming language. One of the main disadvantages of this approach is that students may feel cheated being in a technology class where the tools in use are the pencil and paper.

2.4.1.2. Building Blocks approach

The building blocks approach introduces the language constructs one at a time in isolation before combining them. This approach uses a specific programming language. The main disadvantage of this approach is that mastering individual construct behaviours may not transfer to the building activity required for a meaningful logical algorithm since it is possible for an individual to master a concept but then be unable to use it to produce a solution to a meaningful task. Hence the building block approach needs to be followed up by a simple problem activity so that the problem would provide context where the constructs are meaningfully applied.

2.4.1.3. Simple units:

The third approach to the teaching of programming identified in literature is the simple units approach. In this approach constructs are grouped together to form units of code that can be reused. This approach is similar to learning to speak a foreign language by mastering a phrase book with limited vocabulary before combining the phrases to create meaning. The student would master solving small problems by using the pre-packaged units of code. This approach was found to be quite useful for weak students since it usually gives them a starting point from which to develop solutions.

2.4.1.4. Full systems

The full systems approach is analogous to learning a foreign language by immersion. Using this approach students are provided with a program which they have to read and alter. This approach places greater emphasis on design skills rather than on mastering syntax. The programming concepts and language constructs are introduced only when the solution to the problem requires their application and so the choice of the initial problem is crucial for this approach to succeed. At first glance it may appear that this approach is overwhelming for the students. However the advantage of this system is that the students can be presented with real problems that they are already familiar with and for which they would already have conceptual models. For example in the study carried out by Campbell and Bolker (2002) an ATM simulator was presented to the students who then worked on extending it.

2.4.1.5. Adopting a problem-based learning approach

Reflecting on the four approaches to teach programming discussed above only the Code Analysis and the Full Systems approach teach the students how to read before learning to write. Learning to read code is an important skill in itself since typically software is written by teams of programmers who have to read each other's code to be able to extend it (Busjahn, Schulte 2013). The Full Systems approach has the added benefit of using a real programming language rather than pseudo code and shows the implementation of the programming constructs to solve a real life problem that the students are familiar with.

Problem-based learning requires students to work in collaborative groups to resolve complex, realistic problems under the guidance of a teacher (Allen, Donham et al. 2011). Problem based learning knows its origin in the medical arena where teams of students work together to diagnose and suggest treatment for case histories of real patients. Educators following the problem-based learning approach must find or create good problems based on clear learning goals that not only present the students with issues that matter to them but also foster their development. The teaching of computer programming is suitable to problem-based learning since the teaching of programming is about training a deductive way of thinking (Kay, Barg et al. 2000, Peng 2010). The full systems approach to introduce programming subscribes to a problem-based approach since the students are immersed in a problem and are then helped to extend it.

2.4.1.5.1. Pedagogic recommendation on introducing programming

Kölling and Rosenberg (2001) outline a series of pedagogic recommendations on how to introduce programming. The recommendations are based on introducing object oriented programming to first year university students however most of these recommendations are relevant to other languages, such as Scratch, and to a younger audience too. In the section below I outline the pedagogic recommendations which are relevant to a non-object oriented approach language such as Scratch. The recommendations presented by Kölling and Rosenberg are well suited to the full systems approach of introducing programming outlined by Selby (2011).

2.4.1.5.1.1. Don't start with a blank screen

Starting with a blank screen is a very difficult exercise. Writing code involves a design exercise where a system is broken down into smaller units which interact with one another. Kölling and Rosenberg (2001) recommend that students start by amending existing code and if they are to write new code they do it as part of extending an existing project.

2.4.1.5.1.2. Read Code

Students can learn a lot from studying well written programs. Programs can be an inspiration for coding style and also present the students with an approach to design. Hence it is important that all programs presented to the students for reading purposes are well written and are worth being emulated.

2.4.1.5.1.3. Use large projects and show program structure

Students can learn a lot from seeing how a problem was decomposed into smaller parts. The program shared with the students should contain a set of objects that are related with each other. The structure of the application is crucial to the quality of a solution and so it is important for the teacher to discuss how the system was structured and how objects relate to one another.

2.4.1.5.2. Implementing a Full Systems approach

Kölling (2008) outlines a sequence of progressively more complex activities that can be used in a full systems approach to introduce students to programming. In the following list I adapt the sequence suggested by Kölling to work with Scratch, the programming language that will be used in this project.

2.4.1.5.2.1. Get your feet wet: executing code

The first phase in implementing the full systems approach should be to introduce the students to a project designed to achieve two things: to familiarise the students with the authoring environment and to convey the basic concepts of the program. In the case of Scratch the students need to become familiar with:

- how to execute a game written in Scratch
- Sprites which are the basic building blocks of a Scratch program. In a game developed in Scratch all the game characters will be implemented as sprites
- where the code to manage the sprites will be placed
- where the image and sound assets for a sprite are placed
- how sprites can communicate together

All these concepts will be introduced to the children by executing the program and highlighting code reading.

2.4.1.5.2.2. Manipulate source code

The second step in the approach to introducing programming should be to introduce the students to code manipulation. The original project might have intentional bugs implanted into the project which the students might need to solve by changing the existing code.

2.4.1.5.2.3. Create a new behaviour and adding more building blocks

After fixing the code in the previous step, the students would be asked to extend the project by adding new behaviour to the existing sprites or adding new building blocks in the form of new sprites.

2.4.1.5.2.4. The master test

In the final step of the project the students are asked to create a whole new application from scratch. In this last step only a brief description of the problem is given and the students have to go through the whole development process with guidance from the teacher.

2.4.1.6. Conclusion

In this section I discussed different approaches that can be used to introduce students to programming. Most of the approaches found in literature are aimed at a different age group than the students who participated in this research project. However the ideas remain relevant even when adopted with a younger audience.

As discussed earlier in the introduction of this section building a digital game exposes the children to computational thinking. In the next section I explore the concept of computational thinking and its relevance to the building of computer games by children.

2.4.2 Computational Thinking

Computational thinking is a reasoning skill set that can be applied to solve problems from various fields be it sciences, arts, economics or humanities (Bundy 2007, Perković, Settle et al. 2010, Wing 2006). Activities such changing a tire, brushing teeth, cooking from a recipe and following instructions to construct a table can also be tackled using computational thinking (Cortina 2007, Henderson, Cortina et al. 2007, Wing 2006). A number of researchers (Wing 2006, Wu, Richards 2011) argue that students should be exposed to methods which foster computational thinking even though this poses pedagogical challenges due to computational thinking's nature (Fletcher, Lu 2009, Lee, Martin et al. 2011). Ever since this reasoning skill set aimed at problem solving was first identified by Jeannette Wing in 2006, there have been a number of initiatives aimed at exposing students to computational thinking (Barr, Harrison et al. 2011, Barr, Stephenson 2011, Hambrusch, Hoffmann et al. 2009, Morreale, Joiner 2011, Wolz, Ouyang et al. 2011).

There are various reasons cited in literature why computational thinking should be integrated into schools. Integrating computational thinking in education helps students with different academic inclinations to be better problem solvers and critical thinkers (Fletcher, Lu 2009). Giving importance to computational thinking skills in schools ensures that the future citizens employ computational thinking skills in both formal and informal settings (Fletcher, Lu 2009, Wing 2008).

It is important for students living in a knowledge-driven society to experience situations where reflective engagement, creativity and innovation are the order of the day (Wing 2006, Allan, Barr et al. 2010, Ioannidou, Bennett et al. 2011). Wu and Richards (2011) contend that reflective engagement whilst learning scientific subjects is manifested when the student is able to identify the relationships between the variables, can forecast emergent behaviour, can formulate new problems and can devise computational models to solve them (Aho 2011).

2.4.3 What constitutes computational thinking

Computational thinking is a set of skills which overlap computer science, mathematics and engineering skills (Wing 2006, Isbell, Stein et al. 2010) . Wing (2006) defines computational thinking as "solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science." (p. 33). I especially subscribe to the definition of computational thinking as proposed by Cuny, Snyder and Wing (2010) when they state that Computational Thinking is "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent." I would like to stress three main points from this definition of computational thinking:

Computation thinking is about ***thought processes*** and not about a computer. A computer can be useful but the thinking that goes behind is what is really useful.

Formulating problems and their solutions: Computational thinking is not just about finding a solution to a given problem but it is mostly about identifying a problem and then finding a solution to it. As discussed in section 2.3.3.2 (page 61) the process of formulating problems and asking questions and then investigating a possible answer is fundamental for instilling creative practices in children.

That can be effectively carried out by an information processing agent: The form that is used to provide a solution to the given problem is important since this form must be in a way that a digital device (an information processing agent) can use it to solve the problem identified. This reminds me of Papert's technological fluency concept where the child is in control and tells the computer what to do.

This definition fits perfectly with the game making activities carried out by the children in this research project. Computational thinking will be used to design and build a game. Through the process they will encounter problems and solve them whilst expressing themselves in a way that the machine can use to give life to the game they create.

The question that springs to mind is: what are the thought processes that are identified as computational thinking skills? In a review of literature about computational thinking (Ahamed, Brylow et al. 2010, Barr, Stephenson 2011, Bers 2010, Bryant, Chinn et al. 2009, Bundy 2007, Deng, Huang et al. 2009, Denning 2009, Dierbach, Hochheiser et al. 2011, Good, Romero et al. 2008, Henderson, Cortina et al. 2007, Howland, Good et al. 2009, Hu 2011, Ioannidou, Bennett et al. 2011, Perković, Settle et al. 2010, Qin 2009, Weller, Do et al. 2008, Wing 2006, Wing 2008, Yadav, Zhou et al. 2011) I identified a list of these skills. The list of skills listed below are limited to those that pertain to building computer games.

2.4.3.1. Problem Decomposition

Problem decomposition is the process of breaking a problem into smaller more manageable problems. The whole process of creating a game signifies sub-dividing the game into different characters and then creating code to enact each character. The process of decomposing the game also identifies the rules that govern the game and events in the game timeline which trigger characters to appear or disappear.

2.4.3.2. Abstraction

Abstraction refers to the process of separating ideas from specific instances of those ideas at work in order to avoid duplication of code. Each level in a game might contain a score, hence rather than recreating the score logic in every level, the score logic can be abstracted to work with all levels. Abstraction also introduces the concept of reusing and remixing. Ideas from other games can be reused and altered to fit in the current game being built.

2.4.3.3. Algorithms

Another computational thinking skill identified in literature is algorithms. Algorithms are defined as a series of ordered steps taken to solve a problem or achieve a goal. The order the steps are placed in might involve linear sequencing where steps are placed one after the other, condition sequencing where steps are performed based on the result of some condition and looping where steps are repeated until some condition is met. In the case of games, the implementation of the character movement on screen, their state in the game and rules which govern the game are all implemented using algorithms devised by the game creator.

2.4.3.4. Automation

Automation is defined as having the machine implement repetitive tasks. Automating the rules that govern a game involves the computer repeating the algorithms that checks the rules coded by the game developer repeatedly over the course of the game, until the rule which governs the end of the game is activated.

2.4.3.5. Parallelisation

Parallelisation involves executing routes simultaneously. In games this skill is required since multiple rules enforced in a game need to be enforced at the same time. There is usually more than one character that is enacted in the game and each of these need to be handled at the same time.

2.4.4 Developing computational thinking whilst building a game

For this research project I follow a computational thinking framework presented by Brennan and Resnick (2012). This framework looks at how design-based learning activities in particular programming interactive media supports the development of computational thinking in young people. Brennan and Resnick's model also focuses on the use of the programming language Scratch which is also used in this research project.

The model proposed by Brennan and Resnick presents three computational thinking dimensions: computational thinking concepts, computational thinking practices and computational thinking perspectives and then explores how to assess the computational thinking dimensions the children gained whilst building interactive media. I link the thinking dimensions identified by Brennan and Resnick with the computational thinking skills identified in section 2.4.3, in section 2.4.4.4 below.

2.4.4.1. Computational Thinking Concepts

Brennan and Resnick define computational thinking concepts as the concepts that designers employ as they program their interactive media artefacts. These concepts relate to how the children place the Scratch statements (referred to as building blocks in the Scratch environment) to achieve a result in the interactive media artefact (game in this project) they are building. These concepts are present in a number of programming languages. In the list below I provide a brief explanation of each concept, along with an example based on the popular game *Pacman*.

Sequences: Creating an effect by using a series of building blocks in a serial fashion. Pacman, the round yellow character most of us are familiar with from the arcade game with the same name, is animated on a screen as a character that moves whilst opening and closing its mouth. This movement would be created by a programmer by issuing four commands one after the other:

1. a move command to displace Pacman a few pixels
2. a change image command to show Pacman with its mouth open
3. a wait command to do nothing for a few milliseconds
4. a change image command to show pacman with its mouth closed

Conditionals: An important concept of making interactive media is the ability to perform an action based on certain conditions. In the case of the Pacman game, Blinky, Pinky, Inky and Clyde change colour to blue when Pacman picks up a power pellet.

Loops: In the Pacman example, the four-step sequence would be repeated until the Pacman character reaches a wall in the maze. Loops allow the game creator to repeat a sequence of commands for a number of times or until a condition is reached.

Events: The change of colour in the Pacman game is triggered by Pacman picking up a power pellet. The picking up of the power pellet is known as an event and code can be executed when the event is triggered.

Parallelism: Parallelism refers to executing two or more code blocks concurrently. In the case of the Pacman game when Pacman is moving, Blinky, Pinky, Inky and Clyde are moving too. A different routine is used to handle the movement of each character with the routines executed in parallel.

Operators: Operators support mathematical, logical and string manipulation. These are needed at different instances throughout the life time of a game. Whenever Pacman eats a dot the score is incremented. The incrimination of the score occurs through a mathematical operator.

Data: Data refers to the storage, retrieval and updating of values. The score in the Pacman example is a type of data used in the game.

2.4.4.2. Computational Thinking Practices

The second dimension of the model presented by Brennan and Resnick looks at the practices adopted by the children whilst building interactive media artefacts. Whilst observing and interviewing children building their interactive media artefacts Brennan and Resnick observed four main set of practices:

Being incremental and interactive:

The process of building interactive media tends to be an iterative process. The plan tends to change in response to experiences and new ideas and the children reiterate through the building process until they are happy with the interactive artefact created.

Reusing and remixing:

Children tended to reuse and remix projects created by others. In their discussion Brennan and Resnick identify two areas of reuse. Children tended to get the idea for their project from other projects on the Scratch website and they also tended to get coding ideas from other projects they download and remix.

Abstracting and Modularising:

Whilst building their projects the children tended to subdivide the projects into smaller building blocks. This important practice was useful for design and problem solving purposes.

Testing and debugging:

When developing a game or an interactive media artefact, it is quite normal to encounter problems with the code developed. Brennan and Resnick identified the following strategies adopted by the children they observed and interviewed:

- Reading through the script in a bid to locate the problem.
- Trial and Error experimentation.
- Finding example scripts that worked and then adapting the logic used to the situation they were trying to solve.
- Getting support from the knowledgeable others.

2.4.4.3. Computational Thinking Perspectives

The third dimension presented in this model is the computational thinking perspectives dimension. These are the viewpoints formed by the designers about their relationships with others and the technology world around them. Brennan and Resnick list three perspectives they identified in their studies: expressing, connecting and questioning perspectives (Brennan, Resnick 2012).

Expressing

The children experienced building interactive media as a means of self-expression. The computer is seen as a medium with opportunities to express oneself by creating something new.

Connecting

The experience of building interactive media was enriched by interactions with others. The student's interaction was two way with students working *with* others as well as working *for* others.

The working with others interactions ranged from having questions answered in the discussion fora to studying code of media creations uploaded by other Scratchers. At times the students also formed partnerships with other students to work together on projects.

The students also found themselves working for others. The *for others* occurred when the students were asked to create assets to be used in the interactive media created by others. This was not the only instance of *for others* connections. Students built an audience which they interacted with. It was quite usual for children to create polls about ideas on their next games or to create tutorials on how particular features worked.

Questioning

Young people do not feel the disconnect between the technology that surrounds them and their abilities to negotiate the realities of the technological world (Brennan, Resnick 2012, p.11). Instead they feel empowered to question about and with technology.

2.4.4.4. Relating Computational Thinking Framework with Computational Thinking Skills

The three computational thinking dimensions (computational thinking concepts, computational thinking practices and computational thinking perspectives) identified by Brennan and Resnick are closely related to the computation thinking skills identified in section 2.4.3. Decomposing a problem involves identifying similar traits in other projects in a bid to reuse the ideas. The decomposition process tends to be incremental and iterative and this leads to the solution being more modularised. In Table 3 below I link the computational thinking skills identified in the literature with the computational thinking skills used in the framework presented by Brennan and Resnick:

Problem Decomposition	• Being incremental and iterative • Reusing and remixing • Identifying events • Abstraction and modularizing
Abstraction	• Abstraction and modularizing
Algorithms	• Sequencing • Loops • Conditions • Operators • Data
Automation	• Using Events • Testing
Parallelisation	• Parallelism

Table 3 Relationship between computational thinking skills (2.4.3) and the model proposed by Brennan and Resnick (2012)

2.4.4.5. Assessing computational thinking

How to assess computational thinking has received considerable attention over the past years (Basu, Kinnebrew et al. 2014, Brennan, Resnick 2012, Franklin, Conrad et al. 2013, Grover, Cooper et al. 2014, Werner, Denner et al. 2012).

Brennan and Resnick used three assessment approaches to assess the development of computational thinking in young people who are engaged in design activities with Scratch.

2.4.4.5.1. Project Analysis

In this approach an artefact is analysed to identify the different constructs used to build it. In the study by Brennan and Resnick a tool called Scrape was used to list the different constructs used. Although this approach was usuful to identify which Scratch construct were used it revealed nothing about the process of developing the project and hence could reveal nothing about the computational thinking practices that were employed.

2.4.4.5.2. Artefact based interview

The second approach used was to interview the students. Students were interviewed about their experiences of Scratch and about their motivations for building the artefact.

The students were also asked to explain the process of creating the project, how they got started and how the project evolved over time. They were also asked about what was important in order to build the artefact, the problems they encountered throughout the process and how they dealt with these problems.

This approach highlighted a weakness in the first approach. Although the use of a construct might have indicated apparent fluency it was only after talking to the students that significant conceptual gaps were unearthed. Brennan and Resnick were not present whilst the artefact was being built and hence they relied on what the students remembered rather than seeing the practices in real-time. At times they met students who simply said that they did not encounter problems whilst building the artefacts. Not being present whilst the artefact was being built was perceived by Brennan and Resnick as a weakness of the artefact based interview approach.

2.4.4.5.3. Design Scenarios

The third approach used was the design scenarios approach. Three sets of Scratch projects were created. Each set was increasingly more complex than the preceding one. Each set included two projects each engaging with the same concepts but had different aesthetics to appeal to different interests. The students were asked to select one project from each set of scenarios. For each project they were asked to:

- Explain what the project does
- Describe how it can be extended
- Fix a bug
- Remix the project by adding a feature

This approach offered notable strengths when compared with the other approaches.

1. Through the questions on each project the interviewer had the opportunity to systematically explore the abilities of the children to critique, extend, debug and remix an existing artefact whilst testing the fluency with different concepts and practices.
2. The design scenarios were designed to be increasingly complex and hence the interviewer could gauge how the student faired using a developmental approach.

3. The scenarios emphasised a *process in action* rather than a *process via memory* approach.

2.4.4.6. Adopting the framework for this project

None of the assessment approaches identified by Brennan and Resnick are ideal. Artefact analysis on its own does not bring to the forefront the process adopted whilst building the game. An interview at the end of the process relied too much on the children's memory whilst the framework approach asked the children to work on ready-made scenarios hence denying the children the opportunity to express themselves by building an artefact.

For this project the children were given the opportunity to express themselves by building a game artefact they designed themselves. I analysed the games at different stages throughout the building process. The children were asked to save a version of the game at least once a week in a bid to capture the process of building the game. I also held regular informal discussion with the students rather than an interview at the end. In this way I was able to see and analyse the process as it happens rather than relying on the memory of the children.

2.5 Adopting a workshop model for this research project

A challenge I face in this research project is to integrate teaching for creativity in a creative way (see section 2.3.3 page 58), promote systems and design thinking (see section 2.2.6.1 page 38) and introduce the children to Scratch whilst promoting computational thinking (see section 2.4.2 page 72).

Keeping in mind the problem-based approach to the introduction of programming (see section 2.4.1.5 page 69) I decided to introduce Scratch by demonstrating a game project that works and get the children to try the game project out by playing the game. Rather than introducing specific computation thinking constructs in Scratch, I used the game to stimulate discussion with the students on how features within the game were constructed. In this way the students were able to experience the constructs used in action. The children could see a working product immediately and experiment with modifying the constructs and seeing the effect this modification had on the game. In line with implementing a full system approach (see section 2.4.1.5.2 page 71) building a product from scratch was only introduced at the last stage of the process. The sequence of progressively more complex activities that can be used to introduce students to programming whilst using a full systems approach suggested by Kölling (2008) have been amended and adopted for this project. In the next sections I outline the sequence of activities followed throughout the workshop.

2.5.1 Get your feet wet: executing code

Whilst keeping in mind that programming is seen as a difficult skill to master, I decided to create a simple game to keep the programming to a minimum so as not to create an information overload. However the basics traits of digital games that is, to have a goal, rules and a feedback system (see section 2.2.4 page 32) were all included. The simple game contained two characters a shark operated by the player and a fish operated by the machine. The goal of the game was to eat as many fish as possible. Whenever a fish was eaten a new fish appeared at the centre of the screen. The rules governing the game were very simple, a player had to move the shark to eat a fish and in so doing score points. A number of feedback mechanisms were included in the game. A point system was included to display the points attained by the player whenever the shark eat a fish. The shark was animated to open and close its jaws whenever a collision with a fish occurred. The opening and closing of the shark's jaws was accompanied by a sound effect too so that a multimodal feedback was created.

Characters	Shark: Player operated Fish: Machine operated
Goal	Shark must eat as many fish as possible an in the process scoring points
Rules	When a shark eats a fish a point is scored *Shark* is moved with left arrow is pressed on the keyboard *Shark* turns in a clockwise direction when the up arrow is pressed on the keyboard When a Fish is eaten it is re-spawned at the centre of the screen
Feedback	A score counter is displayed on screen When is in same position as Fish, Fish is hidden to simulate eating When *Shark* is eating *Fish* the image of the Shark is switched to show a shark with open jaws. A sound is played when a *Shark* eats a *Fish*

Table 4 Defining traits of Shark and Fish

2.5.2 Manipulate source code

In the manipulate source code phase of the project students are usually given tasks to change the source code to solve a problem intentionally placed in the original game. In this project rather than provide the students with ready-made tasks I asked the students to come up with suggestions themselves. I then sorted the suggestions according to popularity and ease of implementing the tasks. We started discussing tasks that were popular and at the same time did not require the addition of new sprites. In this way the students could add the new requirements by simply changing the existing code.

2.5.3 Create a new behaviour and adding more building blocks

In the next phase of the workshop the students implemented changes to the game that required the addition of new sprites. Here again the suggestions on what new features to add to the game originated from the children themselves.

2.5.4 The master test

The last phase of the workshop consisted of the children creating their own game from scratch. Here again the children were kept at liberty on the kind of game to create. This phase of the workshop was the longest phase spanning over six weekly sessions.

2.5.5 Teaching for creativity

In section about teaching for creativity (section 2.3.3 page 58) I outlined a series of suggestions found in literature on how to teach for creativity and how to foster possibility thinking. One of the main suggestions included passing control to the student and encouraging innovative contributions. *Shark eat Fish* was intentionally kept simple and I was sure that the children could see a number of limitations in the game and hence could provide suggestions on how to improve the game. After playing the simple game the children were asked to provide suggestions on how the game can be improved. In this way even though the children were provided with a ready-made game they could claim ownership for any modifications they suggested. Placing value in the student's ownership was another of the suggestions found in literature. Some of the more popular suggestions provided by the children were used to kindle a discussion on how to implement the new requirements. Reference was made to the way the features already in the game were implemented so that the children could understand how Scratch works and how to leverage the Scratch features to implement the suggestions they outlined. One of McWilliam (2008)'s suggestions was to devote less time giving instructions and spend more time being a useful co-worker in the tick of action. In order to follow this suggestion the discussion on how to implement the feature and a demonstration of the implementation was held during the initial part of the session. After the discussion the students were allowed to implement their own changes to a copy of the Shark and Fish game. The children were allowed to implement the changes they wanted and to make the game their own. In this way experimentation was encouraged and the children were encouraged to believe in their creative identity. In order to consolidate the learning that occurred during the discussion sessions a video on how to implement the suggestions to the game was uploaded to the game workshop website. In this way reinforcement of learning was provided and the children could review how game features discussed in class were implemented from the leisure of their home.

Throughout the whole process opportunities were created for the children to debate and discuss their thinking. Children were encouraged to identify problems, pose questions and to discuss possible solutions. The role of the teacher was more of a collaborative critic asking open-ended questions on the work being done by the children rather than someone who provides all the instructions and answers all the questions posed.

Co-operation was central to the design of the gaming workshop. Children were encouraged to test each other's game and help each other out. Children could choose to work together with another member of the workshop or on their own.

2.6 Conclusion

As discussed in the introduction the aim of this research project is to use game development as an avenue for children to express their creativity whilst engaging with computational thinking.

In this literature review I looked at the Maltese situation and the value placed in the NCF on the notions of making in the design and technology strand, programmed control in digital literacy strand and creativity and innovation in the education for entrepreneurship, creativity and innovation strand.

Given the importance of gaming for the Maltese children, I explored the notion of gaming as an expression of play and looked at various definitions found in literature for the term digital game. Through the analysis of these definitions I identified tangible characteristics that can be used to describe the games created by the children during this research project.

Whilst building their capital about games by playing and discussing games children become versed in a new kind of literacy based on game design which Zimmerman (2007) calls gaming literacy. The notion of gaming literacy was explored in section 2.2.6 (page 37) by examining the main attributes of systems, play and design. Reflecting on Gee's assertion that there is more to gaming than the game in section 2.2.7 (page 43) I explored the notion of affinity space (Gee 2004). The properties of an affinity space were used in this project to influence the design of an online space what the children used throughout the workshop in order to extend the experience outside the classroom walls.

I concluded the section on games by looking at a model that was created for planning and helping educators understand more what children were doing whilst playing games. I use this model to help me understand the gaming literacy practices which the children picked up whilst building their games.

Since during this research project the children experience the process of making a game the next section of the literature review explored the literature around making and creativity and how teachers can teach for creativity. Creative processes were discussed and compared to the processes children go through whilst creating games found in literature.

The games will be created using Scratch a multimedia programming language aimed at children. Through this literature review I reflected on the possible approaches that can be used to introduce children to programming. Similarly to the literature about creativity, the available literature about introducing programming to children was used to shape the design of the game making workshop. Programming is part of a wider concept, that of computational thinking. Children will be engaging in computational thinking skills whilst building their game. The concept of computational thinking was explored and a framework that will be used to gauge the level of computational thinking skills explored by the children was discussed in section 2.4.4.5 (page 80).

In the next section I discuss the methodology adopted in this research project and how I structured the project to answer the guiding research questions.

3. Methodology

3.1 Introduction

This research deals with evaluating the creative processes that students go through whilst authoring video games. Through the project I reflect on the game literacy benefits the students attain whilst authoring the game. Given the nature of this project I adopted a qualitative research methodology considering how Denzin and Lincoln characterise qualitative research

> Qualitative research involves the studied use and collection of a variety of empirical materials that describe routine and problematic moments and meanings in individual lives. (Denzin, Lincoln 2005, p.3)

Researching the creative process that students go through whilst creating digital games and the game literacy benefits that ensue fits the criterion of "routine and problematic moments and meanings in individual lives". Qualitative research is based on the premise that individuals construct their reality by interacting with their social worlds (Merriam 2009). Thus qualitative researchers are interested in understanding how people make sense of their lives and their experiences. According to research, video games are an integral part of the children's world as they spend a significant portion of their time playing games (Brand 2012, Busuttil, Camilleri et al. 2014, Games 2008, GameVision Europe 2010, Lenhart, Kahne et al. 2008). In this research I want to understand how the gaming literacy they are picking up whilst playing the video games and other skills students bring with them are reflected in the games they create. This qualitative project focuses on a group of eleven year olds as they interact together and build computer games. Since this study works on a bounded system I decided to follow a qualitative case study approach.

3.1.1 Case Study Research

The term case study research has a host of different meanings in different disciplines (Carter, Sealey 2009, Simons 2009) with almost every author on the topic presenting his/her own definition of case study research (Swanborn 2010) . However notwithstanding the differences there seems to be consensus on key elements of case study research in the definitions supplied by Simons (2009), Yin (2008), Swanborn

(2010), Merriam (2009) and Hamilton (2013) For these authors case study research deals with:

3.1.1.1. A bounded system

Merriam states that “the single most defining characteristic of case study research lies in delimiting the object of study” (2009, p. 40). In my research the case under study is the group of students who volunteered to stay on after school every Friday to create their own digital games. Although my research questions look at the creative processes the students go through whilst creating the game and how the game creation process affects their gaming literacy the bounded unit is the group of students.

3.1.1.2. Real-life context

The bounded system, in this research the group of students, does not exist in an inert world but is located in its real context, in this case the school lab where the game creation takes place and the course website where the students using their individual user accounts login and interact with the other members of the group. The students interact with school mates not participating in the workshop and with members of their families, and this interaction feeds back into the bounded system. There are regular interactions between the bounded system under study and the wider world in which the bounded system is situated. I spent time immersed in world of those being researched to capture the complexity of the case.

3.1.1.3. Using several data sources to collect rich data

In case study research a variety of data collection tools are used to collect data and provide depth. Such data sources are typically qualitative in nature and include interviews, participant observations, reflective journals and document analysis.

3.2 Role of the researcher

The role of the researcher in a qualitative study is central. The researcher not a questionnaire is the primary instrument for data collection and analysis (Merriam 2009). The advantage of this is that the researcher can adapt to the situation whilst collecting the information. The qualitative researcher is compared to a “bricoleur”, a quilt maker or a jazz improviser in Denzin & Lincoln (2005) deploying whatever strategies, methods and empirical materials at hand depending on the context and what the researcher can do in that setting.

One could argue that the centrality of the researcher in qualitative research brings with it shortcomings since the subjectivity of the researcher can affect the study. However rather than trying to eliminate subjectivity which is not possible, Simons (2009) suggests concentrating on showing how researchers' values, predispositions and feelings impact on the research.

3.3 Preparatory Project

During the summer of 2012 I led a series of ICT sessions to a group of twenty-two, eleven year old students who were taking part in a summer school. The group of students was composed of a mixed group of boys and girls coming from different schools. Most of the students had attended previous summer schools and knew each other well.

The summer school was held over eight weeks. Throughout the week the children took part in various craft, sports and acting sessions. They also participated in an ICT class once a week. The group was led by the same teacher throughout the day. Although the teacher was present during the ICT classes she did not take an active part in the class.

Since I was given a free hand on the topics to discuss during the sessions, I decided to focus the sessions on game authoring. After trying out a number of game authoring tools suitable for eleven year old children I decided to use the Scratch programming language especially designed for children to use. Although it is easy to get started in Scratch, there is the opportunity to create complex projects over time.

The eight weeks were split into two sections. During the first three weeks of the course I strived to get the students familiar with Scratch whilst during the last five weeks I encouraged the students to team up and design and implement their own mini-games. During the last five weeks I acted as a consultant helping the students whenever they encountered problems whilst building their games.

On the first day of the course I presented the students with a game I implemented in Scratch. The game worked but was not complete. I asked the students to play the game and then suggest improvements. In order to suggest the improvements I asked the students to fill in an online questionnaire. I then collated the suggestions, created a top-ten list of improvements and over the next three weeks showed the students how to implement the changes they suggested. In this way I choose to let the students take the driving seat of their own learning and suggest the features they wanted added to the initial game. Whilst discussing how each feature could be implemented I was introducing the Scratch constructs required to make the feature work.

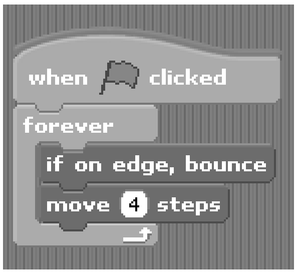
Since I did not seek and attain consent from the students participating in these classes I will not be discussing the games authored by the participants. This exercise was very beneficial in helping me fine tune the structure of the game authoring workshop I conducted as part of my research. Following this experience I decided to:

- Create a website to use throughout the course. This website was meant to increase the contact with the workshop participants between sessions and provide an online space where the students could save their games so that I could view the games too.
- Increase the number of sessions from eight to ten in order to give the students more face to face time whilst authoring their games.
- Limit the number of participants to twelve. Conducting the sessions with twenty-two students did not allow me to allocate enough time to each student.
- Create a series of videos to demonstrate how to create different mini-games using Scratch for the students to view at their leisure. During the summer school session the students were only exposed to the design and implementation of one game.
- In order to encourage participation in the game creation activity I introduced a “game designer of the week” title for the first four sessions. The aim of this title was to use gaming inspired activities to entice the students to engage in the game creation early on in the project.
- Recruit a group of teachers to help with the mentoring of the students whilst they authored their games. In this way I would have more time to observe the students whilst creating the games.

- Approach a co-ed school to participate in an after-school workshop. Most of the schools in Malta are single sex schools[4], however since both boys and girls created games by the end of the summer sessions I wanted to see if both boys and girls participate in a game authoring workshop out of their own will.

3.4 Scratch

Scratch is a graphical programming language developed by the Massachusetts Institute of Technology (MIT) media lab's Lifelong Kindergarten Group. Scratch is named for the way hip-hop DJs creatively combine pieces of code using a technique called *scratching* (Chiu, Lui et al. 2012) . In a similar way programmers in Scratch join different media such as images and sound effects to create games and animations. The Scratch project began in 2003, and the Scratch software and website were publicly launched in 2007 (Maloney, Resnick et al. 2010).



```
when ⚑ clicked
forever
  if on edge, bounce
  move 4 steps
```

Figure 3.4-1 Building blocks in Scratch

3.4.1 Core features of Scratch

Resnick, Kafai, & Maeda (2005) outline the following five core features of this programming environment:

3.4.1.1. Building-block programming.

Scratch programming uses a building-block metaphor, in which learners build procedures by snapping together graphical blocks in a similar way to using LEGO bricks or snapping together pieces of a jigsaw puzzle.

3.4.1.2. Programmable manipulation of rich media.

To help users make their projects personally engaging, motivating, and meaningful, Scratch makes it easy to import or create many kinds of media (images, sounds, music). A built-in paint tool and sound recorder are available in Scratch allowing users to change or create media. This media can then be manipulated using programming blocks

[4] In April 2013 the minister for education and employment announced a pilot project that would eventually see the introduction of co-education in all Form 1 state schools. (MEE 2014)

3.4.1.3. Sharing creations.

The Scratch website provides a social context for Scratch users, allowing users to share their Scratch projects, receive feedback and encouragement from their peers, and learn from the projects of others. Scratch encourages the sharing of creations by including a share menu in the application itself allowing users to upload their creation onto the Scratch website. Scratch projects can be downloaded from the Scratch website and re-mixed by anyone who has a Scratch account. Sharing and remixing encourages users to learn through exploration and peer sharing, with less focus on direct instruction.

3.4.1.4. Integration with the physical world.

Scratch allows programmers to interact with physical objects (such as motors, lights, MIDI synthesizers) in the same way they program virtual objects on the screen. Scratch also allows programmers to use input from physical sensors (distance sensors, motion detectors, sound sensors) to control the behaviour of both physical and virtual objects.

3.4.1.5. Support for multiple languages.

In order to facilitate content creation by programmers who might not be English speakers the Scratch environment is made available in fifty languages. The language can be effortlessly changed allowing a programmer to work and think in a comfortable environment.

3.4.2 The low-floor/high-ceiling/wide-walls mantra

Scratch offers students the possibility of creating any kind of game by importing rich media or by creating the game assets themselves. This was one of the main features which led me to choose Scratch over other game creation environments. Other game creation environments such Kodu and Adventure Author typically offer the students ready-made avatars which the children control. Although this simplifies the game creation experience this limits the children in the choice of games they can create.

Scratch supports the low-floor/high-ceiling/wide-walls mantra. It is easy to get started in Scratch (low floor) and there is the opportunity to create complex projects over time (high-ceiling). Scratch was designed to support different types of projects, so that people with different interests and learning styles can all become engaged (wide-walls).

3.5 Participants and Research site

Following the experience gained during the preparatory project (see section 3.3 page 91) I decided to conduct the gaming workshop over a ten week period.

In order to conduct the after school workshop I had to look for a school interested in participating in this research project.

3.5.1 Recruiting a school

I approached the junior school director of *Aschool*[5] to enquire whether the school would be interested in running an after school game making club. *Aschool* is a private co-ed school catering for early years students up to senior school students. The school has an active after school programme where students stayed after school to participate in organised sports and drama activities. I argued that the game making workshop would be a natural extension to the existing after school programme since a lot of students spend time playing digital games in their free time and would probably be interested in learning how to create their own games. I provided *Aschool* with a project information sheet which I had submitted to the University of Sheffield ethical review process.

Aschool agreed to participate in the project and to make available the school computer lab to be used during the after school game making activity. *Aschool* also agreed to help setup a meeting with all the students in senior one class (11 year olds) so that I could explain to the students the purpose of the research project and explain the process of enrolment into the after school game authoring class. During the meeting we also agreed on the procedure to adopt should more than twelve students apply to join the after school project. We agreed that the first twelve participants will be selected by ballot. The rest of the students will be offered the possibility to join a second club later in the year.

3.5.2 The school lab

The gaming workshop took place in the school's computer lab. The computer lab consists of a longish rectangular room equipped with twenty computers for student

[5] Name of participating school has been changed for confidentiality purposes

use. The student computers face three walls of the room. The fourth wall was reserved for the interactive whiteboard and the teacher's desk and computer. Entrance to the computer lab was from a large glass door next to the whiteboard. The layout of the computer lab did not allow the students to comfortably look at the interactive whiteboard whilst working on their computers. Hence I planned to ask the students to move their chairs to the central part of the room for the first part of the session and to move them back to the computer stations when they had to work on their games.

The lab was used by the senior school students during their school hours. Each class visited the lab for 45 minutes once a week to participate in ICT lessons. The ICT lessons centred around learning how to use the Microsoft office package and the internet in order to sit for the European computer driving license (ECDL). The lab machines were also used by older students (13 to 15 year olds) who choose computer studies as an optional subject. Although the computers were networked to allow centralised access to the internet there was no centralised user accounting system. Each student used a common user account to log into the computer. This meant that any information saved on the computers was easily deleted or changed by other students using the lab at different times. Students were encouraged to save their work on pen drives which they carried with them to school.

3.5.3 Recruiting students

In October I visited *Aschool*. Ms M from the senior school management team introduced me to the senior one students. The meeting was attended by the two senior one classes totalling thirty one students. I asked the students about their digital gaming habits and whether they would be interested in participating in a project aimed at empowering them with game creation skills. I explained that the project will be open to twelve students who would be willing to stay on after school on Fridays. I distributed a project information sheet which included a participation form to the students. Students who were interested in joining the after school club were asked to return the form signed by a parent to Ms M.

