



OULUN YLIOPISTO
UNIVERSITY of OULU

Getting a Child's Perspective on Learning Apps That Teach Programming

University of Oulu
Department of Information Processing
Sciences
Master's Thesis
Zachary Pampel
24/03/2017

Abstract

The purpose of this study is to gain an understanding of how children view learning games and learning applications and games that teach programming. It is rare to find studies that have first-hand experience with children's views on the learning of programming through learning apps and games. This study looks to fill that gap by asking children directly through a questionnaire given to a class in Finland.

Overall, the study found that the students in this study found programming favorable. Most students enjoyed programming and liked programming for a number of reasons including: to do something different, they like electronics, that there isn't a lot of rules, can be fun, express creativity, can see what you have done on the computer, can use what you have done, like the feelings of writing something that works, and the challenge it presents. Some of the students wrote they disliked programming because it can be boring, can be too hard, and can be confusing.

Many of the students liked learning apps because they offer a different way of learning than what they usually do. They seem to like the freedom that comes from using learning apps and which differs from learning from the teacher. They liked using the devices that learning apps were on. Some of the students also said that learning apps can be fun and entertaining. Some of the things the students didn't like about learning apps was technical issues, such as apps that crash a lot. That the learning apps sometimes don't help. That the learning apps can be confusing. That the apps can have too many ads. Some of the students also stated that they can be boring. And some said that they can be too easy. Some of the students' suggestions on improving in learning apps were: make them harder, to use them more, make them more fun, have less ads, learn actual code syntax, and to include language options.

Keywords

Programming, Learning Apps, Game-based Learning, Serious Games, Children's Education

Supervisor

Professor, Marianne Kinnula

Foreword

First, I want to thank Samantha, who allowed me to use her classroom to perform my study. I also want to thank all the children in Samantha's class for their great responses. I want to thank my supervisor, Dr. Marianne Kinnula, for her help with my thesis. I want to thank my wife, Rachel, for her motivation. And I want to thank the University of Oulu for my master's education and allowing the opportunity to conduct a master's thesis.

I also hope this study finds use by those interested in designing learning programming apps and games for children, and by educators and parents selecting these apps for their students.

Zachary Pampel

Oulu, March 24, 2017

Contents

Abstract	2
Foreword	3
Contents.....	4
1 Introduction	5
2 Literature review	6
21 Overview of Game-based Learning/Serious Games.....	6
211 Overview of Game-based Systems.....	10
212 Effectiveness of Educational Games	11
22 Designing Learning Apps/Games for Children	15
23 Learning of Programming.....	18
231 Programming Learning Tools.....	18
232 Student’s Perceptions of Programming	20
24 How to Evaluate Learning Games/Serious Games/Educational Games with Children.....	22
25 Summary	26
3 Research Methods	28
31 Questionnaire with Children.....	28
311 Qualitative	28
312 Quantitative	28
313 In Practice	29
32 Analysis.....	31
321 Content Analysis.....	31
322 In practice	31
4 Analysis	32
41 Qualitative Data	32
42 Quantitative Data	38
5 Discussion	42
6 Conclusion.....	44
References.....	45
Appendix A. Questionnaire used in this study.....	50

1. Introduction

The purpose of this study is to gain an understanding of how children view learning apps and learning games that teach programming. Studies are rare that include first-hand experience with children's views on learning programming through learning apps and games. This study looks to fill that gap by asking children about their experience and reactions directly. This study also targets Finnish students. A questionnaire was given to a class in the international school in Oulu, after they were asked to play a programming learning application. The first part of the study includes a literature review of game-based learning, designing learning apps/games for children, the learning of programming, and how to evaluate learning games, as well as the articulation of the research methods. Following this, I explain meeting with the international school class in Oulu and observing them playing a learning game (Scratch Jr.) that teaches programming. They were given a questionnaire after playing the game to get their thoughts on programming in general, learning apps/games, the most important aspects of learning apps, and the learning game they played. The questionnaire collected both qualitative and quantitative data from the children. This data will be analyzed using content analysis then compared with the results of previous literature. Finally, the conclusion will explore the limitations of the study and possible future research. The research question and goals are presented below:

Research question: What are children's perceptions of programming and learning apps that teach programming?

Goals:

- To get a better understanding on what children want/expect from learning applications.
- To get a child's perspective on the value of learning apps and games.
- To learn what children most like and dislike about learning applications.
- To gain insight on the children's perspective on learning programming.
- To learn what are the most important aspects of programming learning apps from children.

2. Literature review

The first part of the study researched and lays an overview for: game based learning/serious games, designing learning apps/games for children, learning of programming, how to evaluate learning games/serious games/educational game, and how to design learning games for programming. Papers were chosen based on relevance, citation count and related keywords. Some search strings included: “serious games”, “game-based learning”, “design learning applications for children”, “learning of programming”, “how to evaluate learning games”, “effectiveness of game-based learning”, etc.

2.1 Overview of Game-based Learning/Serious Games

The merit of video games has been a hot topic of debate since their creation. Game-based learning can be used to motivate and promote learning by using game-based logic, thinking, and techniques. In Prensky, M. (2001) *Digital Natives, Digital Immigrants*, Prensky advocates that *"Our students have changed radically. Today's students are no longer the people our educational system was designed to teach."* (Prensky, 2001). He advocates that our current educational system is out of date and is in need of change. He says educational video games can be used a tool to bridge this gap between digital natives and digital immigrants, that those who have been brought in the digital age and those who have adapted to newer technologies (Prensky, 2001). Game-based learning can be a powerful tool used to teach subjects, such as programming, however the extent of the effectiveness of game based learning continues to be discussed. Game-based learning can be used as an additional teaching aid or as a standalone to present new material to the student. Most studies have found game-based learning to be effective at teaching programming (Egenfeldt-Nielsen, 2007).

According to Fullerton (2014) a **game** can be defined as: “A closed, formal system that engages players in structured conflict and resolves its uncertainty in an unequal outcome” (Fullerton, 2014). However, the last point stated is not always true for games that may be cooperative or games designed for learning purposes that may not be competitive in nature (Fullerton, 2014). Game-based systems in the context of this paper can be therefore defined as *systems that use game elements but may not contain all aspects of games*. This overarching term will be discussed in more detail in the next section. For now, there are many sub-terms for the various aspects of game-based systems in a learning situation, which mainly fall under game-based learning and serious games categories. The actual definitions of game-based learning and serious games, and the following differences between the two terms, are important, though they can often can be used interchangeably. The primary difference is one of hierarchy. Serious games are an umbrella term within the realm of game-based systems. Susi et al. (2007) in “Serious Games – An Overview” briefly describes game-based learning, serious games, and related concepts in a way I find helpful. There are many terms in the domain of serious games which continue this trend of overlapping use, for example: game-based learning, digital game-based learning, edutainment, e-learning (Susi et al., 2007). For the purposes of this thesis, these terms are defined below.

Serious games can be loosely defined as games for the purposes other than entertainment. Computer video games for purposes other than entertainment have existed for quite some time but have grown increasingly more popular in recent years (Susi et al., 2007, p.2).

Game-based learning can be viewed as a branch of serious games where the applications have defined learning outcomes. As discussed, this term is often used synonymously with serious games. Game-based learning can be used for activities such as training, as training typically has a clear and expected outcome (Susi et al., 2007, p.2).

Digital game-based learning is very like game-based learning, but restricts the domain to digital games. So, digital game-based learning games can be understood as serious games only in digital format and may have more defined expected outcomes (Susi et al., 2007, p.2).

E-learning can be defined as a general concept that incorporates many different terms such as computer-based learning, interactive technology, computer-enhanced learning, and distance learning. So, E-learning can be viewed as more formal learning that takes place virtually (Susi et al., 2007, p.2).

Edutainment can be defined as education through entertainment. Edutainment was a popular trend in the 1990s, but was not as successful as many saw these games as boring or just “drill and practice” learning games. Edutainment can be defined more broadly as any education that also entertains, however, edutainment is more commonly understood as video games with educational goals (Susi et al., 2007, p.2).

Another definition that should be discuss in this domain is **gamification**. A definition from Deterding et al. (2011) is: The use of design elements characteristic for games in non-game context, explained in detail: *“To summarize: “Gamification” refers to the use (rather than the extension) of design (rather than game-based technology or other gamerelated practices) • elements (rather than full-fledged games) characteristic for games (rather than play or playfulness) in non-game contexts (regardless of specific usage intentions, contexts, or media of implementation)”*.

And finally, in this study **learning application** can be seen a blanket term for an application, program, or game that its primary purpose is to educate the user.

Susi et al. (2007) discusses how there are many different definitions of serious games from literature but in general there is a consensus that serious games are games that are used for purposes other than entertainment. The Table below from Susi et al. (2007) shows the differences between games for entertainment and serious games:

Table 1 Differences between entertainment games and serious games.

Table 1. Differences between entertainment games and serious games.

	Serious games	Entertainment games
Task vs. rich experience	Problem solving in focus	Rich experiences preferred
Focus	Important elements of learning,	To have fun
Simulations	Assumptions necessary for workable simulations	Simplified simulation processes
Communication	Should reflect natural (i.e., non-perfect) communication	Communication is often perfect

(Susi et al., 2007, p. 6)

Table 1 highlights the major differences between entertainment games and serious games

and clearly shows how they are different in certain aspects. Serious games are focused on learning while entertainment games are focused on having fun and aspects of each of the game types are changed to meet the focus of the game. There are also positive and negative effects of these games. One obvious advantage of digital serious games or related terms (e-learning, edutainment, game-based learning, digital game-based learning, etc.) is that learners can experience environments that would be impossible in the real world, for reasons such as the restrictions of cost, time, safety, etc. Serious games are also argued to have positive effects on players' development of multiple different skills according to Susi (2007). Below is a Table from Susi et al. (2007) that shows the reported effects of serious games from past studies:

Table 2 Overview of examples on reported effects.

Table 2. Overview of examples on reported effects.