3.5.4 Recruiting teachers

In order to recruit two teachers to help me during the project I decided to approach former students of mine who were now working as teachers of computer studies in secondary schools and enquire whether they would be willing to participate in this project by acting as mentors to the students. I decided to recruit two teachers rather than one since this gave me the flexibility of continuing the project even if one of the teachers had to miss a session due to some other commitment.

The first stumbling block I encountered whilst recruiting the teachers was the timing of the after school sessions. The school day at *Aschool* ended earlier than other schools and the workshop was scheduled to start at a time when the teachers were still at the school they taught in.

The location of *Aschool* was also an issue with some of the teachers who worked on the other side of the island. They raised their concern that travelling to *Aschool* in the after school traffic would take time.

Notwithstanding these issues two teachers agreed to participate in the project if their schools had to release them early on Fridays. I wrote to their respective head of schools explaining the purpose and importance of the project, and the important role that the teacher would play in it. Both heads of school agreed to release the teachers to participate in this project.

3.5.5 The online space

Since the workshop was to be conducted once a week and I had no contact with the students in between the workshop sessions, I decided to create a web portal to be used by the workshop participants. The web portal was designed with the features of affinity space in mind (see section 2.2.7 page 43).

As Gee stresses, an affinity space is “not an all-or-nothing thing” (Gee 2004, p.85). Although affinity spaces do not segregate people by age due to the nature of this research project the students were all from the same age bracket.

In this research project the common endeavor so central to an affinity space is the game making activity. The children have volunteered to join the workshop demonstrating their interest in the activity. Teachers and students shared the physical classroom space as well as the general discussions area of the online portal and students were encouraged to reply to questions generated by their peers in both the physical and online space. In an affinity space content is transformed by interaction. In this workshop the students had the possibility of suggesting changes to the initial game scenario and hence effect which topics are discussed in class. Individual and distributed knowledge was encouraged with children using each other as resources

One of the central principles of teaching for creativity is the passing of control to the learner and this ties itself quite well with the concept of an affinity space since the children were empowered to take on leadership roles especially when mentoring other students about how particular features in their games were created.

To create the online space I purchased a domain to be used during this project and built a portal[6] using Google sites. Google sites offered me the flexibility of designing the portal according to my design and to secure different areas of the site so that only workshop participants could access these areas. I considered security to be an important feature of this portal given the young age of the participants.

The portal included:

- A session summary page
- A discussion area
- A resource section
- A student's area
- A game maker of the week facility
- A team area

I discuss each area in further detail in the sections below:

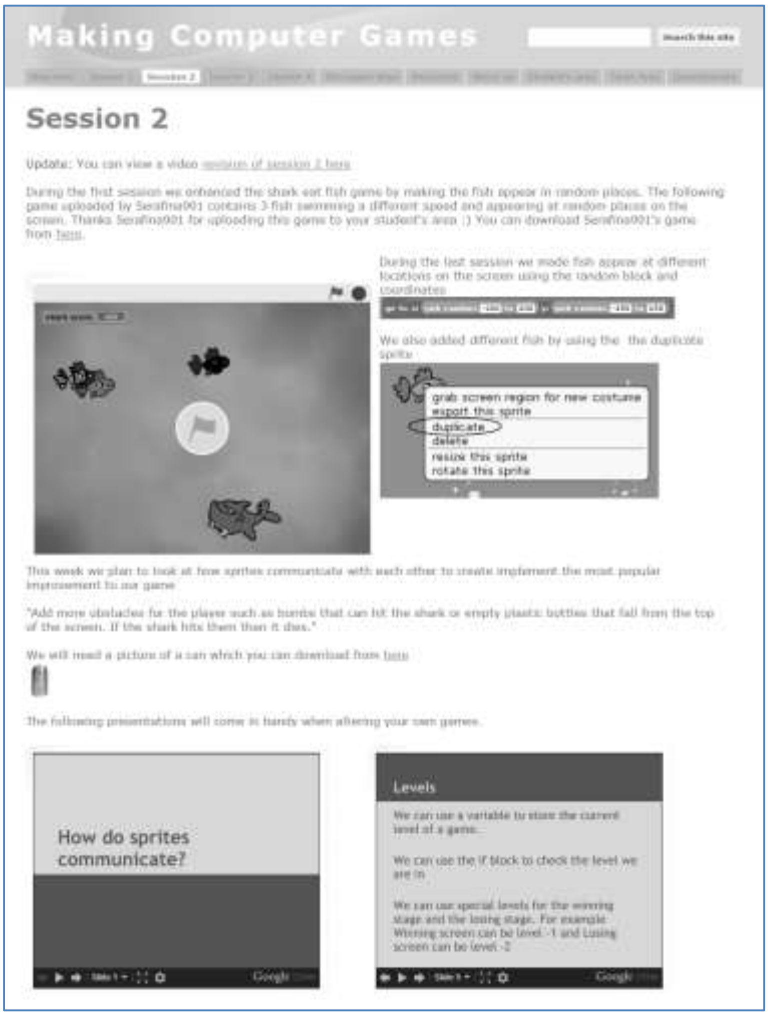
3.5.5.1. Session summary page

Before every session the portal was updated with a new page which included presentations to be used during the session and a description of the features to be discussed (see Figure 3.5-1 on page 99).

The pages also included a link to one of the games created by one of the students at the end of the previous session. This game provided a base line so that students returning to the page at a later stage could download the game and practice the features discussed on it.

[6] The portal for this research project is located in http://www.makingcomputergames.com

After every session I recorded a video outlining the features discussed. This video was uploaded to the video sharing site *YouTube* and linked to the page. In this way the students could review the session at a later stage.



Making Computer Games

Session 2

Update: You can view a video revision of session 2 here

During the first session we enhanced the shark eat fish game by making the fish appear in random places. The following game uploaded by Serafina001 contains 3 fish swimming a different speed and appearing at random places on the screen. Thanks Serafina001 for uploading this game to your student's area :) You can download Serafina001's game from here.

During the last session we made fish appear at different locations on the screen using the random block and coordinates

We also added different fish by using the the duplicate sprite

This week we plan to look at how sprites communicate with each other to create implement the most popular improvement to our game

"Add more obstacles for the player such as bombs that can hit the shark or empty plastic bottles that fall from the top of the screen. If the shark hits them than it dies."

We will need a picture of a can which you can download from here

The following presentations will come in handy when altering your own games.

How do sprites communicate?

Levels

We can use a variable to store the current level of a game.

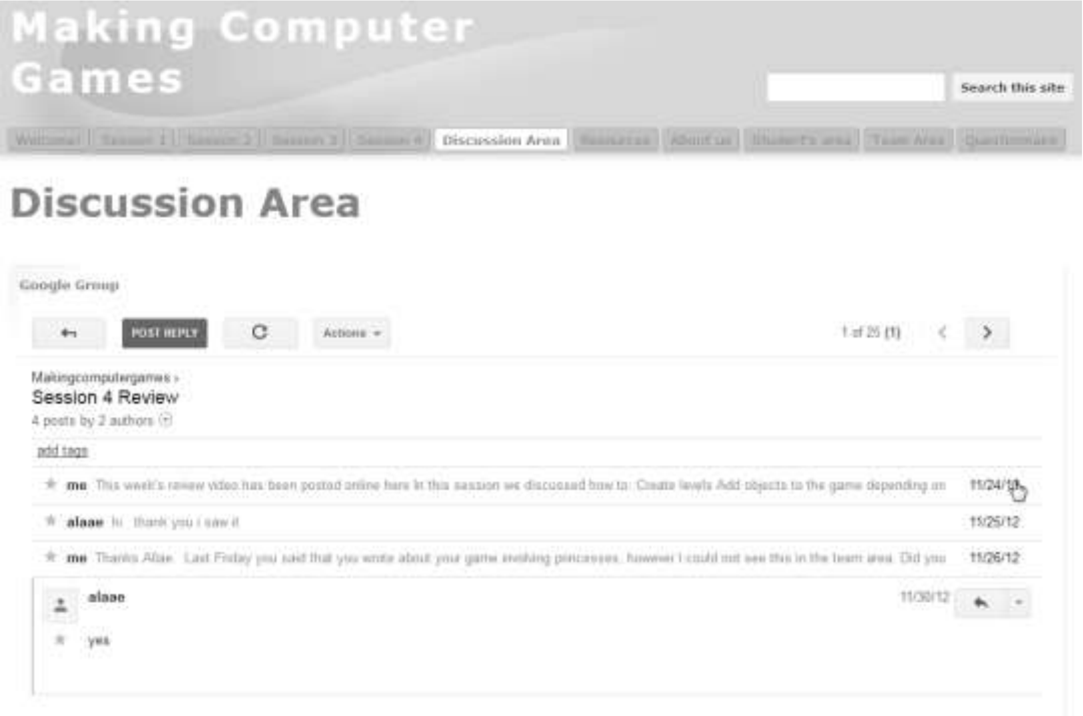
We can use the if block to check the level we are in

We can use special levels for the winning stage and the losing stage. For example Winning screen can be level -1 and Losing screen can be level -2

Figure 3.5-1 Session summary page

3.5.5.2. Discussion area

The game making workshop was held on a weekly basis. In order to allow the participants and the teachers to communicate whenever the need arose during the week, a google group was setup and embedded in the website. Each participant had to log in to the site using the security credentials provided at the beginning of the workshop. In this way all messages remained private to the group participating in the workshop. The teachers helping with the project and myself logged into the portal regularly to ensure that any questions posted by the participants were promptly answered by one of the teachers or by another student.



Makingcomputergames ›
Session 4 Review
4 posts by 2 authors

add tags

me This week's review video has been posted online here In this session we discussed how to: Create levels Add objects to the game depending on 11/24/12

alaae hi thank you i saw it 11/25/12

me Thanks Alae. Last Friday you said that you wrote about your game involving princesses, however I could not see this in the team area. Did you 11/26/12

alaae 11/30/12

yes

Figure 3.5-2 Discussion area

3.5.5.3. Resource section

Following the preparatory project (see section 3.3 page 91) I decided to add step-by-step video tutorials on how to build different games from the one being discussed during the initial part of the game making workshop. I added three games to the resource section of the portal: (1) Penalty shoot-out game, (2) Year 3578 Saving the earth and (3) Shooting a vase.

3.5.5.3.1. Penalty shoot-out game

The video outlines how to design and create a simplified version of the popular 'penalty shoot-out' game. In this game the player plays against the computer trying to score goals while the computer controlled character dives and tries to catch the ball. After using the mouse to choose the spot where the ball is aimed the player clicks on the left mouse button to shoot the ball. The goalie then dives and tries to stop the player from scoring. The objective of this video was two-fold. I wanted to introduce and reinforce programming constructs in Scratch ensuring the students are familiar with the constructs whilst the constructs are applied during the building of a game and also introduce game design features present in digital games.

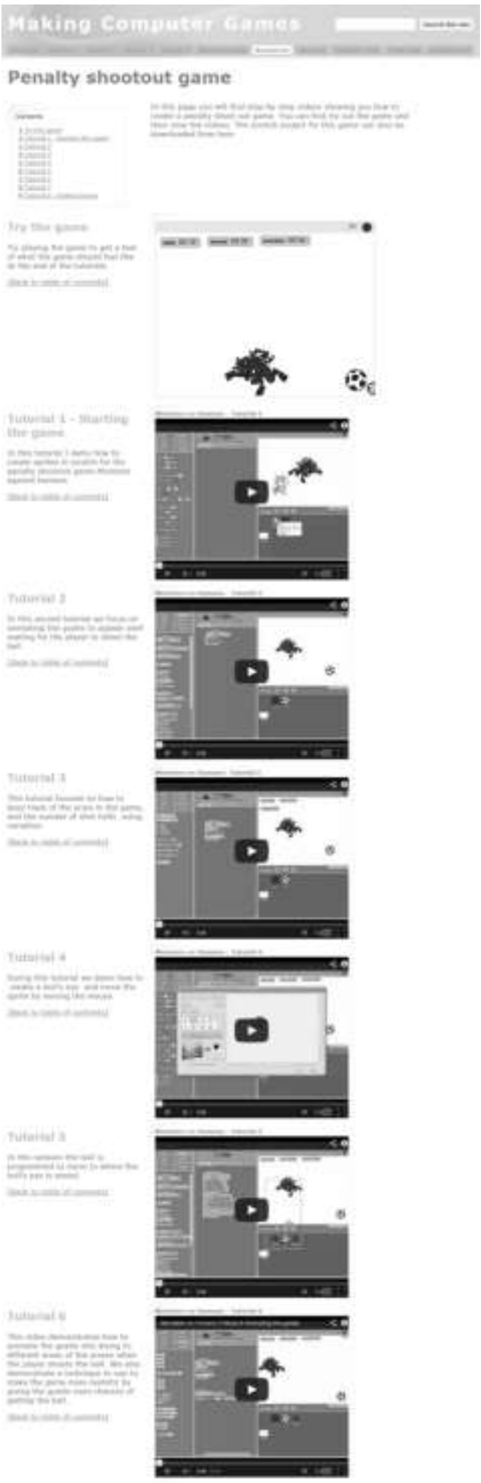


Figure 3.5-3 Resources section - Video tutorials

In this demo I wanted to introduce the students to using the mouse to control a character on the screen, since in the face-to-face sessions the game characters were always controlled by pressing different keys on the keyboard. I wanted to reinforce the use of variables to keep track of the number of balls shot by the player, the score of the game and the number of balls saved by the monster goalie. Finally I wanted to introduce the participants to a construct in Scratch which allows a sprite to glide to a location on the screen rather than simply appearing in the location. This was an important feature in animating the goalie and the ball since in real life a goal keeper jumps for the ball and does not simply appear in a place but moves to it.

In this game tutorial I wanted to introduce the participants to an important feature in games – managing difficulty. Csikszentmihalyi (1996) introduces the state of flow a state of concentration or complete absorption. For flow to occur the task at hand cannot be too easy or too difficult. The state of flow is especially relevant to the design of an engaging video game. If the game is too easy to play the player becomes bored. If on the other hand the game is too difficult, the player becomes anxious (Adams, Rollings 2007). If the goalie in the penalty shoot-out game always catches the ball, preventing the player from scoring the game would be too difficult, in reality impossible for the player to win. If on the other hand the goalie simply jumps to a random passion on the screen the game would be too simple for the player, resulting in a boring game. During the video tutorial I explained how to reach a compromise and let the goalie always guess the horizontal position of the location selected by the player and to randomise the vertical position. In this way the position is partially random. The goalie does not always get the ball but is always near, and has a better chance of getting the ball without making the game unwinnable.

3.5.5.3.2. Shoot the vase

The main objective behind this this video tutorial was to demonstrate how to code shooting and play sounds. The video game implemented is based on a simplified version of the popular game *Space Invaders*.

In this game a vase is displayed in random locations on the upper parts of the screen. The vase only stays in the same place for a short period of time (between one and three seconds). During this period the player has to move a paddle displayed at the lower part of the screen, position it just under the vase and shoot a bullet. Once the bullet hits the vase a bell sound is played. A player cannot shoot more than one bullet at a time making the game more challenging.

During this demo I wanted to show how a sprite can be made invisible. The bullet was a sprite that was only shown when the player pressed the spacebar and then turned invisible again once the bullet hits the vase or ends outside the screen. I also wanted to demonstrate that although invisible, a sprite is still on the screen and can have its position altered. The bullet was made to move in synchrony with the paddle at the bottom of the screen so that when the player presses the spacebar to shoot, the bullet would appear as if it is shot from the middle of the paddle.

3.5.5.3.3. Year 3578: Saving the earth

In this game the player is the commander of a space ship travelling through space trying to collect special energy rocks vital for the surviving humans on earth. The player must make sure to avoid meteors. Once a meteor hits the space ship it is turned to dust and the player loses the game. As the game progresses and the player collects enough *gigarock* more meteors start appearing making each level harder to play.

Through this video I wanted to introduce the participants to:

- Showing an introduction screen with the story of the game and instructions on how to play the game.

- Creating the graphics to be used in the game
- Animating an image to simulate rotation whilst the object is moving
- Shooting and animating an explosion once the bullet hits the meteor or *gigarock*
- Increasing score once *gigarock*s are collected
- Ending the game if a meteor hits the space ship
- Moving to a new level once the score hits a threshold score
- Reinforce another aspect of managing difficulty in a game that of increasing the difficulty in the game as the player gains in-game experience. I discuss the concept of managing difficulty in more detail in section 5.3.1.1.1 (page 160).

3.5.5.3.4. The three games

I choose to make videos showcasing how to make these three specific games for two main reasons. The workshop length meant that there was only time to discuss the creation of one game. The videos allowed the children to see the creation of three other games from start to finish.

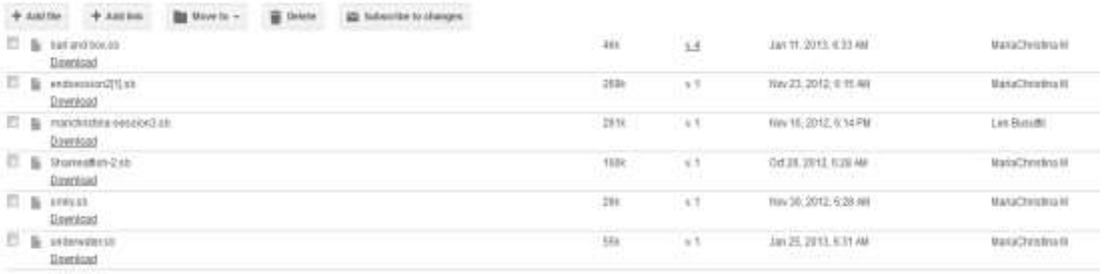
In these videos I also wanted to reinforce the importance of variables in the creation of games and to introduce a series of other items which we had no time to discuss in class:

- Gliding a sript to a location rather than moving to a location
- Using an introduction screen in a game
- Using the mouse
- Creating graphics from scratch
- Animating graphics to simulate an explosion
- Creating an ending screen
- Adding multiple levels to a game

Viewing these videos was not a mandatory task for the students. I left the students free to decide on whether to spend time watching these videos. I planned to release the videos on the workshop websites on three different weeks and to just mention the releases during the workshop.

3.5.5.4. Student’s area

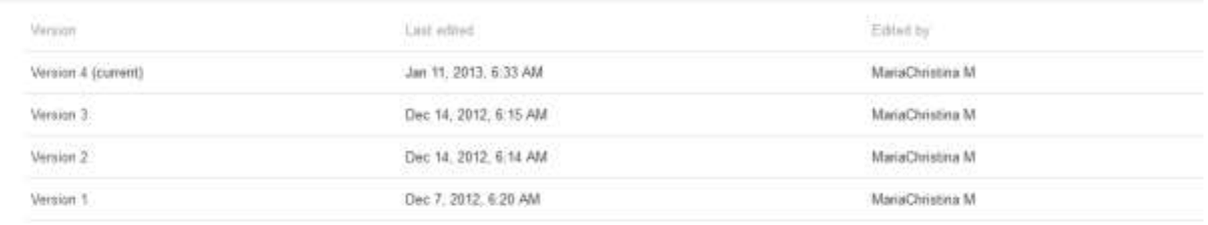
Since the students shared the same account to login to the computers in the computer lab (see 3.5.2) I asked the students to upload their work in a section of the website called the student’s area. This online space was secured using an individual username and password which I handed to the students on the first session. Each student area was accessible to the individual student and the teachers.



ball and box.sb Download	44k	v.4	Jan 11, 2013, 6:33 AM	MariaChristina M
endsession2[1].sb Download	269k	v.1	Nov 23, 2012, 6:15 AM	MariaChristina M
mariachristina-session3.sb Download	291k	v.1	Nov 16, 2012, 6:14 PM	Len Busuttil
[illegible]-2.sb Download	158k	v.1	Oct 28, 2012, 6:28 AM	MariaChristina M
[illegible].sb Download	29k	v.1	Nov 30, 2012, 6:28 AM	MariaChristina M
underwater.sb Download	55k	v.1	Jan 25, 2013, 6:31 AM	MariaChristina M

Figure 3.5-4 Game repository in the Student's Area of the website

The Student’s area section allowed me to view the work of the participants as they progressed throughout their work. Whenever the students uploaded a new version of the game the older game was kept as a different version allowing me to not only look at the final product but also look at the different stages of the game as a “work in progress”

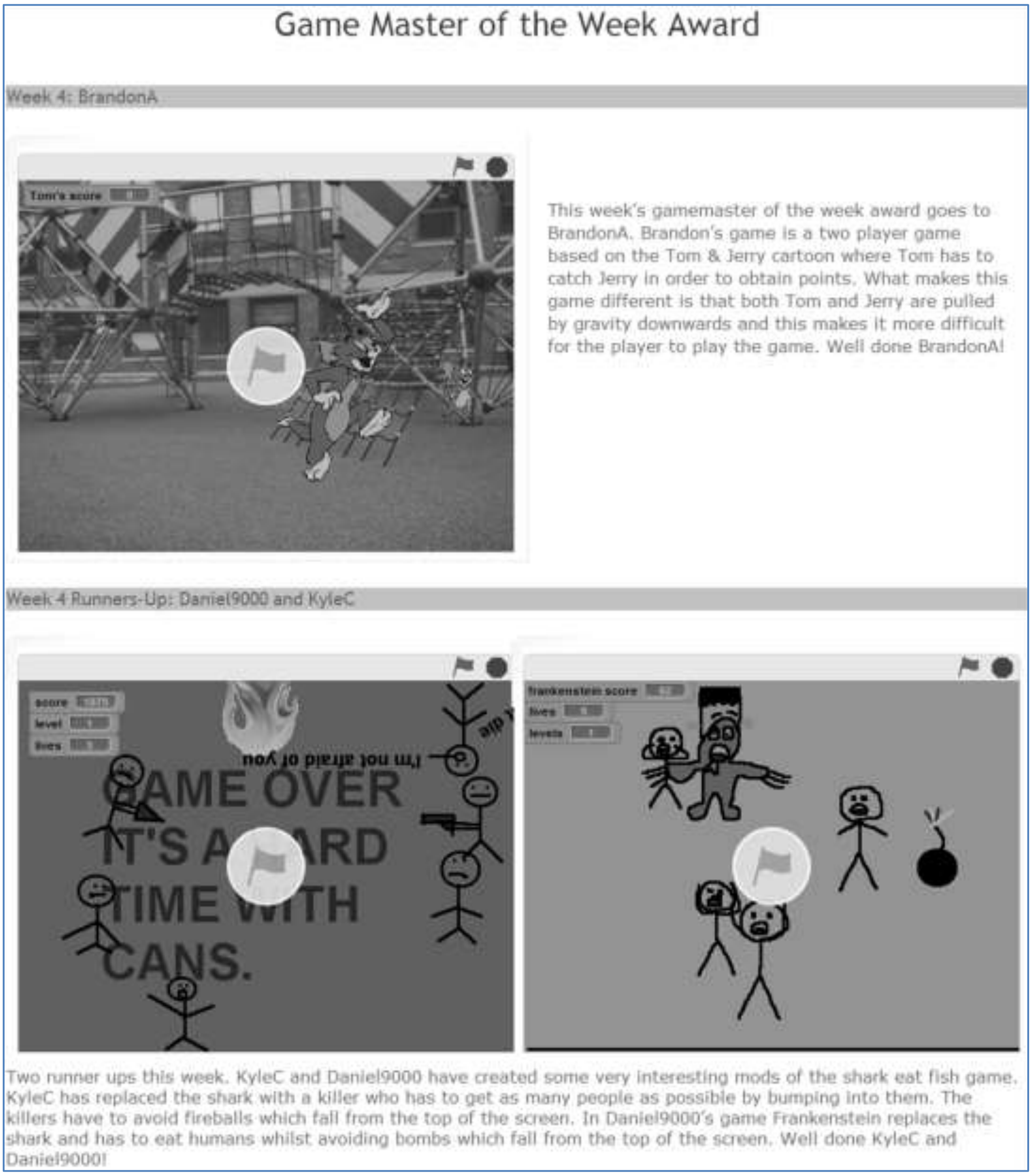


Version	Last edited	Edited by
Version 4 (current)	Jan 11, 2013, 6:33 AM	MariaChristina M
Version 3	Dec 14, 2012, 6:15 AM	MariaChristina M
Version 2	Dec 14, 2012, 6:14 AM	MariaChristina M
Version 1	Dec 7, 2012, 6:20 AM	MariaChristina M

Figure 3.5-5 Versions of the games were stored in the portal

3.5.5.5. Game Master of the week

On every Thursday evening the teachers reviewed the games uploaded by the participants to their respective student areas and chose one to feature as the “game master of the week”. The game chosen was prominently advertised on the home page of the website together with the reason for the choice of the game. The game master award was included to encourage the students to work on their games throughout the week between workshop sessions and to experiment with their ideas. The participants eagerly took up this challenge and teachers found it hard to assign the award to just one student so by the fourth session two runner up games were also selected.



Game Master of the Week Award

Week 4: BrandonA

This week's gamemaster of the week award goes to BrandonA. Brandon's game is a two player game based on the Tom & Jerry cartoon where Tom has to catch Jerry in order to obtain points. What makes this game different is that both Tom and Jerry are pulled by gravity downwards and this makes it more difficult for the player to play the game. Well done BrandonA!

Week 4 Runners-Up: Daniel9000 and KyleC

Two runner ups this week. KyleC and Daniel9000 have created some very interesting mods of the shark eat fish game. KyleC has replaced the shark with a killer who has to get as many people as possible by bumping into them. The killers have to avoid fireballs which fall from the top of the screen. In Daniel9000's game Frankenstein replaces the shark and has to eat humans whilst avoiding bombs which fall from the top of the screen. Well done KyleC and Daniel9000!

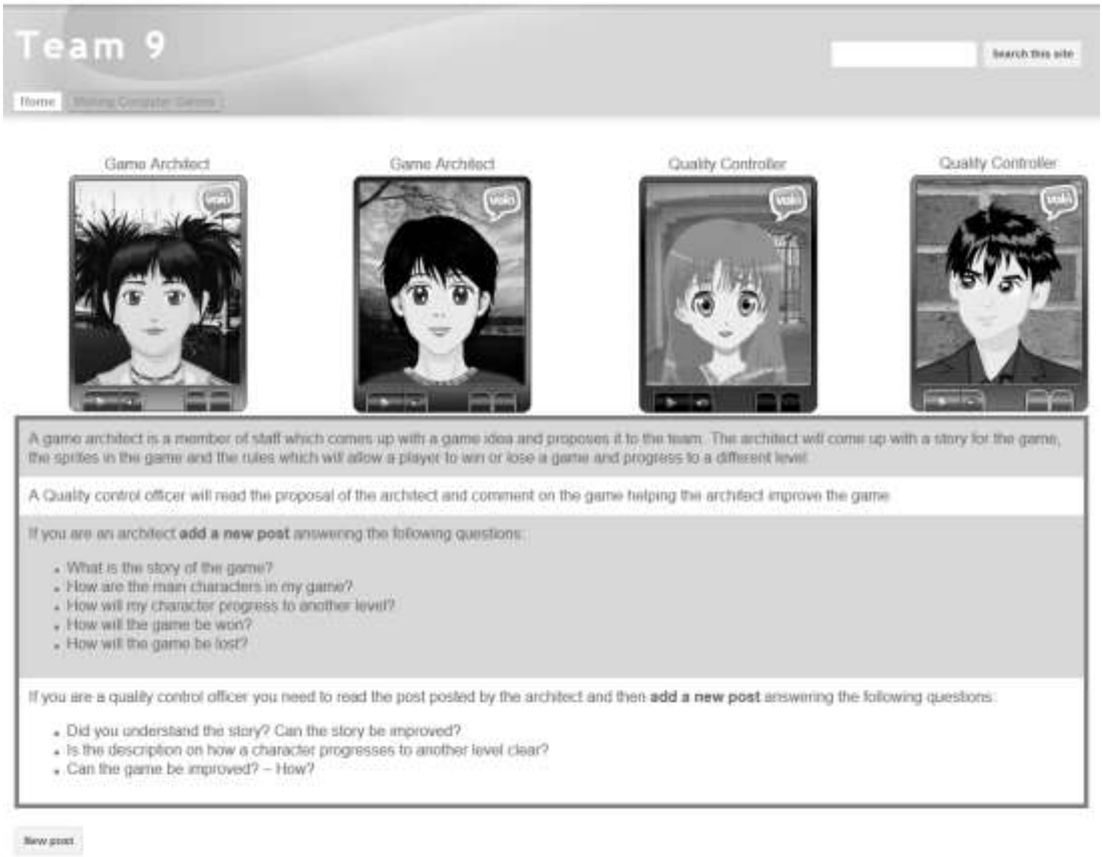
Figure 3.5-6 Game Maker of the week award

This practice was discontinued once the participants started to design their own games. The strategy of discontinuing the game master of the week was adopted on purpose. The games created in the initial part of the workshop were mods of *The Shark eats Fish* game and so had comparable features. Once the students started creating their own games I felt that if I had to continue with the game maker of the week award I would be imposing my judgement on the creative endeavours of the students. Instead during this point of the workshop each participant, or group of participants working together on a game, were encouraged to improve their game. The teachers made it a point not to judge the different games being constructed.

Figure 3.5-6 on page 105 shows the game master of the week award for the fourth week. In this week BrandonA's game was chosen as the game master of the week whilst KyleC and Daniel9000's games were chosen as runners up.

3.5.5.6. Team Area

After the first four sessions were concluded the participants were paired up into teams. Each team consisted of a game architect and a quality controller. The titles chosen were job titles from industry so that the students could start familiarising themselves with industry roles normally present in the game creation industry. Each participant was placed into two different teams with a different role in each team. The objective behind this area of the portal was to get the participants to discuss with their peers the game they were creating in the remaining sessions of the workshop. The participant playing the role of the game architect was asked to write about the story of their game, the main characters in the game, how the game progresses from one level to another and finally how a game is won or lost. The quality controller had to read the entries posted by the game architect and suggest improvements to the game.



A game architect is a member of staff which comes up with a game idea and proposes it to the team. The architect will come up with a story for the game, the sprites in the game and the rules which will allow a player to win or lose a game and progress to a different level

A Quality control officer will read the proposal of the architect and comment on the game helping the architect improve the game

If you are an architect **add a new post** answering the following questions:

- What is the story of the game?
- How are the main characters in my game?
- How will my character progress to another level?
- How will the game be won?
- How will the game be lost?

If you are a quality control officer you need to read the post posted by the architect and then **add a new post** answering the following questions:

- Did you understand the story? Can the story be improved?
- Is the description on how a character progresses to another level clear?
- Can the game be improved? – How?

Figure 3.5-7 The team area

In order to render this exercise more game like *Voki* was used to create speaking avatars representing game architects and quality controllers and embedded in the team area. Participants had the option of listening to the avatar explaining what they had to do in this exercise.

3.5.6 The face-to-face space

The 10 week period was split into two phases. The participants were introduced to Scratch during the first four weeks of the workshop whilst the concluding six weeks were devoted to helping the students build their own games.

Similar to the preparatory project the workshop was structured in two phases. Phase one consisted of the children altering a game I had written. The *Shark and Fish* game was purposely not complete and not very engaging in order to encourage the students to come up with suggestions to modify the game and make it more engaging (see section 2.5.1 page 83). The game consisted of a two character game where the player had to control a shark and eat a computer controlled fish. The fish was always displayed in the middle of the screen and swam along a horizontal line. Whenever the shark ate a fish, a point was added. After playing the game the students were asked to list changes they would implement in the game to improve it.

Each session in both phases of the workshop was split in two. The first part of the session typically lasted about half an hour. In this part of the session the children sat in a group at the front of the room. During the first part of the workshop we discussed how to implement an improvement in the *Shark and Fish* game. The improvement was selected from the list of suggestions the students came up with during the first session. Following the discussion the children made their way to a computer where they worked at implementing the change discussed in their game. The children were left at liberty on whether to implement the change as discussed. Creativity was encouraged and the teachers helping out in the workshop went round the class answering questions and helping children in their tasks.

During the final six weeks the session format was retained. The first half hour was reserved to a class discussion on issues that children were encountering in their quest to build a game with the rest of the session devoted to working on the games they were building.

3.6 Ethical Considerations

Ethical concerns encountered in educational research in particular can be extremely complex and subtle and can frequently place researchers in moral predicaments which may appear quite irresolvable (Cohen, Manion, & Morrison, 2000, p.50). Murphy and Dingwell (2001) distinguish between two approaches to ethical considerations in research. The consequentialist approach looks at the outcomes of the research and determines whether the participants were harmed, and if they were, whether this harm been outweighed by the research’s benefit. On the other hand deontological approach looks at the inherent rights of the participants such as the right to privacy, the right to respect and the right to self-determination. Murphy and Dingwell (2001) stress that in order to address the consequentialist and deontological concerns; one should abide by the following set of principles to guide the research practice:

- **Non- maleficence**: researchers should avoid harming the participants
- **Beneficence**: research on humans should only be carried out if it will produce some positive and identifiable benefit
- **Autonomy or self determination**: researchers should respect the values and decisions of research participants
- **Justice:** people should be treated equally. (Murphy, Dingwall et al. 2001 p.339)

A few days after the information meeting I was informed that fourteen out of the thirty one students present for the meeting had returned the duly filled in forms. Since there were only two applications over the twelve student limit I had imposed I decided to accept all the applications. I felt that leaving out two students would have been unfair on the students who were not selected. The group of fourteen students was made up of five girls and nine boys.

The research project empowered the students to create their own digital games. The students might feel attached to the games they create and hence might want their real names to be used in the research. For this reason I included a clause in the consent form which allowed the students to choose whether they wanted

- to use their real names in the research
- to pick an alias
- have their names anonymised.

Out of the fourteen participants in this research eleven choose to retain their real name, whilst three choose to use an alias of their choice. The choice made by the students has been respected whenever reference is made to any of them in this thesis and will continue to be respected in any additional publications.

During the research, pictures were taken of the workshop participants whilst creating their games. One student asked not to appear in any of the pictures, whilst another student asked to have his/her face blurred.

3.7 Collecting the data

Yin (2012) lists various sources of evidence collected in doing case studies:

- Direct observations (e.g. human actions or a physical environment)
- Interviews (open ended conversations with key participants)
- Archival records (e.g. student records)
- Documents (e.g. newspaper articles, letters emails and reports)
- Participant –observation (e.g. being identified as a researcher but also fulfilling a real-life role in the scene being studied)
- Physical artefacts (e.g. computer downloads of employee's work)

In this research I based my data sources on:

3.7.1 Questionnaires

During the initial and final sessions of the workshop I asked the participants to complete three short questionnaires. The first questionnaire collected information about the participants' and their gaming habits whilst in the final questionnaire the children were asked to answer a series of questions about the game they authored including instructions for people playing the game for the first time. The data collected through these questionnaires could have easily been collected using face to face questions, however asking the questions in the questionnaires using a face to face approach would have taken a considerably longer time out of the session.

The second questionnaire was provided to the students after they played the *Shark and Fish game*. After playing the game the participants were asked about the actions they would take to improve the game. I wanted the children to iterate between playing the game and providing suggestions and allowing the students to jot down their suggestions in an online form allowed them to play and provide suggestions at the same time. The data collected through this exercise provided an indication of the game features they were familiar with and which they would add to the games.

3.7.2 Participant observations during the workshop

As Merriam (2009) accentuates, observational data represents a first-hand encounter with the phenomenon of interest rather than a second hand account of the world obtained through an interview. Conscious that perception is highly subjective and that different witnesses might end up with different accounts of how an incident occurred, I strived to become a careful systemic observer by following the recommendations set by Patton (2002). These recommendations include learning to pay attention, learning how to write descriptively and practicing the disciplined recording of field notes. In order to serve as an aid memoire I used a video camera to record the group discussions in the initial part of each workshop session.

3.7.3 Informal conversations with the participants during the course

During the second part of the workshop sessions I used to circle the class and observe the participants as they worked on their games, as they discussed features with other participants or the teachers and as they played the games being created by their peers. Whilst conducting these observations I held informal conversations with the participants. I used my tablet to audio record the conversations. The recordings were later transcribed and analysed as discussed in section 3.8.3 (page 113).

3.7.4 Discussions with the teachers helping out during the course

After every session I used to share my observations with the teachers facilitating the sessions. In this way I could further confirm my observations and also benefit from opinions offered by the two teachers helping out with the sessions. This sharing session was audio recorded and then transcribed.

3.7.5 Interactions in the online space

All the entries posted by the workshop participants in the discussion area, the student area and the team area can also provide valuable information about how the students were interacting with the game making activity whilst outside the workshop sessions. Whilst designing the online space I also enabled Google Analytics. Google Analytics allowed me to measure how often the participants visited the site and hence deduce if the game building activity was continuing at home or not.

3.7.6 The artefacts (games) created by the participants during the course

The games created by the participants contain a wealth of information which I analysed from a game literacy and a computational thinking perspective. Since I asked the participants to upload the games created during every session I have a continuum of games created by the participants.

3.8 Analysing the Data

The data collected was split into four batches. The first data collection consisted of statistics about the online space use. The second data collection consisted of the descriptive questionnaires filled by the course participants over the course of the workshop. The third collection was all the qualitative data collected during the workshop including the participant observation, informal conversations with the students, discussions with the teachers who helped facilitate the workshop and the interactions which occurred in the discussion areas of the online space. The last collection consisted of the games authored by the children throughout the workshop. In the following sections I outline the procedures used to analyse the different data collections.

3.8.1 Using the online Space

As discussed in section 3.7.5, Google Analytics was enabled on the website created for this workshop. Google Analytics is one of the most popular clickstream data tools (Cutroni 2010). Clickstream analytics is the process of collecting, analysing, and reporting aggregate data about which web pages were visited. The tool also captures the order the pages are visited, the websites people were viewing before visiting the site and the bounce rate that is the number of times viewers left the website after viewing one page only. Clickstream analysis is generally used to analyse trends in web site traffic and for e-commerce analysis.

In this project I was mostly interested in finding out if the children were using the workshop website from outside the classroom, and if they were visiting the site, which areas of the website they were visiting. I was also interested in finding out if there was any relation between the phases of the workshop and the visits to the site. Since the pre-prepared reports Google Analytics made available were mostly targeted at e-commerce sites I exported the raw data files showing the number of visits per day broken down by page name and worked out the statistics using Microsoft Excel.

3.8.2 Descriptive Questionnaires

Three questionnaires were used in this research project (see 3.7.1 page 110). In all cases Google Forms was used to collect the data. This allowed me to embed the questionnaire in the workshop website and to then download the entries filled in by the children directly into Microsoft Excel.

The first questionnaire was used to identify the gaming habits of the children. Microsoft Excel was used to calculate the aggregations of data to answer basic questions such as:

- How many hours per week do the children following the workshop spend playing games?
- Is playing a solitary activity?
- Do the children play with their family and friends?

The children were also asked to list five of their favourite games. Through this list I wanted to create a list of games the children enjoy playing in order to try to identify any similarities with the ones designed by the children during the workshop.

The second questionnaire was used during the workshop to try to elicit from the children changes they suggested carrying out on a game to make it more challenging (see section 4.5.1 page 125).

The final questionnaire was used to build a list of the games created by the children to be analysed. This list of games as described by the students was used during the game analysis process (section 3.8.4 page 114) in order to further understand the games the children created.

3.8.3 Observations, interactions in the online space and conversations

Observations, interactions in the online space, conversations with the students and the teachers all fall within the realm of qualitative data. As Merriam stresses, all qualitative data analysis is primarily inductive and comparative (Merriam 2009, p175). I had to make sense out of the data by consolidating, reducing and interpreting the information collected. In order to help me with the analysis process I uploaded all my videos, transcriptions of conversations with students and teachers, online interactions and notes to self onto the software package Nvivo.

Data analysis was not a phase that was conducted after the workshop ended, but was conducted in parallel to the workshop. I tried to make sense of the observations and discussions during the ten weeks when the workshop was taking place. I reviewed the videos of the initial part of workshop session and interviews after each workshop trying to identify segments of the data which are responsive to the questions I was trying to answer with my research whist at the same time keeping my eyes wide open on the lookout for other themes. Whenever I identified a segment of data which looked interesting I marked it as a node in Nvivo. Subsequent reviewing of my data allowed me to categorise other pieces of data under the same node or to create other nodes as required. In this way categories were discovered, verified, tested and confirmed. Through this approach I was following the advice of Merriam (2009) and moving from the inductive to the deductive phases of qualitative data analysis. At times I would come up with two nodes to merge them into one at a later stage. I validated the categories created using Merriam (2009) suggestions:

- Categories should be responsive to the purpose of the research
- Categories should be exhaustive. All data should be able to fit in one category or sub-category
- Categories should be mutually exclusive. Data should fit in one category only.
- Categories should be sensitizing. The name of the category should be sensitive to the data in the category.
- Conceptually congruent, that is the categories should be characterised by the same level of abstraction.

3.8.4 Analysing the games

Analysing the games required me to spend time playing the games. After playing each game a number of times, I tried to identify:

- The type of game or genre
- The characters making up the game
- The goal of the game
- The rules that the game implemented
- Feedback mechanisms adopted by the game
- Similarity of the game to the games created in class or discussed in the videos uploaded to the resource section (see 3.5.5.3 page 100)

I then analysed the code making up the game looking for computational concepts adopted in the game. Since each game was saved on the web portal over a period of time, whilst it was being built, I also looked at how the game interface and code evolved over the weeks.

3.9 Conclusion

Using multiple methods is characteristic of qualitative research since this reflects an attempt to secure an in-depth understanding of the phenomenon in question (Denzin, Lincoln 2005). In this research I strived to collect data from multiple sources but I remained aware that the primary instrument, that is me, remained the same and hence my subjectivity could affect the study. Since as asserted by Denzin objective reality can never be captured, triangulation can be used as an alternative to validation. Richardson et al take exception at the term triangulation since triangulation presupposes a fixed point that can be triangulated (Richardson, Adams St.Pierre 2005). Instead they propose using the metaphor of a crystal since a crystal is a prism that has multiple faces and what one sees from a crystal depends on the angle of repose.

In this research I tried to look through the crystal from multiple angles and tried to make sense of what I saw. In the next section I describe what I saw through the crystal when looking for the game making practices adopted by the children whilst building their games.

4. Game Making with the 11 year olds

4.1 Introduction

The children were eager to participate in this study and were delighted to share their game making experiences with me and the teachers helping out in the sessions. Meeting the workshop participants for ten weekly sessions and collaborating online with them provided abundant information about their experiences.

In this chapter I present and analyse:

- the statistics collected from the web portal used during the workshop
- the themes that stand out from the transcriptions of informal conversations with the workshop participants and the discussions with teachers helping out during the workshop
- participant observation during the workshop
- questionnaires the participants completed during the initial and final sessions of the workshop.

4.2 Using the online resource

In a bid to maintain contact with the participants between the weekly sessions I created a web portal. I asked the participants to upload their projects onto the web portal. This ensured the participants could work on their projects whenever they wanted to, during the sessions as well as at home. This setup also allowed me to maintain an archive of all versions of their games; any game uploaded did not overwrite previous versions so I could use the archive to explore see how participants' ideas were developing.

Google analytics allowed me to track student activity on the website. Through this feature I tracked which areas of the site were mostly used and the days when the site attracted most traffic.

The participants visited the portal regularly with most visits being logged during the Friday sessions. The participants visited the website on other days too with Saturday and Sunday being the more popular days, followed by Wednesday and Thursday. It seems the children predominantly visited the site mainly before and just after each session.

As Figure 4.2-1 shows the participants mostly visited the web portal during the workshop weeks with the number of visits logged reducing during the Christmas recess and after the workshop ended. There was also a reduction in visits during phase 2 of the programme, that is, once the participants started building their own games.

One of the most popular sections of the online portal was the resource section. This section included step-by-step video tutorials on how to build three different games. Three videos *Penalty Shoot Out Game*, *Year 3578 Saving the Earth* and *Shooting a Vase* were uploaded to the video sharing site YouTube and then embedded into the workshop web portal on three different instances. The most popular video was the *Shooting a Vase* video which demonstrated how to implement shooting in a game. This video remained popular throughout the workshop.

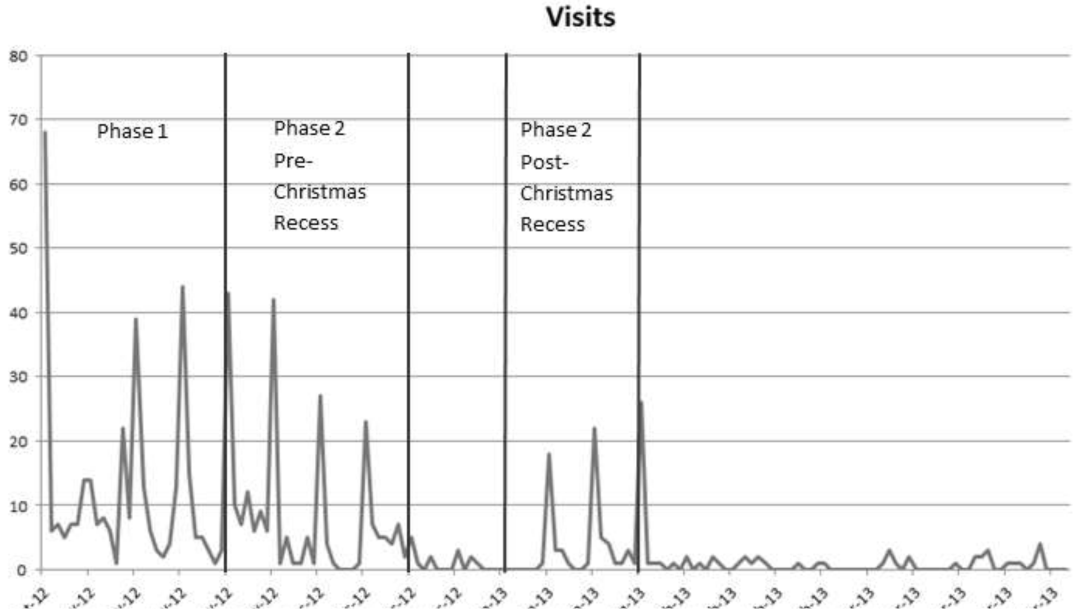


Figure 4.2-1 Visits to the Web Portal

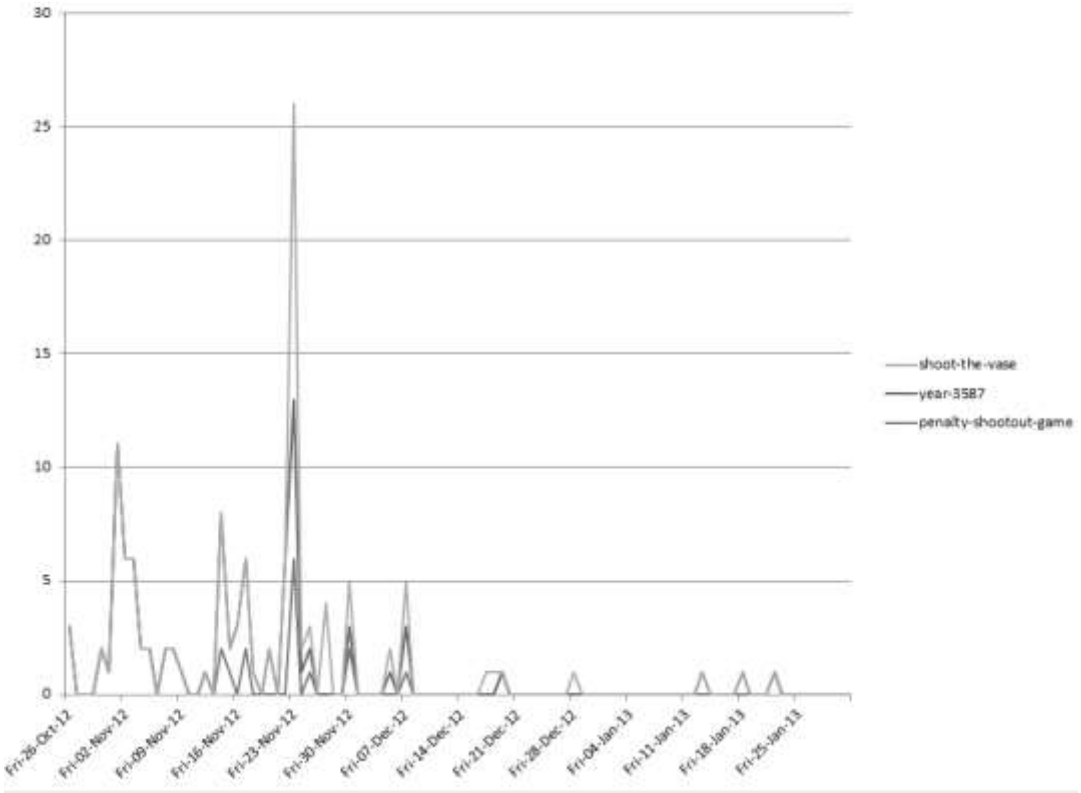


Figure 4.2-2 Visits to the Resource Sections

4.3 Importance of game making for the participants

There is ample research (Brand 2012, Busuttil, Camilleri et al. 2014, Games 2008, GameVision Europe 2010, Lenhart, Kahne et al. 2008) that affirms that game playing is a central activity in the lives of young people. Throughout the experience of conducting the game making workshop I was engulfed with an aura of excitement which suggests that the children were enthusiastic to make the leap from game players to game makers, from reading and writing the games by playing them, to writing their own games from scratch.

4.3.1 Joining the workshop

I was hit with the wave of enthusiasm from the first time I met the students to explain the aims of the project and what they needed to do if they were interested in participating in the project. I had initially capped the number of students joining the workshop to twelve. The participation procedure required the students and the parents to sign and hand the consent form to the school's assistant head. Since the number of participants was capped participation was on first come first served basis. Most of the questions I had to answer as soon as my short presentation was over were about when the sign up procedure would open. A student asked whether he could get his mother to sign the consent form straight after school when she called to pick him up beating his peers who used the school transport to get home. A second student asked for the time when the school administration arrives in the morning to ensure that he would be the first to submit the participation form.

In the evening I received an email from a parent of one of the children enquiring about the time of the workshop. Her son was scheduled to attend football training after school but wanted to attend the game-making workshop too. He was deeply concerned that the two activities would clash. Luckily football training started just after the workshop ended and he could attend both.

Fourteen out of thirty one students (45%) chose to attend the workshop. This is a very high rate considering that parents were made aware of the game-making workshop only a week before commencement date. It is quite usual for parents to make all the necessary arrangements for extra curricular activities before the school term starts.

During the first session I asked the workshop participants to answer a few questions about their gaming habits. Through this questionnaire I wanted to analyse the gaming habits of the workshop participants. All the workshop participants play for three to five times a week with most of the participants (twelve out of fourteen) playing on a daily basis. Most of the participants played alone with only two out of fourteen stating that they played with friends.

4.3.2 Interacting with members of the family

During informal conversations with the workshop participants I encountered various situations where the children were show casing their work to their parents. In the third session of the workshop BenL turned the *Shark and Fish* game into a multiplayer game where a second player could control the fish using the keyboard keys a,x,w,d. Before BenL modified the game, the player could control the shark using keyboard keys whilst the fish were controlled by the computer using rules programmed by the game developers. BenL explained that when he changed the game into a multiplayer game he asked his mum to play with him. She took over the shark character whilst he was the fish trying to escape. Whilst BenL was recounting this episode I could sense the pride of this child who authored a game which was good enough for him to share with his mum.

Similar incidents were recounted by Daniel9000, HallieH and Serafina. Daniel9000 kept adding levels to his maze game which he invited his friends to try out. The activity of trying out the game was not limited to the workshop session but flowed into the home environment too. Daniel9000 was proud that his mum never went past level 6 of his 15 level game. Serafina too shared the game she was creating with her sisters and came back to the workshop with ideas on how to improve the game. These were all instances where the children were experiencing the joy of building – in this case building a digital game.

4.3.3 When I grow up I want to be a game designer

The importance of games in the lives of the workshop participants was further highlighted in the informal chats I had with two of the workshop participants during the first workshop session. BenL explained how he had combined his game playing time with creating videos about his game play to post online

I downloaded this thing called Fraps and I can now while I am playing some online game like I make them do something cool and I record it and then put it on YouTube… when I grow up I want to be a computer game designer (BenL).

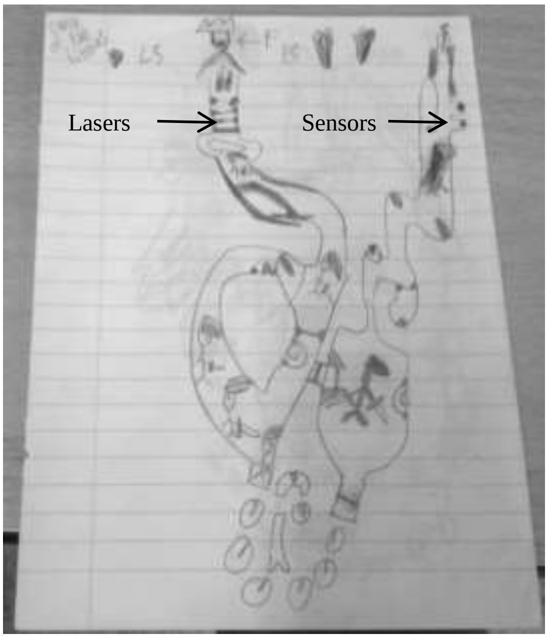


Figure 4.3-1 Daniel9000’s Clash of the titans

This desire to shift to game making from game playing was also highlighted by another participant Daniel9000. During one of the sessions Daniel9000 explained that for years he had wanted to make his computer game. Since he did not know how to create a computer game he resorted to drawing his maze games on paper. The maze games he created were a series of A4 papers on which he drew mazes. The player played these games by traversing the game using his finger as a character in the game. Each maze had a title written on the top of the page and each game consisted of a series of levels. Figure 4.3-1 shows level 5 from the game Daniel9000 called *Clash of the Titans*. The maze has an entrance on the top right hand corner of the paper and one exit marked with an F (for finish) Daniel9000 explained that to arrive to the finishing exit for a level, one had to go through the maze. Most of the passages in the maze were guarded by coloured laser beams. The player had to avoid touching these beams as touching them would result in the player being burnt to death. These beams could be switched off by touching coloured sensors, however some of these sensors were also guarded by laser beams hence the player had to figure out the order of switching off the laser beams in order to successfully make it to the end of the maze without being burnt. Daniel9000's game also included a series of stick figures holding swords which the player had to dodge to arrive to the finish.

The mazes created on paper by Daniel9000 were amazingly detailed and creative. Gaming was such an important activity for him that the drawings he came up with, well before the gaming workshop was even announced, were blue prints for games waiting to be enacted. Most of the features Daniel9000 had drawn in his maze game were replicated in the digital game he created during the final stage of the game creation workshop. Daniel9000's game is analysed in a subsequent chapter (see 5.2.2 Bob's Adventure).

4.4 Game inspired activities

The game creation workshop itself was designed to include game like activities so that participants could learn whilst having fun. These activities included the game master of the week award and allowed students to interact with others during the building of their games by moving away from their computers and testing the games created by their peers.

4.4.1 Game Master of the week

In a bid to encourage the students to use the online resources in between the workshop sessions a *Gamemaster of the Week Award* was introduced. Every week we used to pick out a game which stood out from the rest of the games uploaded in the students' areas. This game would then be featured on the front page of the website with a brief note on why it was chosen as the *Gamemaster of the Week* (see Figure 3.5-6 page 105). This feature of the website was popular and at times it was difficult choosing just one game to feature as the *Gamemaster of the Week Award* so we introduced runners up. I used to put up the gamemaster of the week award on the website just before the session started on a Friday, whilst the students were following other lessons in school. The students would not have had time to see who got the award before the start of the workshop and this meant that they would come in asking about who got the award this week. The competitive attitude so evident in video game play had permeated itself into this part of the workshop with children competing in getting their game featured on the front of the website as the game of the week award. The competition was a healthy competition with the children collaborating with each other during the workshop.

4.4.2 Glitch!

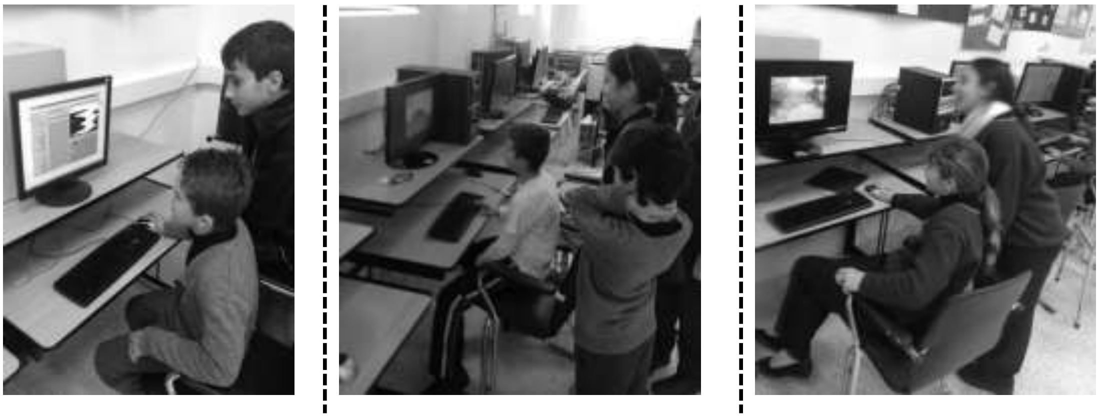
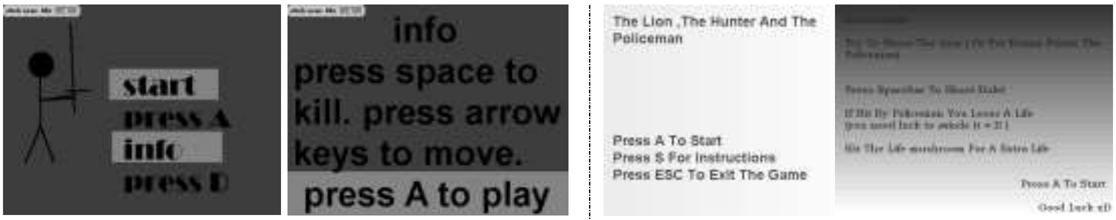


Figure 4.4-1 Play testing games of other workshop participants

A central activity of the workshop sessions was play testing the games created. Whenever a participant felt that their game was good enough to be played by peers, a friend would be asked to play test the game. This usually resulted in a small group of children watching from behind the player's back whilst the player played the game. The game creator usually sat on the side watching in earnest whilst his/her creation was tried out in front of what had become a game making community.

The testing activity quickly developed into an important routine for the group, with the testing period being used to showcase the new features the game presented. One of the most popular testers BenL came up with the term Glitch! Whenever BenL discovered a bug in a game he used to shout the word Glitch! It would be back to the game creator to solve the problem unearthed by the tester in time for another session of play testing.

The ingrained procedure of play testing the games as soon as the game creator added a new feature, resembles the test driven methodology adopted in industry. This software development process is based on the repetition of very short development cycles where the developer first defines a test from the requirements and user-specifications and then develops code to pass the test. In the methodology adopted by the students the game designer/maker first came with a new feature for the game, then developed this feature and had fellow workshop participants test the feature. The additional benefit of the approach adopted was that the game testing acted as a source of ideas for the testers. This cross fertilisation of ideas can be seen in the games created by BenL and Serafina. BenL added a welcome screen to his game where the player could choose a key to start the game and another key to read information about the game. The information consisted of game instructions. Serafina saw this feature whilst play testing BenL's game and asked BenL to help her create a welcome screen for her game too. At the end Serafina's game had a very similar welcome and help screens to that of BenL's game. Testing the games had become a space for transacting in the gaming capital (see 2.2.5 page 36).



start
press A
info
press D

info
press space to kill. press arrow keys to move.
press A to play

The Lion ,The Hunter And The Policeman

Press A To Start
Press $ For Instructions
Press ESC To Exit The Game

[illegible]

Try To Shoot The Lion [illegible] The Policeman)

Press Spacebar To Shoot Bullet

If Hit By Policeman You Loose A Life
(you need luck to avoide it = D)

Hit The Life mushroom For A Extra Life

Press A To Start

Good luck xD

Figure 4.4-2 The initial and help screens in BenL and Serafina's games

4.4.2.1. Cheating the way out of a Glitch!

Consalvo (2007) has written extensively about the cheating phenomenon in digital gaming (see section 2.2.6.2.1 page 41). Cheating strategies range from consulting strategy guides whilst playing a digital game to purchasing game cheating cartridges which, when attached to a gaming console, allow a player to use cheat codes and attain super powers. Even though the students following the game making workshop were making games and not just playing them they were observed adopting cheating strategies, not to win a game but to bend the rules to iron out bugs which occurred in the game.

Testing a multi-level game can be time consuming especially when a bug is identified in the latter levels. Rather than playing the game to reach the level that contains the bug I observed the students using a number of strategies to bend the rules and skip levels.

A game can be run in *Scratch* by using one of two modes. The first mode requires the player to expand the game screen to full screen and then pressing the green flag to start the game, whilst the second mode allows the player to press the green flag whilst the programming environment is still in view. If the second mode is chosen, a player can use the mouse to move the sprites whilst the game is running. BenL was observed using the second mode in his game *Stick with a sword.* Rather than using the keyboard keys to move *Stick* he clicked on the *Stick* sprite and moved him straight onto the dragon. In so doing BenL made sure that *Stick* was not burnt by the fire which blew from the dragon’s nostrils. BenL had managed to skip part of the game to test the game sequence when *Stick* can use his sword on the dragon.

Another instance when cheating was adopted was in the case of *Racer.* Manoeuvring the car around obstacles in *Racer* was quite cumbersome due to the size of the pathways and the car. When debugging KyleC was observed reducing the size of the car so that he could easily manoeuvre it to get to the area where the error was occurring.

Indeed cheating became quite a handy practice for the children especially since solving bugs require the child game maker to test the same aspect of the game numerous times to ensure that the glitch is ironed out.

4.5 Cashing in the gaming capital

Gaming capital is acquired by video game players who play their favourite digital games, and transact this capital with other players in paratext spaces. The game making workshop offered a possibility to the participants to demonstrate the gaming capital acquired by improving a ready-made game as well as building their own games.

4.5.1 Upgrading a game

The first session of the game-making workshop allowed the students to bank on their gaming capital by suggesting ways to improve a simple game provided to the participants. The students had to play a game provided and suggest ways on improving the game. Table 5 shows the suggestions made by the participants after they played the game for a short period. The participants' suggestions varied in details from short phrase suggestions such as *make the fish go faster* to more elaborate suggestions such as:

> *Once you eat 1000 fish and 100 sharks you fight a champion shark. you can attack sharks by biting at their tales and fins. Each shark will leave after thirty seconds, but will come again later. You grow stronger when you eat fish, and bigger when you kill sharks. Getting bigger will help you eat up to five fish at a time and getting stronger helps you inflict more damage to bigger sharks. You start out with seven lives per level. There are up to twenty levels.*

The suggestions made were reduced into a number of features as outlined in Table 5 below.

Feature	References
Add more obstacles for the player such as bombs that can hit the shark or empty plastic bottles that fall from the top of the screen. If the shark hits them then it dies.	17
Add more levels to the game	8
Make the fish go faster	7
Make the fish appear in different locations once the shark eats them	5
Add more prey to the game	5
Make the game a race against time so that if the player does not eat enough fish after some time the game comes to an end	3
If you get enough points the shark should get super powers.	3
Add more controls to the shark	2
Make the game a multiplayer game	2
Add lives to shark so that if a shark dies, a life is lost but the game does not end.	2
Make the fish run away from shark	1
Add more sharks to control	1
Different fish, each harder to get and worth more points	1

Table 5 Improving the Shark and Fish game

The new features suggested by the workshop participants were all based on experience the participants had acquired by playing games and transacting in the gaming capital. The workshop built on this capital by spending the first four weeks implementing the suggestions made by the students. In this way the students were introduced to the *Scratch* programming language by building on their interests.

4.5.2 Building new games

The gaming capital acquired by the workshop participants was also manifest in the games created after the fourth week of the workshop. An example of this manifestation of this gaming capital picked up through the long periods of playing other digital games and discussing these games with peer players in online and face-to-face spaces can be seen in the game designed by KyleC. KyleC designed a racing game, similar to games he played online. Looking at the early versions of the game saved in the online website, one can note that the game included a screen for an online shop. The plan was that the player would collect coins by racing against the time and once the enough coins were accumulated the player would be allowed to purchase car models to use in the race.

The scheme of exchanging coins collected in-game with upgrades to be used in the game is similar to popular games such as *Subway Surfers*. In *Subway Surfers* the player has to run from a railway inspector collecting coins and other power-ups whilst avoiding a series of obstacles such as trains, light posts, wooden barricades, tunnels and more. The coins collected during play can be used to purchase one-time use items such as hovering surfboards and paint powered jetpacks. Whilst designing his game KyleC was drawing on experiences and knowledge from outside the school. His experiences as a gamer and a consumer were shaping the way he was designing his new game. He was testing the boundaries and trying to add new dimensions to the game.

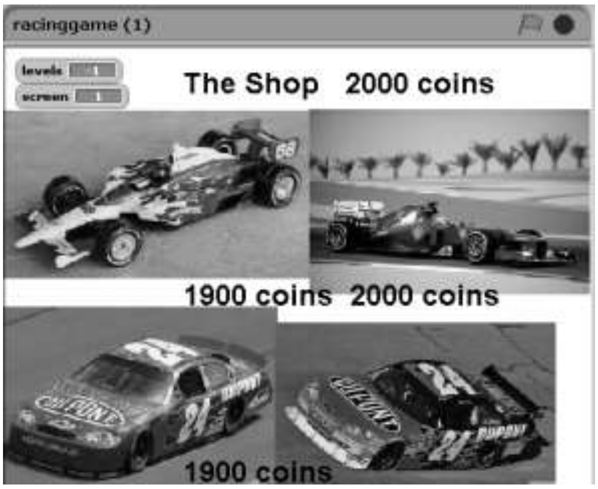


Figure 4.5-1 In game shop in KyleC's games

The planned shop was eventually removed from the later versions of KyleC’s game. The skills required to create the online shop proved to be too difficult and KyleC decided to focus more on making his racing game work. The attitude of playfulness adopted whilst testing the boundaries of the software lead him to understand the limits of what is possible in the tool he was using to build the game.

4.6 Remixing the Narrative of games

As Jenkins (2013) argues, the process of digitalisation has led a large number of us to create new materials by manipulating, appropriating, transforming, and recirculating existing media content. One has to simply visit video sharing sites such as *YouTube* or *Vimeo* to come across user generated content which is a remix of content created by others. This process of remixing is not just the preserve of digital content. (Upon reflection, it is quite usual for me not to stick to a recipe whilst cooking and to substitute ingredients, or alter the procedure outlined in the recipe).

During the course of the first session of the game creation workshop I was amazed at the rapidity the workshop participants started to alter the narrative of the game. The participants had just been introduced to *Scratch* and shown how to play the game *Shark and Fish* and immediately I could notice a number of the participants altering the game and remixing it to create a new game without changing the rules of the game.

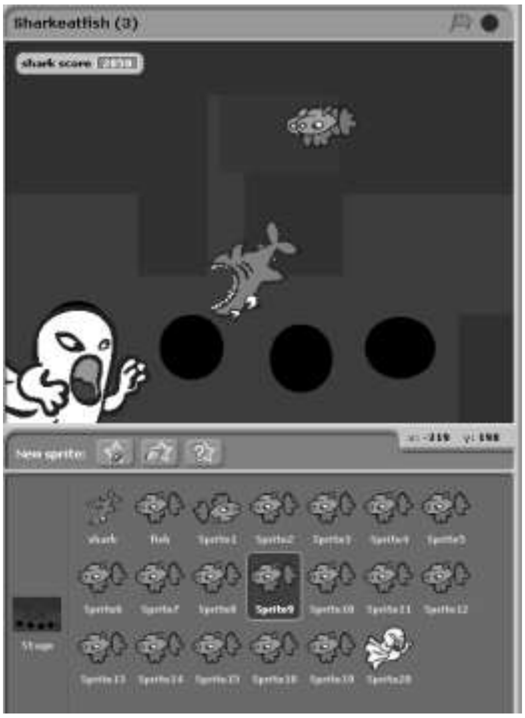


Figure 4.6-1 MiguelB's Shark and Fish

The remixing took various grades. MiguelB’s and HellieH’s mods of the game retained the same theme of a shark hunting fish under the ocean. To modify his game MiguelB copied the fish sprite to create a large number of fish which the shark character could feast on. He also drew a sunken ship in a drawing program (*MS Paint™*) and imported the image into *Scratch* to be used as a background to the game. A ghost sprite was added to the game to show that the ship was haunted.

HellieH used a browser to locate pictures of a shark, a diver and an underwater background shot. She then edited the shark and diver images to replace the background with transparent space. The shark and diver were used to replace the shark and fish images of the original game. The cartoonish background of the original game was substituted with an underwater background shot showing a dolphin (see Figure 4.6-2 HallieH's Shark and Fish).

4.6.1.1. Funds of Knowledge

MiguelB and HallieH were quick to transfer skills they had acquired from other software packages such as *MS Paint™* to make the sample game provided their own. They were not just using the resources provided in *Scratch* instead they were using the internet to find existing pictures which they would modify or use other tools to draw the resources.

HellieH did not simply download an image off the internet and use it. She meticulously modified the shark and diver images by removing the background from the images downloaded, retaining only the part of the image needed. The children demonstrated that they could easily draw on skills they were using in different software packages such as image editors, drawing tools and search engines. This knowledge was not confined to software skills. Children drew on experience gained whilst playing games, as in the case of KyleC's design of an in-game shop. They were also ready to fluently use technology to document their daily experiences as demonstrated by BenL during his recordings of game play on YouTube. It was evident that technology had become ingrained in the processes of everyday life. The knowledge accumulated throughout the years was acting as "Funds of Knowledge" (Gonzalez, Moll et al. 2005).

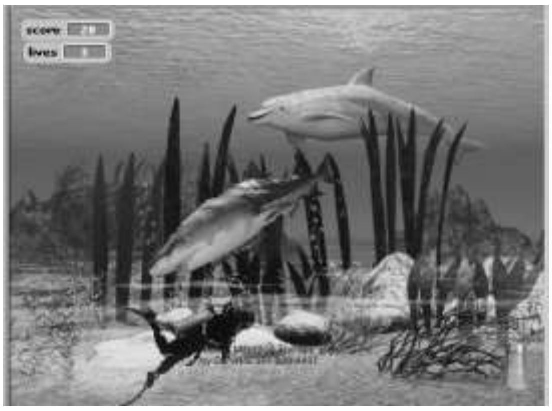


Figure 4.6-2 HallieH's Shark and Fish

I feel that in this aspect of the workshop I missed out on a learning opportunity to discuss with the children issues of copyright and attribution. HellieH and other children participating in the workshop were quick to download media off the internet and modify it without asking for permission or acknowledging the source. Game making provided an avenue where such a topic could be discussed in a meaningful way since the children perceived a need to use media created by others.

4.6.2 Including a portrait as an actor in the game

Whilst the remixes developed by MiguelB and HallieH retained the same theme of fish swimming under the ocean other workshop participants changed their games to totally remove the reference to the ocean and instead replacing the ocean backdrop with other spaces.

BenL produced a number of remixes of the shark eat fish game. In the first remix shown in Figure 4.6-3 the ocean was replaced by outer space with a large sun in the background. The game includes two sharks and two fish which now adorn a spacesuit to match the environment where the game is being played.

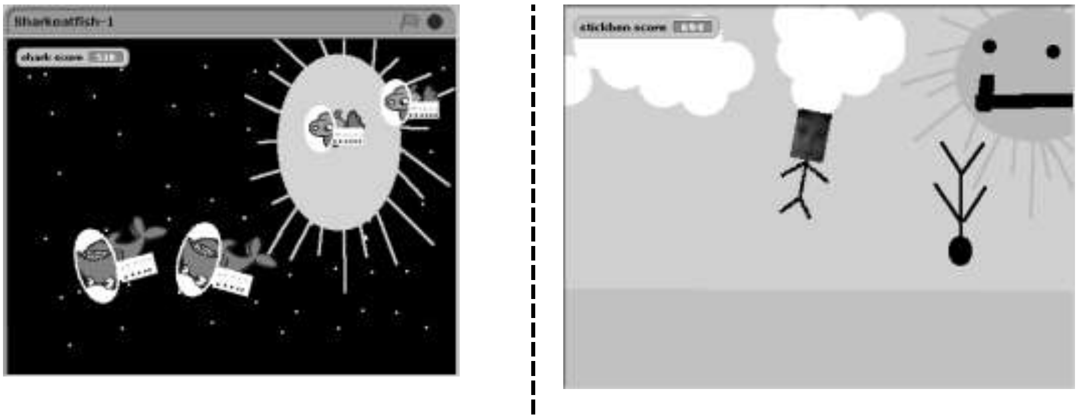


Figure 4.6-3 BenL's remixes of the Shark and Fish game

In the second remix included in Figure 4.6-3 the ocean floor is replaced by grass and a bright blue sky. The sun with face like features, including eyes and a smile, grace the sky. The shark and the fish are replaced with stick figures drawn by BenL which like the shark and the fish in the original game float in air.

The shark in the second remix of the *Shark and Fish* game was replaced by a stick figure with a picture of BenL’s face replacing the stick figures face. This was not the only instance when children decided to include themselves in the game. Daniel9000 in *Bob’s Adventure* recorded himself saying *Ha Ha.* This sound bite was then played whenever a player lost the game. It was as if Daniel9000 wanted to personally make fun of the player for failing to win *Bob’s Adventure.*

Children putting themselves in the game reminded me of the famous painting *Martyrdom of Saint Ursula* by Carravagio. In this painting Carravagio includes a self-portrait as the witness on the right hand side of the painting. This act is different from drawing a self-portrait or simply taking a selfie where the subject of the picture is the person. Through these actions the children were feeling so part of the game they were creating that they decided to feature in it themselves.

4.6.3 The monster claps as the stick figures try to flee

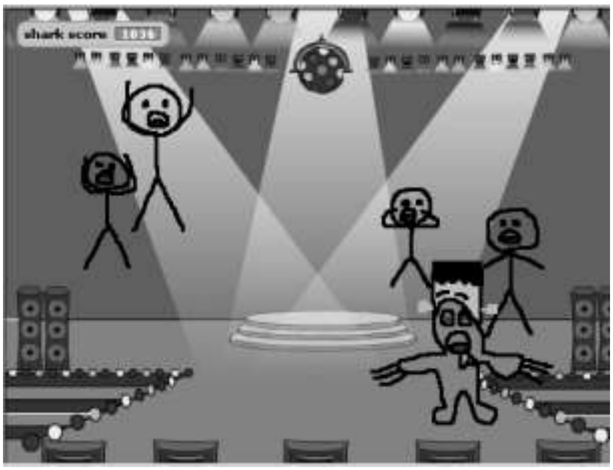


Figure 4.6-4 KyleC's remix of the Shark and Fish

KyleC’s remix of the shark eat fish game contains no aesthetic reference to the original game, however the game mechanics, the way the characters operate are exactly the same as the *Shark and Fish* game. In this game KyleC replaced the ocean floor with a spotlit stage. Stick figures replaced the fish trying to escape the monster that replaced the shark. Interesting in this remix is the animations of the monster and the stick figures. Four stick figures replaced the fish. Each stick figure was drawn with arms in different positions. A stick figure has its hands high up in air, whilst another has its hands against its face as if it is screaming in terror. The figures are displayed moving across the screen at a fast speed simulating a crowd of people terrorised by a monster trying in every way to escape the ordeal. The monster on the other hand is moved by the player to capture the stick figures. The shark’s jaw movements whenever it catches a fish were also remixed by KyleC to show the monster clapping its arms whenever it captures its prey.

4.6.4 Tom and Jerry meet Pacman

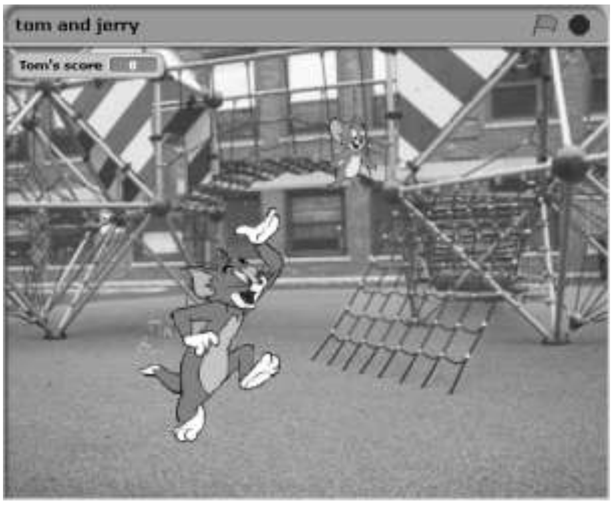


Figure 4.6-5 BrandonA's Tom and Jerry

Other games created by the workshop participants were not a remix of the game originally provided to the participants in the first session, however they still contained remixes of characters the participants met in other games and cartoon series. The game created by Daniel9000 included the Pacman character. Pacman in Daniel9000's game is a machine operated character which guards a maze passage. The player has to dodge Pacman to continue in the pursuit of winning more levels. The Pacman character was remixed by Daniel9000. There is no mention of pac-dots so synonymous with Pacman in the traditional arcade game. The character moves from being controlled by the player to becoming a machine operated character, however the movement of the character retains the original sequence that of moving side-to-side whilst opening and closing its mouth.

The multiplayer game by BrandonA focuses on Walt Disney's cat and mouse arch-enemies Tom and Jerry. Two players can play this game with one player controlling Tom as he tries to catch Jerry, whilst the second player controls Jerry and tries to escape Tom. Similar to the cartoon series, Tom retains the predator role whilst Jerry does his best to outmanoeuvre Tom. The remixing in this game is not just in the characters used but also in the coding itself. Tom and Jerry are remixes of the gravity cat sprite included in *Scratch*. The gravity cat sprite was changed so that key stokes by the player alters the position of the sprite. As soon as the player releases the key the gravity effect resumes and the sprite is pulled to the bottom of the screen. The original gravity cat script leaves a trace on the screen showing the path the cat takes whilst it moves on the screen. The trace is removed from the Tom and Jerry characters used by BrandonA. This remix of the game highlights the fluency that BrandonA had acquired in such a short period of time. This fluency allowed him to read and understand code created by others, in this case a demonstration project included in Scratch, and change it to suit his needs in the new game he was creating.