	Motor skill/ spatial	Educational/ Informational	Social	Physiological
Backlund et al. (2006)	X			
Enochsson et al. (2004)	X			
Guy et al. (2005)	X			
Radford (2000)	X			
De Lisi and Wolford (2002)	X	X		
Navarro et al. (2003)		X		
Squire and Steinkuehler (2005)		X	X	
Baldaro et al. (2004)			X	X
Durkin and Barber (2002)			X	
Hong and Liu (2003)		X		

(Susi et al., 2007, p. 9)

Susi et al. (2007) claims that, despite the finding shown above in Table 2, there is no conclusive answer to the benefits and potential problems of games and game play. So, from past studies we can see some of the benefits and negatives of serious games but it is hard to make strong conclusions from these studies as there are many different results. Susi et al. (2007) goes on to say that, according to Van Eck (2006), the proponents of digital game-based learning have effectively conveyed the message that educational games are beneficial, so now proponents of digital game-based learning need to take the message further in the future. He says that new research should be focused on explaining why games are engaging and effective and that there needs to practical guidance on how to implement digital game-based learning and learning processes to maximize their learning protentional. Van Eck argues games are effective “*not because of what they are, but because of what they embody and what learners are doing as they play a game*” (*ibid.*, p.18).” (Susi et al., 2007) The argument can be made that one reason why games are effective is because the learning in the game takes place in a context that is meaningful within the context of the game. And it can be argued that learning in a meaningful and relevant context is more effective than learning undertaken outside that context. There is also debate as to whether games should take place in a similar setting to school and if that has an impact on the amount of learning the game can achieve (Susi et al., 2007). Though educational games are, overall, argued as beneficial in the literature, it can still be hard to determine which games are the most beneficial and why (Susi et al., 2007).

Squire (2003), states from Malone's (1981) study, the characteristics that make video games fun are: "challenge, fantasy, and curiosity"; and the aspects educational programs should include are: "using surveys, interviews, and observations". Malone (1981) made guidelines on how to create enjoyable educational programs and argues educational programs should also include:

- Clear goals that the students can find to be meaningful
- Goals and scoring can give good feedback to the students of their progress
- Difficulty setting allows for learners at different skill levels to allows the program to be better tailored to the student
- Random elements
- A meaningful fantasy element related to the game skills (as cited in Squire, 2003)

Squire (2003), states that video games have evolved quite a bit over the years, in terms of the many different genres, as well as emphasizing that increasing levels of immersion (the ability to 'lose oneself' while playing) sets good games apart. Squire states that video game design is much more complex than in the past and therefore provides a much richer experience than games like Pac-Man. The many different genres of video games today include action games, simulations, strategy, role playing, sports, puzzles, adventure, etc. Squire (2003) argues that good video design throughout these genres of video games is what immerses users in these digital worlds.

Squire (2003) describes the various forms and genres of games being used in education:

- **Drill and practice** games that are used for learning by enriching factual recall exercises in a playful way.
- **Simulation** games can be used to simplify complex systems. Low-fidelity, or hi-fidelity simulations can be used according to the type of lesson to be taught.

Drill and practice games have been used historically because they can more easily be integrated in traditional curriculum and can be used as just an extension of traditional teaching. Drill and practice can be used by students to recall information they had already learning in the classroom but new meaningful learning can be harder to accomplish using this technique. Simulation games, on the other hand, can be used to teach through a game in an environment that can be similar to reality. They split into hi-fidelity and lo-fidelity systems. A hi-fidelity can be used to model very aspect of a real-life system, for example training pilots. A hi-fidelity simulation wants to be as close to the actual system as possible which can be costly but produce good results. A low fidelity system is generally a simplified model of a system to highlight the key components of a system. Low-fidelity are often used when there is an emphasis on developing a conceptual understanding, allowing students to interact with a simplified complex system. A few lo-fidelity examples are: the laws of physics, ecosystems (Sim Earth), or politics. (Squire, 2003)

Squire (2003), states many ways that simulations and edutainment video games can be beneficial to learners:

- Students can manipulate variables that may not be possible in real life examples.
- Students can experience a phenomenon from a different perspective.
- Students can observe a systems behavior over time and relate that to a real-life example.
- Students can propose questions to a system, then test the system under different conditions.

- Students are given the opportunity to visualize a system.
- Students can compare simulations with their previous understanding of a system.

2.1.1 Overview of Game-based Systems

		Authoring				Visualisation				Play									
		2006 :: RoboCode	2010 :: Alice2	2010 :: Greenfoot	2010 :: Scratch	2003 :: Jerroo	2006 :: objectKarel++	2007 :: C-Sheep	2008 :: Bomberman	2006 :: TurtleT	2007 :: Catacombs	2007 :: Saving Sera	2009 :: EleMental	2009 :: Resource Craft	2009 :: YWU's Castle	2010 :: Prog & Play	2010 :: Robozzle	2011 :: Lightbot	2011 :: PlayLOGO 3D
Style of Programming	procedural							X	X	X				X			X	X	X
	functional		X		X									X					
	object-based	X		X		X	X				X	X	X	X	X	multiple			
Programming Language	mainstream language	X	X					X	X	X	X	X	X	X	X	X			
	bespoke language		X	X		X	X										X	X	X
Concepts Taught	conditional		X	X	X	X	X	X		X	X	X	X	X	X	X	X		
	for			X	X	X	X	X		X	X	X	X	X	X	X			
	while		X	X	X	X	X	X		X	X			X	X				
	variable		X	X	X	X	X	X		X			X	X	X	X			X
	parameter		X	X	X	X	X	X		X			X	X	X	X			X
	function / method		X	X	X	X	X	X		X			X	X	X	X	X	X	X
	user-defined data type																		
	recursion							X					X				X	X	
collections / array	Java language	X	Java language			X		X	C language				X	X					
Code Representation	text	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X
	pictures																X	X	
Program Construction	typing code	X	X			X	X	X	X	X	X	X	X	X	X	X			X
	assembling graphical objects		X	X													X	X	
	selecting / form filling						X			X									
Microworld Metaphor	general (changable)		X	X	X														
	specific	X				X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figure 1: Overview of Existing Game-Based Systems for Programming Education

Figure 1 Overview of Existing Game-Based Systems for Programming Education

(Li and Watson, 2011)

Li and Watson (2011) identify existing game-based systems for teaching programming shown above in Figure 1. They categorize the game-based system for programming education into three categories: authoring, visualization, and play (Li and Watson, 2011). Authoring uses computer game development as the student's primary learning activity within the application. This is a constructivist technique that allows the students to learn by creating. They wrote that this technique has had success in previous studies and students demonstrate a greater understanding of programming principles, self-belief in their abilities, higher satisfaction and higher retention rate of material compared to traditional learning techniques of lectures and exercises. Scratch is a popular and good example of authoring developed by MIT and is used in the international school in Oulu. Authoring can be seen as sugar coating the learning process of programming but students will still encounter similar difficulties found in traditional programming and learn programming principles in their work. Students also are unable to see the visual effects of code execution using this technique. (Li and Watson, 2011)

Another technique is the play-based approach. With this approach student learning is attained by conveying programming principles through direct gameplay. The game can include missions which are related to specific game concepts. The students learn the programming concepts by creating strategies to complete the tasks and can see their execution of their solution in the game. The writing of actual code is less emphasized in this method and may be used only for completing a function or to specify commands to control a character through missions. This method is usually targeted at specific programming concepts, such as loops or functions, rather than full programming languages. Lightbot is an application that uses this technique. Syntax, or proper ‘grammar’ in coding, is also not as important within this teaching technique. (Li and Watson, 2011)

The third and final method is the visualization-based approach. This approach is in between authoring-based and play-based methods. Game construction is not included and neither are game elements such as levels, scoring, story, etc. Visualization-based game systems use micro-worlds for concept visualization and try to demonstrate code execution in a visual context. This method allows for a visual representation of code written to get a better understanding of what the code is doing and is often used as a teaching aid for higher levels of programming learning. (Li and Watson, 2011)

These three techniques encompass the majority of variations between the large number of learning apps for programming. They also span the wide range of learning application types in general. A serious game can be play-based or an edutainment game can be authoring. The terms explored in the last two sections therefore offer a foundation to approaching educational game systems and their characteristics, which will enable readers to begin the process of selection and evaluation of the best options for learning programming.

2.1.2 Effectiveness of Educational Games

Egenfeldt-Nielsen (2007) compared many past studies on the effectiveness of educational computer games. He states that the results from studies that look at the learning outcome from educational computer games show optimistic trends. However, he also warns that the results should be examined carefully, because many studies have several flaws such as no control group, short exposure time, research bias, no integration with prior research, etc. *“Overall, this undermines the strength of each study -- the incremental learning process within the research field is as of yet still weak (Egenfeldt-Nielsen, 2007, p.4).”* Below in Figure 2, a range of studies that investigated the learning outcome of educational computer games are summarized.

Author(s)	Year	Genre	N	Subject	Results
Levin	(1981)	Action	-	Math	Computer games are motivating, engaging and ultimately successful in teaching children the planned math concepts. Computer games may be especially suitable for teaching different ways of approaching math that caters for individual differences.
Dowey	(1987)	Puzzle	203	Dental health	Children learn best from a combination of teaching and computer games but although they learn about dental hygiene this does not transfer into change of everyday practice.
McMullen	(1987)	-	37	Science	The drill-and-practice computer game was not found to have any effect on the learning, neither short-term nor long-term. However the students playing the computer game indicated that they thought they had learned more.
Jolicoeur & Berger	(1998a; 1998b)			Fractions Spelling	You learn from computer games, but educational software is more effective.
Wiebe & Martin	(1994)	Adventure	109	Geography	They find that there is no difference in learning geography facts and attitudes between computer games and teaching activities not on a computer.
Sedighian and Sedighian	(1996)	Strategy	200	Math	The learning outcome is critically affected by teachers' integration of computer games and traditional teaching, but computer games prove highly effective.
Betz	(1995)	Strategy	24	Engineer	Finds that computer games increase motivation and learning
Thomas et al.	(1997)	Adventure	211	Sex education	Students learn from playing the computer game both on specific knowledge items and in self-efficacy.
Brown et al.	(1997)	Action	59	Diabetes	The study finds that children can learn about diabetes from computer games changing everyday habits.
Klawe	(1998)	Adventure	200	Math	Computer games are effective in teaching students about math.
Adams	(1998)	Strategy	46	Urban geography	Computer games increase motivation and teach students about the role of urban planners (affective learning)
Noble et al.	2000	Action	101	Drug education	Students taught by the computer games, found the experience motivating and wanted to play the computer game again.
Turnin et al.	(2000)	-	2000	Eating habits	Computer games can teach students about eating habits and lead to significant change in everyday habits.
Feng & Caleo	(2000)	-	47	Spelling and math	Children that played computer games learned better than peers not using computer games, mostly in spelling.
Becker	(2001)	Action	-	Program.	The study testifies to the increased motivation in connection with computer games. Games are found to be more effective and motivating than traditional teaching
Lieberman	(2001)	Action		Asthma, diabetes,	A review of a number of research projects that support that you can learn from computer games.
Rosas et al.	(2003)	Action	1274	Reading and maths	Computer games increase motivation, and there is a transfer of competence in technology from using the computer game.
McFarlane et al.	2002	-	-	All subjects	The study finds that teachers in general are sceptical towards the learning of content with computer games. However the learning of general skills was appreciated.
Gander	(2002)	Strategy	29	Program.	The study finds that computer games are effective for especially teaching specific knowledge.
Squire et al.	(2004)	Simulation	96	Physics	Students using the simulation game performed better compared to the control group.

Figure 2 An overview of the studies into the effectiveness of educational use of computer games (Egenfeldt-Nielsen, 2007, pp.4-6)

In Figure 2 for most of the studies they used computer games as the only teaching method. The studies explore whether computer games can be a practical supplement to traditional teaching approaches. Overall, the current findings are positive, but because of the methodological flaws previously discussed, anything more than the affirmation that you can learn from computer games is unsupported. There is little substantiated evidence to argue for a replacement of traditional methods from these studies alone. Egenfeldt-Nielsen (2007) does state that there is overwhelming support for the potential of learning from computer games and that most students like educational games. The studies so far do not ask the hard questions about educational use of computer games, such as comparing games to other teaching methods to determine concrete differences and determine if educational games are worth the initial efforts of setup. Most of the studies were one-off studies and had weak relationships to previous research. Egenfeldt-Nielsen (2007) argues that we need “raise the bar” for the use of educational computer games and we should ask in what circumstances do we learn and how do computer games compare to other learning practices. Egenfeldt-Nielsen (2007) states: “*The real question is what computer games offer that set them aside from existing educational practice.*” These concerns are taken into account in the design of this thesis research within the limited resources of the study.

Egenfeldt-Nielsen goes on to show different ways to look at the educational use of games and how they differ. In Figure 3, Egenfeldt-Nielsen (2007) illustrates how different generations view the educational use of games and learning theories across different areas and show the historical progression of learning theories and the connection with educational games.

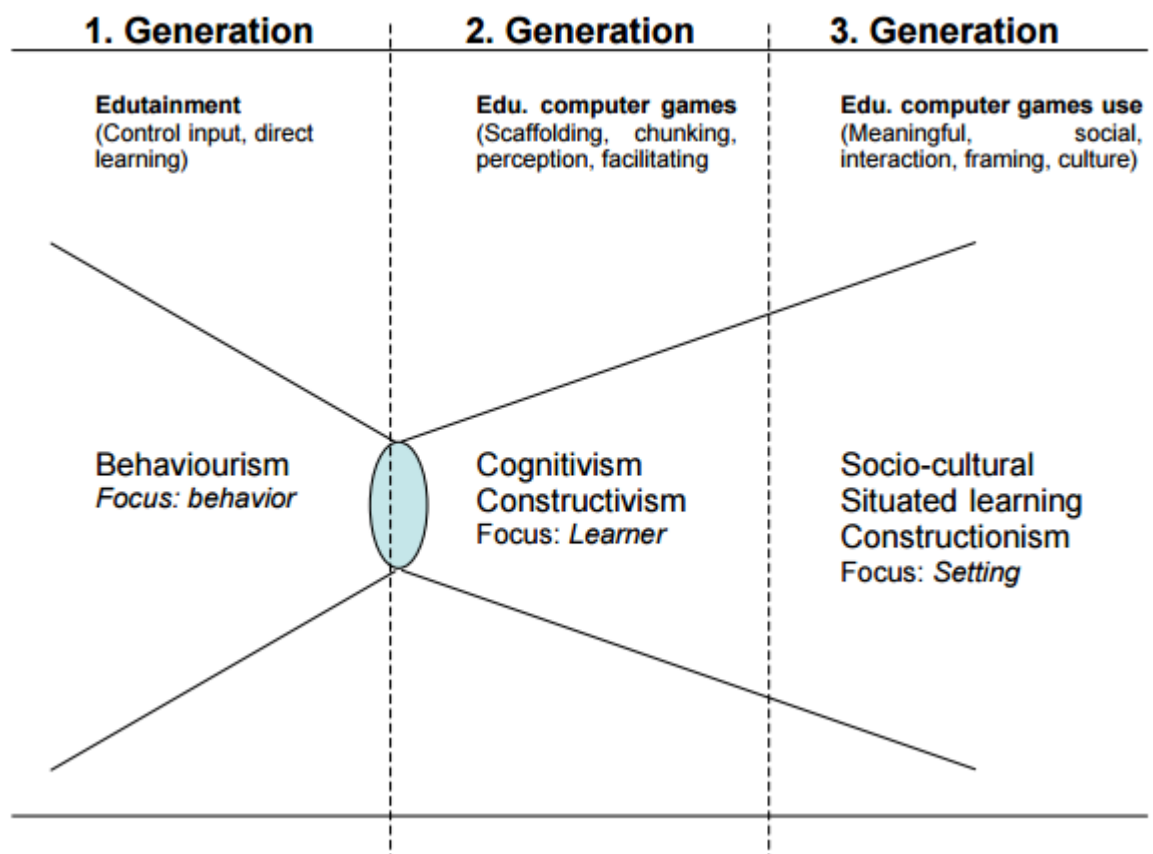


Figure 3 The model shows the different generations characteristics, and how they emphasize different learning theories. (Egenfeldt-Nielsen, 2007, pp.8)

The first generation focuses on learning by means of externally changing the behavior of students, called behaviorism. This theory claims that learning is accomplished by the practice of skills and content through reinforcement and conditioning. This theory claims that you will learn with enough practice and repetition of skills. This is related to process of edutainment, which assumes learning occurs when you unreflectively practice skills enough times. The cognitivist approach used by the second generation criticizes the automatic relation between stimuli and response described in behaviorism and places the learner as the center of attention. The cognitivist approach argues that learners have underlying “schematas” that represents what they have learned and that these schematas are developed differently for everyone (Egenfeldt-Nielsen, 2007, pp.8). This approach argues that focus on behavior is one sided, neglecting other important aspects of learning, especially cognitive structures behind the responses, which is argued to be crucial for gaining motivation and meaningful learning. This motivation can be attained using computer games for educational purposes. Interaction with this form media allows for the user’s progression at their own speed and ability. This approach also focuses on gaining meta-skills such as problem-solving, analyzing, perceiving, and spatial ability, etc. (Egenfeldt-Nielsen, 2007, p.8-9)

The third generation focuses on social aspects and communities within games. This stage is about applying what is learned in games to outside of the game and doesn’t focus on specific games, but looks at the bigger purpose of computer games within education. Egenfeldt-Nielsen explains the theories of this generation and maintains that constructionism is the bridge between the second and third generation as the approach focuses on the immediate learning process while still taking the setting into account. Construction of knowledge in a social context is important in the third generation. Egenfeldt-Nielsen argues that instead of conceiving content, skills, and attitudes coming from the user, knowledge is transferred to tools, communities, and culture. Learning takes place by participating in communities and negotiating what is considered knowledge, attitudes, and skills. The third generation focuses on providing a social context that helps the user ask the right questions and go to the right places. The teacher becomes more of a facilitator rather than trying directly transfer knowledge. (Egenfeldt-Nielsen, 2007)

Egenfeldt-Nielsen (2007) states that we need to take a broad learning approach with computer games that includes several layers. He stresses that the beliefs of each generation are carried over to the following generations, though not to the same degree. Factors of behaviorism are still in the next-generation, but they are implemented on a broader scale and not as specific. Each approach offers insights on the use of game-based systems. Utilizing these insights can offer better understanding of the educational use of computer games and lead to the further innovation within the market of educational games (Egenfeldt-Nielsen, 2007). In summary, the actual definition of serious games and/or game-based learning is often up for debate, but can usually be defined as games used for something other than entertainment. There are several different game-based systems that have their own benefits and disadvantages, though there is little hard evidence of the comparative effectiveness of these systems. A worthwhile study of educational game systems should consider all of these factors. This thesis attempts to do so through the clarification of terms and goals in relationship to the existing literature. The next sections will explore the basics of design for learning applications and the specifics of learning programming using those applications.

2.2 Designing Learning Apps/Games for Children

In the literature reviewed, some key beliefs about how to approach the design of learning apps came forward. Children have their own personalities and distinct likes and dislikes and should not be considered a homogeneous group. Children are also not “just short adults” and instead should be treated as an entirely different user group with their own unique set of needs, cultures, norms, and complexities. Despite this trend, developers often ask parents and teachers for advice on what their children are looking for, rather than asking the children directly to receive more straightforward and relevant answers. Children are of course dependent on their parents for basic needs and their teachers for their education experience, but that doesn’t mean their voices shouldn’t be heard first-hand. Also, designers tend to have their own biases, which can lead to incorrect assumptions. Everyone has also gone through many years of schooling themselves and may not be aware of the differences that are present in the current school system as compared to their past experiences. Everyone also has personal experience with children, but the information isn’t always up to date and is rarely broad in range. It may be harder for a child to articulate their thoughts and opinions, but their input may prove invaluable to the developer, in terms of a direct perspective from their target audience when they are designing for children. Druin (2002) stated that: *“In addition, as we know, young children have a more difficult time verbalizing their thoughts, especially when it concerns abstract concepts and actions (Piaget, 1971; Piaget, 1973). While children can be extremely honest in their feedback and comments concerning technology, much of what they say needs to be interpreted within the context of concrete experiences (Druin, 1999).”* Because of this children’s role in designing new technologies has been limited. However, children input on design should not be ignored or avoided just because it may be more difficult to acquire their input.

Druin (2002) argues that designers need to question how new technologies are built in regard to children and how they challenge themselves and investigate their environment. He believes the designer should create new technologies that offer children more control in the world, when their control is usually very limited. We must first understand the role children can play when designing technology. This will lead to better technologies being designed to better serve the needs of children and further their knowledge and growth. Druin (2002) states that based on their research and analysis of literature, children play in four major roles of the design process: user, tester, informant, and design partner. Figure 4 is an illustration of those roles.