4.7 *What if* – avenues for possibility thinking

Throughout the game-making workshop I could observe various instances where participants engaged into possibility thinking (Craft 2010, Craft 2001). The participants were engaging with the game creation process by using their imaginative skills to think of new possibilities for the game being created. They were also posing questions and proposing multiple solutions to the questions.

> Inherent in possibility thinking is a willingness and capacity to be immersed, to pose and respond to questions, to make connections, to use imagination, to innovate and to take risks.(Craft 2010, p.20)

In the next section I present three situations where the participants engaged in possibility thinking during the workshop.

4.7.1 Creating a refuge for the fish

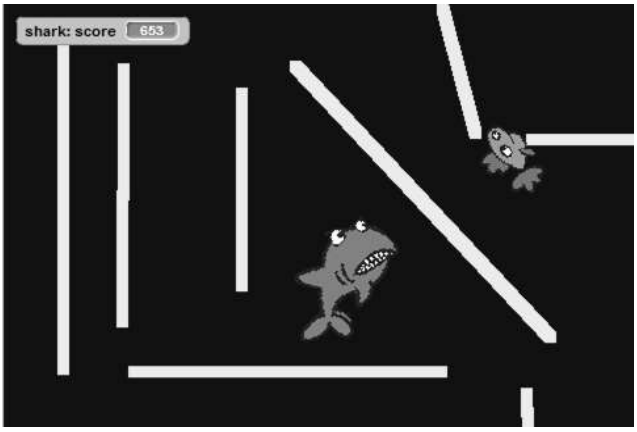


Figure 4.7-1 Building a refuge for the fish.

One of the suggestions one child made to improve the *Shark and Fish* game, was to make the game harder for the shark by providing obstacles. In one of the game modifications uploaded to the portal, BenL added a series of yellow lines to his game (see 4.7).

During one of the discussion sessions Daniel9000 inquired how to code the game so that the yellow lines could act as barriers behind which the fish could hide. I bounced the question onto the whole group and asked if anyone could think of a solution. Immediately the participants started to provide possibilities. BenL pointed out that the lines were meant to be barriers but he could not figure out how to get them to work. One idea he had, but which he did not implement, was to turn each bar into a sprite. In this way if the shark hits the bar it would not be able to move on, similar to how a fish can be made to disappear when it hits a shark. In order to stimulate the discussion I praised BenL for using this approach but pointed out that with this approach we would need to create 8 sprites since there were 8 lines and code logic for each sprite. So although this approach was a good approach I asked the group again if there were other approaches we could come up with. MariaChristinaM suggested storing the Cartesian coordinates[7] of the bars and then coding logic in the shark sprite to stop the shark from moving into these locations. Again I pointed out that this approach was a great one but we would have to change the code if one of the lines were moved during the design stage. BenL pointed out that whilst experimenting he came across a block "if touching colour" which could be used to stop the shark if it hit the colour of the bars. I summarised that out of the three methods suggested this method would be the better one since if we had to add more lines or change the locations of the lines the code would still work. At this point Jacques pointed out that the bars had to share the same colour and that this technique might be confusing if a new fish is introduced in the game with the same colour as the bar since the shark would not be able to eat the fish.

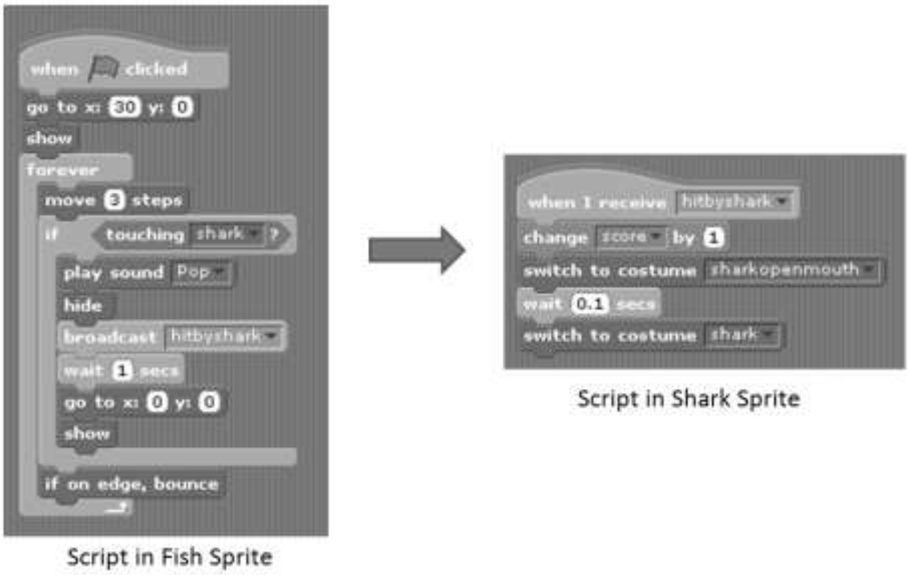
As a teacher I have learnt to use questioning techniques to stimulate discussion in class in order for students to become more engaged. What intrigued me in this episode was the fluency with which the workshop participants not only asked questions but also came up with different possibilities to solve the problem identified by their peer. This episode occurred during the start of the third session. The students had only been working with Scratch for two weeks yet they were highly engaged and confident enough to propose valid solutions to a problem. It usually takes me a lot of prodding for students to come up with ideas when I try this technique with other groups of students. Yet in this case this group of students managed to think outside the box and come up valid ideas in a relatively short time. Their engagement with games and their drive to make this game better was providing a fertile ground to foster their possibility thinking.

[7] Cartesian Coordinates (X,Y locations) are used by *Scratch* to note the location of sprites on the computer screen. A sprite placed in location (0,0) is placed in the middle of the screen.

Another interesting observation was the use of Cartesian coordinates in the technique suggested by MariaChristinaM. The workshop participants were introduced to Cartesian coordinates in Maths. The coordinates are used to locate a point whilst drawing graphs, however most students tend not to associate these coordinates with anything else. Game making was providing an opportunity to the workshop participants for this mathematical notation to come to live. The students were seeing another use for these coordinates other than using them to plot inert graphs. They were using them to display sprites in different locations or in this case to stop a sprite from advancing into an area of the screen. Maths was coming to life.

4.7.2 Who changes the score – The shark or the fish?

In order to introduce the idea of game development, using *Scratch* I created a purposely simple game. The game only had two characters - a shark and a fish and whenever a shark collided with a fish, the fish sprite disappeared and sent a message to the shark sprite. The shark sprite changed the shark image to simulate the shark opening and closing its jaws and changed the score.



```
when ⚑ clicked
go to x: 30 y: 0
show
forever
  move 3 steps
  if touching shark ?
    play sound Pop
    hide
    broadcast hitbyshark
    wait 1 secs
    go to x: 0 y: 0
    show
  if on edge, bounce
```

Script in Fish Sprite

```
when I receive hitbyshark
change score by 1
switch to costume sharkopenmouth
wait 0.1 secs
switch to costume shark
```

Script in Shark Sprite

Figure 4.7-2 Script to simulate the shark eating a fish

Whether the score is changed by the shark or the fish in this simple game does not really matter. In both cases the score is increased by a point once the fish is eaten by the shark. However when the narrative of the game is changed as happened when a villain fish was added or in the instance when two sharks were used then which sprite attributes the point is of central importance as I explain shortly. The change in narrative acted as a good catalyst for further possibility thinking.

4.7.2.1. Adding a villain fish

One of the game improvement suggestions the children made, was to add several fish each worth a different amount of points. They quickly found a way to create additional fish by duplicating the first fish however they soon realised that all the fish changed the score by one point. During the start of the second session KyleC queried why the shark sprite updated the score rather than the fish sprite. He elaborated that since he wanted to add different fish each worth different points the logic had to change so that the points were attributed by the fish and not the shark. From the discussion that ensued it was clear that the possibility of fish each worth different points was shared by a group of the participants and that whilst thinking on how to make this possible KyleC had homed on the possibility of altering the code to make this feature possible.

KyleC: But how come the points don't come from the fish and come from the shark?
Me: OK because it is the shark who's eating
KyleC: Yes but I thought the fish would get points. Cause like I did the red fish would be the one that gets most points
Me: Ok we can change the place where we add the points
HaydenB: Mela [then] not always the orange fish there'll be a red one and you get bonus points for it.
KyleC: Yes
HallieH: Hey but that was my idea.

4.7.2.2. Adding two sharks

Another discussion which got the workshop participants questioning the validity of the logic implemented in the game was when BenL added a second shark

BenL: I have one question about this. I made 2 sharks but when I control them they move both but only one can eat.
BenL had hit on a problem caused by the way the initial game which I had shared with the students was coded. I had coded the fish sprite to reduce the available live by one when it hits the shark sprite and to then respawn to a new location. *Scratch*'s collision detection block relies on the name of sprites. Since the new shark sprite had a different

name from the original sprite when the fish hit the new shark the code to reduce the available lives and respawn the fish was not triggered.

The urge to change the game, to make it his own, was making BenL engage in active problem solving , taking risks in changing the code to introduce a new feature which he wanted to add to this game. Even though this incident was early on in the workshop BenL was exploring the Scratch environment looking for a solution to his problem. He eventually solved the problem with the help of a teaching assistant using an *or block* so that the code is triggered in both cases - when a fish collides with the original or the new shark.

4.7.3 Stickman duelling with the dragon

Stick with a sword was one of the games developed by BenL. In this game the player is a character called stickman who has to fight a dragon with a sword to progress to a second level. The dragon blows fire from its nostrils every 1.2 seconds. The player has to kill the dragon by using a sword at the same time to avoid getting burnt by the fire.

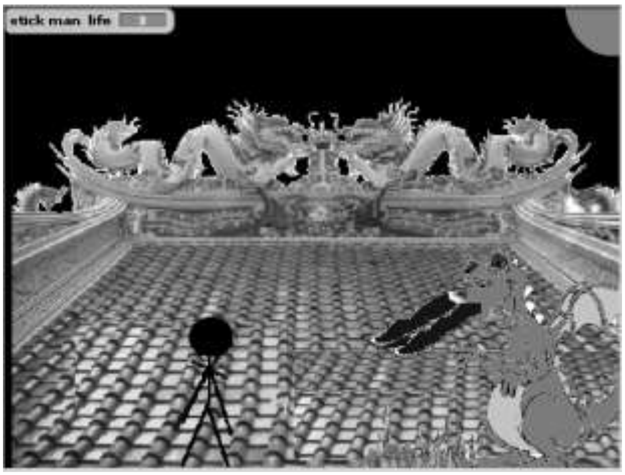
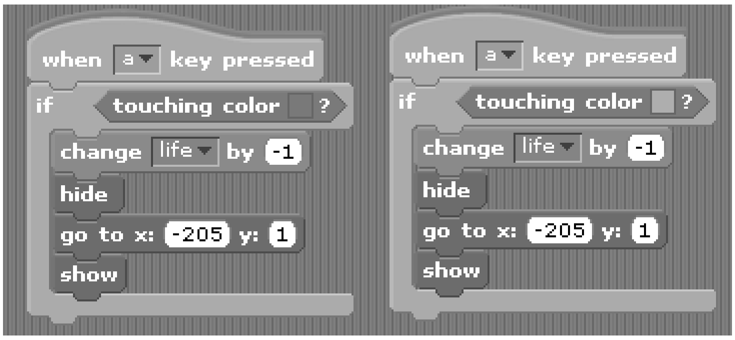


Figure 4.7-3 Stickman and the dragon in Stick with a Sword

What I found extremely interesting in this game was the way BenL handled the fight between the dragon and stickman. *Scratch* does not distinguish between a dragon image and a stickman image. For Scratch they are just images and when one image is placed in the same location as the second image a collision occurs which *Scratch* has to deal with according to the logic coded by the developer. But in this case if part of the image, the fire, hits the stickman then the stickman loses a life whilst if the stickman hits the dragon then the dragon dies.

Rather than using the collision detection block - touching dragon block, as we had done in class with the *Shark and Fish* example, BenL opted to use touching colour block (see Figure 4.7-4 page 138) He edited the tip of fire to be shades of orange and red and then put in logic in the stickman sprite so that when the stickman touches the orange or red colours a life is deducted from the player and the stickman respawns back to life by reappearing in a new location on the screen.



```
when [a ▾] key pressed
if <touching color ▨ ?>
  change [life ▾] by (-1)
  hide
  go to x: (-205) y: (1)
  show
```

```
when [a ▾] key pressed
if <touching color ▨ ?>
  change [life ▾] by (-1)
  hide
  go to x: (-205) y: (1)
  show
```

Figure 4.7-4 Is the fire burning stickman?

Here again the game-making environment had acted as fertile ground for the workshop participants to engage in active possibility thinking. At this stage in the workshop the participants were actively creating their own games rather than adapting games created by someone else. BenL was in control of the game narrative. He could have easily changed the narrative to avoid moving into this uncharted territory of dealing with collisions between images in two different ways depending on where the collision occurs. He could have stayed on solid familiar ground by using collision detection as he had done before. However gaming is all about taking risks and so BenL was all for trying new approaches, experimenting and taking risks. In so doing he developed an innovative way of dealing with the problem at hand.

4.8 Conclusion

In this chapter I discussed the main themes that emerged from the data collected through observations and informal conversations with workshop participants.

I highlighted the importance of game making for the participants (see section 4.3) and how game playing infused into game making (see section 4.4). I also looked at the ways the children extended and used their gaming capital (see section 4.5) and how they took to changing the narrative of the example games whilst retaining the same ludology (see section 4.6) . Finally this chapter included a discussion about how game authoring allowed the workshop participants to engage in possibility thinking (section 4.7).

What is missing is an in depth analysis of some of the games created by the children. In the next chapter I analyse a selection of three games made by three different workshop participants during the second phase of the workshop, first from a gaming literacy perspective and then from a computational thinking perspective.

5. Simple Narratives, Complex Thinking

5.1 Introduction

In this chapter I analyse a selection of three games made by the workshop participants. I start off by looking at what makes these artefacts a game and analyse the similarity of these games to the game created during the initial part of the workshop or games discussed in the videos posted in the resource section of the workshop portal.

I examine the games from a game literacy perspective and then look at computational concepts and practices that the participants adopted during the game construction exercise.

In the final section I reflect on the parallels between the game development practices adopted by the children and the software development practices adopted in the industry.

5.2 The Games

In the last session of the workshop I asked the participants to provide some information about the games they worked on during the final 6 weeks of the workshop. I asked them to write a general description of the game and instructions on how to play it. Table 6 contains the list of games provided by the children. This list does not include remixes of the *Shark and Fish* game which the children modified during the first 4 sessions of the workshop.

Name/s	*Game*	*Description/Instructions:*
AlaaE and MariaChristinaM	The Ball and the Box	You have a box which you move using the left and right arrow keys. You need to catch the falling basket balls. Boom = -1 life and Spotted Ball = +1 life. have fun playing and good luck playing
BenL	Stick with a sword	Stick with a Sword is a game in which you need to kill the dragons it is a short but fun game pleas ENJOY it ;].

the instructions are in the game hahahahahahaha

Name/s BrandonA

Game arkanoid taikem

Description/Instructions:

My game is a simple game.You need to simply break blocks with the ball and the paddle needs to hold it from falling into the black part.To win you'll have to destroy all the blocks and you'll lose when all your lives are out.You are the paddle.

Name/s Daniel9000

Game Bob’s Adventure

Description/Instructions:

its about bob on an adventure you are bob and you have get to the finish
press space to turn and -> to move

Name/s Hellieh

Game Newgame

Description/Instructions:

their is a bird flying and you have to shoot it to win
press the arrow buttons to move the shooter
then you press space bar to shoot and try and hit the bird

Name/s JacquesC

Game soccer cup

Description/Instructions:

world cup final and you are losing 7-0

Name/s KyleC

Game Racer

Description/Instructions:

You have to drive a car around a track with obstacles and not touch white and complete it before time runs out.
You have to press the top arrow key to drive, the right arrow key to go right, the left arrow key to go left and press the back arrow key to reverse.

Name/s MichelaA

Game Click the Zebra

Description/Instructions:

You are the lioness. You have to click on the zebra to get points and you have to avoid the cans....

Name/s MiguelB

Game Bear Escape

Description/Instructions:

there is this bear trying to esape this military lab and the bear needs to encouter various mazes to esape the game is impossiple to win so if you beat any levels contact me so I make you champion of the level the instructions are to move space key to rotate is down arrow cheat skip levels uparrow	
Name/s Serafina901	*Game* The Lion, the Hunter And the Policeman
Description/Instructions: There is a hunter who is trying to shoot ta lion for hunting season but this year the policemen are sneaky and can easily catch you shooting . The hunter needs you to help shoot the lion and avoid the policemen press space bar to soot bullet	

Table 6 The games as described by the workshop participants

Although some of the games were variants of the games discussed on the workshop website, all the games created were fun to play and uncover game literacy practices adopted by the children. *Soccer Cup* was very similar to *Penalty Shoot Out Game* whilst *The Lion, the Hunter and the Policeman* was a remix of *Shooting a Vase* with more characters added on. HellieH and MichelaA worked together throughout the workshop to create *Click the Zebra*. This game allowed the player to use the mouse to click on zebras which appear randomly on the screen and gain points. In the final session of the workshop HallieH decided to create her own game which turned out to be a remix of the *Shooting a Vase* game available from the workshop portal Another two children AlaaE and MariaChristinaM decided to work together to build a three level game they called *The Ball and the Box.*

Stick with a sword and *Arkanoid Taikem* were short one-level games. *Stick with a sword* had a very interesting narrative of a stickman battling with a dragon (see section 4.7.3 Stickman duelling with the dragon page 137) whilst *Arkanoid Taikem* was a port of the popular game Arkanoid. BrandonA was such a fan of the traditional arcade game that he decided to recreate it in *Scratch.*

Two of the more popular games to play-test during the workshop were *Bob's Adventure* and *Racer*. *Bob's Adventure* became so popular that MiguelB decided to create a similar game which he called *Bear Escape*. The narrative of *Bear Escape* was different from *Bob's Adventure* but the ludology was basically the same.

Picking three games to discuss in depth in this chapter was not an easy task. At the end I decided to analyse in depth following three of the games created by the children:

- *The Ball and the Box* created by two workshop participants. AlaaE and MariaChristinaM.
- *Bob's Adventure* created by Daniel9000
- *Racer* by KyleC

I chose *Bob's Adventure* and *Racer* because they were two of the most popular games with the children following the workshop. *The Ball and the Box* was special because it was developed by a pair of students. The students were not constrained into working in groups or on their own. Indeed MariaChristinaM and AlaaE worked on their own during the first part of the workshop but then decided to team up when the students were asked to create their own game.

5.2.1 The Ball and the Box

The game created by two of the workshop participants AlaaE and MariaChristinaM is a ball and paddle action game similar to traditional games such as *Pong* and *Breakout*. In these games the player uses controls on the keyboard or game console to move a paddle and hit a ball which falls from the opposite direction of the paddle. Similar to the physical sports tennis or squash, in *Pong* the player would lose a point if the ball got past the paddle without being hit.

The narrative in *The ball and the Box* is a very simple one, as demonstrated by the game's name chosen by the authors. The player has to catch a ball dropped from the top of the screen in order to score points. Whilst catching balls one must shun bombs and catch bonus balls. Although *The Ball and the Box* is very similar to the traditional Pong, there are some notable differences. In *The Ball and the Box* game a ball is dropped from the top of the screen whilst a box is used instead of a paddle. Rather than bouncing the ball with the paddle, the ball needs to be caught in the box for the player to gain a point. If the player misses a ball no points are lost the ball will simply drop off the screen and reappear in a new location at the top of the screen. The end effect on the game of failing to catch a ball is that the player will take more time to advance to the next level. In the second level bombs are added to balls. A player must avoid the bombs and collect as many balls as possible to move on to the third level. Hitting a bomb would reduce lives with the possibility of bringing the game to an end once all the lives are taken up.

Consalvo (2007) emphasizes that gaming capital changes over time and across types of players and games. Through the game-making experience AlaaE and MariaChristinaM found an avenue where they could express their gaming capital. Through the game created they demonstrated knowledge about digital games they picked up through years of gaming and transacting in gaming capital with their peers and online. In the next section I look into the defining characteristics of *The Ball and the Box* game to show how the game created subscribes to the defining traits of a digital game

5.2.1.1. Defining traits of The ball and the box

McGonigal (2012) extenuates that when one removes the genre differences, all games share four defining characteristics:

- goal
- rules
- feedback system
- voluntary participation

Voluntary participation relates to the participation in the game through playing the game, however the other three characteristics are all design related and are present in the game by AlaaE and MariaChristinaM.

5.2.1.1.1. Rules

In *The Ball and the Box* the player has to move the box and collect as many balls as possible whilst dodging bombs which appear in level 2. A point is scored whenever a ball is collected by the player. If a bomb hits the box a live is lost. When the number of lives reaches zero the game ends. A player can replenish lives by collecting special balls which appear in the third level of the game. When a special ball is collected no points are scored but the number of available lives is increased by one.

The player progresses to level 2 by scoring 20 points and to level 3 when the score reaches 50. The game is won when the player gains 80 points.

5.2.1.1.2. Goal

The goal of this game is quite simple, that of arriving at the winning screen by collecting 80 points without losing all the lives.

5.2.1.1.3. Feedback

This game uses three feedback mechanisms: levels; points, and lives as follows.

Lives	When the game starts the player is awarded five lives. Whenever the playing character hits a bomb a live is lost and the feedback system is updated accordingly.
Points	This feedback mechanism displays the amount of points accumulated so far by the player.
Levels	When the player accumulates enough points to move to a new level the level number displayed in the middle of the screen changes to show the current level of the game.

Table 7 Feedback in The Ball and the Box

5.2.1.1.4. Characters

The Ball and the Box includes a number of characters through which the gaming unfolds. The box is the character which the player, or what Galloway (2006) calls 'operator', uses to play the game. The other three characters are all operated by the machine. Figure 5.2-1 displays the four characters used in this game: the ball, the bomb, the bonus ball and the box.

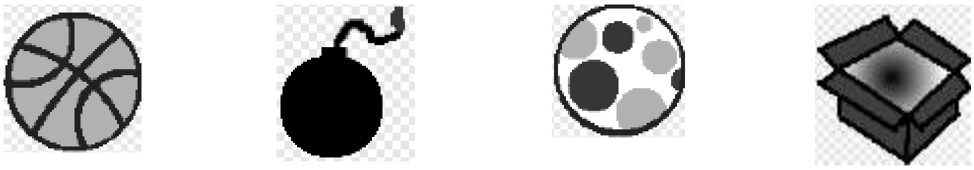


Figure 5.2-1 The Characters

5.2.1.1.5. Machine Operated Characters

The ball is a central character in this game since it is present throughout the game. It is displayed at random locations on the top of the screen and falls to the bottom of the screen. When it reaches the bottom of the screen or hits a box it is displayed again in a random location on the top of the screen to start the journey to the bottom of the screen.

Bombs appears in level 2 and stay on throughout the game. When a bomb hits the box, a life is lost. Code is used to animate the bomb so that when a bomb hits a box the graphic momentarily changes to an explosion image.

Bombs appear in random places on the top of the screen and fall slowly to the bottom of the screen at a constant speed. Once the bomb is out of view it is displayed again at a new random position on the top of the screen.

The bonus ball is introduced in level 3 and is used to replenish lives. Similarly to the ball and the bomb it appears in random positions at the top of the screen and falls to the bottom of the screen at the same rate as bombs.

5.2.1.1.6. Player Operated Characters

The player character in this game is a box which the player can move on a horizontal axis at the bottom of the screen by using the left and right arrow keyboard keys. When the box reaches the end of the screen it is not allowed to proceed out of screen.

5.2.1.2. Background Scenes

The characters in the game operate with a scene in the background which is changed whenever a level is won. In this game the background scenes are linked to the perceived difficulty of each level. The scene showing a blue sky with two white clouds was reserved for the first level where the player does not loose points by missing balls. The second background showing grey clouds and rain was used in the second level of the game where the player runs the risk of losing the game by hitting bombs instead of catching balls. The third level allows the player to recoup lives by catching bonus balls and the background changes from the rain adopted in level two to a sunset background with birds flying in the sky.

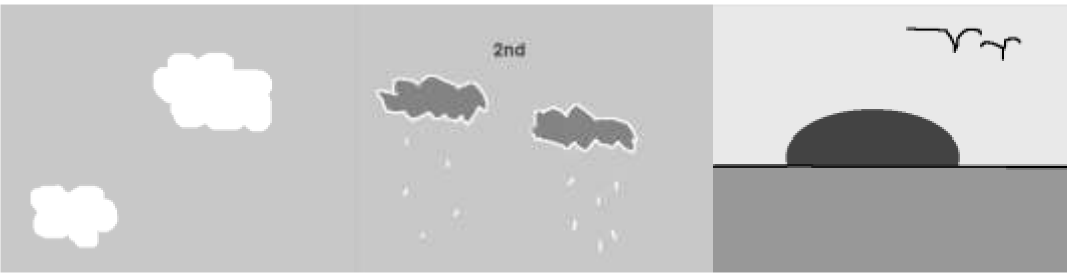


Figure 5.2-2 Background scenes in The ball and the Box

5.2.1.3. Similarity of the game to the game built during the workshop

A number of the suggested improvements for the shark eat fish game put forward by the workshop participants centred on the concept of making the life of the shark harder by providing obstacles for the shark. A number of participants suggestions included:

- "*making it a bit more challenging, there could be obsticals* [sic] *like bombs*"
- "*obstacle* [sic] *like sinkind* [sic] *rubbish*"
- "*the higher the level you go you will get more things in the way like bombs and stuff you have to dogde* [sic] *like cans and plastic bottles*."

Since these suggestions were quite popular, during the second session of the workshop we discussed the implementation of an obstacle in the form of a can which someone drops into the sea and enters our game world at the top of the screen sinking down to the bottom of the sea. If this can hit the shark, the player would lose a life and the shark reappears at a different location on the screen.

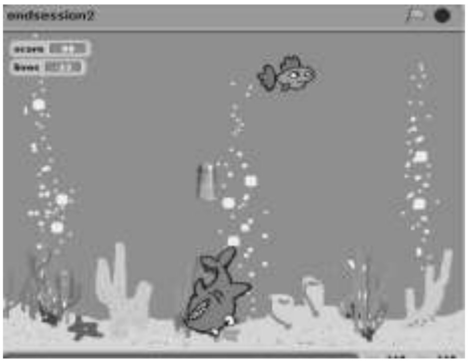


Figure 5.2-3 Obstacles in the Shark

In *The Ball and the Box* game, the feature of the dropping can was adapted for the three non-playing characters, that is the bomb, the ball and the bonus ball. These non-playing characters were similarly coded to drop from the top of the screen, however they all had different effects on the game. The ball and the bonus ball both increased the chances of the player winning the game by increasing the score or adding lives respectively. On the other hand the bomb reduced lives and the instance of the bomb hitting the box is followed by an animated explosion. The process of creating an animated explosion was not discussed in class but was shown in the video *Year 3578: Saving the earth*. In this video, uploaded to the resources section of the workshop website, I show how meteors can be made to explode when hit by bullets shot by the game player. The same technique was used to animate the explosion that is displayed when a bomb hits the box.

The code used to create the animation of the can falling to the bottom of the sea and the meteors exploding have been appropriated by AlaaE and MariaChristinaM to create their own characters in the *The Ball and the Box* game, similar to how a musician borrows elements of a composition to create a new piece of music. The game creation exercise provided AlaaE and MariaChristinaM the avenue to participate in social creativity (Fischer 2005).

5.2.2 Bob's Adventure

Bob's Adventure is a maze game developed by Daniel9000. In the first few sessions of the game making workshop Daniel9000 had already shown a big interest in maze games which he used to draw on A4 sized sheets of paper (see section 4.3 page 119).

Bob's adventure shows Bob, a player operated character, which has to navigate a series of mazes outplaying machine operated characters. Bob has to switch off laser beams to reach the door to the next level placed at the bottom of the screen. Daniel9000's fascination with the popular game *Minecraft* is evident throughout *Bob's Adventure*. *Bob's Adventure* includes a number of characters found in *Minecraft*. Most of these characters are stationary characters which are simply placed in some of the levels. *Creeper*, one of the characters which spawns in the *Minecraft* world at night, makes an appearance in *Bob's Adventure* too as a machine operated character and the player has to use skill to outmanoeuvre *Creeper* to make it to the next level.

This fusion between the player's game playing world and the game he created is also evident with the inclusion of another character from another epic maze game – *Pacman*. Daniel9000 decided to include the characters that inhabit the games he enjoys playing in the game he designed and created.

Daniel9000 himself features in the game. Whenever the player moves *Bob* over the maze's border and loses the game, Daniel9000 can be heard saying "Ha Ha" in a voice-over he recorded and inserted in the game.

5.2.2.1. Defining traits of Bob's Adventure

Similarly to *The Ball and the Box, Bob's Adventure* includes the three design related characteristics making up a digital game.

5.2.2.1.1. Rules

There is one simple rule in *Bob's Adventure*, avoid the obstacles and get *Bob* safely to the finish line of each level. To do this *Bob* must avoid lasers which block the way, and machine operated characters which appear in some levels and patrol the screen. If *Bob* hits one of these obstacles, the game is lost and Bob has to start again at level 1.

Bob can be moved using the right arrow and turned using the space bar. Although there is welcome screen as in other games created by BenL and KyleC the keys the player has to press to move Bob on the screen are shown on the first screen on the game (see section 4.4.2 page 123).

5.2.2.1.2. Goal

The goal of this game is to get *Bob* to the next level. There is no winning screen in this game. As soon as the player wins level 15, the level number is changed to 16 but the background and obstacles do not change. The player can keep playing the same level, even though the level number is incremented by 1 whenever the player wins the level. It's as if Daniel9000 did not want the game to end, with *Bob* predestined to spend its existence in this ever lasting journey in a maze that never ends.

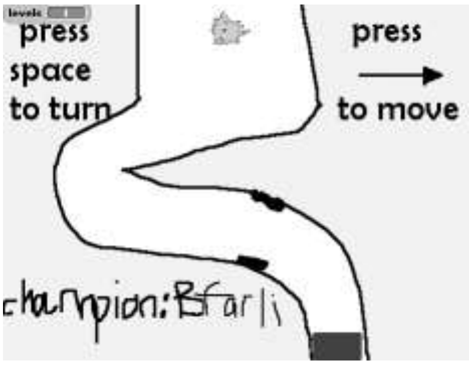


Figure 5.2-4 First screen of Bob's

5.2.2.1.3. Feedback

This game uses one on-screen feedback mechanism: level. The level indicator on the top of the screen shows the screen that the player has managed to arrive to.

Scratch, the game development language used by Daniel9000 did not provide the ability to store information from one game execution to the next. As soon as the player stopped playing the game the feedback information disappeared. This made it impossible for Daniel9000 to store the name of the player who advanced most in the game. To make up for this feature Daniel9000 wrote the name of the player on the background of the opening screen as can be seen in Figure 5.2-4.

5.2.2.1.4. Characters

Bob meets a number of characters whilst traversing the maze screens in the different levels. By characters I am not simply referring to objects that move in the game but to objects that participate in what Salen and Zimmerman (2003) call the *system*.

Creeper and *Pacman* come straight out of games played by Daniel9000 whilst *Dangerboy* is inspired from these games. Other characters such as the finishing line and lasers fit more with the definition of an object which makes *Bob's adventure* more exciting.

The use of *Creeper* and *Pacman* in *Bob's Adventure* reminds me of what Jenkins (2006) had written about fans: "One becomes a 'fan' not by being a regular viewer of a particular program but by translating that viewing into some kind of cultural activity … for fans consumption naturally sparks production, reading generates writing" (p.41). Daniel9000 is passionate about digital games. Being a fan for Daniel9000 does not only mean playing the games but also writing his own games based on the characters of the original games. This does not imply that the game created by Daniel9000 shares the same game plot as *Minecraft* or *Pacman* however the borrowed characters placed in *Bob's adventure* share many of the original characteristics as I elaborate shortly.

5.2.2.1.4.1. Bob

Bob is the player operated character which the player has to guide to the finishing line in each level to succeed in moving onto the next level. Daniel9000 used a standard graphic available from the library of images installed by default with Scratch and also used the graphics's name as the name of the character. *Bob* can be operated by using two key strokes, the right arrow to move forward and the spacebar to turn anti-clockwise. The cryptic movement makes navigating *Bob* in the maze harder for the player. Most of the passage-ways *Bob* has to navigate through are quite narrow and it is very easy to hit the black border which instantly ends the game. Had Daniel9000 used an easier navigation mechanism, for example one which allows *Bob* to be turned in both directions, the difficulty of the game would have been drastically reduced interrupting the flow that the player gets into when playing the game.

5.2.2.1.4.2. The finishing line

The finishing line is a character which is present in every level. This character is the one which drives the game since whenever Bob touches the finishing line the background changes and a new level starts again.

5.2.2.1.4.3. Lasers

When *Bob* enters the third level, the player finds the passageway blocked by laser beams. No instructions are provided on how to deactivate the laser beams and the player has to rely on previous gaming experience gained in other games to deactivate the laser beams. The player soon learns that the laser beams can be deactivated by navigating *Bob* onto the coloured patches of the screen.

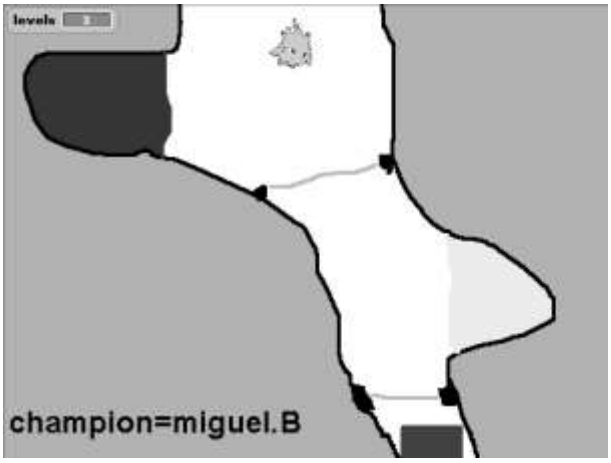


Figure 5.2-5 Laser beams in Bob's Adventure

Daniel9000 shows great skill in the design of *Bob's Adventure* by using the player's experience gained by deactivating the lasers in level three to enhance game play in latter levels of the game. In levels seven, ten and eleven Daniel9000 introduces booby traps in the form of coloured patches. The patches are outlined by a black border. As soon as the player tries to navigate *Bob* onto the coloured patches, in a bid to unlock some feature of the game as happened in level three, *Bob* hits the black border and the game restarts. In level fifteen the same tactic is used to activate another game character *Dangerboy* (see section 5.2.2.1.4.6 page 154).

5.2.2.1.4.4. Creeper

The *Creeper* character was borrowed from the popular game *Minecraft*. In *Minecraft Creepers* are the equivalent of suicide bombers, hostile characters who sneak up to the player and blow up inflicting a lot of damage to the players, animals and blocks in the vicinity. Daniel9000 used the same character in level twelve and thirteen of *Bob's Adventure*.

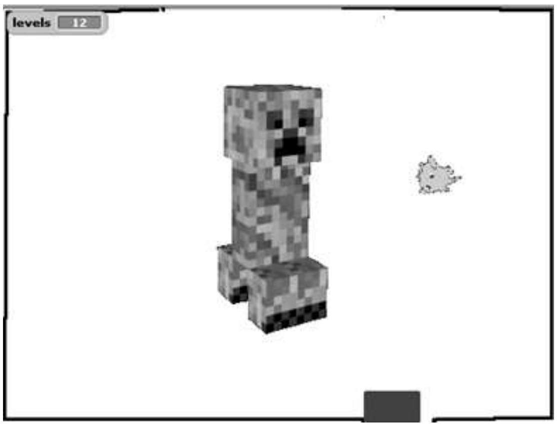


Figure 5.2-6 Creeper in Bob's adventure

Creeper patrols the screen of levels twelve and thirteen by moving horizontally across the screen. *Bob*, the player operated character, has to dodge *Creeper* to make it to the finishing line of the screen. The game is lost if *Bob* accidentally bumps into *Creeper*

Although *Creeper* retains the property of inflicting damage to the player when physical contact occurs, there are a number of differences between the two characters. In *Minecraft Creepers* tend to sneak up to the player and explode. Players have to use a number of tactics to run away from *Creepers*. In *Bob's Adventure* C*reeper* moves along a horizontal line. This makes its movement predictable to the player. However the size of Creeper in *Bob's Adventure* is much larger when compared to the character in *Minecraft*, hence the surface area and possibility of impact is greatly increased. This design approach adopted by Daniel9000 balances out the ease of predicting the movement *Creeper* will adopt since *Bob* has to choose the appropriate time to cross across the screen so as not to collide with *Creeper*.

5.2.2.1.4.5. Pacman

Another character borrowed from a popular arcade game is *Pacman*. In the traditional arcade game *Pacman* is the player operated character which the player uses to eat pacdots and gain points. In *Bob's Adventure Pacman* is a machine operated character that acts as a threat to *Bob's* journey across the maze.

Pacman traverses the screen in a horizontal direction threatening to collide with *Bob* and sending him back to level 1. As in the case of *Creeper*, the size of *Pacman* is used to offset the predictability of Pacman's movement. In level thirteen Daniel9000 added the *Pacman* character to the level whilst retaining *Creeper*. These characters both patrol the screen on the same horizontal line, however *Pacman* moves with a slightly faster pace than *Creeper*. This makes it even more difficult for *Bob* to traverse the screen without being hit as the player has to time his attempt to cross the screen so that both *Creeper* and *Pacman* are a considerable distance away.

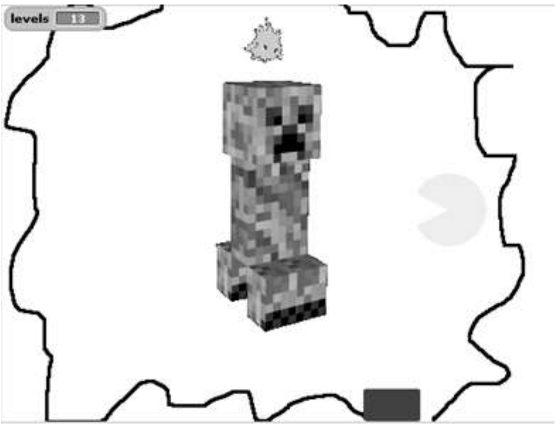


Figure 5.2-7 Pacman makes Bob's adventure a bit harder

5.2.2.1.4.6. Dangerboy

Dangerboy is another machine operated character present in *Bob's Adventure*. Similarly to *Creeper* and *Pacman*, the player must dodge *Dangerboy* to make it to the finishing line of the fifteenth level. What makes *Dangerboy* different is that way it appears on the screen. *Dangerboy* is not displayed immediately in the fifteenth level but appears only when the player moves *Bob* over the area of the screen that is coloured in light blue. It is possible for the experienced player of *Bob's Adventure* to avoid activating *Dangerboy*, however given the previous experience with deactivating laser beams a novice player is bound to guide Bob straight into the booby trap. *Dangerboy* demonstrates Daniel9000 skill in increasing the flow of the game by exploiting the player's previous experience in the game.