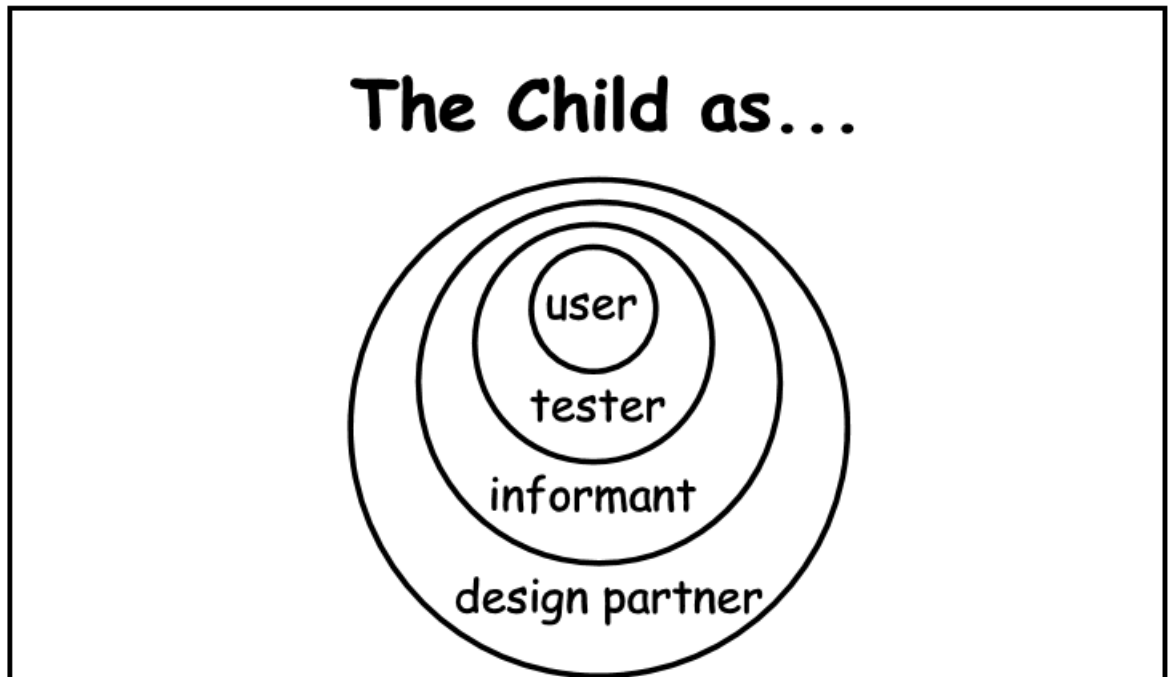


Figure 1: The four roles that children may have in the design of new technologies

Figure 4 The four roles that children may have in the design of new technologies. (Druin, 2002, p.4)

For the role of user, children contribute to research and development by using a technology, while being observed or tested on skills gained. As a tester, children can be given a new technology that may not yet be released and be asked for comments and suggestions on their experience with the product they are testing. Informants may be asked for their input on existing technologies or on low-tech prototypes or design sketches. As design partners, children are seen as equal stakeholders in the design process of new technologies and are treated as partners and contributors. These roles can also overlap depending on the amount of child involvement in a project. For example, a child in the role of informant may act as a tester on a prototype and observed as a user when using a competing software. Table 2 is a chart of strengths and challenges of each role of the child in the design process and when that role started to be used in research. (Druin, 2002)

Table 2 Summary of the roles of children in the design process.

Role of child	Began	Strengths	Challenges	In use by
User	Late '60s/ Early '70s	<ul style="list-style-type: none"> • Easy to include children • Researcher in control • Can suggest future directions in HCI & education areas 	<ul style="list-style-type: none"> • Less direct impact on changes in technology • Children have less say in changes • Educators need time to accomplish 	Primarily academic researchers
Tester	A few examples in the 1970s—Began primarily in the late '80s/ early '90s	<ul style="list-style-type: none"> • Begins to empower children • Quicker input for changing technology • Methods can be done in and out of schools 	<ul style="list-style-type: none"> • Children don't have input until later in the design process • Can offer surprises to adults • Adults decide what can be done given limits of schedule 	Academic researchers & Industry professionals
Informant	Mid 1990s	<ul style="list-style-type: none"> • Empowers children • Brings children's input into the start of the development process • Flexible when children and adults work together 	<ul style="list-style-type: none"> • Adults still decide when to bring children into the design process • More time is needed to work with children 	Academic researchers & Industry professionals
Design Partner	Mid 1990s	<ul style="list-style-type: none"> • Empowers children throughout development experience • Children and adults can change and learn from the experience • Instant feedback from children throughout the design process 	<ul style="list-style-type: none"> • Team decisions must be negotiated between adults & children • More time is needed to work as partners • School environment is difficult to work within • Difficult finding researchers that can work with children 	Primarily academic researchers with industry professionals beginning

Table 2: Summary of the roles of children in the design process

(Druin, 2002, p.28)

Depending on the research situation, a certain role may be more appropriate than another for children when designing new technology. There are also roles children should not be considered for just like any other person. For example, a developer, a teacher, and a designer all bring very different skills and background when approaching the design process for technology. So, the children cannot be expected to program as well as a software developer; however, children can inform the designer about what engages them and what can be boring to them and this feedback can be vital in the design process. New technologies can give children a much louder voice and enable them to do things that would otherwise be impossible. In summary, children should be involved in design process of interactive products for children, but the extent of their involvement is dependent on the situation and may not be the same across different products and different companies. (Druin, 2002)

2.3 Learning of Programming

2.3.1 Programming Learning Tools

Today there are many tools that self-motivated students can use to learn programming on their own. There is software that teaches many different programming language and also teaches the user about the basic syntax and logic of a programming language, as well as basic programming principles. Here is a list of free and paid programs aimed at teaching the user programming from young children to working professionals:

- A free example is Code Academy (<https://www.codecademy.com/>), which teaches many coding languages, especially languages used for web development.
- Lynda.com (<https://www.lynda.com/>) is a paid service that teaches coding for everyone.
- Khan Academy (<https://www.khanacademy.org/>) is a service that offers a wide range of courses subjects including free programming classes, and provides many tutorials.
- Tree house is a paid service that is engaging and good for beginners.
- CodeHS (<https://codehs.com/>) is another paid service that is good for starting coding education.
- Code School (<https://www.codeschool.com/>) is a paid service that has a lot of video courses and offers insights from industry professionals.
- Free Code Camp (<https://www.freecodecamp.com/>) is a free service that provides coding lessons shows opportunities to apply those lessons to the real world.
- Inform 7 (<http://inform7.com/>) is a free design system that is designed to have code that reads English and is used to create text-adventure games.
- Twine (<https://twinery.org/>) is a game development tool that can be used with little coding knowledge to make interactive stories.
- General assembly Dash (<https://dash.generalassemb.ly/>) is a free service that provide a good introduction to web development with in depth courses and show real world implications.
- Sitepoint (<https://www.sitepoint.com/>) is a paid service that offers online classes about coding and also offers eBooks as well to learning programming.
- Code Avengers (<https://www.codeavengers.com/>) is a service that offers both free and paid programming courses.
- Google for Education (<https://developers.google.com/training/>) is coding classes made by Google that is for current developers with experience. These can be more complicated and possibly not as fun as other courses aimed at beginners.
- Udemy (<https://www.udemy.com/courses/>) is a service that offers a wide range of courses including programming courses with both paid and free courses and offers many video lessons that can also be accessed on mobile.
- Youth Digital (<http://www.youthdigital.com/>) is a set of computer sciences courses for children.
- Hopscotch (<https://www.gethopscotch.com/>) is an iPad app used to teach kids programming with a kid-friendly interface, but is still open-ended enough to allow the kids to be creative with their programming.
- Code Combat (<https://codecombat.com/>) is a game that teaches kids some basics of code while they the play the game.

- Lightbot (<https://lightbot.com/>), is a game-based app focused on teaching programming principles to children through game play. This involves moving a robot through a level. The levels get more difficult and eventually the player will need to use things such as functions and loop to be able to complete the level with the amount of code space allowed to the user. There is both a free and paid version of Lightbot, and it is available on Android, iOS, Windows, Mac and Kindle. The app was built with the “Hour of Code” initiative in mind, that aims to introduce many students to on hour of computer programming. There is also a Lightbot Jr. (<https://itunes.apple.com/us/app/lightbot-jr-4+-coding-puzzles/id858640629>) aimed at younger children for iOS, android and Kindle.
- Move the Turtle (<http://movetheturtle.com/>) is an app for iPad that targets young children that gives access to some programming principles where they move the turtle through a puzzle or draw pictures.
- Daisy the Dinosaur (<http://www.daisythedinosaur.com/>) is another iPad app for young children that can teach some very basic coding principles. (Minor, 2016)
- Scratch by MIT Media Lab (<https://scratch.mit.edu/>) is a free program to teach coding to children. It is a web application that presents code as puzzle pieces that kids put together to write code and can share and see what others have created.
- BeeBots (<https://www.bee-bot.us/>) is used to teach programming principles to young children.
- Koodaustunti (<http://koodaustunti.fi/>) is based on the “Hour of Code” global movement for teaching programming to children worldwide. The “Hour of Code” is a movement that tries to emphasize that anyone can write some code in an hour and that coding isn’t as difficult or inaccessible as commonly imagined.

These new tools can make programming much more enjoyable to learn, and make it easier for a wider range of students to become involved and engaged. This is especially important as learning programming is, as of 2015, a required part of the national curriculum in Finland. This has made many teachers interested in learning the best ways to teach programming because many have never taught or even learned about programming.

Massive Open Online Courses (MOOCs) have become very popular recently. Course materials include recorded lectures, reading materials, and assignments distributed via the internet. They are therefore able to reach vast audiences with minimal resources committed. Udacity (<https://www.udacity.com/>), EdX (<https://www.edx.org/>), and Khan Academy (<https://www.khanacademy.org/>) are examples of MOOCs. One problem with MOOCs, though, is that evaluation is usually either automated or peer-based. Automation is only feasible for some assignments and peer evaluation has inherent problems, such as unequal participation. Tillmann et al. (2013) used the Pex4Fun MOOC (<http://www.pex4fun.com/>) which is a browser-based software that is experienced as a game by the user and teaches programming principles. The use of Pex4Fun in their research had good results from students and teachers, though empirical effectiveness was not evaluated. Empirical evidence of effectiveness is rarely collected for learning-based games, which leads again to the question of the effectiveness of learning through games (Tillmann et al., 2013). The large number of tools available has the potential to support more thorough empirical research in the long run.

2.3.2 Student's Perceptions of Programming

Maloney et al. (2008) study is an implementation of learning for programming. The researchers used Scratch in an afterschool program and pursued concrete empirical findings. In their study, they found an increase in engagement with programming among the urban youth at a Computer Clubhouse when using the learning app. They also noticed that the youth began to use more complex programming concepts over time though their instructor had no prior programming experience as shown in Figure 5 below:

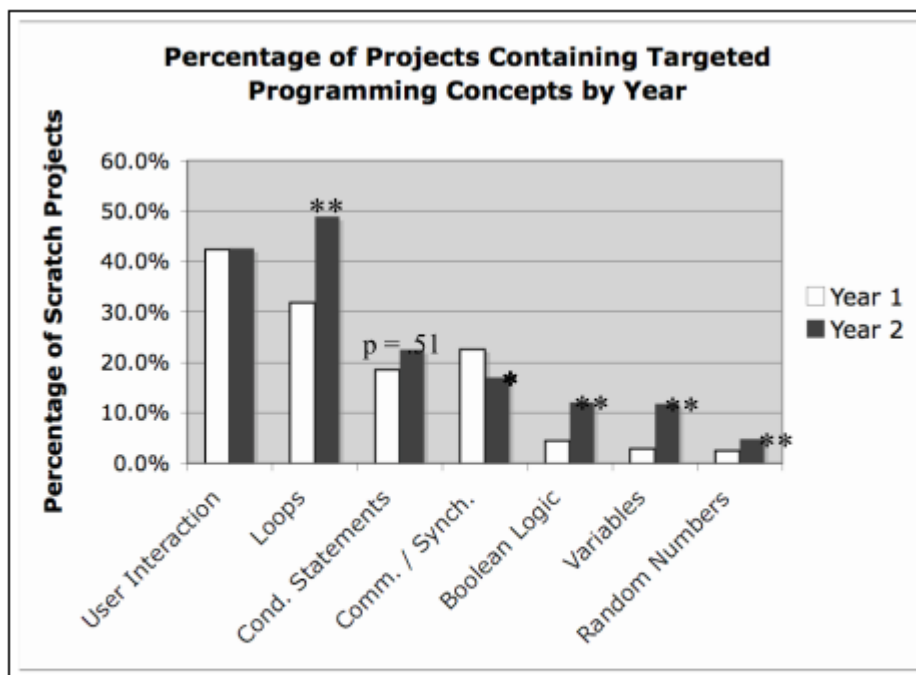


Figure 5. Graph demonstrating the change in the percentage of projects that used various programming concepts over time
****p < .001 *p < .05**

Figure 5 Graph demonstrating the change in percentage that used various programming concepts over time. (Maloney et al., 2008, p.4)

Tan et al. (2009) state that programming is a difficult subject to learn and that there is a high dropout and failure rate in programming for undergraduates. They investigated the initial claim that learning programming is difficult due to the undergraduates' lack of understanding about how programs are executed and proposed that a visualization tool could be used to help increase their understanding. They used a questionnaire whose results showed which example programs were rated as the most useful materials when learning programming. Table 3 ranks these materials. (Tan, et al., 2009)

Table 3 Materials that would help to learn programming.**TABLE VII. MATERIALS THAT WOULD HELP TO LEARN PROGRAMMING**

	Mean	Std. Deviation
Example programs	4.35	0.70
Exercise questions and answers	4.00	0.88
Interactive visualizations	3.65	0.94
Programming course book	3.61	0.93
Still pictures of programming structures	3.59	1.01
Lecture notes/copies of transparencies	3.51	0.96
Forums	3.40	1.00

(Tan et al., 2009, p.4)

As can be seen in Table 3, example programs followed by exercise questions and answers is what students felt to be the most and significantly more useful when learning programming. Also, students found forums to be the least useful and thought interactive visualizations, programming course books, still pictures of programming structures and lecture notes/copies of transparencies to be of similarly helpful. This shows that students really value example problems and exercises and more concrete solutions which are similar to real code. In Table 4, is a list of factors that lead to poor performance in programming courses. (Tan, et al., 2009)

Table 4 Factors that lead to poor performance in programming subjects.**TABLE IX. FACTORS THAT LEAD TO PERFORM POORLY IN PROGRAMMING SUBJECTS**

	Mean	Std. Deviation
Less examples in practical use are shown	3.68	0.96
Syllabi focuses too much on theory	3.62	0.90
Syllabi coverage per semester is too wide	3.60	0.95
Students lack of interest to learn	3.59	1.04
Learning environment that is not conducive	3.36	0.83
Presentation of instructors and their attention on students	3.36	0.87
Teaching methodology is less effective	3.35	0.87
Computers provided in labs are not functioning well	3.30	1.05

(Tan et al., 2009, p.4)

As we can see from the questionnaire results shown in Table 4, the factors student thought what led to the poorest performance was having less practical examples shown to them. This shows that the students really valued having practical resources. This is also shown in the next two highest rated factors, syllabi that focus too much and theory and syllabi coverage that is too wide, indicating that these students would like more practical teachings and more focused learning (Tan et al., 2009). These two studies and their methods form the basis of the thesis's approach to student perceptions of learning applications for programming.

2.4 How to Evaluate Learning Games/Serious Games/Educational Games with Children

One consistent problem with educational games is the difficulty in empirically measuring the effectiveness of the software. There are few empirical studies on how to evaluate educational games, but there quite a few qualitative studies. The PLU model, which was

first discussed by Read (2004), defines and outline how children interact with technology. It defines children as players, as learners, and as users. For children as players, the child must see the interactive product as a plaything, and its main purpose is to entertain the child. For child as learners, the interactive product is seen as a substitute for school or a teacher and the product is expected to instruct, challenge, and reward. For children as users, the child is supposed to see the interactive product as a tool. The product should enable the child and makes tasks easier. The PLU model is used to evaluate interactive technology and used to map out the purpose of the interactive product (Markopoulos et al., 2008). The PLU model is illustrated below in Figure 6:

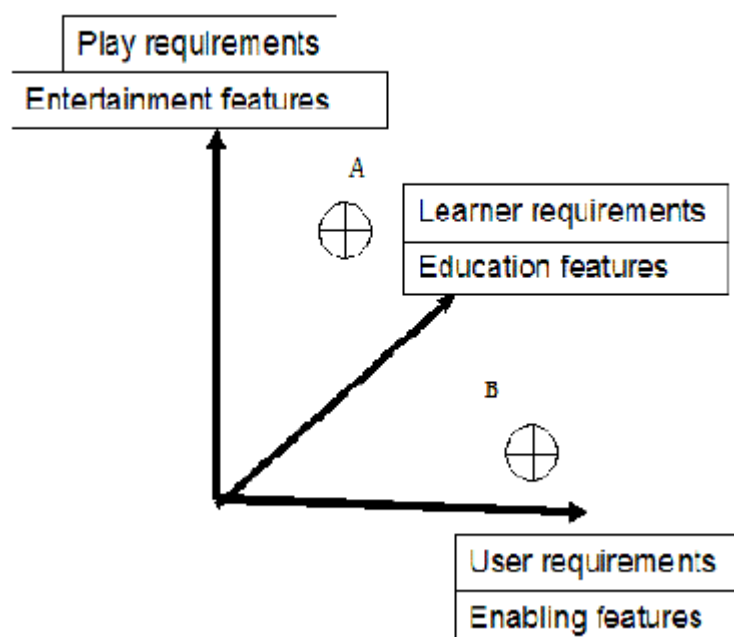


Figure 6 PLU model illustrated. (Read, 2004, in Markopoulos et al., 2008)

Baauw et al. (2005) evaluated children's computer games using a predictive evaluation method. Their study was to test the assumption that adults are able to predict the problems children will come across while playing the children's computer game. They stated that there are two different types of evaluation methods that exist for the assessment of interactive products, empirical evaluation methods and predictive or analytical methods. The advantage of the empirical method is that the actual users of the software are evaluated, whereas the predictive method can be applied to detect potential problems before end use. Using the predictive model, Baauw et al. (2005) developed, with the help of experts, the Structured Expert Evaluation Method (SEEM). This method has two main aspects, the plans and actions aspect and the evaluation aspect, which involves the child's ability to perceive and interpret the feedback from their inputs. The model uses the assumption of goal-driven behavior, which is applied to both the behavior of children and of computer games. SEEM is a checklist of questions that help identify potential problems children may have with the learning application. The SEEM questions presented in the study are as follows:

- "1 Do children understand the goal?
- 2 Do children know what to do in order to accomplish the goal?
- 3 Are children able to perform the physical actions easily?

- 4 Can children perceive the feedback? This includes feedback (if any) from both correct and wrong actions, and whether children can click to stop the feedback.
- 5 Do children understand the feedback? This holds for both visual and auditory feedback from correct and wrong actions.
- 6 Will children keep on going until they reach the goal? This includes whether children will like the sub game and if the level of difficulty is okay for young children.
- 7 Are the navigation possibilities and the exits from a (sub) game clear?
- 8 Are there other objects in the Game Interface that will cause problems?

The global questions are:

- 1 a. Is the challenge right for the target group? b. Is the curiosity of children stimulated? c. Are the story and the interface tuned to the fantasies of children?
- 2 a. Is it clear whether a sub game is optional or obligatory? b. Does the flow of the game meet the expectations? Is the story line logical? c. Is it clear when a child should be either passive or active during the game?" (Baauw et al., 2005)

Both the thoroughness and validity of the model were tested and had fairly good results at predicting the actual problems children experienced while playing computer games. They found that about 74% of the problem predictions were correctly related to a predictive question from SEEM. The thoroughness was found to be very good, but the validity was not as good because the number of false positives discovered by the experts. They made an updated set of questions as follows:

- "1 Goal: Can children perceive the goal? Do children understand the goal? Do children think the goal is fun?
- 2 Planning and translation into actions: Can children perceive and understand the actions they have to execute in order to reach the goal? Do children think the actions they have to execute in order to reach are fun?
- 3 Physical actions: Are children able to perform the physical actions easily?
- 4 Feedback (after correct and wrong actions): Is the feedback (if any) perceivable? Do children understand the feedback? Is the feedback motivating?
- 5 Continuation: Is the goal getting closer fast enough? Is the reward in line with the effort children have to do in order to reach the goal?
- 6 Navigation: Are the navigation possibilities and the exits from a (sub) game clear?
- 7 Are there other (e.g. technical) problems?" (Baauw et al., 2005)

In a study by Sim et al. (2006), the researchers again establish that, despite the growth of educational and leisure software, there is no clearly established methodology for evaluation software for children. They then identify what they consider to be the important aspects of learning software. The aspects focused on were:

1. The national curriculum in England
2. Learning, assessment, and feedback
3. Fun in educational software
4. Usability in educational software

Their study reiterated the difficulty of measuring learning effects of educational software designed for children, as well as maintaining that a long experiment would be hard to perform as there are many uncontrollable variables that can contribute to the children's

learning, for example the amount of time spent reading books. How fun and usable the products were weakly correlated, though with a significant link. A strong and significant correlation was found between the software the children thought was the most fun and the software they would choose to use. There was no significant correlation between reported fun or usability with learning effectiveness of the software. Usability problems also led to children appearing to have less fun. Young children, around 7-8, were however able to distinguish the concepts of ease of use, fun, and learning. (Sim et al., 2006)