5.2.2.2. Background Scenes

The background scenes in *Bob's Adventure* do not just provide a backdrop on which the game is played but also serve an operative function. The backgrounds provide a path bound by black lines in which Bob can operate. Whenever Bob hits the black boundary, the game comes to an end and Bob is respawned to level 1.

Daniel9000 uses the background scenes as a canvas on which to draw his maze where Bob can navigate. The maze is not just a pathway; it includes stick figures and other characters which *Bob* has to circumvent to get to the finishing line of each screen. The influence of the popular game *Minecraft* can be seen in a number of background scenes in *Bob's Adventure*. *Steve*, the player in *Minecraft*, and the mobs *Creeper* and *Spider* are all included as stationary images in levels 7, 8, 9 and 10.

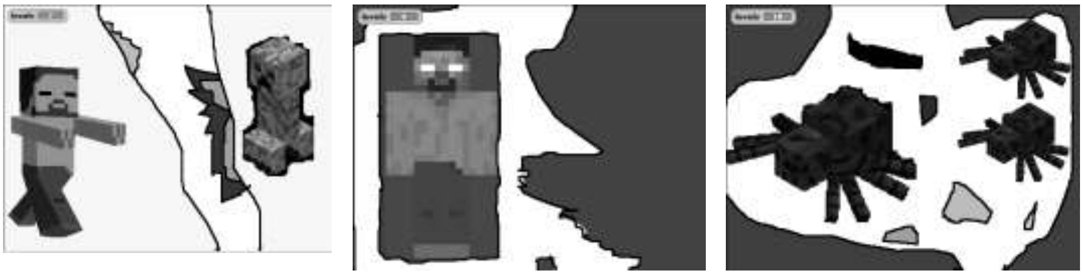


Figure 5.2-8 Minecraft inspired background scenes in Bob's Adventure

The background scenes were also used by Daniel9000 as a canvas where to provide textual information for the player. The first screen of the game contains the keystrokes to use to navigate *Bob* as well as the overall champion of the game that is the player who managed to complete the levels of the game. Typically games would contain a hall of fame where the players who attain most points would be listed. Daniel9000 did not have enough programming knowledge to be able to implement a hall of fame and therefore resorted to writing the names of the players on the background. Some of the initial levels included other players listed as champions. These levels were the more complicated of the series of levels making up *Bob's Adventure* and so merited their own champion. The awkward navigation controls which Daniel9000 used for Bob made navigating the maze quite difficult and levels 2, 3 and 4 were especially challenging given the curvy nature of path Daniel9000 chose to implement.

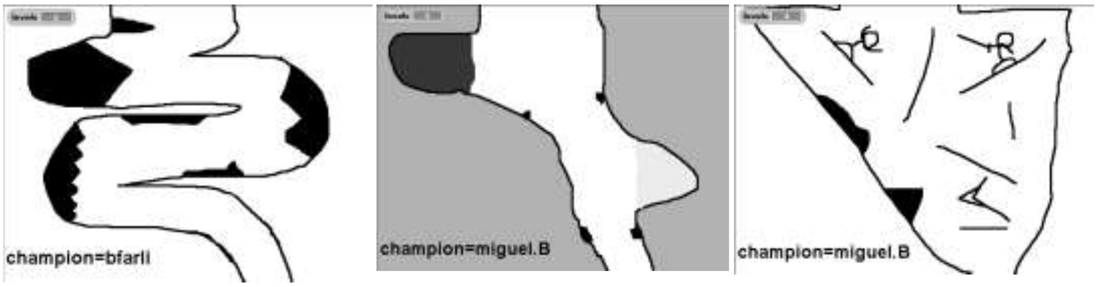


Figure 5.2-9 Champions for different levels

5.2.2.3. Similarity of the game to the game built during the workshop

Bob's Adventure had been planned by Daniel9000 well before the start of the workshop. Daniel9000 had drawn the scheme of the game (see Figure 4.3-1 page 121) which he had initially called *The Clash of the Titans*. Notwithstanding this, *Bob's Adventure* does contain similar features to the *Shark and Fish* game developed in class. In the *Shark and Fish* game the shark is moved using two keys, the left arrow to move the shark and the up arrow to turn the fish. This made it harder for the player to manoeuvre the shark and eat more fish. In *Bob's Adventure* Daniel9000 adopted the same strategy of not providing more controls to move *Bob*, making the game more challenging as the awkwardness of the controls made it quite difficult to manoeuvre *Bob* in the tight passages of some of the levels.

The machine-operated characters *Creeper*, *Dangerboy* and *Pacman* also used movement which was similar to the fish in the *Shark and Fish* game. However the fish in the *Shark and Fish* game was a prey which the player operating the predator had to eat, whilst in Daniel9000's game the machine operated characters were predators which the player had to outmanoeuvre.

5.2.3 Racer

Racer is a racing video game developed by KyleC. KyleC listed *Formula 1 2012* as his preferred game when filling in the initial questionnaire upon joining the workshop. The game he played nearly every day inspired him to build a game of the same genre. *Formula 1 2012* is a semi-simulator racing game based on the 2012 Formula One season and features all the twenty four circuits and Grand Prix included in the 2012 championship. The game claims to be a faithful virtual edition of driving a racing car.

KlyeC's *Racer* is not a simulator, it does not include the complex mathematical code that is embedded in simulators that render driving the simulator as close as possible to driving the real car. However *Racer* includes a series of features which make playing the game incredibly challenging. In *Racer*, the player is presented with a top view perspective of a racing track where the car being driven is viewed from the top. The player has to drive the car to the finishing line whilst driving around stationary obstacles. Whenever the car hits an obstacle or is driven off-track the game is lost and a *Game Over* screen is displayed explaining the reason why the game ended. To make the game more challenging the player has to complete the track within a hard-wired time interval.

5.2.3.1. Defining traits of Racer

Racer too subscribes to the characteristics making up a game (see section 2.2.4 page 32).

5.2.3.1.1. Rules

The main rule in *Racer* is to arrive to the finishing line of the racing track in under 1250 time-ticks[8] whilst ensuring that the car is not driven off-track or into one of the numerous white objects which litter the racing track. The fixed value of 1250 was not randomly selected. During the workshop sessions I could observe KyleC fine tuning the number until only some of the players testing the game could finish the game in time.

The car can be navigated forward or backward by using the up or down keyboard arrows, whilst the car can be stirred towards the left or the right using the left/right

[8] A time-tick is a measure of time which is dependent on the speed of the computer where the game is running.

arrow key on the keyboard. This allows the player to reverse the car if the car is navigated into a tight spot. However when this happens valuable time is lost by the player making the possibility of winning the game even harder.

The size of the car and objects were engineered in a way to make it very challenging for the player to drive the car without hitting an obstacle. Even the curvature of the track was drawn in a way that a player can easily go off track when the screen changes from one portion of the track to another. All these game features increase the challenge offered by the game as the game is still winnable. I must admit that it took me well over an hour playing this game until I finally managed to safely guide the car to the finishing line.

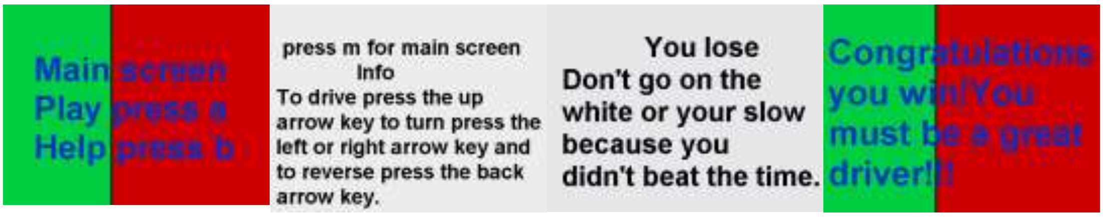
Although there are no levels in *Racer* the track is split up into a number of screens with the screens increasing in complexity as the game progresses. It was evident that KyleC managed to bank on his extensive prior gaming experience to build this game.

5.2.3.1.2. Goal

The goal of this game is to arrive to the finishing line of the racing track without hitting any white obstacles on the track and in under the limit time.

5.2.3.1.3. Feedback

There are a number of feedback mechanisms implemented in this game. These include a number of screens shown at various stages of the game and a time counter.



press m for main screen
Info
To drive press the up arrow key to turn press the left or right arrow key and to reverse press the back arrow key.

You lose
Don't go on the white or your slow because you didn't beat the time.

Congratulations you win!You must be a great driver!!!

Figure 5.2-10 Menu, Help and Ending screens in the game Racer

The game’s opening screen displays a menu and allows the player to display a help screen. The help screen includes the instructions needed to navigate the car in the maze. The screen does not include information about the obstacles that need to be avoided or the requirement to keep the car on track all the time. This information is instead provided the first time the car hits an obstacle or is driven off-track in the *Game Over* screen. KyleC uses the same screen for all the conditions that end the game that is driving into an obstacle, driving off track, as well as being too slow in completing the game. Hence the *Game Over* screen acts as a help screen too as it informs the player on what actions to avoid in order to win the game at the next attempt. The final feedback screen is displayed whenever the player manages to navigate the car to the finishing line and win the game. This screen congratulates the player on achieving this and being a "great driver".

Another feedback mechanism adopted by KyleC is the time counter shown at the top of the racing track. This counter shows the time elapsed playing the game. This feedback mechanism is a very important one given that winning *Racing* is a game against time. At no point does KyleC list the time limit imposed on the players playing the game. It took KyleC a number of sessions observing peers play testing his game to arrive to the optimal time limit value for his audience.

5.2.3.1.4. Characters

The only character in *Racer* is the player operated racing car. The image used for the racing car was downloaded from the internet and was initially drawn on a white background. KyleC spent a considerable amount of time cleaning the image to replacing the white background with a transparent background.

Initially KyleC had designed a shop where the player could purchase different cars to replace the standard red card (see section 4.5.2 page 127). This shop was not implemented.

5.2.3.2. Background Scenes

The background scene in *Racer* is the racing track where the car is navigated. KyleC uses a top view where the player can see the part of the track where the car is currently placed. The hand drawn racing track is divided into five sections. Each section leads into the next one and as soon as the car moves into the new region the background is switched to show the new section. In Figure 5.2-11 I stitched the five sections together by placing them next to each other to show the entire racing track. The track resembles a typical racing track with parts which are straight, were the cars can pick up speed, and others which are curved and tend to pose a bigger challenge for the driver to keep control of the car.

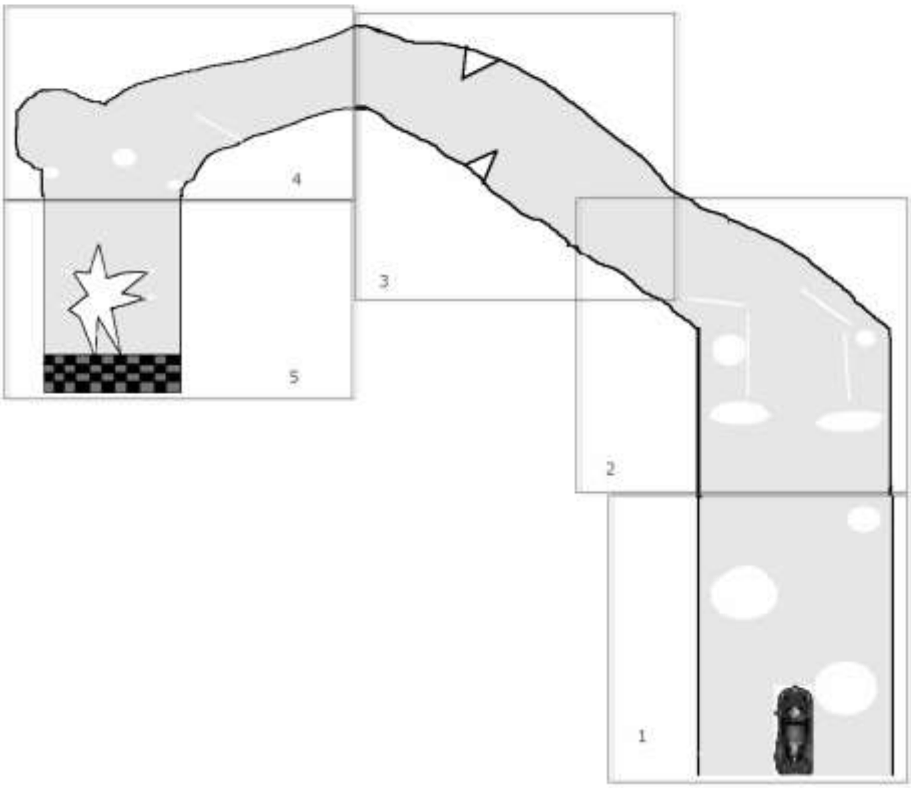


Figure 5.2-11 Racer's racing track background

Each section is progressively more difficult for the player to navigate the car without crashing into an obstacle. The last section is the most difficult to navigate because the white lines which protrude from the white star in the middle of the track leave barely enough space for the car to squeeze by to get to the finishing line.

The track in the initial versions of the game did not include the obstacles. Obstacles were added in later sessions by KyleC to add to the challenge offered by the racing game.

5.2.3.3. Similarity of the game to the game built during the workshop

Racer is significantly different from *Shark and Fish* game. The obstacles were not implemented as different characters as in all the games discussed in class or uploaded as videos on the website. Instead KyleC decided to use a technique that we had discussed in class whilst looking at different ways of building a refuge for the fish in the *Shark and Fish* game (see section 4.7.1 page 133). This technique made the coding considerably easier for KyleC and achieved the same result of having obstacles the car needed to avoid to get to the finishing line.

5.2.4 Conclusion

The three games presented above are a representative sample of the games created by the children during the latter part of the workshop. The narrative of these games was quite simple, however the simplicity of the narrative allowed the children to think in complex ways and to express their knowledge of games in the artefacts they created. In the next section I analyse the games from a games literacy perspective.

5.3 Games as Literacy

In their model for games and literacy Beavis and Apperley (2012) focus on two intertwined perspectives on games as literacy: games as text, and games as action. When applying this model to *The Ball and the Box, Bob's Adventure* and *Racer* one can notice that the games, and the journey to build them, resonate with a number of themes (sectors) in this model:

5.3.1 Games as text

5.3.1.1. Knowledge about the games

One of the sectors from the games as text layer of the model which comes to the foreground in these games is the **knowledge about games** sector. For the game to be effective the game designers drew on their knowledge of games, the gaming capital they built through the years.

5.3.1.1.1. How difficult should a game be?

Games should be hard fun and evoke eustress - a combination of well-being and stress. Hard fun results in Fiero or what we feel when we triumph over adversary (McGonigal, 2011, p. 33). Managing difficulty in a game is a crucial aspect of creating a challenging game. Players tend to give up on games which are too easy or too difficult to achieve and hence a game designer needs to walk the tight rope of using the right amount of difficulty to achieve a balanced game (Habgood & Overmars, 2006). A well-balanced game should lead to the state of flow (Csikszentmihalyi 1996).

The concept of managing difficulty was not discussed in the workshop sessions. I briefly touched upon the topic in the videos on creating the *Penalty Shoot Out Game* and *Year 3578: Saving the earth* uploaded to the workshop website. Notwithstanding the lack of discussion in class all the games reviewed managed to tackle the issue of difficulty successfully when developing their games.

Whilst playing *The Ball and the Box* I got absorbed trying to catch balls whilst avoiding bombs. The game started off with just balls dropping from the top of the screen only to add another challenge in the form of bombs in the second level just when the game was becoming unchallenging for the player. The game took another twist with the addition of bonus balls to the bombs when it started getting difficult to reach the goal of winning the game in level 3. AlaaE and MariaChristinaM demonstrated that their *knowledge about games* helped them master an important notion in the design of digital games.

A similar experience was observed whilst playing *Bob's Adventure* and *Racer*. It was evident that *Scratch* did not allow Daniel9000 and KyleC to implement all the features they wanted to include in their games. Daniel9000 included the *Minecraft* character *Creeper* in the game. In the original game Daniel9000 played, *Creeper* sneaks behind the back of the player before exploding and inflicting damage on the player. *Creeper* does not simply move to the player's location. It first wanders around in *Minecraft* and then homes into the player once it is near the player. Programming a character like *Creeper* requires more elaborate programming skills which the students clearly did not possess. But this did not discourage Daniel9000. Instead he enlarged the image of *Creeper* so that the area of impact between *Bob* and *Creeper* was large enough to pose a challenge for the player operating *Bob*. The next level included *Pacman* to make the game even harder ensuring that the absolute difficulty of challenges included in the game were increased over time.

Awareness of difficulty management can also be seen in *Racer*. The track in the game starts with a straight part and a small number of obstacles. The track then turns into a curved one in sections two to four, with the last section of track containing a star shaped obstacle with protruding lines making it quite difficult to navigate around (See Figure 5.2-11 *Racer's* racing track page 159). KyleC then went on to add a race against time element to the game to ensure that the in-game experience gained by the player does not make the challenges start to feel as if they are getting easier.

Adams and Rollings (2007) differentiate between different types of difficulty of challenge in games. The absolute difficulty of a challenge is calculated by comparing the skill required to meet a challenge and the amounts of stress the challenge imposes when compared to a trivial challenge of the same type. However as players progress through a game and gain in-game experience the easier they will perceive a given type of challenge to be.

Guidelines for a balanced game	Game Examples
The absolute difficulty of challenges included in the game should be increased over time.	*Racer* starts off with a straight track and over the course of the game obstacles are added increasing the difficulty of the game
The power available for players to meet the challenges should be increased at a lower rate than the rate of increase of absolute difficulty	In *The Ball and the Box* bonus balls are added in level 3 to provide lives for the player. The bombs in this level were retained.
A game designer should ensure that the player does not gain in-game experience so fast that the challenges start to feel as if they're getting easier rather than harder	In-game experience was used to the player's disadvantage in *Bob's Adventure* when coloured areas were used to activate *Dangerboy* when earlier they were used to switch off laser beams
Games should be play tested to ensure that there are no dramatic spikes or dips in perceived difficulty of challenges.	All games were play tested and features were fine-tuned based on the feedback obtained by peers.
A sharp unanticipated rise in game difficulty will discourage players and so the game difficulty should be increased at a slow rate without dramatic spikes.	All the three games are designed to have an ascending level of difficulties without major spikes in difficulty changes.

Table 8 Designing a balanced game

Adams and Rollings conclude that it is the perceived difficulty, that is the difficulty the player actually experiences, that a game designer should actually be concerned with. They provide a series of guidelines that can be used by game designers to ensure a balanced game in terms of difficulty. As can be seen in Table 8 (page 162) all the guidelines were tackled by the students further demonstrating the effect knowledge about games had on their ability to design balanced games.

5.3.1.1.2. Multimodal semiotics

A captivating feature of *The Ball and the Box* is the use of multimodal semiotics (Kress 2013). The game does not have an instructions page. However by looking at the characters one can immediately grasp what is required of the player. The box is to be moved to collect the balls. The balls fall, the player has no power on them other than collecting them to score points. The image used for the character that can lose you points is a bomb. Bombs have to be avoided at all costs as bombs explode. The images used for the characters in this game are all available from the Scratch library. However the backgrounds of the game were drawn by the participants. Even here the semantic of the image used is important. The linking of the background image to the difficulty of the level further stresses the importance attributed to meaning making by looking at the screen. Rain is reserved for the level in the game where one can lose points and eventually the game, whilst a blissful sunset is reserved for the level where the game can be won and the lives restored by collecting bonus points. The game building exercise has offered MariaChristinaM and AlaaE the possibility of practicing their multimodal knowledge. During the game making workshop these important features of the game were not explained, however it is evident that both participants regarded the meaning of images as important given the care they both took to design backgrounds that impart meaning.

5.3.1.2. Learning through games

The sector learning through games explores the ways in which games are used to teach explicitly curriculum topics through the use of commercial games or serious games (Beavis 2012). At face value this sector of the model is the one which least applies to the workshop. The children were not asked to create a game about a particular curricular topic as was done in the research by Kafai (1995) and Baytak, and Land (2011). Instead they were asked to create a game they wanted to. However there was a lot of learning about topics in the curriculum which took place too. There are a lot of curricular topics in subjects like mathematics that the children might see as inert, unapplied. Cartesian coordinates are a case in point. Children were already introduced to the idea in their main stream subject however whilst creating the game the topic came to life as they could see a practical application of the notion of coordinates.

Indeed KyleC used it a lot with another notion which he came across in mathematics but which he had to use extensively to build his game: comparison operators. He used “the greater than” and “the less than” operators to identify when the car was in a particular portion of the screen that merited the background to be switched to show another part of the track.

5.3.1.3. World around the game

This sector in the model focuses on literacy practices that surround the game and the world around it. The children building their games were working on their games and play-testing the work done by their colleagues. They were using the experience they gained whilst playing other games to design their own games. However this process was also affecting their gaming capital. The success they were having in designing their game also resulted in increased status within the group. *Bob’s Adventure* was one of the more popular games built in the workshop. Children used to queue to be able to play test it whenever Daniel9000 added a new feature or tweaked an existing one. The success workshop participants had whilst building their games in turn boosted their social status within the group and hence their gaming capital.

5.3.1.4. Me as game player

Whilst creating their games MariaChristinaM, AlaaE, Daniel9000 and KyleC went through iterative phases of playing and creating the games. The “Me as game player” in previous games not only effected their gaming capital and their knowledge about games but also provided them with what I’ll call *game making identity affordances*.

Gibson had coined the term affordances as a relationship between an environment and an animal. He had stated that “the affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill.” (Gibson 1986, p.127). The game authoring activity that the students experienced during the workshop offered affordances which allowed the participants to shift from a player role to a creator role.

The children gained membership in a community of game producers by sharing thoughts and experiences with fellow players/designers. Identity and games have been closely linked in research. Gee emphasizes that video games recruit identities and encourage identity works and reflections on identities in clear and powerful ways (Gee 2003, p.46) .Taylor (2006) looks at how virtual identities adopted during play and real identities merge in the game world and out of the game reality. The identities the workshop participants took on during the game making sessions allowed them to gain fluency in a specialist language linked with game development allowing them to explore a new form of writing.

The workshop offered the participants game making identity affordances by allowing them to wear the hat of a game designer, a system architect, a programmer and a quality assurance officer during the different phases of the game. The game making activity fitted itself into the *box of tools and signs* which make up the funds of identity (Esteban-Guitart, Moll 2014).

5.3.1.4.1. Game Designer Role

Whilst wearing the hat of the game designer the children had to decide on the plot, the characters and action to adopt for their game. AlaaE and MariaChristinaM decided on creating a ball and paddle game involving a box for a paddle and a bomb, a ball and bonus balls for objects to catch. Daniel9000 stuck to the plot he had been drawing on paper well before the start of the workshop. He decided to build his game and to invite into it characters from his favourite games too. KyleC built on his racing car game interests to build his own racing car game. The game designer role is not something the children took on at the start of the project only. They kept going back to it to tweek their game and improve it. The versions of all the three games show that the design of the game changed throughout the course of the workshop sessions. KyleC kept altering the shape of his racing track and adding obstacles. Daniel9000 kept adding levels well after the workshop came to an end. The fact that his game does not have an ending screen gives the impression it is still a work in progress.

During this phase of the game creation process the children also devised the rules which governed the action in the game. This involved deciding how the characters interacted, whether a character would be driven by the machine or by the game operator. As game designers the children also made decisions about the images, sounds and animations to use for this game. They had to plan the feedback that the game had to provide the player, so that the interaction between game and player provided an optimal gaming experience.

5.3.1.4.2. System Architect Role

Another role assimilated by the children creating the games was the systems architect role. Building on the decisions they took whilst designing the game, the children had to take decisions on the logical structure of the Scratch program. They had to decide the number of sprites to use and how the sprites would interact. Whilst wearing the game designer role the children decided what rules to adopt in this game and how to provide feedback to the player. Whilst adopting the systems architect role they had to decide how to implement the rules and feedback designed.

The way the three games were structured by their authors were significantly different. AlaaE, MariaChristinaM and Daniel9000 decided to use a number of sprites. *Bob* had to dodge a number of other characters in form of sprites which were architectured by Daniel9000 to make the game harder. *The Box* had to collect the *ball* and *bonus ball* sprites whilst dodging the *bomb* sprites. KyleC resorted to use only one sprite the car sprite. Even though *Racer* contained only one sprite, the car still had to work its way around the obstacles in the track. KyleC decided to design the architecture of his game so that the obstacles were implemented as blotches of colour rather than sprites.

Whenever the children revisited the structure of their game by taking on the game designer role, the children had to come back and wear the systems architect role hat again. Any change in the game design inherently resulted in changes in the way the game was architected.

5.3.1.4.3. Programmer Role

The programmer role adopted by the children built on the decisions taken at systems architect stage. After deciding on the number of sprites to use, the children adopting the identity of programmers had to create the sprites and code them to support the rules they had designed whilst wearing the hat of the systems architect. Whilst taking on the role of programmers the children had to identify which programming construct to use to achieve an architectural choice they made whilst designing the game. Even when taking on this role there was diversity in the way issues where handled. KyleC decided to use the *Scratch* block *touching colour* to check whether the racing car was off track or touching an obstacle. AlaaE, MariaChristinaM resorted to using the *touching sprite* block to check whether the *box* had picked a *ball* or hit a *bomb*. Daniel9000 used both approaches. He used the *touching sprite* block to check if *Bob* hit *Creeper, Pacman* or *DangerBoy* whilst he decided to opt for the *touching colour* block to check if the car was off track. The diversity of the approaches used by these children game designers is testament to the attitude of testing the boundaries of *Scratch* as they are accustomed to do as game players whilst playing digital games.

5.3.1.4.4. Quality Controller Role

At various stages in the game creation process the children had to adopt the role of quality controllers. They had to play test the game to ensure that the features designed whilst wearing the game designer hat were implemented correctly. This role was not reserved to their game only. It was custom to see children play test each other's games. This not only lead to problems being identified in games but was also served as a mechanism for cross fertilisation of ideas (see section 4.4.2 page 123)

5.3.2 Games as Action

5.3.2.1. Situation(s)

Situation refers to the context in which the digital game is played. This sector of the model focuses on the spaces where the digital games are enacted and the learning and sociality that takes place during the experience of gaming (Beavis 2012).

The situation where these games were being built was that of a workshop. The workshop was held in the school's computer room after school hours. The sessions started off with a short session where the students could ask questions about issues they encountered during the week. The teacher acted as a consultant facilitating the discussion. Following the discussion the children moved on their computers where they could continue working on their computer games. At this point the teacher role switched to a "meddler in the middle" role. The atmosphere in the workshop was game-like where an attitude of exploring and experimenting was encouraged. The students were taking risks and testing the boundaries by trying new things. Sometimes these risks produced great results as in the case of *Stickman duelling with the dragon* (see section 4.7.3 page 137) where BenL experimented with *Scratch*'s mode affordances (Kress 2013) to come up with an approach that allows the stickman to kill dragon by using his sword. This approach also allows the dragon to inflict damage on stickman by burning him with fire from the dragon's nostrils. In other cases these risks drove the student into an alley due to the lack of capability of the software as in the case of KyleC's inclusion of a shop to purchase racing cars in the game *Racing.* The playful approach of engaging with game making allowed KyleC to take this failure in his stride, modify the game structure and remove the reference to the shop in newer iterations of the game.

The *situation* where the workshop took place contributed to a great atmosphere where children play tested each other's games (see 4.4.2 page 123). A lot of learning was occurring in this situation. The children were not only expanding their knowledge of games whilst having fun but also being introduced to the exciting realm of programming (see 5.4 page 170)

5.3.2.2. Actions

Action in the game authored by the children is the most evident sector present in the game they authored from the game as text, game as action model.

The importance of action in digital games is amply explained by Galloway when he states: "If photographs are images and films are moving images then video games are actions" (Galloway 2006, p.2). The digital game comes into being through actions performed by the operator (game player) and by the computer. These two work together in synthesis to produce a computer game. Galloway maps game action on two orthogonal axis *Machine – Operator*, *Diegetic – Nondiegetic* to produce a game action analysis module.

There are two aspects of the games that need to be considered: the Operator–*Nondiegetic* action of creating the game itself and the actions whilst playing the game. The creation of the game itself by the children can be seen as an operator initiated activity that is *Nondiegetic* as it does not exist in the game world. This action is similar to the one taken by a player customising an avatar whilst preparing to play a game on a gaming console. The action is *nondiegetic* to the game but still important to the gameplay because it positions the player in the gameplay. The actions entrenched in the games created by the children are significantly important. The actions created demonstrate once more the affordability the game building experience offered to the children. It allowed them to put into practice information about the design of games they picked up during the years of game playing.

	Diegetic	
	Operator	**Machine**
The Ball and the Box	Moving the ball Collecting balls Dodging/Hitting bombs Collecting bonus balls	Keeping track of score/levels Adding/Reducing Lives Changing the background scenes Animating exploding bombs, falling balls and falling bonus balls
Bob's Adventure	Navigating *Bob* Deactivating laser beams Activating *Dangerboy* Dodging/Colliding with *Creeper, Pacman* and *Dangerboy*	Keeping track of levels Restarting the game when *Bob* traverses a black line Showing laser beams when active Displaying *Dangerboy* Animating *Creeper, Pacman* and *Dangerboy* Changing the background scenes
Racer	Navigating Car Avoiding obstacles	Displaying the relevant part of track depending on position of car Ending the game when the car goes off track / hits an obstacle Displaying the time taken Ending a game when the time spent exceeds 1250 time ticks

Table 9 Diegetic actions in The Ball and the Box, Bob's Adventure and Racer

The three games being analysed contain actions which fall in the four quadrants of the model. As can be expected the areas which contain most actions are the Operator-Diegetic and the Machine Diegetic actions (see Table 9). KyleC's attempt to add a shop to *Racer* would have resulted in expanding the Operator-Nondiegetic quadrant for the *Racer* game. This addition would have allowed operators to purchase cars to race with during the game. The lack of programming skill and the mode affordance of *Scratch* constrained KyleC into removing this feature; however it is noteworthy that the original plan was to include this action too. KyleC did add a menu at the beginning of the game as well as a help screen. This feature also allows the operator to interact with the game in a *Nondiegetic* manner.

5.3.2.3. Design

The final sector of Games as text, games as actions model mostly relevant to the games created, is the design sector. During this activity the workshop participants designed a game from the ground up and went through an iterative process of tweaking their game until they were satisfied of the outcome. The tweaking of the game had a social component to it as the participants influenced and were influenced by peers whilst playing the games of others and discussing how some of the features were implemented (see section 4.4.2 page 123).

The participants not only made aesthetic choices as is most commonly possible in off the shelf games which support modding but also designed and implemented the rules for their games. The creation of the game from the ground up introduced the participants to game authoring skills which move beyond game literacy and which I will discuss in section 5.4. below.

5.4 Computational thinking concepts

The importance of programming as a twenty first century literacy skill has been highlighted by Jenkins(2006). Rushkoff (2010) takes this a step further by stating that we have two choices to make, to program or be programmed. The argument Rushkoff makes throughout his book is that digital technology is biased towards those who make the technology. Rushkoff is not suggesting that everyone should become a programmer. He does however stress the importance of realizing that a computer is operating in a particular way because someone programmed it to act that way.

Although I agree with the importance of becoming aware of programming, I am convinced there is much more to it than just programming. Programming is the skill of piecing together code in a programming language. However what is important is, to use Rushkoff's words "the awareness that a computer operates the way it does because someone programmed it to act that way" and this is best picked up if we give the students the chance to build systems in an authentic and meaningful environment rather than constraining them to build simple computer programs. From my experience whilst observing students in Maltese schools the introduction of programming typically involves exercises such as listing the multiplication tables using a routine, or simply displaying a text message on a screen. Allowing the students to build systems in an authentic and meaningful environment allows the students to engage in open-ended problem solving experiences, and this is precisely what happened with the workshop participants. They engaged in something which was meaningful to them, game making, and in so doing they were immersed in an open-ended problem solving experience. They designed the game and came up with the problems to solve. Through this method they engaged in problem solving that mattered to them.

The workshop participants were not engaged in programming only. They were building a system and in so doing they engaged in what Wing (2006) calls computational thinking. Wing defines computational thinking as "solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science." (p. 33). Indeed the children were posing questions and providing answers. They were reflecting on the gaming capital accumulated throughout the years and using their funds of knowledge to create their digital games.

Brennan and Resnick (2012) outline a series of frameworks for studying and assessing the development of computational thinking when building artefacts based on *Scratch*. The group of three frameworks are based on assessing computational concepts, computational practices and computational perspectives. Computational concepts are the concepts the designers develop whilst they construct the artefact, in this case the game. Computational practices are the practices the designers pick up whilst they build the artefact. Finally the computational perspectives describes how the students understanding of themselves, their relationship to others and the technological world around them changes whilst they build the artefact.

In the following sections I analyse how *The Ball and the Box*, *Bob's Adventure* and *Racing* maps against the computational concepts framework. I also analyse how the practices observed by AlaaE, MariaChristina, Daniel9000 and KyleC during the game creation sessions map against the computational practices framework.

5.4.1 Computational concepts

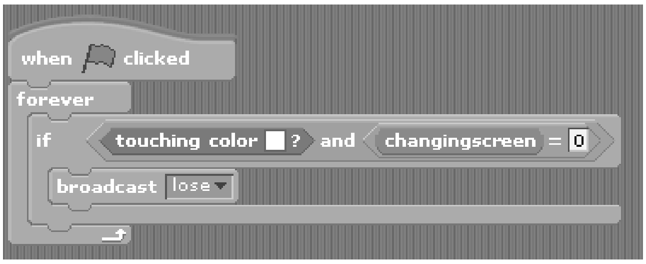
Whilst AlaaE, MariaChristina, Daniel9000 and KyleC built their game they engaged in computational concepts which are common in many programming languages. Game making provided a meaningful environment where the concepts they picked up by analysing the *Shark and Fish* game and other games were moulded together to produce their games. I provide examples of seven computational concepts listed by Brennan and Resnick in their computational concepts framework (Brennan, Resnick 2012).

5.4.1.1. Sequences

Programming in many ways is similar to cooking. When cooking, a recipe is followed step by step to achieve the end result. Similarly in programming a task is subdivided into steps which are executed one after the other. Sequencing was used a lot during the games developed by the children. In all games whenever the game was started a sequence of steps were executed to change the background to the first screen of the game and to initialise the score and levels.

5.4.1.2. Loops

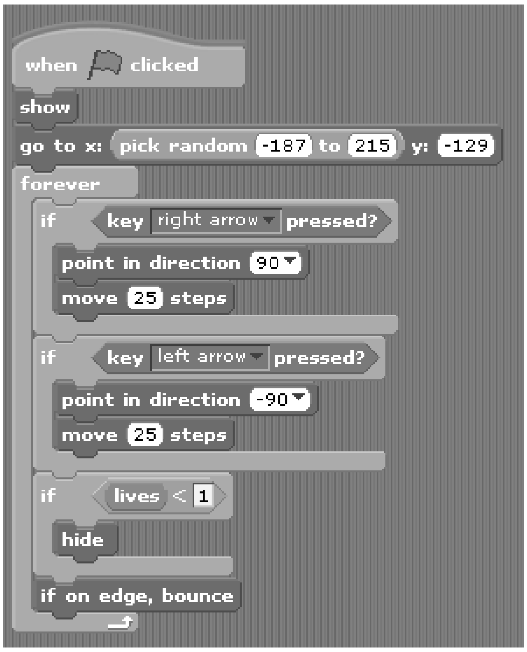
Another key computational thinking concept which the children picked up during the construction of their games is iteration, or repeating a set of steps until a condition is met. This concept has been used numerous times by the children. In the example in section 5.2.3 KyleC used a loop to check if the car hit a white colour throughout the game execution. KyleC drew the off-track space and obstacles in white. In this way if the car went off track or hit an obstacle the game came to an end.



```
when ⚑ clicked
forever
  if  touching color □ ?  and  changingscreen = 0
    broadcast lose ▾
```

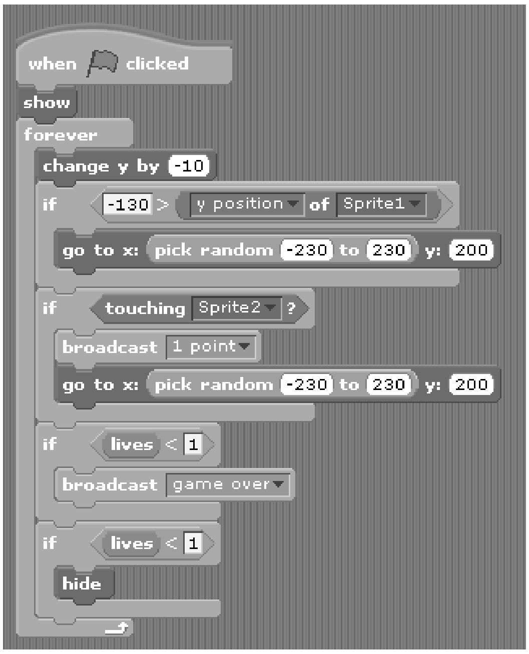
Figure 5.4-1 Loops

5.4.1.3. Parallelism



```
when ⚑ clicked
show
go to x: (pick random -187 to 215) y: -129
forever
    if <key right arrow pressed?>
        point in direction 90
        move 25 steps
    if <key left arrow pressed?>
        point in direction -90
        move 25 steps
    if <lives < 1>
        hide
    if on edge, bounce
```

Code block for the box character



```
when ⚑ clicked
show
forever
    change y by -10
    if <-130 > (y position of Sprite1)>
        go to x: (pick random -230 to 230) y: 200
    if <touching Sprite2 ?>
        broadcast 1 point
        go to x: (pick random -230 to 230) y: 200
    if <lives < 1>
        broadcast game over
    if <lives < 1>
        hide
```

Code block for the ball character

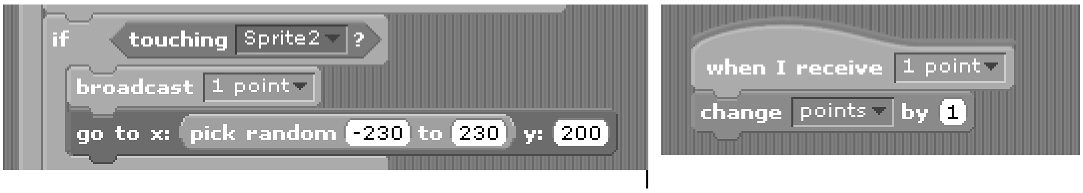
Figure 5.4-2 Parallelism

Another key concept when constructing software is parallelism or executing a number of code sequences in parallel. Figure 5.4-2 shows an example of parallelism used in *The Ball and the Box*. From the first level the code block for the box and the ball are executing at the same time to display the effect of the ball falling from the top of the screen whilst the box responds to the player pressing the left and right arrow keys to move on a horizontal axis.

Similar code was used in *Bob's Adventure* when Daniel9000 was using code to allow the player to animate Bob whilst at the same time animating *Creeper*.