Donker and Markopoulos (2002), found that “thinking aloud” can help children report more about usability factors than other methods and state that this is consistent with earlier findings with adult users. The results didn’t seem to be affected by different verbalization capabilities or by how extroverted the children were. This was in contrast to their initial expectation that the think-aloud method would require high verbalization skills and be less effective with younger children. The study also found that girls reported more issues during usability testing than the boys. The study concluded with a suggestion for greater use of verbal protocol analysis for usability testing with children. (Donker & Markopoulos, 2002) Below are the criteria to evaluate for interactive products from Markopoulos, et al. (2008) some based on ISO 9241-11 (ISO, 1998):

- Usability is defined by ISO 9241-11 (ISO, 1998) as the “*extent to which a product can be used by specified users to achieve goals with effectiveness, efficiency, and satisfaction in a specified context of use*” (as cited in Markopoulos et al., 2008, p. 74). Usability isn’t a single aspect of an interactive product but more of a framework of related aspects related to the interaction with the product.
- Effectiveness can be defined as how accurate and complete can a user achieve goals in a certain setting.
- Satisfaction is defined by ISO 9241-11 (ISO, 1998) as “*freedom from discomfort, and positive attitudes toward the use of the product*” (as cited in Markopoulos et al., 2008, p. 75). Satisfaction can be seen as the users’ overall feelings, attitudes and perceptions of the product and how their interaction with product makes them feel.
- Learnability can be defined how easy it is for the user to learn how to use the product. A good competence level can be when the user is able use the product effectively without help from another. This can be important as it can determine how quickly a child can use a product without help from an adult.
- Fun can be hard to define, *Carroll (2004) defines fun as: “Things are fun when they attract, capture, and hold our attention by provoking new or unusual emotions in contexts that typically arouse none, or [by] arousing emotions not typically aroused in a given context.”* (as cited in Markopoulos et al., 2008, p. 75). Fun can be viewed as a user’s enjoyment of the product and as user that is having fun can often be more engaged with the product.
- Accessibility can be defined as the extent a product can be used by many different potential users. Accessibility is often considering when designing for user with disabilities.
- Safety for interactive products with children two main concerns, the first is that the product is physically safe, so it doesn’t present any physical danger to the child. The other main concern is that the product doesn’t expose the child to inappropriate content or give their information to strangers.

Some tips from “Evaluating children's interactive products: principles and practices for interaction designers” by Markopoulos, et al, 2008 are:

- Triangulate at all levels when doing research. Which can involve combining methods, involving different children, different test admins, and task and comparing the results against others.
- Informal qualitative input from a few children in the early stages of research can be more beneficial than formal evaluations in later stages.
- Before doing an evaluation it is important to spend a lot of time with the product or prototype and it can be beneficial to let some of the children to use the product in an informal setting first.
- It can also be beneficial to involve children as often as soon as possible when in design process. (Markopoulos et al., 2008)

Finally, a study by Hong et al. (2009) offers a framework for the evaluation of the educational value of digital games, which can be used to help aid in the assessment of digital games that can be used to encourage or discourage students to play certain games. The researchers argue that digital games are playing a progressively vital role in learning in children. This is important to account for, as some games have little educational value because they lack pedagogical design and game-based learning principles (Prensky, 2001). Hong et al. (2009) argue that game-based learning can be an efficient tool if designed correctly and take into account pedagogical and learning needs in a realistic educational setting and that educators and developers should work closely together in the design process to make these realities. To examine the educational value of digital games in Taiwan, an educational value index for digital games was created using an expert panel, which included five game scholars and five professional game designers. The researchers identified and divided learning games into five game mode categories: Drill and practice, single combat, sTable contest, evolutionary contest, and scenario. The panel identified 74 items into 7 categories to create game evaluation indices. The seven categories were: Mentality change, emotional fulfillment, knowledge enhancement, thinking skill development, interpersonal skills, spatial ability development, and bodily coordination. They then used the game evaluation indices on an online vocabulary game called “Super Word Searching Contest.” The assessments by both the game scholars and designers found that the game structure helped players develop flexible thinking skills, improve interpersonal skills, and broaden vocabulary. The researchers suggest again that educators, researchers, and developers should collaborate in all levels of the education system to ensure games are effective as a learning tool. The research also determined that games must have clear roles and goals and must be fun to play to help the user gain value out of the game. (Hong et al. 2009)

These are some good examples of potential ways to evaluate learning games for children. However, all models are not without their faults and there is no one model that can provide the best answer for all children’s games. These models offer approaches that help identify potential problems with the learning application or game and identify areas that need improvement. The lessons from these approaches will be used in the analysis section of the thesis.

2.5 Summary

There are many different terms for educational games, such as game-based or serious games that have been defined differently in different studies, but can mostly be considered games that are for more than just entertainment. The literature review makes apparent that educational games are effective, but there is little research on the exact effectiveness of educational games in comparison to other teaching methods. Some serious flaws of

previous research that lead to this lack of empirical evidence are: researcher bias, no control group, short exposure time, and no integration with prior research. A common problem discovered in the literature is that educational video games are sometimes made without pedagogy in mind which can limit the educational value of the game or application. Designing interactive products for children can be a tricky task with many steps involved. The amount of a child's involvement in design process of design interactive products for children can vary based on the situation. However, child perspectives should be considered for at least some of the design process. Children's feedback can be very useful when designing for children. There are many important aspects of learning games and applications, such as the amount of learning supported, fun, challenge, the usability of the application, and feedback from the application. There is no universally accepted method to evaluate learning games, but there are several different methods that can produce helpful results. There are many different applications and games currently available that help teach programming which might be used in studies that take all of these factors into account.

3. Research Methods

This research is both qualitative and quantitative in nature as both qualitative and quantitative data was collected using a questionnaire with children. Content analysis was used to further examine the data collected from the questionnaire.

3.1 Questionnaire with Children

Creating a questionnaire that intended for children can be a tricky task and must be considering carefully. Below is a list of tips for conducting questionnaires with children from “Evaluating children's interactive products: principles and practices for interaction designers” by Markopoulos, et al, (2008) that helped guide this research:

- The first tip is to limit the writing. Sometimes children find it hard to express themselves with writing if their language skills are not yet proficient.
- The second tip is to use appropriate tools and methods when conducting a questionnaire with children. It is important what methods and tools to use before conducting research and insuring the method is appropriate for research with children and to help reduce the effects of suggestibility.
- The third tip is to make the questionnaire fun, so the children do not lose interest in the questionnaire.
- The fourth tip is to expect the unexpected, as data that is gathered from children can be different than what is expected and children could also answer questions incorrectly.
- The fifth tip is to not take the process too seriously, as reading too much into data from one group could be counterproductive and likely not generalizable.
- The last tip offered was to be nice. A researcher may get better responses from the children if they treat the children with respect and are more approachable. (Markopoulos et al., 2008)

The questionnaire created using these tips can be found in Appendix A.

3.1.1 Qualitative

Strauss & Corbin (1998), defines qualitative research as: “*any type of research that produces findings not arrived at by statistical procedures or other means of quantification.*” (Strauss & Corbin, 1998). Qualitive research can be about people’s lives, experiences, emotions, feelings, perceptions, social aspects, culture, etc. Qualitative research methods are often chosen because of the nature of the research question, as collecting quantitative data that is related to some research questions can be very difficult or impossible.

3.1.2 Quantitative

Aliaga and Gunderson (2002) defines quantitative research as, “*Explaining phenomena by collecting numerical data that are analysed using mathematically based methods (in particular statistics)*” (as cited in Muijs, 2010, p.2). Muijs (2010), explains that

quantitative can be collected to answer questions that are typically not suited for quantitative data, for example numerical scales can be used to ask questions about the participant's attitude towards a subject.

3.1.3 In Practice

Quantitative and qualitative data are held as necessary aspects in learning application research, to respond to both the social and evaluative aspects. A questionnaire given in person to the children after undertaking a relevant activity seemed to be a good fit for this study. Giving out the questionnaire in person after having an activity allowed me to get more engaged answers from the children and higher response rates to all the questions asked (Strauss & Corbin, 1998). Using the tips and these concerns, a questionnaire was formed to ask the children questions to get a better understanding of the research question: "What are children's perceptions of programming and learning apps that teach programming?" The questionnaire allowed for easy collection on how the student felt about aspects of learning applications and their feelings towards programming. The qualitative data in this study was collected via free response questions in the questionnaire conducted. The data from these questions is similar to that of a structured interview. The quantitative data collected in this study was collected via the rating questions and yes and no questions in the questionnaire. This study also uses scales in the questionnaire to offer more nuanced data.

During the practical section of the thesis research, I visited a classroom in the International School of Oulu which had 22 students. The students were first asked to play with an app (Scratch Jr.) on iPads in pairs. Scratch Jr. was selected for this study because it was readily available from the school and the students were familiar with Scratch. Scratch aims to teach basic programming principles through play and construction of games by the user using puzzle piece like snippets of code that the user combines. Afterwards, a questionnaire was conducted on the children from the classroom in the International School of Oulu. A questionnaire was also chosen because the age group was 11-12 years old, and at this age children can provide fairly sophisticated responses to written questions. The questionnaire explored the students' thoughts on programming, learning apps/games, the most important aspects of learning apps, and the learning game with which they interacted. The questions were explained and responded to one by one so that if the students did not understand the questions, they could ask myself or their teacher for clarification. The students' names were not taken to protect their confidentiality.

The questionnaire, as shown in Appendix A, shows all the questions asked to the students all questions were carefully chosen to help answer the research question. A questionnaire with qualitative and quantitative questions was chosen for this study because it allowed for the gathering of fairly extensive data from the whole classroom within a limited time. The free response questions allowed good insight into the children's opinions on programming and learning apps. For the quantitative questions in the questionnaire a scale from 1 to 6 was used to give a good range of options to the students and eliminate the tendency to pick the middle option (such as 3 when given a range from 1 to 5).

- "What is your favorite subject in school?" (short answer) was asked to get a baseline of the student's interest and see if their favorite subjects in school correlated with their like or dislike of programming.

- “What is your least favorite subject in school?” (short answer) was asked for the same reasons previously stated and to see if their least favorite subjects affected their view on programming.
- “When did you start learning programming?” (short answer) was asked to discover when the students had begun programming and see if it affected their opinion on programming.
- “Have you done any programming outside of school?” (Y/N) was asked to gauge the students’ interest in programming and see if it was related to how much they liked programming.
- “How much do you like programming?” (scale) was asked to get the students’ general perceptions on their view of programming.
- “Why do you like or dislike programming?” (free) was asked to give the students the opportunity to express why they felt a certain way towards programming with an open answer.
- “What aspects of learning applications (apps) are the most important to you?” (scale) which listed Fun; Challenge; Learning; How easy it is to use the application; and Feedback from the application; allowed the students to rate all of these aspects of learning applications. This question was asked to learn what the students felt were the most important aspects of learning applications.
- “What is the best thing about learning apps?” (free) was asked to allow the students to answer freely what they thought was good about learning apps and get their perspective on what makes learning apps a good learning tool.
- “What is the worst thing about learning apps?” (free) was to get the students’ perspectives on what they thought was bad about learning apps.
- “How much did you like Scratch Jr., the app you used today?”; “How difficult was Scratch Jr?”; “How fun was Scratch Jr?”; “How easy was it to use Scratch Jr?”; and “Do you feel like you were learning something while playing Scratch Jr?” (scale) were asked to get the students’ perspectives on the programming learning application that had used that day in the classroom. Those questions were also asked to relate a real experience with what aspects of learning applications they found to be the most important.
- “What program do you like best for learning programming concepts?” (multiple choice) was asked to learn what program the students preferred listing the two programs they had used in school Scratch, Koodaustunti and allowing them to put other options if they had done some programming learning outside of school that they preferred.
- “What one thing would you do to improve the learning applications you use in school?” (free) was asked to allow the students to voice what they would do to change learning applications and how they think learning applications could be improved.
- Finally, the students were asked if they had any other comments, where they could write any last thoughts that they didn’t or couldn’t express earlier in the questionnaire.

The activity portion of the research seemed to go very well. The students were active and engaged with one another and the application. The questionnaire was received with only a few simple questions. However, some students did not complete their questionnaires fully, whether due to disinterest or confusion is not known.

3.2 Analysis Method

3.2.1 Content Analysis

Morgan (1993) describes content analysis as:

“In qualitative content analysis, data are categorized using categories that are generated, at least in part, inductively (i.e., derived from the data), and in most cases applied to the data through close reading (Morgan, 1993). (as cited in Forman & Damschroder, 2007, p.40). Content analysis can involve counting details, words or categories and analyzing the data to detect any patterns in the data, and then further examining those patterns to understand what those patterns could mean. Content analysis is used to make inferences based on the data collected in the study. Content analysis allows researchers to get a deeper understanding of the data collected by carefully analyzing it for any patterns that may be present in the data (Forman & Damschroder, 2007).

3.2.2 In practice

Content analysis was used in this study to make inferences from data collected using the questionnaire. The data collected was also compared to previous literature. In this study, the content of the data was analyzed to identify the answers to the questions that multiple students had similar views on. The qualitative data collected in the study was also quantified to be able to display the data in easily understood graphs, to emphasize the free response answers that students shared.

4. Analysis

As previously mentioned, this study consisted of a questionnaire that collected both qualitative and quantitative data from children. First the qualitative answers will be reviewed, then the quantitative data will be examined.

4.1 Qualitative Data

In Table 5 are the answers from the free response questions of the questionnaire. The first column gives the students' responses to the questions: When did you start learning programming? Have you done any programming outside of school? and How much do you like programming?. This is to give context to the following five free response questions:

Table 5 Table of qualitative data collected.

When did you start learning programming ?	Why do you like or dislike programming?	What is the best thing about learning apps?	What is the worst thing about learning apps?	What one thing would you do to improve learning applications you use in school?	Any other comments?
Have you done any programming outside of school?					
How much do you like programming (1-6)?					
4+ years, Yes, 5	"It can be pretty fun and you can learn stuff."	"It can be more fun then learning from books."	"I don't know"	"Don't know"	"No comments"
3 years, Yes, 5	"I like the though and feel of something working that I did."	"Simplicity and the way they teach things easily and getting the concept clear."	""	""	"No."
3 years, Yes, 5	"I like it because then you know what you have done with it and also with computer."	"Then you get some new info. Also I like coordinating."	"I don't know."	"Make in them do more possibilities to use them."	"I liked to coordinate today."

When did you start learning programming?	Why do you like or dislike programming?	What is the best thing about learning apps?	What is the worst thing about learning apps?	What one thing would you do to improve learning applications you use in school?	Any other comments?
<p>Have you done any programming outside of school?</p> <p>How much do you like programming (1-6)?</p> <p>2 years, Yes, 5</p>	"I like programming because I like the aspect of challenge and I like computers and I use HTML (a bit)"	"The best thing about learning apps is the capability to teach someone about something and getting someone interested in it."	"When they do no teach anything."	"I don't know"	"No"
0 years, No, 4	"I like it because when I program something and it works I'm really surprised."	"That you get to test them."	"If you don't understand them."	"I would like if there was language options."	"Scratch was a fun app."
0 years, No, 3	"I'm in the middle because sometimes it is confusing"	"When learning apps teach me something I never knew before."	"Some of the learning apps are really boring."	"Maybe to make them to be funer learning apps"	"That's about it"
1 year, No, 6	"I like programming because it is more fun than sitting and writing."	"It is more entertaining."	"I'm not sure."	"To use them more."	"Rott, Rots."
2 years, Yes, 5	"I like programming because you can use what you do in some."	"I don't know."	"Nothing."	"How to program."	"Coding is very fun."
2 years, Yes, 4	"I like it because it's fun!"	"It's sometimes fun!"	"It's boring!"	"Nothing really!"	"No!"
1 year, Yes, 6	"Because when you know how to program you can fix bugs on your phone etc."	"They're fun and entertaining"	"They might have technical issues."	"They could include guns or parkour."	"No!"

When did you start learning programming ?	Why do you like or dislike programming?	What is the best thing about learning apps?	What is the worst thing about learning apps?	What one thing would you do to improve learning applications you use in school?	Any other comments?
Have you done any programming outside of school? How much do you like programming (1-6)?					
0 years, No, 5	"I like it because you get to see what you have done and with a computer doing it."	"You get to learn in a different way."	"They are sometimes too easy."	"I would not change anything."	"No."
0 years, Kind of, 5	"I like programming since you get to express your own creativity in any way you want."	"Maybe that I can do basically anything I want when I learn the basics."	"Maybe learning advanced things and having to delete everything because of a corrupt file. Or something like that."	"I would like to improve on learning code and remembering where brackets should be."	""
0 years, No, 3	"It is kind of hard and boring."	"They might help you SOMETIMES."	"Sometimes they don't help."	"Make them more difficult."	"No."
4+ years, Yes, 4	"I like programming I like doing something weird."	"To try something new."	"When you download and you thought it was fun but it wasn't."	"I don't know, maybe fun."	"Umm ...nope (lol)."
0 years, Yes, 45	"I think that when I have times, it's nice. I don't know do programming a lot."	""	""	""	""
0 years, Yes, 55	"I don't know."	"I don't use."	"Confusing."	"I don't use."	""

When did you start learning programming ?	Why do you like or dislike programming?	What is the best thing about learning apps?	What is the worst thing about learning apps?	What one thing would you do to improve learning applications you use in school?	Any other comments?
Have you done any programming outside of school?					
How much do you like programming (1-6)?					
0 years, Yes, 6	"I like programming because you can make it do almost anything."	"That you can learn things. And they're fun."	"Nothing."	"Nothing."	"No."
2 years, Yes, 3	"It's just a bit boring and a bit fun."	"I don't know."	"They're boring."	"Nothing."	"Not really."
0 years, No, 6	"It's fun because you get learn how to program things."	"Don't know."	"Don't know."	"Don't know."	"Nope."
1 year, Yes, 6	"I like programming because there is not much rules."	"Sometimes you can mess around."	"It gives you too many adds."	"Not to put too many adds."	"No!"
3 years, Yes, 6	"I like because I like electronics and electricity."	"Takes no time and No TEACHERS."	"they crash A LOT sometimes."	"Make them harder."	"No."
3 years, Yes, 6	"I like it because I like electronics."	"You get to use electronic devices."	"Nothing."	"Use them more."	"Nope."

The students stated they like programming for a number of reasons: to do something different; they like electronics; that there aren't a lot of rules; it can be fun; express creativity; can see what you have done on the computer; can use what you have done; like the feelings of writing something that works; and the challenge it presents. Some of the students wrote they dislike programming because it can be boring; can be too hard; and can be confusing. Overall, the students seem to be very positive towards programming. Many of the students liked learning apps because they offer different ways of learning than what they usually do. They seem to like the freedom that comes from using learning apps and not just learning from the teacher, saying that learning apps can be fun and entertaining. They also specifically liked using the devices on which the apps were installed.

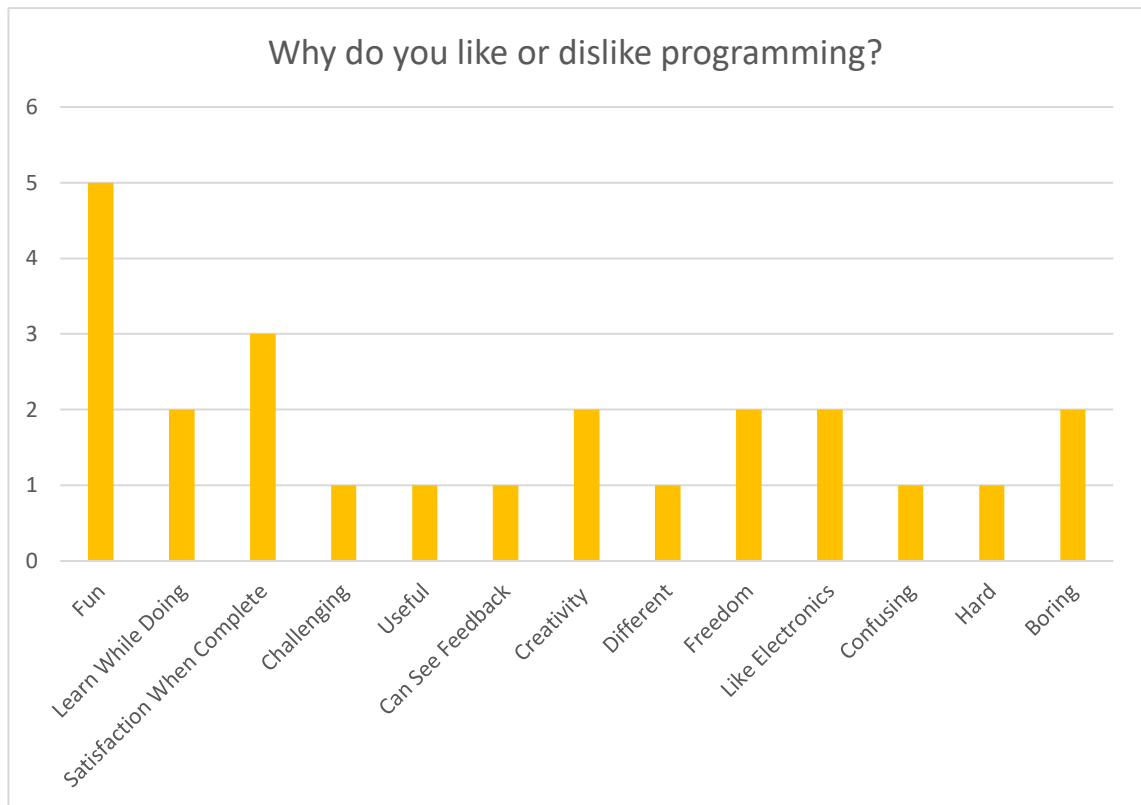


Figure 8 Students' response to the question: "Why do you like or dislike programming?"