5.4.1.4. Events

Similar to Lego blocks that are snapped together to build a structure, the children had to snap instructions together into code blocks. Each code block is designed to perform a specific function such as increasing the score or changing a level. The code blocks in the games are activated when a particular situation arises in the game. This triggering of code execution is known as event driven programming in the computing world and the children used it effectively throughout the game to change levels, to add points, decrease and increase available lives and to end the game once the number of lives are exhausted



```
if <touching Sprite2 ?>
  broadcast 1 point
  go to x: (pick random -230 to 230) y: 200
```

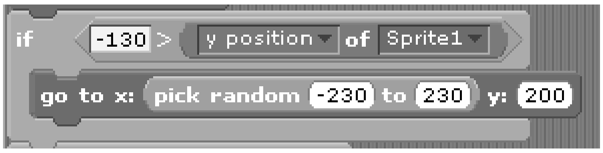
```
when I receive 1 point
change points by 1
```

Code block for the ball character triggers an event to increase the points by 1

Figure 5.4-3 Events

5.4.1.5. Conditionals

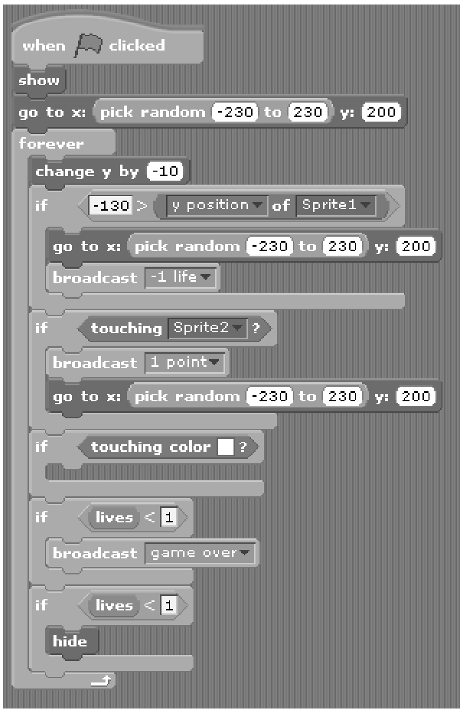
Rules are central to games. In order to implement the diegetic and non-diegetic actions in the games created, the participants used a number of conditional statements.



```
if <-130 > (y position of Sprite1)>
  go to x: (pick random -230 to 230) y: 200
```

Figure 5.4-4 Using logic conditions, randomisation and a conditional block to manage location of ball

AlaaE and MariaChristina used the conditionals to check the ball's position on the screen so that once the ball fell outside the screen it could be dropped from a new location from the top of the screen. Conditionals were placed in the code by AlaaE and MariaChristina to check if the number of lives reached zero and hence the game needed to end. Rules were also used for collision detection, that is to check if the ball hit the box, in which case a point is attributed or if the bomb hits the box, in which case a life is lost. Game authoring offered a lot of possibilities for the children to master this computational concept.



```
when ⚑ clicked
show
go to x: (pick random -230 to 230) y: 200
forever
  change y by -10
  if <-130 > (y position of Sprite1)>
    go to x: (pick random -230 to 230) y: 200
    broadcast -1 life
  if <touching Sprite2 ?>
    broadcast 1 point
    go to x: (pick random -230 to 230) y: 200
  if <touching color □ ?>
  if <lives < 1>
    broadcast game over
  if <lives < 1>
    hide
```

Figure 5.4-5 The game changes through the iterations

5.4.1.6. Operators

Whilst discussing aptitudes for learning a subject Papert (1994) points out that some people explain their lack of understanding of mathematics to not being "mathematically minded" or lacking mathematical intelligence. He then draws a parallel with students who have difficulty learning a foreign language such as French. In this case no one claims that the student lacks French intelligence, since in all probability the student would have picked up French in no time if the student had been born in France. The argument here is one of relevance. All the participants of the workshop had been introduced to Cartesian coordinates in maths since Cartesian coordinates are used to draw line graphs, however whilst programming their game in *Scratch* the participants found a concrete use for these coordinates since Cartesian coordinates are used to locate positions on the screen for different sprites. Similarly the participants were familiar with comparison operators such as >, < ,= however in the game making process they found a relevant place where to use these operators rather than inert maths exercises. *The Ball and the Box* is full of logic and mathematical operators which allowed the participants to check the location of the ball on the screen and increase the points whenever a ball is collected in the box. Another operator which the participants found useful in this game was the randomisation operator. Most of the students are familiar with randomisation since this is used extensively in games of chance and board games. Game building provided an avenue where the participants could practice their previous knowledge of numbers and logic.

5.4.1.7. Data

A fundamental concept in computational thinking is the concept of storing data. Scratch features two methods for storing and manipulating data: variables and lists. Variables can hold a single piece of data which can be a number or string whilst lists can store a collection of numbers or strings. Building a game provided motivation to the children to experiment with variables.

AlaaE and MariaChristinaM used a variable to store the number of lives available for the player and a second variable to store the points gained by the player. The content of the variables was altered throughout the game whenever a ball was collected by the box and when a life was lost by hitting a bomb sprite. Both variables were displayed on the screen as a means of providing feedback to the player.

KyleC used a number of variables in *Racer* however only one of the variables was used as a feedback mechanism with the other variables used to implement the rules and animations of the game. The variable *time* was used to calculate the time taken by the player to drive the car and to provide feedback on the top of the screen. KyleC used a variable called *screen* to store the current screen graphic being displayed. Whenever the car arrived at the end of the track being displayed the *screen* variable was incremented. The routine that displayed the track then checked the value of *screen* to display the appropriate graphic. As discussed in section 5.3.1.4.2 (page 165) KyleC detected if the car was driven off track or hit an obstacle by checking the colour that the car touched. This method of detection presented a cumbersome problem. Whenever *Scratch* changes from one graphic to another, the car ends up being placed temporarily on a white screen. The change of screen is too fast for the human eye to detect but the rule that checks which colour the car is touching was being triggered. This resulted in the player losing the game after completing just the first part of the track. KyleC realised there was a problem with the game and together with the help of one of the teachers he identified the cause of the problem. To solve this problem KyleC used another variable which he called *changingscreen*. *Changingscreen* was set to 1 just before the screen was being changed and back to 0 soon after. The routine which enforced the rule that checks which colour the car is touching was amended by KyleC to stop enforcing the rule whenever the value of *changingscreen* was 0. KyleC had managed to gain full benefit of the use of variables and had used them to solve a problem which cropped up in his game.

Daniel9000 too made use of variables in his game. Similarly to other games *Bob's Adventure* displayed the current level being played on the top portion of the screen. This information was held in a variable Daniel9000 called levels. Daniel9000 used another variable called *laser on/off* to control the workings of the laser beams in use in level 3.

Variables are an abstract topic for new developers to understand. However the familiar territory of game making allowed the children to pick the concept up and to use them to solve problems within the games they created.

5.4.2 Computational practices

5.4.2.1. Being incremental and iterative

One of the computational practices observed whilst the children were building their games was the iterative and incremental approach they adopted. Designing a project, especially a game, is rarely a clean sequential process (Brennan, Resnick 2012). It is an adaptive iterative process where a feature is added to a game, tested through play and the resultant game is then amended or further developed.

Playing a game is closely linked to reflection. Every action taken, is taken in light of the current state of play and what effects the action will have on the future of the game. Gamers playing a game do not simply use a trial and error approach but if an action does not lead to a desired goal they reflect on their action subconsciously to choose the next action to be taken. Salen (2007) links game play with reflection in action (Schön 1983) and describes gamers as researchers reflecting on their practice whilst playing the game. This reflection in action is done throughout the game play in an iterative fashion. The children as game designers in the workshop adopted the same strategy whilst designing their games. They were asked by the teachers to save their game into the portal adopted for the workshop after every lesson. This allowed me to view different snapshots, versions, of the game whilst it was being created. The progressive versions of the same game demonstrate an iterative process to building the game where by the children were creating part of the game, playing it (alone and with peers) and then modifying/adding on a feature.

The first version of *The Ball and the Box* was authored in December 2012 and consisted of the first level only. In this version of the game the player lost a life whenever the ball missed the box similarly to the traditional pong game (see Figure 5.4-5 page 174). This feature made the game more challenging than the final game produced by the end of the workshop. An easier approach, whereby no lives are lost when a ball is missed by the player, was adopted in subsequent game iterations. Bombs were introduced in the second level providing adequate challenge for the game.

Racing too changed considerably throughout the sessions. The first version contained a screen for an in-game shop which was then dropped in the final version. The race track did not contain any obstacles. Obstacles were only added in subsequent versions. The timer was also a feature which was added in the final version of the game. Initially Racing did not include the race against time element making it easier to win.

5.4.2.2. Reusing and remixing

Another computational practice which is equally important and was observed whilst the participants were building their games was the practice of reusing and remixing. Reuse is not simply the reuse of graphics, which was amply done by the workshop participants, but is also the reuse and adaptation of ideas. The ludology adopted in the games is not a novel one. There are plenty of games out there which adopt similar strategies. However what is different in these game creations is that the children took stock of their gaming capital, merged them with their existing funds of knowledge to create their game. The multimodal resources they used were a mixture of ready-made images, images which they drew and sounds which they recorded. The pieces of code were assimilated from other projects discussed in class and learnt through experimenting with other block constructs found in Scratch. The game artefact was constructed out of a collage of gaming ideas and multimodal artefacts glued together with coding logic. All this contributed to the children making their own little c creative artefacts (see section 2.3.2.1 page 55).

5.4.2.3. Testing and debugging

All the games discussed in section 5.2 were mostly developed in a workshop setting. The amount of time spent making games at home was limited. The workshop participants went through periods of development which were quickly followed by periods of testing. Initial testing was performed by the children themselves who were developing the games, however testing was observed to be a social practice too. Whenever the children added a new feature they found plenty of volunteers from their peers to test the game. The participants quickly got into the routine of ironing out problems in their code by reading code and understanding logic. Taking a hint from game making cheating was occasionally used as well (see section 4.4.2.1 page 124) to speed up the process of identifying bugs (problems) and testing a fix. The teacher was seen as a consultant ready to give advice whenever the bug encountered proved to be a tough nut to crack or to help come up with ideas which could help identify the source of the bug. Through the game making activity the participants managed to assimilate the testing and debugging computational practice.

5.4.2.4. Abstracting and modularizing

The children approached the computational practice of abstracting and modularising from different angles. AlaaE, MariaChristinaM and Daniel9000 decided to split their game into a number of sprites. Each sprite was coded as a module on its own with sprites communicating together through events. The abstraction employed by the children allowed them to divide the main task into smaller tasks an important practice for design and problem solving techniques. Abstracting and modularizing also made the task of debugging and reading code easier for the participants since the code chunk was more manageable in terms of length.

On the other hand KyleC decided to use only one sprite the main character of *Racer* – the car. All the obstacles were implemented as part of the background and drawn in one colour. This method too had its advantages. KyleC could add new obstacles by simply drawing them on the background screen without altering the code, as long as the obstacles were drawn in white. Notwithstanding the fact that the game consisted of only one sprite the code within the sprite was split into three modules. The first module dealt with placing the car in its right place and displaying the first screen when the game was started. The second module checked if the car hit an obstacle whilst the third module dealt with the process of changing the track whenever the car arrived to the end of the currently displayed track portion.

5.5 Being Agile

Observing the children going through the cycles of game making, game testing with peers and then going back to add new features in their games or fix problems that were unearthed during test play made me reflect on the similar work practices adopted in the software development industry. There is a growing movement in the software development industry that promotes the use of an agile development philosophy. The Agile group of software development methods knows its origin in 2001 when a group of software developers called the Agile Alliance came up with a philosophy for developing software called the Agile software development manifesto (Fowler, Highsmith 2001). This manifesto offered an alternative to the documentation driven, software development processes present up to then and brought about unprecedented changes to the software engineering field (Dingsøyr, Nerur et al. 2012). One of the main principles behind this manifesto was that software progresses in response to user feedback, rather than as a reaction to a fixed plan (Hunt 2006). This does not mean that there is no fixed plan, but that the plan is altered through frequent releases of software which are discussed with the end users. The feedback is then fed into the next cycle of software development.

This process is similar to how the games took shape in class. Daniel9000 had a clear idea for his game, well before the start of the workshop. The main structure of the game remained the same through the game making process however the game was shaped following the discussions with peers whilst they tested his game. New levels and characters were added whilst difficulty of each level was fine tuned. The same observation can be made about *Racer*. When I reviewed the different versions of *Racer* uploaded to the website throughout the six weeks that KyleC was building it I could observe the changes that were made in the game. All the changes were a reflection of the peer interaction with the audience of his game, his peers. The social aspect indeed was a very important aspect that helped the developers shape their games.

5.5.1 Working together to build the game

The Ball and the Box game was created by a pair of students, AlaaE and MariaChristinaM rather than by a student working on his or her own. It is interesting to note that out of the five girls participating in this study four decided to work in groups of two whilst creating their game. On the other hand all the boys worked individually. One might think that the girls found the game creation process harder to master and hence decided to team up to help each other whilst creating the game. However after analysing the games I can see that the game features implemented are comparable to the features in other games created in the workshop.

Working in pairs whilst developing software is becoming an increasingly common practice in the software industry especially in teams following the eXtreme Programming (XP) methodology, part of the Agile software development philosophy whilst practicing pair programming, one developer takes the role of a driver, the person who writes the code, whilst the other developer plays the role of the navigator watching for problems, thinking of alternatives and asking questions (Shore 2008). Throughout the lifetime of the team the roles taken on by the developers change: a driver moves on to become a navigator whilst navigators take over the driving seat. I could observe this happening during the workshop. Most of the times MariaChristinaM took over the driver seat coding the game with AlaaE on the side suggesting improvements and testing the game. During other sessions AlaaE was the one writing the code with the navigator role taken over by MariaChristinaM. At times the coding paused with both participants discussing the features to be added to the game whilst discussing how to add new features with the teacher facilitating the workshop.

5.6 Conclusion

In this chapter I analysed a selection of three games authored by two individual participants and a team of two participants. The games were analysed from a game literacy perspective. The computational concepts and practices that the participants adopted during the game construction exercise were explored. In the final section I drew a parallel between the game development practices adopted by the children whilst creating the game and the Agile software development practices adopted in industry.

In the next chapter I reflect on how the knowledge I gained whilst working with the children participating in the game making workshop helped me provide an answer to the guiding research questions that were set at the start of this research project.

6. Findings and Reflections

6.1 Introduction

When I embarked on this research journey I outlined two questions which I wanted to answer through my research (see section 1.2 page 11). The first question was related to the processes eleven year olds follow whilst building a digital game whilst the second question was about the benefits children gained out of participating in this research from a gaming literacy perspective.

Whilst analysing the data for this research I tried to answer the guiding research questions whilst at the same time keeping my eyes open on the lookout for other topics which might not be directly related to the guiding research questions. In this way I realized that the learning benefits were not from a gaming literacy perspective only. The children learned a lot from a computation thinking perspective too. The ludic attitude was pervasive throughout the workshop with the children working together whilst pushing each other to accomplish their creativity in the game created. Game making also featured as a potential “funds of identity” (Esteban-Guitart, Moll 2014).

I this chapter I also reflect on the structure of the workshop and how this played an important part in fostering creativity in the children. Finally I look at the way the children were introduced to game making by programming Scratch and reflect on the pedagogical implications this might have for future projects which introduce programming to young children.

6.2 The processes eleven year olds follow whilst building a game.

One of the main objectives of this research was to trace the process the children go through whilst building their games. The children were observed going through six phases whilst building their game. Figure 6.2-1 outlines the flow between the different phases.

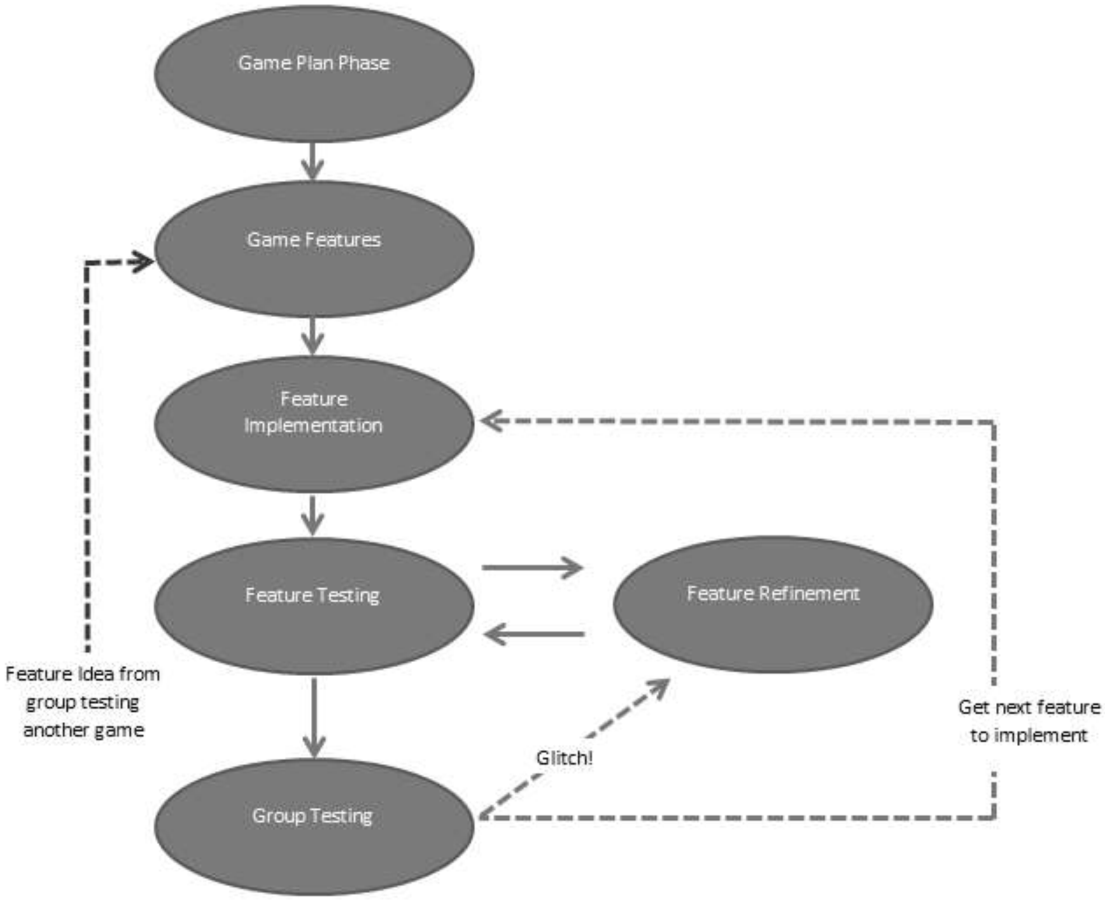


Figure 6.2-1 Stages in game development

The first phase of game making, the *Game Plan Phase*, consisted of the creation of a general plan for the game. In this phase the students created the general plot of the game whilst identifying the characters and the basic rules of the game. As demonstrated in *Racer,* the game created by KyleC (see 5.2.3 page156), this plan is not cast in stone. KyleC initially planned to create a shop where the player could exchange coins collected in-game with upgrades but then decided to drop the idea of a shop at a later stage in the game development.

The game plan phase is followed by the *Game Features* phase. In this phase the children split the main game into multiple building blocks consisting of the characters, background screens and feedback mechanisms making up a game. For example in *The Ball and the Box* (see section 5.2.1 page 144) MariaChristinaM and AlaaE split the game plan into the ball, bomb, bonus ball and box characters, the feedback mechanisms such as the lives, points and levels and the background screen for each level.

A feature is selected and the child starts working on it. Whilst working on the feature the child goes though successive testing sessions where problems with the code are identified and solved. When the child is happy with the feature, peers are asked to play test the game in the *Group Testing* phase. The Group Testing phase might be delayed until the game starts taking shape. During the initial stages of *Bob's Adventures* BenL created the first background scene, selected the image for the game character Bob and added coding to Bob so that the movement of the character depended on the keys pressed by the player. It was only after the first screen was ready that the peers were invited to play test the game.

Game play testing resulted in one of two outcomes. If a *Glitch* (see 4.4.2 page 123) was identified by the game testers the game developer focused on solving the glitch by refining the feature being implemented. If on the other hand the feature is accepted by the gaming community then the game maker moves on to implement a new feature. The group testing phase was beneficial to the testers as well as the game developers. On more than one occasion testers got ideas for game features from the games they tested for their peers (see 4.4.2). Hence the process of making a game whilst testing other games created an atmosphere that enabled the cross fertilisation of ideas. Similarly to how the space in which a game is enacted effects the learning and socialising that takes place, the atmosphere in the game making workshop affected the students in a way they acquired ideas from each other.

6.2.1 Similarities and differences with other game development models

There are a number of similarities between the model outlined in section 6.2 (page 183) and similar models found in literature (Resnick 2008, Robertson 2011). These models were discussed in section 2.3.5 (page 63).

Both models started off with a stage where the project is defined. This stage, the *Imagine* stage in Resnick's model and *Problem Finding* stage in Robertson's model is similar to the *Game Plan* stage in the model observed in this research. Both models make no reference to subdividing the project identified in this stage into smaller components and instead proceed to the Problem solving stage in Robertson's model and the Create stage in Resnick's model.

The subdivision of the game into features to implement is an important stage in the model identified in this research. It is at this stage that the children go through the problem decomposition stage, an important skill from the computational thinking perspective. This phase also demonstrated that the children already viewed the game as a system made up of a web of interrelated subsystems. Rather than developing the game at one go the children were observed worked on one feature at a time.

The play / share stage in Resnick's model and the internal validation / external validation in Robertson's model are similar to the Feature Testing and Group Testing stages in this model. The Feature refinement stage in this research's model was inserted to distinguish between the work done by the game creator when coding a new feature versus the work done when fixing a problem with the feature. The problem can be identified by the game creator during the feature testing stage or identified by another player in the group testing phase.

A key difference between the model identified in this research and the other models in research is the effect group testing has on the tester. Group testing of each other's games can lead to the identification of features to add to one's own games. The social aspect of the game creation was an important factor throughout the game making workshop. As discussed in section 4.3.2 the social interaction was not confined to the game making workshop. Children tended to demonstrate the games they were building in the workshop to their family. They played the games with their brothers and sisters and came back with suggestions of features to add to the game.

6.3 The benefits creating digital games has from a gaming literacy perspective

Zimmerman (2007) defined Game Literacy as an approach to literacy based on game design. The three concepts of systems, play and design on which Zimmerman had argued game literacy is based were all present throughout the gaming workshop.

6.3.1 Systems

Whilst building the games the children were experiencing first-hand the design of a game system. The game was decomposed into a series of objects. Daniel9000 divided

Bob's Adventure into seven objects. These objects included Bob the main character operated by the player and the computer operated characters Creeper, Pacman and Dangerboy. Daniel9000 added lasers and the finishing line as different objects too. Although these two objects were not characters, they were important components of the game system Daniel9000 built. All the characters and game components were placed on another object, the game stage. Each of these objects had attributes which defined them. The background attribute of the game stage was changed whenever a level changed. The laser beams had attributes which defined whether they were on or off. When the laser beams were on *Bob* could not traverse them, however the player could switch the laser beams off by navigating Bob onto the appropriate colour on the screen background.

The children showed mastery in implementing the game system by defining the interactions between the various objects in order to define the rules of the game. When Bob was traversing an area guarded by a laser, the laser's property was checked to see if the laser was switched on or off. The building of the game further reinforced the notion of a system being made of smaller systems which share complex and constantly changing relationships. Through the process of game making the children gained the skill of applying systematic thinking. They moved from thinking about systems to experiencing systems by creating them. They were able to subdivide systems into subsystems and to then map the relationships between the subsystems.

6.3.2 Play

That playing digital games is an important activity in the lives of the participants was not a surprise. Research has been telling us this for the past years. However I never expected to experience the enthusiasm from the children to participate in such a workshop. Through this workshop I came to realise that digital games are so important for these children that some of them spent time creating video game walk-throughs for others to experience and drawing game plans on paper with the hope that the games would one day become actualised (see section 4.3.3 page 121). Previous gaming experience had a dual effect on the activity of making a game. The prior gaming experience contributed to the game design as I discuss in the section about design (see section 6.3.3 page 190) but the prior gaming experience, the ability to see the world's structures as opportunity for playful engagement, also contributed to making the game building exercise as hard fun - hard work that's satisfying (McGonigal 2012). Playful engagement permeated this project and this could be seen in the competitive spirit during the initial part of the workshop, the playing with rules whilst developing the games and the playful engagement whilst testing the games.

6.3.2.1. Competition in the initial part of the workshop

Competition is one of the central characteristics to playing digital games (Whitton 2010). Feedback systems in games such points and levels are all aimed at identifying winners from losers. These feedback systems increase competition between players who in turn play to win. Competition can be used to optimise individual contributions by pitting one's talents against another (Reeves, Read 2009). Achievement systems such as badges, trophies and accolades are also used to increase competition between players (Montola, Nummenmaa et al. 2009). During the gaming workshop a *GameMaster of the Week Award* was used as an achievement system (see section 4.4.1). This award was designed to encourage the children to use the online resources during the first phase of the workshop. The competitive aspect of game playing translated itself quite well to objective with the website receiving most of the visits during the first phase of the workshop (see section 4.2 page 117). There was also enthusiasm and a healthy competition between the students. This aspect of the website was discontinued in the second phase of the workshop. In the second phase of the workshop the children were asked to create their own game and in the spirit of creative teaching I could not reward one game over another. The hits on the workshop website during the second phase of the workshop declined. The discontinuation of the *GameMaster of the Week Award* could have been a contributing factor to this decline. Even though one could argue that the game master of the week award should have been kept throughout the workshop in order to maintain interest in the online space component of the workshop, I still think that I took the right decision to stop the award. Maintaining the award would have required teachers to make judgment calls on the creative expression of the children.

6.3.2.2. Group Testing

Although the *GameMaster of the Week Award* was discontinued in the second part of the workshop the competitive element was still present in the workshop. Every student wanted to make a game which was enjoyable to play by his peers. As described in section 6.2 (page 183) whilst discussing the processes eleven year olds follow whilst building a game the addition of every new feature was generally followed by a session of group testing where the students tested each other's game. The group testing phase involved competition and cooperation. Every person presenting his game to his peers wanted his game to be the best. And this led students to compete with each other on making their games the most enjoyable to play by their peers. However the session also involved cooperation. It was quite usual for the students who tested the games to include a feature from the game being tested into their own game. The cooperation involved the student who added the original feature who would usually explain to his peers how the feature was programmed.

The playful engagement was also extended to testing games. The children invented their own term for bugs – *Glitch!* (see section 4.4.2 page 123) which they used to shout whenever the tester identified a problem with the game being tested. Here again the competitive element in games pervaded the game making activity with children doing their best to identify bugs in the games developed by their peers. In a way this competition was also cooperation because by identifying a bug they were making their peer's game a better game.

6.3.2.3. Playing with the rules

As discussed in section 2.2.6.2 (page 40) play can take two connotations in gaming. Play can be play within the rules which is achieved when players play a game whilst abiding by the rules of the game. However play can also be playing with the rules, that is, when players bend the rules to win a game. It was evident that students following the gaming workshop were accustomed to bending the rules when playing games and used similar approaches to play test their games.

Debugging or removing bugs whilst developing games can be a lengthy process since it involves playing the game to arrive up to the same point where the error occurs. Rather than breaking the rules to win a game by using one of the methods listed on page 41, the students were creative and banked on their knowledge of games to devise methods to bend the rules to make the debugging process shorter. As discussed in section 4.4.2.1 (page 124) students were observed to change the size of characters to make it easier for them to avoid hitting obstacles and to enable them to use the mouse to move characters to other areas of the screen and in so doing skipping obstacles.

6.3.3 Design

The concept which was the most evident during the game making workshop was the concept of design. The students showed that they were well versed in the possibilities games offer through their previous gaming experience. Through the game making workshop they engaged in designing games which were complete. The games produced show that the students were well aware of the multimodal possibilities at play in games.

6.3.3.1. The completeness of the games

Not only were all the games created by the children complete, they all had a goal and had rules which governed them, as well as adequate feedback systems. The rules varied in complexity from game to game. In *The Ball and the Box* the same rules were in force through the game. The play had to collect balls throughout the game whilst dodging bombs in the second level. In *Bob's Adventure* the rules enforced were more complex and varied according to the level in the game. In level three the player could deactivate laser beams by guiding Bob onto coloured patches which deactivate the laser beams. However the same coloured blocks are used to activate *Dangerboy* a character an evil character which could kill Bob and restart the game.

The feedback mechanism employed by all the game creators were evidence to the importance of the *Knowledge about games* (see 2.2.8.2 page 48) that the children acquired whilst playing games. The children implemented the feedback mechanisms which they were familiar with in the games they played. Similar to the rules that governed the games the feedback adopted also varied from game to game. The feedback adopted in *Bob's Adventure* was minimalistic with only a level counter shown on screen, whilst *The Ball and the Box* employed lives and points in addition to levels. The feedback adopted was also appropriate for the game developed. Levels was an appropriate mechanism to use for the games *The Ball and the Box* and *Bob's Adventure* but given the nature of *Racer* a race against time was more appropriate. For this reason KyleC the game developer of *Racer* decided to display a counter of the time taken by the player whilst navigating the car through the race track dodging obstacles. The children also found ways of circumventing the limitations of the software used to author the games. Since it is not possible to keep a list of top players in a game in Scratch, Daniel9000 resorted to write the names of the best players on the screens instead. This approach was also adopted by other students.

6.3.3.2. Multimodality

One of the main things that struck me out of this research project was the multimodal awareness that the student game players in my workshop had built throughout their years of game playing. As early as the first session when the students had barely heard of Scratch, the students set out to change the narrative of the *Shark and Fish* game to a different game by changing the graphics of the game. As Kress aptly states the graphics are not just decorative. They are used to convey meaning as much as writing conveys meaning (Kress 2003). The students were well aware that games can share the same game mechanics but be different by having different graphics and be set in a different environment. The combination of these semiotic objects is not just the sum of these objects together but a new creation, a new text. So they set out to exploit their funds of knowledge (Gonzalez, Moll et al. 2005) the knowledge they had acquired on how to use different software packages such as image editors, drawing tools, search engines and their knowledge on games to turn the game they were given into their own game. The game they remixed in the first phase of the workshop and the games they created from scratch in the second phase of the workshop shared a series of modes of representation (Jewitt, Kress 2003) which I expand upon below:

6.3.3.2.1. Size ratio of characters

The size of the characters was very aptly used by the students throughout the games they designed. The dragon in *Stick with a sword* is larger when compared to the player operated *Stickman*. The size ratio adopted by BenL denotes the size ratio usually adopted in films and games where the dragon is depicted as an evil creature with larger dimensions than the human. The same can be seen in other games. The size ratio of the car in *Racer* was realistic when compared to the width of the racing track or the size of obstacles on the track. In other occasions the size ratio of the characters did not follow that found in other games. A case in point is the game *Bob's Adventure*. Bob's Adventure used a number of characters present in other games. The machine operated character *Creeper* was copied from the game *Minecraft*. In *Bob's adventure* Daniel9000 increased the size of *Creeper* to increase the difficulty of the level. A larger character increases the likelihood of impact between *Creeper* and the player operated

Bob. In this instance Daniel9000 used the size property of the character not only as a visual prompt but also as a means of managing difficulty.

6.3.3.2.2. Alignment of game world with the theme of the game

The background scenes depicting the game world in which the game evolves were very well managed by the students creating the game. The scenes were purposely chosen from ready-made graphics the children had found on the internet or which were provided in the Scratch gallery. In some of the games the background scenes were drawn from scratch or edited using the inbuilt graphic editor or an external graphic editor such as MSPaint. The choice of the background scenes again demonstrate the multimodal intelligence that the children had built throughout the years. The background scene in *Racer* was that of a racing track whilst that of Bob's *Adventure* was of a maze that Bob had to traverse. In these games the scene was not just a backdrop but an integral component of the game since the games had rules inbuilt that worked on the background. If the car hits the track border drawn on the background scene in *Racer* the game is lost and the player has to start all over again.

In other games the background was more of a backdrop. However even in these games the choice of the background images were very well thought out. For example in *Click the Zebra* the background chosen by MichelaA was an image of a forest downloaded from the internet whilst JacquesC's *Soccer Cup* shows the goal posts of a football ground.

In The Ball and the Box the background scenes were tied to the perceived difficulty of the level. A blue sky reserved for the first level, grey clouds and rain for the level where the player has to dodge bombs and a scene with a sunset and birds flying in the sky reserved for the level where the player can collect bonus points.

Through the choice of images used in the game the children showed that they were well versed in social semiotics (Jewitt, Kress 2003).

6.3.3.2.3. The affectivity of music

Another mode of meaning making used in the games created by the children was the use of sound effects. As a researcher I never expected this mode to feature in the games created by the children. Although basic sound effects were included in the *Shark and Fish* game used to introduce the children to game authoring the sound feature was not devoted much time in the discussion sessions of the workshop. Notwithstanding this, a number of the games included sound effects and one of the game - *Bob's Adventure* also included sound effects recorded by the game maker Daniel9000.

6.3.3.3. Gaming instructions

The students showed awareness of the importance of guiding the player to play the game by providing instructions to play their games. The details of the game instructions varied from game to game. Most of the games included a start screen as shown in Figure 4.4-2 on page 124. In these games the detail offered in the screen varied from informing the player about the keystrokes to use to stating the goal of the game and wishing the player good luck. Other games included instructions on the main screen. The Ball and the Box did not include any instructions; however the students making the game relied on the social semiotics offered by the symbols for the player to grasp what the rules of the game are (see section 5.3.1.1.2 on page 163)

6.3.3.4. Feedback types

Another prominent design feature adopted by the game making students was the variety of feedback types adopted in the games they created. The feedback types ranged from screens to inform the user of the outcome of the game to score, level and life counters which provided players with on-going visual feedback during the game. Not all games contained levels and score. The game *Racer* was designed as a race against time again demonstrating the flexibility with which the students approached the subject of feedback in games.

In the game The Ball and the Box feedback was also provided through the choice of background chosen for the level. The background changed according to the difficulty of the level providing another means of feedback to the player.

6.3.3.5. Managing difficulty

Conflict and challenges are central to digital games (Crawford 1984, Juul 2003, Salen, Zimmerman 2003, Whitton 2010). Challenges and conflict should be sufficiently difficult since mishandling difficulty can break a game. Players give up on games which are too easy as there is no satisfaction in playing a game that has no challenge. On the other hand it feels bad to always lose, so games which are too difficult for the player's skill tend to be discarded as well (Habgood, Overmars 2006). A game maker must keep in mind that the perceived difficulty of a game depends on the skill of the player. To ensure that a game is balanced in terms of difficulty game developers tend to follow guidelines (Adams, Rollings 2007).

Even though the topic of difficulty in games was not discussed in class, the games designed and created by the students were all balanced games. As outlined in Table 8 (page 162) the students' prior experience with games ensured that the guidelines set by Adams and Rollings were followed to the letter.

6.3.4 Conclusion

The question I set out to answer was whether there are benefits, from a gaming literacy perspective, to introduce students to game making. As shown through the participation in this project children tended to benefit from all the three aspects of gaming literacy as defined by Zimmerman. They experienced first-hand how to design a game system. They designed games which were subdivided into subsystems which were interrelated and worked together to enact a game with its rules and goals. They turned their "systematic understandings" (Squire 2011, p. 36) into practice by authoring their own system and subsystems. The experience gained through participation in this gaming workshop also provided them with an avenue where they could express their gaming experience in a different practise from playing a game. They showed that they could transfer the skills to play with rules to think outside the box and speed up their debugging skills. Their playful engagements with the game making process allowed them to compete with each other whilst at the same time helping each other out. Finally the area which gained mostly from participation in the game making was the design aspect. This mirrors research by O'Mara and Richards (2012) who had concluded that the design aspect of the Games as Action dimension was the most prevailing aspect of the model that featured when children created their own games using the software GameMaker (see section 2.2.8.3 page 48). The students' prior gaming experience allowed them to design games which were complete, had appropriate feedback systems and were well balanced from a difficulty perspective. However the aspect which is most outstanding in the design of the games is the fluency with which the children engaged with multimodal aspects of game designs. Their prior gaming experience, their experience in using other software packages acted as funds of knowledge which the students then used skilfully to express themselves and design games which included game worlds aligned with the theme of the game, animations of characters as well as sound effects.

6.4 Potential funds of Identity

Fourteen out of the school population of thirty one students for this age group, nearly one out of every two, volunteered to stay on after class every Friday to attend the game making workshop. All these students actively wanted to join this workshop with some of the students going to great lengths to make sure they were selected to attend (see section 4.3.1 page 119). A number of students talked about their desire to learn how to make games and to join the game making profession in the future. This desire was present well before I met the students at the school to advertise the workshop. A student had been drawing game blueprints on paper since he was not aware how to make games whilst another was creating videos of his play time and posting them online (see section 4.3.3 page 121).

Esteban-Guitart and Moll (2014) coined the term funds of identity whilst referring to the previous research on funds of knowledge. They define funds of identity as follows "funds of knowledge become funds of identity when people actively internalize family and community resources to make meaning and to describe themselves" (p.35). Before attending the workshop the children had spent considerable time playing digital games. They were well versed in using digital technology to look up information on the internet, to create video and post it online, create and amend images. Attending the workshop fulfilled this desire to learn how to create digital games and they could describe themselves as game makers having designed and created their first game. They managed to turn their funds of knowledge into funds of identity.

Attending the workshop provided the children with more than just an opportunity to create a game, it provided the children with *game making identity affordances* (see section 5.3.1.4 page 164). They took on the roles of a game designer whilst designing the game, a systems architect role whilst designing the systematic structure of the game, the programmer role when coding the different sprites and a quality controller when testing the games and raising defects found in the games tested. These roles were taken on during the various stages of the game development process (see section 6.2 page 183).

During the *game plan phase* when the children were creating the general plan for the game the children were taking on the role of a *game designer*. *The systems architect role* was assumed during the games features phase since in this phase the children were taking decisions on how best to split the game into multiple building blocks consisting of characters, background scenes and feedback mechanisms.

The *programmer role* was assumed during the *feature implementation* and *feature refinement* phases. In the *feature implementation phase* the student implements one of the features decided upon at the *game features phase*. It is during this phase of the game development that the child writes code to implement this feature of the game. Following the *feature testing* or *group testing phase* a defect might be unearthed in the feature implemented and the student would need to refine the feature implemented. During the *feature refinement stage* the student assumes the programmer role to identify the error in the code causing the defect and to write a code patch to solve the problem.

During the *feature testing* and the g*roup testing phase* the student would be playing the game to try to identify defects. The only difference in these two stages is that in the *feature testing phase* the student is testing his or her own game whilst in the *group testing phase* the student would be testing the game developed by another participant of the gaming workshop.

Although in the workshop these roles were taken on by the same student in the different phases of the game development process, these roles would be typically assigned to different people in the industry. Hence during the workshop the students were given a taste of what each role in the real world would do.

When designing the workshop I did not plan to follow any particular software development model. Instead I planned to follow the suggestions in literature to foster creative thinking (see section 2.3.3 page 58) by giving more voice to the students whilst gradually introducing the students to making digital games (see section 2.5 page 82). Notwithstanding the children tended to follow a software development model similar to one used in industry – the Agile software development methodology (see section 5.5 page 179). This model, based on frequent software releases to the clients, resembled the software development adopted by the children.

6.5 Pedagogical Implications

One of the central themes of the national curriculum framework is to foster creativity and innovations in the Maltese schools. Throughout this project I strived to propose game making as a possible avenue where the children could expose their creativity. For this reason I tried to adopt strategies found in literature to foster creativity in students. In this section I reflect on how this pedagogic strategy bore fruit.

Another objective of the pedagogy applied in this workshop was to introduce programing, traditionally seen as a difficult task, to young children. In this section I reflect on the affectivity of the approach used and on how this approach differed from others found in literature.

6.5.1 Teaching for creativity

The pedagogy adopted in this workshop was modelled on the literature around teaching for creativity (see section 2.3.3 page 58) I strove to spend as little time as possible giving instructions and instead spent more time near the children discussing the challenges they encountered. I also strove to pass control back to the students and encouraged them to be innovative in their games designs. The children were encouraged to pose and identify questions during the discussion period at the beginning of the workshop sessions and to discuss possible solutions to the issues identified. Cooperation between the student game makers was encouraged at every stage of the game making process but was mostly evident in the game testing stage of the development process.

All these standpoints during the teaching process brought about the expected result from a creative perspective. The self-determination from the part of the children was evident in the *Shark and Fish* mods the children created with no two games being the same (see section 4.6 page 128). The mods created varied from an underwater wreck to a space environment where the fish donned space suits. The children were engaging in question posing to change the behaviour of the characters in the game. Whilst building a refuge for the fish (see section 4.7.1 page 133) the children were not only asking the questions but also brain storming possible ways of solving the issues raised in the question posed. The students came up with three different methods to construct lines which would act as a refuge for the fish. The advantages and disadvantages of adopting each method were discussed until one of the methods was chosen. The students did not just rely on programming constructs present in the original game presented to the students at the start of the workshop. When suggesting using the *if touching colour* block BenL was demonstrating his will to experiment and take risks by trying new things as this block was not explored previously during the workshop. The choice of the block to use showed that the experimentation had taken place before the discussion as the student was already confident of the outcome of the block during the discussion.

Being imaginative led the students to ask questions and to be critical of the way game features were implemented in *The Shark and Fish* game. When KyleC decided to add fish to the game each worth a different amount of points he was being imaginative and innovative (see section 4.7.2.1 page 136). This led him to be critical of code design of the game and to ask questions on why the code was structured the way it was in the initial game. Here again the children did not stop at posing questions but also proposed solutions. The engagement with the game creation process provided the students with fertile grounds where to ask questions and provide answers, where to be imaginative and show self-determination.

6.5.1.1. Creativity in the games produced.

In section 2.3.2 (page 54) of the literature review I discussed various outlooks towards the term creativity and settled on a definition of creativity to be used in this research. The definition adopted for this research was the one proposed by NACCCE where creativity is seen as: "Imaginative activity fashioned so as to produce outcomes that are both original and of value" (NACCCE 1999, p.30). When considering this definition of creativity, the children were all being creative whilst going through the process of creating games.

6.5.1.1.1. Imaginative activity

An imaginative activity is seen as the process of providing an alternative to the expected (NACCCE 1999). In other words it is the ability to think outside the box. The games produced all contained imaginative activities inspired by previous game play. The children immediately showed their ability at imagining alternatives when creating the mods from the initial game *The Shark and the Fish*. It was inspiring to see the children changing the narrative of the games without changing the game rules as early as the first session of the workshop. However the ability to think outside the box became more evident in latter sessions of the game workshop. I will illustrate using two examples.

In the game created by BenL *Stick with a Sword* (see section 4.7.3 page 137), BenL was not only creative in the story used for the game but also in the way he implemented the game mechanics. *Stickman* had to slay the dragon by hitting it with a sword whilst avoiding being burnt to death by touching the fire which the dragon blows from its nostrils at regular intervals. Coding such a rule is tricky even for experienced developers yet BenL managed to code the stickman sprite to achieve the rule he had in mind. One can state that there is nothing imaginative in the narrative adopted. There are a number of games where the player has to battle a dragon. However in this case the imaginative activity was in the way BenL coded the sprites to achieve the intended game play.

Another example where the imaginative activity was very evident was in the game *Bob's Adventure*. In this game Daniel9000 shows his passion for game designing by playing tricks on the player through the rules created in the game. Initially the player learns that moving *Bob* over coloured areas of the screen switches off laser beams which stop *Bob* from proceeding on its quest. At a later level the same control, that is moving *Bob* over coloured areas of the screen, is used to switch on a game character *Creeper* which tries to stop *Bob* from progressing in the game. By using this strategy the game maker manages to think alternative ways of making his game engaging to play.

These two examples demonstrate that the game making activity provided possibilities for the students to be imaginative in multiple ways, in the ways they designed their games as well as in ways of making their designs work through coding.

6.5.1.1.2. Fashioning imagination to produce outcome

Creativity is seen as a process where the imaginative activity is shaped and reshaped to arrive to a creative act. Creative is about making and producing. Whilst making their games the children followed a process as elaborated in section 6.2 (page 183). What is important to stress here is the iterative nature of this process. The children showed that creating is not about coming up with an idea and simply enacting that idea. It is about coming up with an idea and implementing it in iterative steps where in each step the idea gets refined and moulded into the end result. This iterative process approach was quite evident in a number of games created by the students during the workshop when looking at the games taking shape over the course of the weeks. It was most evident in Racer were ideas were initially adopted and then discarded midway through the creation process once the game maker realised that the development environment did not allow him to generate the required outcome. The development environment acted as a constraint but KyleC worked around it to continue his journey. The iterative nature of this process results in an end product which might not be the same as the product imagined initially. The finished game *Racer* was different from the initial *Racer* imagined by KyleC. And this did not just happened to *Racer* or just because of the constraints of the development environment. Children picked up ideas from each other's games whilst testing their products created by their peers. This does not mean that the resulting games were homogenous. On the contrary the games were quite different from each other. The children managed to work through the process of shaping their imagination to arrive to the stage of producing a creative outcome.

6.5.1.1.3. Originality

As discussed in section 2.3.2.5 (page 57) there are multiple ways of looking at originality. An outcome can be original from an individual perspective, relative to the peer group and historically original. What the children had achieved by participating in this gaming workshop was original from an individual perspective. The children were introduced to game making and all ended up creating a functioning game which was engaging to play. This result was original since the children were not capable of creating digital games before attending the workshop. One might argue that some of the games created were also relatively original since when comparing some of the games to the games created by their peers they stood out in terms of features used. *Bob's Adventures* stands out in the rules adopted in the game and in the management of difficulty. It was quite original for Daniel9000 to adopt rules which worked in favour of the player at one point in the game and then enabled an obstacle for the player at the latter stages of the game. Using the size of obstacles to manage difficulty was also an original feature since no other game produced in the workshop used the same approach.

Daniel9000's game was not the only one to be relatively original when compared to the games produced by the peers. KyleC's *Racer* was original in terms of the display and feedback adopted. *Racer* had the racing track split over a number of screens with each screen flowing into the next giving the player the illusion of driving the car on a circular track (see Figure 2.1-1 on page 159). Racing also adopted a race against time approach. This too was original since most of the games implemented in the workshop were structured such that every screen was a different level.

Another game which had original features was *Click the Zebra*. This game allowed the user to use the mouse pointer to move the player operated character rather than the keyboard as used in all the other games created during the workshop. Here too the game makers showed their will to experiment by trying blocks in Scratch which were not discussed in the discussion part of the workshop.

6.5.1.1.4. Value

One of the defining outcomes of creativity is that the product created has to be of value. Creativity is not just about generating ideas; it involves a judgment process which evaluates these ideas (Robinson 2011, p.153). The evaluative process of the product can be shared with others or involve period of quiet reflection (see section 2.3.2.6 page 58). The evaluation process of the games was present throughout the game creation process through the testing phases. There were various phases of testing. The game makers evaluated the value of the game they were creating by testing each feature meticulously before seeking validations by the peers. The group testing stage acted as a validation by the group whereby the group judged the game on their experiences and believes of what works and what does not in a digital game. Features got discarded or improved based on the feedback received during the group testing stage. Students were eager to test each other's games and this was a sign that they valued the games being created. The "ludic attitude that sees the world's structures as opportunities for playful engagement" (Zimmerman 2007, p.27) helped make the evaluative component of the creative process an enjoyable routine. The on-going evaluation was seen as an opportunity for healthy competition where each tester did his best to unearth defects in the game of the game maker with every game maker trying one's best to ensure that the game being created was defect free.

6.5.2 Introducing a programming language to young children

As discussed in section 2.4.1 (page 68) there has been a recent drive towards introducing programming to young children. The motivation behind this drive is an economic one where programming is seen as a skill that will be in demand and hence should be introduced to children at an early stage. As I argued earlier I think that programming should be introduced to children because it provides another means of self-expression rather than just for future economic gain. Irrespective of the motivation to introduce children to programming there has been an emergence of games, environments and physical devices all claiming to introduce programming to children. Although the tools are important using the right pedagogic approach to introduce programming should take a central focus. The literature seems to ignore the methodological implications on the process to use to introduce the children to creating artefacts by using a programming language. Most of the literature found tackles the methodology of introducing programming to first year university students. Although some of the issues faced by educators and students are the same, introducing programming to an eighteen year old is not the same as introducing programming to an eleven year old. Although the methodology to introduce children to programming was not the main focus of this research I feel that the methodology used was successful and it merits further discussion in view of carrying out further research in the future.

As discussed in section 2.4.1 (page 68) literature outlines four different ways to introduce programming. In this project I followed a problem based approach where the children were presented with a full system which partially worked. The students were first shown how to run the game and were then asked to come up with suggestions for improving the game. Through this approach I strove to follow the suggestions of Kölling (2008) by getting the students to wet their feet by executing a ready-made game. Rather than providing the children with a series of worksheets to extend the readymade game the children were encouraged to come with suggestions for extending the game. The suggestions provided by the students were then used to nurture a discussion on how the existing code could be changed and extended. For every session brief videos were provided online consolidating the main items covered during the sessions. These videos were used by the children whenever they wanted to refresh their memory on the methods we used during the workshop to extend and improve the game.

The amount of time which the session was teacher led was limited to the first part of the workshop so that the children could engage with hands on practice to try out the techniques used to extend the game.

In the latter part of the workshop the students were asked to create their own game, what Kölling (2008) calls ‘the master test’. It is through the analysis of this master test, and the process to create it, that one can evaluate the take up of the programming language. There was a good mix of games created in the final session of the workshop. The games created were not simple graphical remixes of the *Shark eat Fish* game used during the initial part of the workshop. This clearly indicates that the children had not only learned to read the code and understand the actions that the code should produce when executed, but they had internalised the meaning of the constructs and found it possible to use these constructs to author their own creations. This internalisation led to a form of fluency where the children felt that they were knowledgeable enough to create the game that at times they had envisioned well before the start of the workshop. The students were fluent enough to be able to experiment and engage in meaningful discussions whenever a problem was encountered in their code. This fluency picked up during the workshop can be seen in the creation of *Racer.* KyleC was detecting whether the car was driven off track or hit an obstacle by checking the colour of the background that the car was touching (see section 5.4.1.7 page 175). This method of collision detection was initially suggested by BenL when discussing how to build a refuge for the fish in *The Shark and Fish* game. However when KyleC was applying this approach he hit a snag. Whenever the background showing a portion of the track was changed for another background Scratch was displaying a white screen. The white screen display was only for a fraction of a second, not long enough for the human eye to detect it but long enough for Scratch to detect that the car had hit a white obstacle or was placed off track. Providing a solution for this problem required KyleC to discuss the problem with one of the helper teachers and to build a solution by using variables. It is important to highlight the role of the teacher at this point. The teacher did not take the role of an expert, knowing where the problem lies and how to put a fix to solve it. Instead she took the stance of an “ignorant co-worker in the thick of action” (McWilliam 2008) and discussed strategies that might be adopted to locate what was making the game fail. These strategies helped in eventually identifying what was causing the problem and putting in a fix to solve the problem. Participating in a discussion and coming up with a solution that worked demonstrated that KyleC had become fluent enough with Scratch to be able to help devise a solution and code it too. I am not suggesting that all children had the same level of fluency at the end of the workshop. The fluency picked up by KyleC and others in the workshop demonstrates that the pedagogy adopted in this workshop was successful with this group of students.

As discussed in section 5.4.1 all the games analysed made use of sequences, loops, parallelism, events, conditionals and data. Most of the students remixed and reused aspects of code from other projects. Their systematic thinking produced games which were modular. Testing and debugging was an active component of the game development process. Hence the pedagogy used during the workshop was effective in fostering an introduction to programming and computational thinking.

6.5.2.1. This approach versus the approaches found in literature.

In section 2.4.1 I discussed four approaches found in literature which are used to introduce programming to students. The approach used in this research project was based on the full systems approach where the students are introduced to a ready-made large system and then are guided to learn how to read the code. The main difference between the approach adopted in this research project and the full systems approach is that the children were given the lead to propose any changes to the initial game provided. This was done in order to foster more creativity; however this approach also had its benefits from a programming perspective since this stance increased the engagement and motivation in the students to learn how to apply the changes they suggested to the game.

It is typical in courses following the full systems approach to provide the students with a series of challenges which they have to solve. There were no ready-made worksheets which the students had to follow in the research project. Instead the suggestions the workshop participants made for improving the game were the ones tackled by the teacher to introduce new features of the programming environment.

In this research project the game features suggested by the students were placed in popularity order (see Table 5 page 126). The most popular feature suggested by the students was to add more obstacles such as bombs or plastic bottles that could potentially kill the shark. This feature was not the easiest feature to tackle since this change demanded the addition of new sprites in the form of plastic bottles or bombs.

As a teacher I had to make the judgment call to start from the third suggestion. The suggestion which was the third most popular with the students was to make fish go faster. This change only required a change in the existing code and hence tallied with Kölling's (2008) suggestion that first students should manipulate code and then move to creating new code.

6.6 Conclusion

In this chapter I explored the main findings for this research project namely:

1. The stages that children go through whilst creating a digital game as described in section 6.2 are similar to a number of phases discussed in previous research. However the stages of development outlined in this research highlight the importance of the social aspect in game development. Group testing not only has an effect on the game being tested but also acts as a source of cross fertilisation of ideas between the students testing the game and the students developing the game.
2. There were a number of benefits from a gaming literacy perspective that children gained whilst participating in this research project.
    - The game making experience provided the students with a possibility to enact their systematic thinking when designing their games as a system made up of interrelated subsystems.
    - Prior gaming experience had a dual effect on the students. The prior experience contributed to the way they designed their games and also contributed to the way they engaged with game making using a playful engagement attitude. The students were competitive yet cooperative whilst making their games. Group testing resulted in a game-like experience with the tester striving to find a *Glitch* whilst the game maker endeavouring to ensure that all defects are ironed out during the feature testing stage. Cheating in game playing also found itself into the game design process as the game designers using similar strategies to speed up the game testing.
    - The area which gained mostly from participating in the game making was the design aspect. All the games were complete and demonstrated

that the children were multimodal savvy. They created games that were well balanced from the difficulty point of view and that provided the game players with instructions on how to play as well as implemented appropriate feedback mechanisms.

3. The game making experience acted as potential funds of identity with the children experiencing different roles during the different game making stages.
4. Teaching for creativity led to a series of benefits. The children were more engaged as a result of learning whilst implementing changes they had suggested themselves. They asked questions and engaged in discussions in a bid to identify possible solutions to the questions raised. They were critical of the way features were coded in the game provided and provided suggestions on how the game could be improved. Finally the students exercised their creativity during the different phases of the game creation process.
5. The workshop structure proved to be an appropriate way of introducing children to programming. All the games analysed made use of sequences, loops, parallelism, events, conditionals and data. The students' systematic thinking produced games which were modular whilst testing and debugging were an active component of the game development process.

In the next chapter I reflect on the implications these findings have on teaching and learning, assessment and the tools that can be used to introduce students to programming and game development. I also reflect on the limitations of this study and suggest further areas to consider in future research

7. Conclusion and Implications

7.1 Introduction

Playing digital games is an important leisure activity for a large number of us. The increasing popularity of digital games has led governments around Europe to acknowledge the contribution digital game production brings to the economies of these countries. In their influential report Livingstone and Hope (2011) outline ten recommendations which schools should adopt in order for UK to retain its current position in the digital game creation market. These recommendations point towards the inclusion of game based learning in schools and to the introduction of school clubs which offer a different environment to that found traditionally in classrooms.

Although playing digital games is quite popular amongst Maltese children, there is very little evidence of game based learning activities conducted in the Maltese schools. It is disappointing that such an interest is not acknowledged enough in schools through the inclusion of game based learning activities.

Three approaches have been adopted in literature to introduce game based learning into schools: the use of serious games in class; the use of commercial games and the opportunity for children to author their own games. In this research a workshop was held after school hours to introduce children to game making. I was surprised with the level of interest amongst children to learn how to develop games. This project allowed the children to work together and provided an avenue where the children could express their creativity and knowledge about games. Two research questions were posed in this research project:

1. What processes do eleven year olds follow to create digital games?
2. What benefits does creating a digital game have from a gaming literacy perspective?

A qualitative case study approach was used in this research project to answer the two research questions posed. The stages that children go through whilst creating a digital game were found to be similar to the ones identified in previous research. However, in this research the importance of the social aspect was more foregrounded. Group testing not only served as validation by the community but also acted as a source of cross fertilization of ideas between students making the game and the students testing it.

A number of benefits were observed from a gaming literacy perspective. Game making enabled the children to apply their systematic thinking whilst designing games as a group of interrelated subsystems. Prior game playing experience contributed to the way the children designed their games whilst also generating a playful attitude with which they engaged with game making. This playful attitude resulted with the children being competitive yet cooperative. A game like experience was observed whilst the students group tested their games and ironed out defects unearthed by their peers.

The gaming literacy area which gained mostly from the children's participation in the game making activity was the design aspect of gaming literacy. The children demonstrated their multimodal savviness through the design of their complete games. The games created provided the players with game instructions and appropriate feedback mechanisms.

Other benefits were identified which do not fall within the guiding research questions for this research. The teaching for creativity approach adopted in this research resulted in students being engaged in discussions whilst trying to identify solutions to the problems they raised. The workshop structure was also found to be adequate for introducing children to programming.

There are a series of implications that can be drawn from the results of this research. These implications are related to the way the NCF's cross-curricular theme of digital literacy is implemented in schools; implications on teaching and learning of digital game making and implications on assessment strategies.

7.2 Digital Literacy as a subject in the Maltese schools

Although the national minimum curriculum framework lists digital literacy as a cross curricular theme, students attending state schools still attend a weekly forty-five minute lesson called ICT. The ICT curriculum adopted in the Maltese state schools is based on the European Computer Driving License (ECDL) curriculum with the students sitting for the ECDL certification at the end of the secondary years, aged sixteen.

The ECDL foundation markets itself as an organisation whose mission is to enable proficient use of ICT that empowers individuals, organisations and society, through the development, promotion and delivery of quality certification programmes throughout the world (ECDL Foundation n.d.). It markets the ECDL curriculum as an externally defined structure that can be readily adopted by schools to embed digital literacy into schools (ECDL Foundation 2010). The curriculum is not exclusively aimed at schools, since the same curriculum and approved courseware are marketed to be used in schools as well as with adults who are seeking employment.

It is important to note the definition that the ECDL foundation reserves for the term digital literacy. This definition explains the rationale behind the stance taken about the courseware structure and assessment strategy adopted. EDCL foundation defines digital literacy to be:

> the set of enabling skills that are required to efficiently use commonly available technology, including computers. Young people need to be equipped with these enabling skills if they are to be able to function both as a participant in the knowledge economy and an active citizen in the knowledge society. (ECDL Foundation 2010)

This definition of digital literacy is what Lankshear and Knobel (2008) call a standardised operational definition where being digital literate is defined as a set of skills to hold. The motivation behind being digital literate is an economic one where digital literacy is seen as a key to participate in the job world. Although I acknowledge the importance of the economy, I think that such an argument is too narrow especially when discussing digital literacy with school children. It is important to consider that, in all probability, today's students will go on to take up careers which are yet to be conceived.

The ECDL curriculum is assessment driven, with the students having to follow segments of knowledge, called modules, which are validated by an electronic test. This approach towards digital literacy has two main drawbacks.

- Knowledge is compartmentalised into modules which the students tend to study in isolation. Through this approach students tend to see technology as an end in itself rather than as a means to an end. A project based approach would allow the students to experience the application of technology to solve a problem or create a product.

- Since the validation occurs through an electronic test, students have to learn how to conduct an operation using one standard pathway. Technology tends to allow multiple pathways to conduct an operation. For example in a word processor the operation of underlining a word can be performed by choosing an option in a menu; by pressing a combination of keys on the keyboard or pressing an icon on a toolbar. Which pathway the operator uses is of little importance since all options result in the same outcome. An automated test will only allow one type of answer. This approach forces the students to learn how to perform a task using one approach only in order to make it through the electronic exam. The method of assessment used by ECDL has been criticised in literature as it promotes the mastery of specific technical skills with scarce emphasis on competencies (Calvani, Fini et al. 2009).

7.2.1 The NCF and ECDL

Given the importance the NCF reserves for the cross curricular themes of creativity and innovation ; learning to learn and cooperative learning I argue that the strategy of basing the ICT curriculum solely on the ECDL certification in Maltese schools needs to be reconsidered. As discussed in section 2.3.3 (page 58) teaching for creativity requires the passing of control to the students, valuing the students' ownership and encouraging the posing and answering of questions. It is hard to conceive doing this with the restrictions imposed by the ECDL approach. The children who participated in this research project built on their pre-existing interest of digital gaming to design and build their own games. Through the process, not only where they introduced to programmed control, a strand which the NCF lists to be part of the digital literacy curriculum, but also collaborated together and engaged in problem solving strategies to create working digital games.

A pedagogic approach similar to the one used in this research that builds on the children's interest in digital gaming and game making, might be a good strategy to adopt in the ICT sessions. This approach could easily be extended to incorporate the traditional topics which currently form part of ECDL. The children could write user manuals about the games created and conduct presentations about the games to their peers. In this way, the children can still acquire competence in traditional software tools whilst at the same time using a project based approach built around their interest in games and game making. The project could also be extended to included discussions on copyright and attribution in an authentic learning environment as discussed in section 4.6.1.1 (page 129).

7.2.2 Implications for teaching and learning

The pedagogic approach adapted for this research project allowed the students to take the driving seat in learning, whilst proposing changes to the initial game and at a later stage whilst designing and authoring their game. This teaching approach resulted in the children being more engaged in posing and answering questions. The method adopted was an appropriate teaching approach to introduce the children to programming without resorting to direct instruction. All the games analysed made use of sequencing, loops, parallelism, events, conditionals and data. The students also engaged in computational practices by being incremental in their designs; reusing and remixing media and code; testing and debugging their games and using a modularised approach to structure their games.

Leading such a workshop required different skills from the teacher than the skills required to teach an ICT class based on the ECDL curriculum. The teachers are accustomed to follow a rigid lesson plan aimed at teaching a specific skill of the curriculum. However, as discussed in section 2.3.3.1 (page 59) the skills required to teach for creativity are very different. As McWilliam (2008) suggests they require the teacher to adopt a meddler in the middle approach. This approach requires the teacher to spend less time giving instructions and more time being a useful ignorant co-worker in the thick of action. It requires the teacher to become an experimenter and risk taker rather than a risk minimiser. The teacher must shift to being a collaborative critic and authentic evaluator from being a counsellor. I am not arguing that the meddler in the middle approach was used exclusively throughout the workshop. During this research the teacher adopted different approaches during the various stages of the workshop. The teacher was a discussion moderator, throughout the discussions that occurred during the initial parts of the sessions. Moreover the teacher took on the role of a consultant on the side of the game makers advising how to approach problems unearthed. However the shift towards being an ignorant co-worker in the thick of action and a collaborative critic, must take place if we value the student adopting a stance to be able to learn how to learn. This approach is very different from the approach teachers are accustomed to adopt in ICT lessons, and this might require re-training and more importantly reflection on practice.

The pedagogical approach used in this research has implications on my practice as a pre-service teacher educator. Further research is required to study whether, conducting a practical unit on game authoring with pre-service ICT teachers will help them model a pedagogic approach similar to the one adopted in this research project once they are enrolled as newly qualified teachers in schools.

7.2.3 Implications on assessment

In section 2.4.4.5 (page 80) I discussed three ways found in literature that are used to assess computational thinking skills: project analysis, artefact based interviews and design scenarios (Brennan, Resnick 2012). These assessment strategies were intended to be used with children engaged in design activities using Scratch. Each of these methods of assessment had its drawbacks and for this research project a different approach was adopted. Games were analysed throughout the building process rather than at the end. During the building process informal discussions with the students were held on a regular basis. In this way, the process used to build the games was analysed in real time without having to rely on the children's memory. This approach points to a formative method of assessment, rather than a summative one, as is the practice with the current ICT subject modelled on the ECDL certification.

7.3 Limitations in this study

The current study has certain limitations that need to be taken into account. In this section I discuss two limitations. The first limitation is about the preparation in terms of programming skills of the teachers conducting this workshop, whilst the second limitation is about the online space used throughout the project.

7.3.1 Programming knowledge of teacher

The teachers recruited to help with this research project, as well as myself as the researcher, had prior programming experience using Scratch. This experience was valuable when discussing with the children possible solutions for problems they encountered. Teachers had to rely on prior game development experience to discuss adequate solutions. Hence the programming experience of any person leading similar workshops is of fundamental importance. Teachers preparing to lead similar workshops need to take time to experiment with the game making software.

7.3.2 Introducing the children to the Scratch online space

Although an online space was specifically created for the children to be used during this research project, it is felt that the children might have benefited more had they been introduced to the Scratch online space during the workshop. The children used the workshop's online space to upload versions of their games and to interact with the other members of the workshop. If the students had uploaded versions of their games to the Scratch online space, they could have also obtained feedback from other experienced Scratchers, rather than just their peers. This might have had beneficial effects on their game making experiences. On the other hand one could also argue that lack of feedback from the online community could have had a negative motivational effect on the children.

7.4 **Recommendations for further study**

Following the outcomes of this research, in this section I propose a series of recommendations for future research.

7.4.1 Game making during school hours

This research project focused on introducing game making to children in an afternoon environment. The sessions were one and a half hours long, with an initial discussion period followed by hands-on time where the children could work on the computer creating their own games. A lot of adaptation to the structure of the workshop would be required to adapt this approach to fit into a weekly forty five minute session. Creating an effective series of sessions that introduce children to game making within the constraints of the school timetable, would merit further research.

7.4.2 Effect on students taking up Computing

This research project introduced the students to programming through game development. It will be interesting to revisit the students after they choose the optional subjects in secondary school to see how many of them have chosen to study Computing and discuss whether participation in the game making workshop had any bearing on their decision making.

7.4.3 One Laptop per child programme

In January 2014 the Ministry for Education and Employment launched a pilot project to introduce tablets in the primary Maltese schools. The goals of the eventual roll out of tablets in the primary classes is to promote literacy skills, numeracy skills and digital literacy (Minister for Education and Employment 2014). Students will be given a tablet each which they will use in class and at home during the scholastic year. There are a number of game making tools such as Scratch Jr which run on a tablet. It would be interesting to research game making with children that can use the same device at home and at school. Tablets have the additional benefits of being equipped with inbuilt multimedia capabilities including video and audio. I am intrigued in observing how the children would use these inbuilt devices whilst building games.

7.4.4 A development environment with a social flair

Commercial digital games are usually built by teams of people with large budgets. The popular game Grand Theft Auto V cost around £170 million to develop and market (Usher 2013). It is estimated that over 300 staff including designers, artists and programmers worked on the production of the game. Similarly games created within the Indie[9] game space with very limited budgets are usually created by small teams made up of developers, artists and designers.

In this research project four out of fourteen students choose to work in pairs to create a game (see section 5.5.1 page 180). Notwithstanding the fact that most of the students worked individually on their game, the game development model discussed in section 6.2 (page 183) highlights the importance of collaboration whilst creating a game.

As a software development program, Scratch does not facilitate working in pairs or in teams. Students could work together during the workshop by taking turns writing code. This would entail one member of the team coding whilst the second member of the

[9] Indie games are games created by independent teams generally without a video publisher financial support. Some indie games such as World of Goo and Minecraft are very successful.

team sits on the side providing suggestions as observed during this research project. However, if the members of the team try to work on the games in parallel whilst away from class they would end up with two distinct games. The children would need to go through the laborious process of trying to merge the individual games by copying the relevant pieces of code from one game to the other. In Scratch one cannot simply copy and paste code from one game to another as is the practice with other software such as word processors. Hence the merging of the individual games tends to be a cumbersome operation that takes time and involves the rewriting of code. Scratch contains all the code for the game in one file. If the code for different Sprites were contained in separate files, potentially one could copy sprites from one game to another by copying the sprite files. This would only partially solve the problem as game makers could work on the same game in parallel and merge their work as long as they work on separate sprites. The same approach of storing all the code in one file is followed by other game development software such as Microsoft's Kodu and GameMaker and hence this limitation is not restricted to Scratch.

This problem does not occur in industry where teams of developers work together on developing the same piece of software. Most development software integrates with a *version control system* which would allow software developers to synchronise code between different versions of the software being developed. Version control systems provide access to a repository of code, maintain historical editions of the software and record all changes in a log (Loeliger, McCullough 2012). If such an option had to be added to a game development environment such as Scratch, children would be able to work together on the same project without ending up with different games which they would need to synchronise manually. Such a development environment would merit further research to investigate how teams of students would put these features to use to create games.

Appendix A
References

AARSETH, E.J., 1997. *Cybertext: perspectives on ergodic literature.* Maryland: John Hopkins University Press.

ACKERMANN, E., 2001. Piaget’s constructivism, Papert’s constructionism: What’s the difference. *Future of learning group publication,* **5**(3), pp. 438.

ADAMS, E. and ROLLINGS, A., 2007. *Game Design and Development: Fundamentals of game design.* New Jersey: Pearson Prentice Hall.

AHAMED, S.I., BRYLOW, D., GE, R., MADIRAJU, P., MERRILL, S.J., STRUBLE, C.A. and EARLY, J.P., 2010. Computational thinking for the sciences: a three day workshop for high school science teachers, *Proceedings of the 41st ACM technical symposium on Computer science education* 2010, ACM, pp. 42-46.

AHO, A.V., 2011. Ubiquity symposium: Computation and Computational Thinking. *Ubiquity,* **2011**(January), pp. 1.

AKCAOGLU, M. and KOEHLER, M.J., 2014. Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. *Computers & Education,* **75**, pp. 72-81.

ALLAN, V., BARR, V., BRYLOW, D. and HAMBRUSCH, S., 2010. Computational thinking in high school courses, *Proceedings of the 41st ACM technical symposium on Computer science education* 2010, ACM, pp. 390-391.

ALLEN, D.E., DONHAM, R.S. and BERNHARDT, S.A., 2011. Problem-based learning. *New Directions for Teaching and Learning,* **2011**(128), pp. 21-29.

ALVERMANN, D.E., 2010. *Adolescents' online literacies: connecting classrooms, digital media, and popular culture.* New York: Peter Lang.

AMABILE, T.M., 1996. *Creativity and innovation in organizations.* Boston: Harvard Business School.

ANCHOR, R., 1978. History and play: Johan Huizinga and his critics. *History and Theory,* **17**(1), pp. 63-93.

APPERLEY, T., 2010. Gaming rhythms: Play and counterplay from the situated to the global. Institute of Network Cultures: Amsterdam. Ashton, D. & Newman, J.(2010). Relations of control: Walkthroughs and the structuring of player agency. *The Fibreculture Journal,* **16**.

APPERLEY, T. and BEAVIS, C., 2011. Literacy into action: Digital games as action and text in the English and literacy classroom. *Pedagogies: An International Journal,* **6**(2), pp. 130-143.

AVEDON, E.M. and SUTTON-SMITH, B., 1981. *The study of games.* New York: John Wiley & Sons, Inc.

BARAB, S., DODGE, T., TUZUN, H., JOB-SLUDER, K., JACKSON, C., ARICI, A., JOB-SLUDER, L., CARTEAUX JR, R., GILBERTSON, J. and HEISELT, C., 2007. The Quest Atlantis Project: A socially-responsive play space for learning. In: B.E. SHELTON and D.A. WILEY, eds, *The educational design and use of simulation computer games.* Rotherdam: Sense Publishers, pp. 161-188.

BARAB, S., THOMAS, M., DODGE, T., CARTEAUX, R. and TUZUN, H., 2005. Making learning fun: Quest Atlantis, a game without guns. *Educational Technology Research and Development,* **53**(1), pp. 86-107.

BARR, D., HARRISON, J. and CONERY, L., 2011. Computational Thinking: A Digital Age Skill for Everyone. *Learning & Leading with Technology,* **38**(6), pp. 20-23.

BARR, V. and STEPHENSON, C., 2011. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads,* **2**(1), pp. 48-54.

BASU, S., KINNEBREW, J.S. and BISWAS, G., 2014. Assessing Student Performance in a Computational-Thinking Based Science Learning Environment, *Intelligent Tutoring Systems* 2014, Springer, pp. 476-481.

BAYTAK, A., 2009. *An investigation of the artifacts, outcomes, and processes of constructing computer games about environmental science in a fifth grade science classroom,* Ph.D thesis, The Pennsylvania State University.

BAYTAK, A. and LAND, S.M., 2011. An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. *Educational Technology Research and Development,* **59**(6), pp. 765-782.

BEAVIS, C. and APPERLEY, T., 2012. A model for games and literacy. In: C. BEAVIS, J. O'MARA and L. MCNIECE, eds, *Digital Games: Literacy in Action.* pp. 12.

BEAVIS, C. and O'MARA, J., 2010. Computer games-pushing at the boundaries of literacy. *Australian Journal of Language and Literacy, The,* **33**(1), pp. 65.

BEAVIS, C., 2012. Multiliteracies in the Wild. In: G. MERCHANT, J. GILLEN, J. MARSH and J. DAVIES, eds, *Virtual literacies: Interactive spaces for children and young people.* London edn. Routledge, pp. 57.

BEAVIS, C., O'MARA, J. and MCNEICE, L., 2012. *Digital games: Literacy in action.* Australia: Wakefield Press.

BEGHETTO, R.A., 2005. Does assessment kill student creativity? *The Educational Forum* 2005, Taylor & Francis, pp. 254-263.

BERS, M.U., 2010. The TangibleK Robotics Program: Applied Computational Thinking for Young Children. *Early Childhood Research & Practice,* **12**(2), pp. 2.

BETCHER, W., 1987. *Intimate play: creating romance in everyday life.* New York: Viking Pr.

BOGOST, I., 2007. *Persuasive games: The expressive power of videogames.* Cambridge: Mit Press.

BÖHM, S. and LAND, C., 2009. No measure for culture? Value in the new economy. *Capital & Class,* **33**(1), pp. 75-98.

BRAHMS, L.J., 2014. *MAKING AS A LEARNING PROCESS: IDENTIFYING AND SUPPORTING FAMILY LEARNING IN INFORMAL SETTINGS.* PhD edn. University of Pittsburgh.

BRAND, J.E., 2012. *Digital Australia 2012 (DA12).* Australia: School of Communication and Media Faculty of Humanities and Social Sciences Bond University.

BRANSFORD, J.D., BROWN, A.L. and COCKING, R.R., 1999. *How people learn: Brain, mind, experience, and school.* United States: National Academy Press.

BRENNAN, K. and RESNICK, M., 2012. New frameworks for studying and assessing the development of computational thinking, *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.* 2012.

BRUNER, J.S., 1966. *Toward a theory of instruction.* Cambridge: Belkapp Press.

BRYANT, R., CHINN, D., HAUSER, G., FOLSOM, M. and WALLACE, S., 2009. Computational thinking: what is it, how is it relevant, who's doing what with it? *Journal of Computing Sciences in Colleges,* **25**(1), pp. 170-171.

BUCKINGHAM, D. and BURN, A., 2007. Game literacy in theory and practice. *Journal of Educational Multimedia and Hypermedia,* **16**(3), pp. 323-349.

BUNDY, A., 2007. Computational thinking is pervasive. *Journal of Scientific and Practical Computing,* **1**(2), pp. 67-69.

BURN, A., 2009. *Making new media: Creative production and digital literacies.* New York: Peter Lang Pub Inc.

BURNARD, P., CRAFT, A., CREMIN, T., DUFFY, B., HANSON, R., KEENE, J., HAYNES, L. and BURNS, D., 2006. Documenting ‘possibility thinking’: A journey of collaborative enquiry. *International Journal of Early Years Education,* **14**(3), pp. 243-262.

BUSJAHN, T. and SCHULTE, C., 2013. The use of code reading in teaching programming, *Proceedings of the 13th Koli Calling International Conference on Computing Education Research* 2013, ACM, pp. 3-11.

BUSUTTIL, L., CAMILLERI, L., CAMILLERI, V., DINGLI, A. and MONTEBELLO, M., 2014. *Digital and Video Game Usage in Malta.* University of Malta:Malta: gamED (Games in Education).

CAILLOIS, R., 1958. *Man, play, and games.* Paris: Univ of Illinois Pr.

CALVANI, A., FINI, A. and RANIERI, M., 2009. Assessing Digital Competence in Secondary Education - Issues, Models and Instruments. In: M. LEANING, ed, *Issues in Information and Media Literacy: Education, Practice and Pedagogy.* California: Informing Science Press, pp. 153-172.

CAMPBELL, W. and BOLKER, E., 2002. Teaching programming by immersion, reading and writing, *Frontiers in Education, 2002. FIE 2002. 32nd Annual* 2002, IEEE, pp. T4G-23-T4G-28 vol. 1.

CARBONARO, M., CUTUMISU, M., DUFF, H., GILLIS, S., ONUCZKO, C., SIEGEL, J., SCHAEFFER, J., SCHUMACHER, A., SZAFRON, D. and WAUGH, K., 2008. Interactive story authoring: A viable form of creative expression for the classroom. *Computers & Education,* **51**(2), pp. 687-707.

CAROLYN YANG, Y. and CHANG, C., 2013. Empowering students through digital game authorship: Enhancing concentration, critical thinking, and academic achievement. *Computers & Education,* **68**, pp. 334-344.

CARR, D., BUCKINGHAM, D., BURN, A. and SCOTT, G., 2006. *Computer games: Text, narrative and play.* Cambridge: Polity Press.

CARRINGTON, V. and ROBINSON, M., 2009. *Digital literacies: Social learning and classroom practices.* London: Sage Publications Ltd.

CARTER, B. and SEALEY, A., 2009. Reflexivity, realism and the process of casing. In: D. BYRNE and C. RAGIN, eds, *The SAGE Handbook of Case-Based Methods.* London: SAGE Publications Limited, pp. 69-83.

CASPERSEN, J., BENNEDSEN, M. and KÖLLING, M., eds, 2008. *Reflections on the Teaching of Programming.* Germany: Springer.

CAUGHEY, J., 1984. Imaginary source worlds: A cultural approach.

CHIU, Y., LUI, A. and KIM PING HUI, E., 2012. *Super Scratch Programming Adventure.* San Francisco: No Starch Press.

CHOE, I., 2006. Creativity - A sudden rising star in Korea. In: J. KAUFMAN and R. STERNBERG, eds, *The international handbook of creativity.* Cambridge, UK: Cambridge University Press, pp. 395-420.

CONSALVO, M., 2009. There is no magic circle. *Games and Culture,* **4**(4), pp. 408-417.

CONSALVO, M., 2007. *Cheating: Gaining advantage in videogames.* Cambridge: The MIT Press.

CORTINA, T.J., 2007. An introduction to computer science for non-majors using principles of computation, *ACM SIGCSE Bulletin* 2007, ACM, pp. 218-222.

CRAFT, A., 2010. Teaching for possibility thinking: What is it and how do we do it. *Learning Matters,* **15**(1), pp. 19-23.

CRAFT, A., 2013. Childhood, possibility thinking and wise, humanising educational futures. *International Journal of Educational Research,* **61**, pp. 126-134.

CRAFT, A., 2011. *Creativity and Education Futures: Learning in a Digital Age.* Staffordshire: England: Trentham Books Ltd.

CRAFT, A., 2005. *Creativity in schools: Tensions and dilemmas.* London: Routledge.

CRAFT, A., 2003a. Creative thinking in the early years of education. *Early Years: An International Journal of Research and Development,* **23**(2), pp. 143-154.

CRAFT, A., 2003b. *Creativity across the primary curriculum: Framing and developing practice.* London: Routledge.

CRAFT, A., 2001. 'Little c Creativity'. In: A. CRAFT, B. JEFFREY and M. LEIBLING, eds, *Creativity in Education.* London: Continuum, pp. 45-61.

CRAFT, A., CREMIN, T., BURNARD, P. and CHAPPELL, K., 2008. Possibility thinking with children in England aged 3-7. In: A. CRAFT, T. CREMIN and P. BURNARD, eds, *Creative Learning 3-11.* Trentham Books Ltd., pp. 65-73.

CRAFT, A., CREMIN, T., BURNARD, P., DRAGOVIC, T. and CHAPPELL, K., 2013. Possibility Thinking: culminative studies of an evidence-based concept driving creativity? *Education 3-13,* **41**(5), pp. 538-556.

CRAFT, A., MCCONNON, L. and MATTHEWS, A., 2012. Child-initiated play and professional creativity: Enabling four-year-olds' possibility thinking. *Thinking Skills and Creativity,* **7**(1), pp. 48-61.

CRAWFORD, C., 1984. *The Art of Computer Game Design.* Berkeley, California: McGraw-Hill/Osborne Media.

CREMIN, T., BURNARD, P. and CRAFT, A., 2006. Pedagogy and possibility thinking in the early years. *Thinking skills and creativity,* **1**(2), pp. 108-119.

CROPLEY, A.J., 2001. *Creativity in education & learning: A guide for teachers and educators.* London: Routledge.

CSIKSZENTMIHALYI, M., 1996. *Creativity: Flow and the psychology of discovery and invention.* New York: Harper Collins Publishers.

CUNY, J., SNYDER, L. and WING, J.M., 2010. Demystifying computational thinking for non-computer scientists. *Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~ CompThink/resources/TheLinkWing.pdf, .*

CUTRONI, J., 2010. *Google Analytics.* California: O'Reilly Media.

DAWS, J.E., 2009. Teachers and students as co-learners: possibilities and problems. *The Journal of Educational Enquiry,* **6**(1),.

DE FREITAS, S. and MAHARG, P., 2011. *Digital games and learning.* London: Continuum International Publishing Group.

DENG, Z., HUANG, W. and DONG, R., 2009. Discussion of ability cultivation of computational thinking in course teaching, *Education Technology and Computer, 2009. ICETC'09. International Conference on* 2009, IEEE, pp. 197-200.

DENNER, J., WERNER, L. and ORTIZ, E., 2012. Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education,* **58**(1), pp. 240-249.

DENNING, P.J., 2009. The profession of IT Beyond computational thinking. *Communications of the ACM,* **52**(6), pp. 28-30.

DENZIN, N. and LINCOLN, Y., 2005. *The Sage handbook of qualitative research.* 3 edn. California: Sage Publications, Inc.

DIERBACH, C., HOCHHEISER, H., COLLINS, S., JEROME, G., ARIZA, C., KELLEHER, T., KLEINSASSER, W., DEHLINGER, J. and KAZA, S., 2011. A model for piloting pathways for computational thinking in a general education curriculum, *Proceedings of the 42nd ACM technical symposium on Computer science education* 2011, ACM, pp. 257-262.

DIGITAL GAMES MALTA, 07/2011, 2011-last update, Gamezing 2011 competition rules. Available: https://www.facebook.com/pages/Digital-Games-Production-Malta/351683102213?v=app_2347471856 [06/2014, 2014].

DINGSØYR, T., NERUR, S., BALIJEPALLY, V. and MOE, N.B., 2012. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, .*

DOUGHERTY, D., 2012. The maker movement. *Innovations,* **7**(3), pp. 11-14.

DQSE, 2011. *TOWARDS A QUALITY EDUCATION FOR ALL The National Curriculum Framework.*

ECDL FOUNDATION, n.d.-last update, ECDL Foundation: About Us [Homepage of ECDL], [Online]. Available: http://www.ecdl.org/index.jsp?p=93&n=94&a=3235 [08/30, 2014].

ECDL FOUNDATION, 2010. *Building a Digitally Literate Europe through Education* . ECDL Foundation.

EGENFELDT-NIELSEN, S., SMITH, J.H. and TOSCA, S.P., 2013. *Understanding video games: The essential introduction.* London: Routledge.

ESTEBAN-GUITART, M. and MOLL, L.C., 2014. Funds of Identity: A new concept based on the Funds of Knowledge approach. *Culture & Psychology,* **20**(1), pp. 31-48.

EUROPEAN COMMISSION, 29/07/2014, 2014-last update, Vassiliou urges Education Ministers to help kids crack the code [Homepage of European Commission], [Online]. Available: http://europa.eu/rapid/press-release_IP-14-887_en.htm [29/07/2014, 2014].

FISCHER, G., 2005. Distances and diversity: sources for social creativity, *Proceedings of the 5th conference on Creativity & cognition* 2005, ACM, pp. 128-136.

FLANNERY, L.P., SILVERMAN, B., KAZAKOFF, E.R., BERS, M.U., BONTÁ, P. and RESNICK, M., 2013. Designing ScratchJr: support for early childhood learning through computer programming, *Proceedings of the 12th International Conference on Interaction Design and Children* 2013, ACM, pp. 1-10.

FLETCHER, G.H. and LU, J.J., 2009. Education Human computing skills: rethinking the K-12 experience. *Communications of the ACM,* **52**(2), pp. 23-25.

FOWLER, M. and HIGHSMITH, J., 2001. The agile manifesto. *Software Development,* **9**(8), pp. 28-35.

FRANKLIN, D., CONRAD, P., BOE, B., NILSEN, K., HILL, C., LEN, M., DRESCHLER, G., ALDANA, G., ALMEIDA-TANAKA, P. and KIEFER, B., 2013. Assessment of computer science learning in a scratch-based outreach program, *Proceeding of the 44th ACM technical symposium on Computer science education* 2013, ACM, pp. 371-376.

GALLOWAY, A.R., 2006. *Gaming: Essays On Algorithmic Culture.* Minneapolis: University Of Minnesota Press.

GAMES, N., 2008. Video gamers in Europe-2008. *Retrieved December,* **2**, pp. 2008.

GAMEVISION EUROPE, 2010. *Video Gamers in Europe 2010.* Interactive Software Federation of Europe (ISFE).

GANDINI, L., 2011. History, ideas and basic philosophy. In: C. EDWARDS, L. GANDINI and G. FORMAN, eds, *The Hundred Languages of Children: The Reggio Emilia Experience in Transformation: The Reggio Emilia Experience in Transformation.* 3 edn. ABC-CLIO, .

GAUNTLETT, D., 2013. *Making is connecting.* Cambridge: Polity Press.

GEE, J.P., 2012-last update, Big G Games. Available: http://www.jamespaulgee.com/node/63 [10/20, 2013].

GEE, J.P., 2003. *What video games have to teach us about learning and literacy.* New York: Palgrave Macmillan.

GEE, J.P. and HAYES, E., 2012. Nurturing Affinity Spaces and Game-Based Learning. In: C. STEINKUEHLER, K. SQUIRE and S. BARAB, eds, *Games, Learning and Society. Learning and Meaning in the digital age.* New York: Cambridge University Press, pp. 129-153.

GEE, J.P., 2007. *Good video games and good learning: Collected essays on video games, learning and literacy.* New York: Peter Lang.

GEE, J.P., 2008. Learning and games. *The ecology of games: Connecting youth, games, and learning,* **3**, pp. 21-40.

GEE, J.P., 2004. *Situated language and learning: A critique of traditional schooling.* London: Routledge.

GIBSON, J.J., 1986. *The ecological approach to visual perception.* New York: Routledge.

GILL, J.H., 2000. *The tacit mode: Michael Polanyi's postmodern philosophy.* New York: New York Press.

GONZALEZ, N., MOLL, L.C. and AMANTI, C., 2005. *Funds of knowledge.* New York: Routledge.

GOOD, J., ROMERO, P., DU BOULAY, B., REID, H., HOWLAND, K. and ROBERTSON, J., 2008. An embodied interface for teaching computational thinking, *Proceedings of the 13th international conference on Intelligent user interfaces* 2008, ACM, pp. 333-336.

GROVER, S., COOPER, S. and PEA, R., 2014. Assessing computational learning in K-12, *Proceedings of the 2014 conference on Innovation & technology in computer science education* 2014, ACM, pp. 57-62.

HABGOOD, J. and OVERMARS, M., 2006. *The Game Maker's Apprentice.* New York: Springer.

HAMBRUSCH, S., HOFFMANN, C., KORB, J.T., HAUGAN, M. and HOSKING, A.L., 2009. A multidisciplinary approach towards computational thinking for science majors. *ACM SIGCSE Bulletin,* **41**(1), pp. 183-187.

HAMILTON, L. and CORBETT-WHITTIER, C., 2013. *Using case study in educational research.* London: Sage Publications Ltd.

HAREL, I.E. and PAPERT, S.E., 1991. *Constructionism.* New York: Ablex Publishing.

HENDERSON, P.B., CORTINA, T.J. and WING, J.M., 2007. Computational thinking, *ACM SIGCSE Bulletin* 2007, ACM, pp. 195-196.

HOWLAND, K., GOOD, J. and NICHOLSON, K., 2009. Language-based support for computational thinking, *Visual Languages and Human-Centric Computing, 2009. VL/HCC 2009. IEEE Symposium on* 2009, IEEE, pp. 147-150.

HU, C., 2011. Computational thinking: what it might mean and what we might do about it, *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* 2011, ACM, pp. 223-227.

HUIZINGA, J., 1955. *Homo ludens: A study of the play element in culture.* Boston: Beacon Press.

HUNT, J., 2006. *Agile software construction.* New York: Springer.

IOANNIDOU, A., BENNETT, V., REPENNING, A., KOH, K.H. and BASAWAPATNA, A., 2011. Computational Thinking Patterns. *Online Submission,* .

IOI, 2014-last update, IOI, Past, Present, Future. Available: http://www.ioinformatics.org/history.shtml [03/18, 2014].

ISBELL, C.L., STEIN, L.A., CUTLER, R., FORBES, J., FRASER, L., IMPAGLIAZZO, J., PROULX, V., RUSS, S., THOMAS, R. and XU, Y., 2010. (Re) defining computing curricula by (re) defining computing. *ACM SIGCSE Bulletin,* **41**(4), pp. 195-207.

JEFFREY, B. and CRAFT, A., 2004. Teaching creatively and teaching for creativity: distinctions and relationships. *Educational studies,* **30**(1), pp. 77-87.

JENKINS, H., 2006. *Fans, bloggers, and gamers: Exploring participatory culture.* New York: NYU Press.

JENKINS, H., CLINTON, K., PURUSHOTMA, R., ROBISON, A.J. and WEIGEL, M., 2006-last update, Confronting the challenges of participatory culture: Media education for the 21 st century. Available: http://www.macfound.org/press/publications/white-paper-confronting-the-challenges-of-participatory-culture-media-education-for-the-21st-century-by-henry-jenkins/ [07/25, 2013].

JENKINS, H. and KELLEY, W., 2013. *Reading in a Participatory Culture: Remixing Moby-Dick in the English Classroom.* New York: Teachers College Press.

JEWITT, C. and KRESS, G.R., 2003. *Multimodal literacy.* New York: Peter Lang.

JONASSEN, D.H., 2000. Toward a design theory of problem solving. *Educational technology research and development,* **48**(4), pp. 63-85.

JUUL, J., 2003. The game, the player, the world: looking for a heart of gameness. *DIGRA Conf.* 2003.

KAFAI, Y., 1996. Learning design by making games. In: Y. KAFAI and R. MITCHEL, eds, *Constructionism in practice.* New Jersey: Lawrence Erlbaum Associates, pp. 71-96.

KAFAI, Y.B., 1995. *Minds in play: Computer game design as a context for children's learning.* New York: Routledge.

KAFAI, Y.B. and BURKE, Q., 2013. The social turn in K-12 programming: moving from computational thinking to computational participation, T. CAMP, P. TYMANN, J. DOUGHERTY and K. NAGEL, eds. In: *Proceeding of the 44th ACM technical symposium on Computer science education,* March 06-09 2013, ACM, pp. 603-608.

KAFAI, Y.B. and PEPPLER, K.A., 2012. Developing Gaming Fluencies with Scratch: Realizing Game Design as an Artistic Process. In: C. STEINKUEHLER, K. SQUIRE and S. BARAB, eds, *Games, Learning, and Society: Learning and Meaning in the Digital Age.* New York: Cambridge University Press, pp. 355-380.

KAFAI, Y.B., PEPPLER, K.A. and CHAPMAN, R.N., 2009. *The Computer Clubhouse: Constructionism and Creativity in Youth Communities. Technology, Education--Connections.* ERIC.

KALIL, T., 2013. Have Fun—Learn Something, Do Something, Make Something. In: M. HONEY and D.E. KANTER, eds, *Design, make, play: Growing the next generation of STEM Innovators.* 1 edn. New York: Routledge, pp. 12-16.

KATO, P.M., 2010. Video games in health care: Closing the gap. *Review of General Psychology,* **14**(2), pp. 113.

KATTERFELDT, E., ZEISING, A. and LUND, M., 2013. Notes on Maker Culture. In: J. WALTER-HERRMANN and C. BÜCHING, eds, *FabLab: Of Machines, Makers and Inventors.* Verlag, Blelefield: Transcript, .

KAUFMAN, J. and BEGHETTO, R., 2009. Creativity in the schools: A rapidly developing area of positive psychology. *Handbook for Positive Psychology in Schools, R.Gilman, MJ Fulong, E.Scott Huebner,* , pp. 175-201.

KAY, J., BARG, M., FEKETE, A., GREENING, T., HOLLANDS, O., KINGSTON, J.H. and CRAWFORD, K., 2000. Problem-based learning for foundation computer science courses. *Computer Science Education,* **10**(2), pp. 109-128.

KE, F., 2008. A case study of computer gaming for math: Engaged learning from gameplay? *Computers & Education,* **51**(4), pp. 1609-1620.

KELLEY, D., 1988. *The Art of Reasoning.* New York: W. W. Norton & Company.

KETELHUT, D.J., CLARKE, J. and NELSON, B.C., 2010. The development of River City, a multi-user virtual environment-based scientific inquiry curriculum: Historical and design evolutions. *Designs for learning environments of the future.* Springer, pp. 89-110.

KÖLLING, M., 2008. Using BlueJ to introduce programming. In: J. BENNEDSEN, M. CASPERSEN and M. KÖLLING, eds, *Reflections on the Teaching of Programming.* New York: Springer, pp. 98-115.

KÖLLING, M. and ROSENBERG, J., 2001. Guidelines for teaching object orientation with Java, *ACM SIGCSE Bulletin* 2001, ACM, pp. 33-36.

KRESS, G., 2003. *LITERACY IN THE NEW MEDIA AGE.* London: Routledge.

KRESS, G., 2013. *Multimodality: A social semiotic approach to contemporary communication.* London: Routledge.

LANGE, P. and ITO, M., 2010. Creative Production. In: M. ITO, S. BAUMER, M. BITTANTI, R. CODY, B. HERR-STEPHENSON, H.A. HORST, P.G. LANGE, D. MAHENDRAN, K.Z. MARTÍNEZ and C. PASCOE, eds, *Hanging out, messing around, and geeking out.* Cambridge USA: MIT Press, pp. 243-293.

LANKSHEAR, C. and KNOBEL, M., 2008. *Digital literacies: concepts, policies and practices.* New York: Peter Lang.

LAVE, J. and WENGER, E., 1991. *Situated learning: Legitimate peripheral participation.* Cambridge: Cambridge University Press.

LEE, I., MARTIN, F., DENNER, J., COULTER, B., ALLAN, W., ERICKSON, J., MALYN-SMITH, J. and WERNER, L., 2011. Computational thinking for youth in practice. *ACM Inroads,* **2**(1), pp. 32-37.

LENHART, A., KAHNE, J., MIDDAUGH, E., EVANS, C. and VITAK, J., 2008. Teens, video games, and civics.

LIN, Y., 2011. Fostering creativity through education–a conceptual framework of creative pedagogy. *Creative education,* **2**(03), pp. 149.

LIVINGSTONE, I. and HOPE, A., 2011. *Next Gen. Transforming the UK into the world's leading talent hub for the video games and visual effects industries.*

LOELIGER, J. and MCCULLOUGH, M., 2012. *Version Control with Git: Powerful tools and techniques for collaborative software development.* California: O'Reilly Media, Inc.

LUBART, T.I., 2001. Models of the creative process: Past, present and future. *Creativity Research Journal,* **13**(3-4), pp. 295-308.

MALABY, T.M., 2007. Beyond play a new approach to games. *Games and culture,* **2**(2), pp. 95-113.

MALONEY, J., RESNICK, M., RUSK, N., SILVERMAN, B. and EASTMOND, E., 2010. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE),* **10**(4), pp. 16.

MARTINEZ, S.L. and STAGER, G., 2013. *Invent to learn: Making, tinkering, and engineering in the classroom.* California: Constructing Modern Knowledge Press.

MASLOW, A.H., FRAGER, R., FADIMAN, J., MCREYNOLDS, C. and COX, R., 1970. *Motivation and personality.* New York: Harper & Row.

MAYER, R.E. and WITTROCK, M.C., 2012. Problem solving. In: D. BERLINER, R. CALFEE, P. ALEXANDER and P. WINNIE, eds, *Handbook of Educational Psychology.* 2 edn. New York: Routledge, pp. 287-303.

MCCANNELL, D., 1976. *The tourist.* New York: Schocken Books.

MCGONIGAL, J., 2012. *Reality is broken: Why games make us better and how they can change the world.* London: Vintage.

MCST, 06/02/2014, 2014-last update, Malta Robotics Olympiad in May 2014. Available: http://www.mcst.gov.mt/all/malta-robotics-olympiad-may-2014 [30/08, 2014].

MCWILLIAM, E., 2008. Unlearning how to teach. *Innovations in Education and Teaching International,* **45**(3), pp. 263-269.

MEE, 14/01/2014, 2014-last update, Co-Ed in Form 1 across all colleges from next Scholastic Year, set-up to remain as is. Available: https://education.gov.mt/en/resources/News/Pages/News%20items/CO-ED-IN-FORM-1-ACROSS-ALL-COLLEGES-FROM-NEXT-SCHOLASTIC-YEAR,-SET-UP-TO-REMAIN-AS-IS-.aspx [08/05, 2014].

MELLOU, E., 1996. Can creativity be nurtured in young children? *Early child development and care,* **119**(1), pp. 119-130.

MERRIAM, S.B., 2009. *Qualitative research: A guide to design and implementation.* San Francisco: Jossey-Bass.

MINISTER FOR EDUCATION AND EMPLOYMENT, Jan 08, 2014, 2014-last update, Tablets Pilot Project launched. Available: http://education.gov.mt/en/resources/News/Pages/News%20items/Tablets-Pilot-Project-launched.aspx [09/2014, 2014].

MINISTRY OF EDUCATION AND EMPLOYMENT, 2012. *A National Curriculum Framework for All.* Malta: Salesian Press.

MITA, 2014-last update, mScratch Competition. Available: http://mita.gov.mt/en/DigitalOutReach/mscratchcompetition/Pages/mScratch-Competition.aspx [08/30, 2014].

MITA, 2013-last update, Only Girls Allowed Competition. Available: https://www.mita.gov.mt/en/DigitalOutReach/onlygirlsallowed/Pages/Only-Girls-Allowed-Competition.aspx [08/30, 2014].

MITA, 2012-last update, The Scratch Competition. Available: http://mita.gov.mt/en/DigitalOutReach/brighterfuture/Pages/The-SCRATCH-competition.aspx [08/30, 2014].

MONTOLA, M., NUMMENMAA, T., LUCERO, A., BOBERG, M. and KORHONEN, H., 2009. Applying game achievement systems to enhance user experience in a photo sharing service, *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era* 2009, ACM, pp. 94-97.

MORREALE, P. and JOINER, D., 2011. Changing perceptions of computer science and computational thinking among high school teachers. *Journal of Computing Sciences in Colleges,* **26**(6), pp. 71-77.

MSTE, 05/2011, 2011-last update, Computing Team runners-up in GameZing 20.10 National Competition. Available: http://www.um.edu.mt/educ/mste/teaching_areas/computing/events [06/2014, 2014].

MURPHY, E., DINGWALL, R., ATKINSON, P., COFFEY, A., DELAMONT, S., LOFLAND, J. and LOFLAND, L., 2001. The ethics of ethnography. *Handbook of ethnography,* , pp. 339-351.

NACCCE, 1999. *All Our Futures: Creativity, Culture & Education: Report to the Secretary of State for Education and Employment, the Secretary of State for Culture, Media and Sport.* DfEE.

NAVARRETE, C.C., 2013. Creative thinking in digital game design and development: A case study. *Computers & Education,* **69**, pp. 320-331.

NEW LONDON GROUP, 1996. A Pedagogy of Multiliteracies: Designing Social Futures. *Harvard Educational Review,* **66**(1), pp. 60-92.

O'MARA, J. and RICHARDS, J., 2012. A blank slate: Using Gamemaker to create computer games. In: C. BEAVIS, J. O'MARA and L. MCNEICE, eds, *Digital games: Literacy in action.* Australia: Wakefield Press, pp. 57-64.

OWSTON, R., WIDEMAN, H., RONDA, N.S. and BROWN, C., 2009. Computer game development as a literacy activity. *Computers & Education,* **53**(3), pp. 977-989.

OXLAND, K., 2004. *Gameplay and design.* Essex: Pearson Education.

PAPERT, S., 1994. *The children's machine: Rethinking school in the age of the computer.* New York: Basic Books.

PAPERT, S., 1980. *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

PATTON, M.Q., 2002. *Qualitative evaluation and research methods.* London: Sage.

PELLETIER, C. and BURN, A., 2005. Making games. Developing games authoring software for educational and creative use. *CAL*, **5**.

PENG, W., 2010. Practice and experience in the application of problem-based learning in computer programming course, *Educational and Information Technology (ICEIT), 2010 International Conference on* 2010, IEEE, pp. V1-170-V1-172.

PEPPLER, K.A. and KAFAI, Y.B., 2007. What Videogame Making Can Teach Us about Literacy and Learning: Alternative Pathways into Participatory Culture. A. BABA, ed. In: *Proceedings of DiGRA 2007 Conference: Situated Play*, 24-28 September 2007, DiGRA, pp. 369-376.

PERKINS, D.N., 2013. *Knowledge as design.* London: Routledge.

PERKOVIĆ, L., SETTLE, A., HWANG, S. and JONES, J., 2010. A framework for computational thinking across the curriculum, *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* 2010, ACM, pp. 123-127.

PHILIP, R.L., 2013. Cultivating creative ecologies: creative teaching and teaching for creativity, S. FRIELICK, N. BUISSINK-SMITH, P. WYSE, J. BILLOT, J. HALLAS and E. WHITEHEAD, eds. In: *Research and Development in Higher Education: The Place of Learning and Teaching*, 1-4 July 2013, Higher Education Research and Development Society of Australasia, Inc, pp. 360-369.

PIAGET, J., 1962. *Play, Dreams and Imitation in childhood.* New York: Norton.

PRENSKY, M., 2003. *Digital game-based learning.* New York: McGraw-Hill Companies.

PRITCHARD, A. and WOOLLARD, J., 2010. *Constructivism and Social Learning.* New York: Routledge.

QIN, H., 2009. Teaching computational thinking through bioinformatics to biology students, *ACM SIGCSE Bulletin* 2009, ACM, pp. 188-191.

REEVES, B. and READ, J.L., 2009. *Total engagement: using games and virtual worlds to change the way people work and businesses compete.* New York: Harvard Business Press.

RESNICK, M., 2008. Sowing the Seeds for a More Creative Society. *Learning & Leading with Technology,* , pp. 5.

RESNICK, M., MALONEY, J., MONROY-HERNÁNDEZ, A., RUSK, N., EASTMOND, E., BRENNAN, K., MILLNER, A., ROSENBAUM, E., SILVER, J. and SILVERMAN, B., 2009. Scratch: programming for all. *Communications of the ACM,* **52**(11), pp. 60-67.

RESNICK, M., KAFAI, Y. and MAEDA, J., 2005. A networked, media-rich programming environment to enhance technological fluency at after-school centers in economically-disadvantaged communities.

RESNICK, M. and RUSK, N., 1996. The Computer Clubhouse: Preparing for life in a digital world. *IBM Systems Journal,* **35**(3.4), pp. 431-439.

RICHARDSON, L. and ADAMS ST.PIERRE, E., 2005. Writing A method of Inquiry. In: N. DENZIN and Y. LINCOLN, eds, *The Sage Handbook of Qualitative Research.* 3 edn. London:UK: Sage, pp. 959.

ROBERTSON, J., 2011. *Towards an Understanding of the Creative Process of Classroom Based Computer Game Design.* Heriot Watt University Computer Science Technical Report.

ROBERTSON, J. and HOWELLS, C., 2008. Computer game design: Opportunities for successful learning. *Computers & Education,* **50**(2), pp. 559-578.

ROBERTSON, J., NICHOLSON, K. and HOWELLS, C., n.d., n.d-last update, About Adventure Author. Available: http://judyrobertson.typepad.com/adventure_author/about-adventure-author.html2011].

ROBINSON, K., 2011. *Out of our minds: Learning to be creative.* Chichester: Capstone Publishing Ltd.

RUNCO, M.A., 1992. Children's divergent thinking and creative ideation. *Developmental Review,* **12**(3), pp. 233-264.

RUNCO, M.A. and RICHARDS, R., 1997. *Eminent creativity, everyday creativity, and health.* Greenwich: Ablex Publishing Ltd.

RUSHKOFF, D., 2010. *Program or be programmed: Ten commands for a digital age.* Berkeley: Soft Skull Press.

SALEN, K., 2007. Gaming literacies: A game design study in action. *Journal of Educational Multimedia and Hypermedia,* **16**(3), pp. 301-322.

SALEN, K. and ZIMMERMAN, E., 2003. *Rules of play: Game design fundamentals.* Cambridge: MIT press.

SCHÖN, D.A., 1983. *The reflective practitioner: How professionals think in action.* Basic books.

SELBY, C., 2011. Four approaches to teaching programming, *Learning Media and Technology : a doctoral research conference*, 4 July 2011.

SHAHEEN, R., 2010. Creativity and education. *Creative Education,* **1**(03), pp. 166.

SHARP, C., 2004. Developing Young Children's Creativity: what can we learn from research. *Topic,* **32**, pp. 5-12.

SHORE, J., 2008. *The art of agile development.* California: O'Reilly.

SIMÕES, J., REDONDO, R.D. and VILAS, A.F., 2013. A social gamification framework for a K-6 learning platform. *Computers in Human Behavior,* **29**(2), pp. 345-353.

SIMONS, H., 2009. *Case study research in practice.* London: Sage Publications Ltd.

SLACK, M., 2012-last update, President Obama Launches a Marshmallow Cannon. Available: http://www.whitehouse.gov/blog/2012/02/07/president-obama-launches-marshmallow-cannon2014].

SOTAMAA, O., 2010. Achievement unlocked: rethinking gaming capital. *Games as Services,* **73**.

SPACKS, P.M., 1986. *Gossip.* Chicago: University of Chicago Press.

SQUIRE, K., 2011. *Video Games and Learning: Teaching and Participatory Culture in the Digital Age.* New York: Teachers College Press.

STAGNO-NAVARRA, K., 2011-last update, Schemes for digital gaming fling doors open to US$45 billion global market. Available: http://www.maltatoday.com.mt/en/businessdetails/business/technology/Schemes-for-digital-gaming-fling-doors-open-to-US-45-billion-global-market [03/17, 2012].

STEINKUEHLER, C., 2007. Massively multiplayer online gaming as a constellation of literacy practices. *E-learning,* **4**(3), pp. 297-318.

STEINKUEHLER, C., 2006. The mangle of play. *Games and Culture,* **1**(3), pp. 199-213.

STEINKUEHLER, C., SQUIRE, K. and BARAB, S., 2012. *Games, learning, and society: Learning and meaning in the digital age.* New York: Cambridge University Press.

STEPHENSON, W., 1967. *The play theory of mass communication.* Chicago: University of Chicago Press.

STEVENS, R., SATWICZ, T. and MCCARTHY, L., 2008. In-game, in-room, in-world: Reconnecting video game play to the rest of kids' lives. *The ecology of games: Connecting youth, games, and learning,* **9**, pp. 41-66.

SUITS, B., 1978. *Grasshopper: Games, Life and Utopia.* Toronto: University of Toronto Press.

SURMAN, D., 2010. Everyday Hacks:Why Cheating Matters. In: R. CATLOW, M.A. GARRETT and C. MORGANA, eds, *Artists Re: Thinking Games.* Great Britian: FACT (Foundation for Art and Creative Technology), pp. 74-78.

SUTTON-SMITH, B., 1997. *The Ambiguity of Play*. Cambridge: Harvard University Press.

SWANBORN, P., 2010. *Case study research: what, why and how?* London: Sage Publications Ltd.

TAYLOR, T.L., 2006. *Play Between Worlds: Exploring Online Game Culture.* Cambridge: MIT Press.

TEGANO, D., MORAN, J.D. and SAWYERS, K., 1991. *Creativity in Early Childhood Classrooms.* New York: NEA Early Childhood Education Series.

THOMAS, D. and BROWN, J.S., 2011a. *Learning for a world of constant change: Homo sapiens, homo faber & homo Ludens revisited.* 7th Glion Colloquium by JSB,.

THOMAS, D. and BROWN, J.S., 2011b. *A new culture of learning: Cultivating the imagination for a world of constant change.* Lexington: CreateSpace.

ULICSAK, M. and WILLIAMSON, B., 2010. *Computer games and learning.* Bristol: Futurelab.

USHER, W., 08/09/2013, 2013-last update, GTA 5 cost $265 Million to Develop and Market. Available: http://www.cinemablend.com/games/GTA-5-Cost-265-Million-Develop-Market-58922.html [09/05, 2014].

VAN ECK, R., 2006. Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE review,* **41**(2), pp. 16.

VOS, N., VAN DER MEIJDEN, H. and DENESSEN, E., 2011. Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education,* **56**(1), pp. 127-137.

VYGOTSKY, L.S., 1978. *Mind in society: The development of higher psychological processes.* Cambridge: Harvard university press.

WALLAS, G., 1926. *The art of thought.* Kent: Solis Press.

WALSH, C., 2010. Systems-based literacy practices: Digital games research, gameplay and design. *Australian Journal of Language and Literacy,* **33**(1), pp. 24-40.

WALSH, C. and APPERLEY, T., 2009. Gaming capital: Rethinking literacy, *Changing Climates: Education for sustainable futures. Proceedings of the AARE 2008 International Education Research Conference,* 30 Nov - 4 Dec 2009, Queensland University of Technology.

WELLER, M.P., DO, E.Y. and GROSS, M.D., 2008. Escape machine: teaching computational thinking with a tangible state machine game, *Proceedings of the 7th international conference on Interaction design and children* 2008, ACM, pp. 282-289.

WERNER, L., DENNER, J., CAMPE, S. and KAWAMOTO, D.C., 2012. The fairy performance assessment: Measuring computational thinking in middle school, *Proceedings of the 43rd ACM technical symposium on Computer Science Education* 2012, ACM, pp. 215-220.

WERTSCH, J.V. and TULVISTE, P., 1992. L. S. Vygotsky and contemporary developmental psychology. *Developmental Psychology,* **28**(4), pp. 548.

WHITE, S., CARTER, J., JAMIESON, S., EFFORD, N. and JENKINS, T., 2007. TOPS-Collaboration and competition to stretch our most able programming novices, *Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual* 2007, IEEE, pp. S2J-20-S2J-24.

WHITTON, N., 2010. *Learning with digital games: a practical guide to engaging students in higher education.* New York: Routledge.

WILLETT, R., ROBINSON, M. and MARSH, J., 2009. *Play, creativity and digital cultures.* New York: Routledge.

WILLIAMSON, B., 2009. *Computer games, schools and young people A report for educators on using games for learning.* Bristol: Futurelab.

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

WING, J.M., 2008. Computational thinking and thinking about computing. *Philosophical transactions.Series A, Mathematical, physical, and engineering sciences,* **366**(1881), pp. 3717-3725.

WOLZ, U., OUYANG, Y. and LEUTENEGGER, S., 2011. Scratching the subject surface: infusing computing into K-12 curriculum, *Proceedings of the 42nd ACM technical symposium on Computer science education* 2011, ACM, pp. 489-490.

WU, M.L. and RICHARDS, K., 2011. Facilitating computational thinking through game design. *Edutainment Technologies. Educational Games and Virtual Reality/Augmented Reality Applications.* Springer, pp. 220-227.

YADAV, A., ZHOU, N., MAYFIELD, C., HAMBRUSCH, S. and KORB, J.T., 2011. Introducing computational thinking in education courses, *Proceedings of the 42nd ACM technical symposium on Computer science education* 2011, ACM, pp. 465-470.

YEE LENG, E., ZAH BTE WAN ALI, WAN and BAKI, R., 2010. Computer games development experience and appreciative learning approach for creative process enhancement. *Computers & Education,* **55**(3), pp. 1131-1144.

YIN, R.K., 2012. *Applications of case study research.* 3rd edn. London: Sage Publications Ltd.

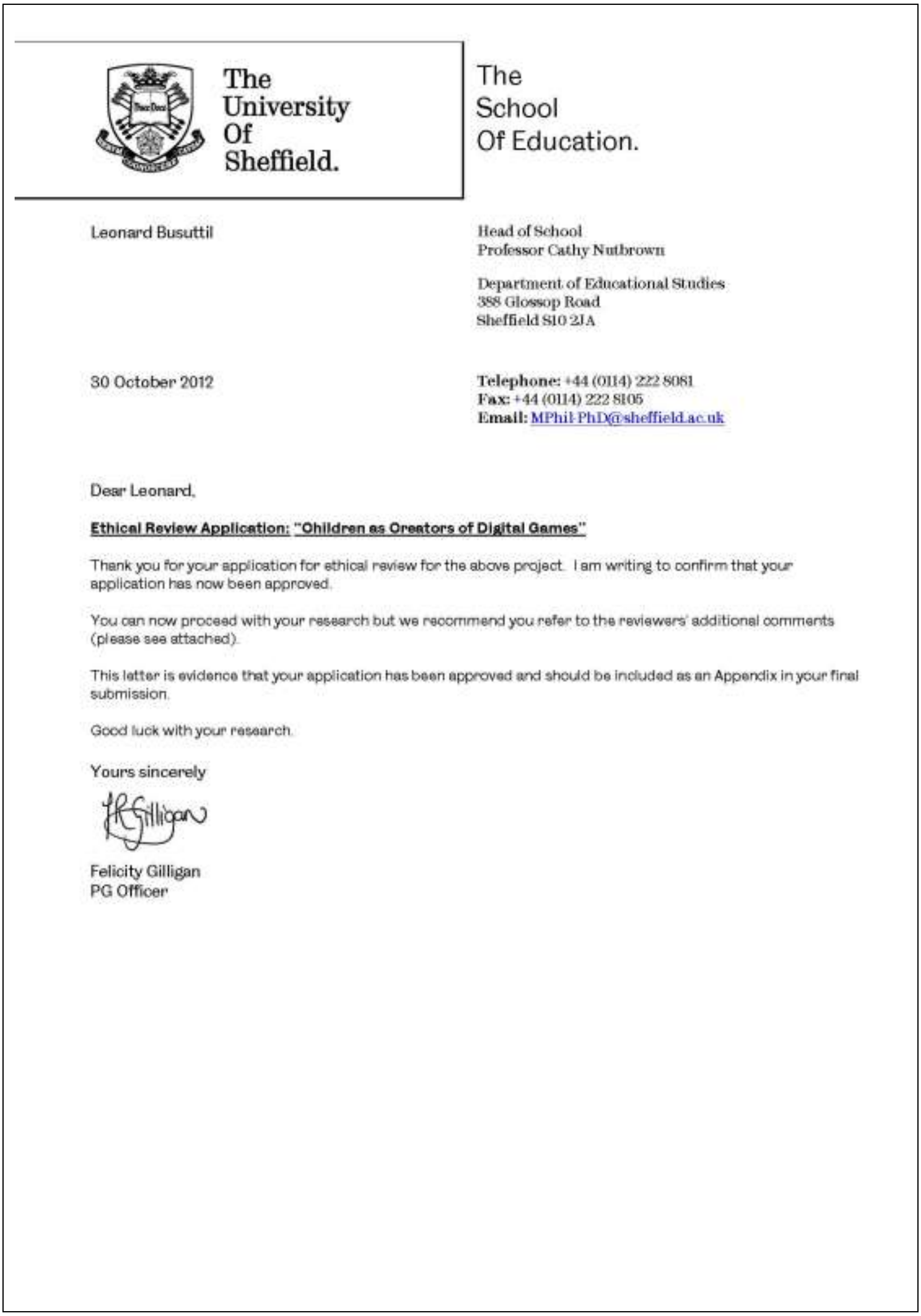
YIN, R.K., 2008. *Case study research: Design and methods.* Sage Publications, Incorporated.

ZIMMERMAN, E., 2007. Gaming literacy. *Harvard Interactive Media Review,* **1**.

ZITTRAIN, J.L., 2008. *The future of the Internet and how to stop it.* Virginia.: USA: Caravan Books.

ZYDA, M., 2005. From visual simulation to virtual reality to games. *Computer,* **38**(9), pp. 25-32.

Appendix B Ethical Review



The University Of Sheffield.

The School Of Education.

Leonard Busuttil

Head of School
Professor Cathy Nutbrown

Department of Educational Studies
388 Glossop Road
Sheffield S10 2JA

30 October 2012

Telephone: +44 (0114) 222 8081
Fax: +44 (0114) 222 8105
Email: MPhil-PhD@sheffield.ac.uk

Dear Leonard,

Ethical Review Application: "Children as Creators of Digital Games"

Thank you for your application for ethical review for the above project. I am writing to confirm that your application has now been approved.

You can now proceed with your research but we recommend you refer to the reviewers' additional comments (please see attached).

This letter is evidence that your application has been approved and should be included as an Appendix in your final submission.

Good luck with your research.

Yours sincerely

Felicity Gilligan
PG Officer