In Figure 8, we can see that the most common reason the students like programming was because they think programming is fun. The next most common response was the satisfaction the student felt when they finished. Some of the other responses were the creativity allowed when programming, freedom when programming, that they liked electronics. Two students disliked programming because they think it is boring.

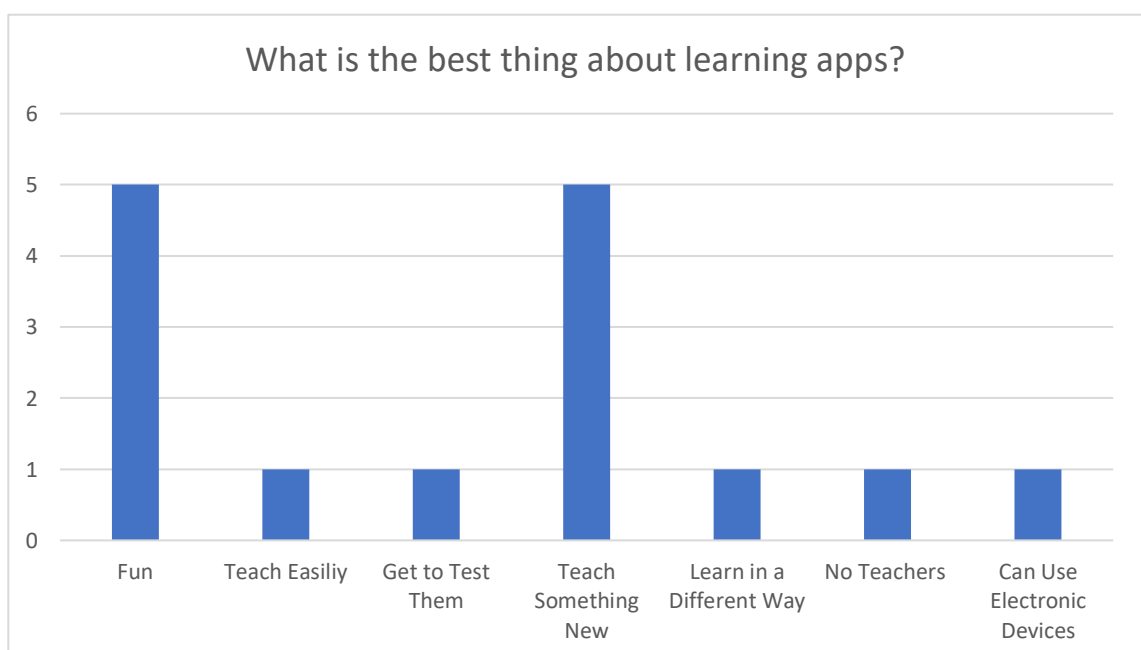


Figure 9 Students' response to the question: "What is the best thing about learning apps?"

In Figure 9 we can see that the most common answers to the question: "What is the best thing about learning apps?" were "fun" and "[to be taught] something new". This shows that the students valued the fun and the amount of learning that learning applications provided. Some of the things the students didn't like about learning apps were technical issues, such as apps that crash a lot, or that the learning apps sometimes don't help them to learn, or can be confusing, or have too many ads. Some of the student also stated that they can be boring or too easy (Figure 10).

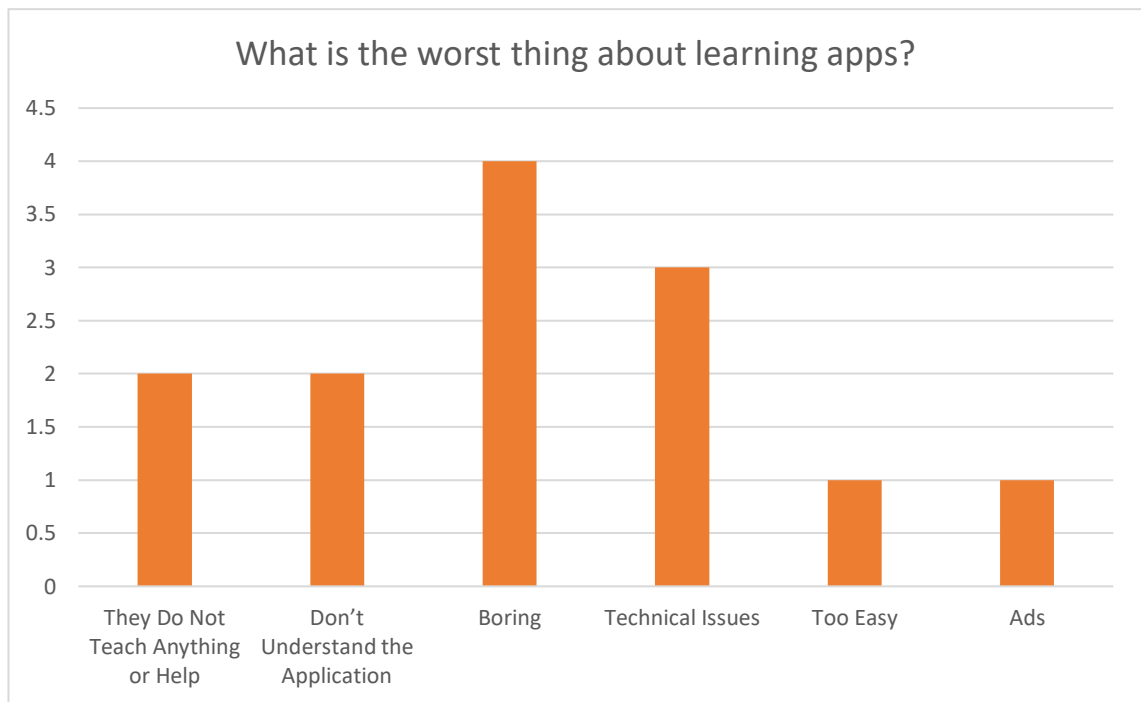


Figure 10 Students' response to the question: "What is the worst thing about learning apps?"

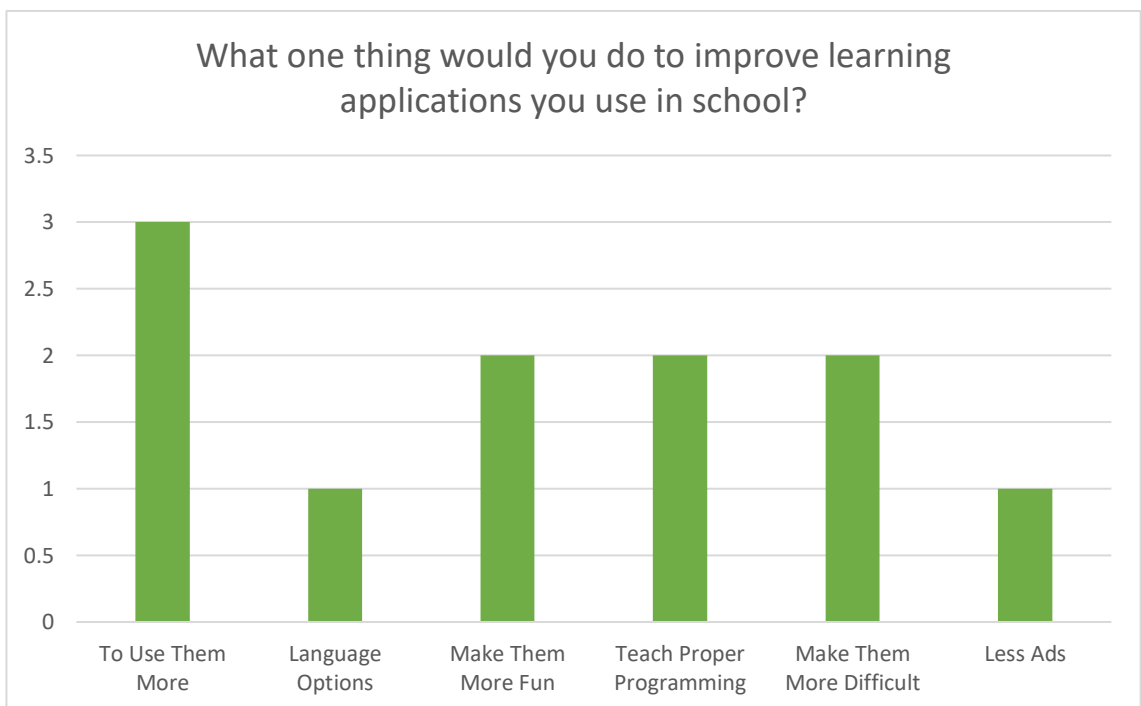


Figure 11 Students' response to the question: "What one thing would you do to improve learning applications you use in school?"

In Figure 11, students responded about things that can be improved in learning apps. Students wrote things such as: make them harder; to use them more; make them more fun; have less ads; learning actual code syntax; and to include language options. This shows that some students like using learning applications and their one improvement would simply be to use them more in their teaching. It also seemed that students would be interested in more challenging learning applications that match better with their skill level, so that they can learn more while using learning applications. Some of the students that like programming would be more interested in learning actual code of a programming language, rather than learning programming principles through games or other means at this age. This is understandable, because many of them probably have not written much “real code” yet, and are curious to do so.

4.2 Quantitative Data

The demographic data collected about the students was that they were all in grade 5 and were 11-12 years old. The first non-demographic quantitative question from the questionnaire was the students’ favorite application to learning programming.

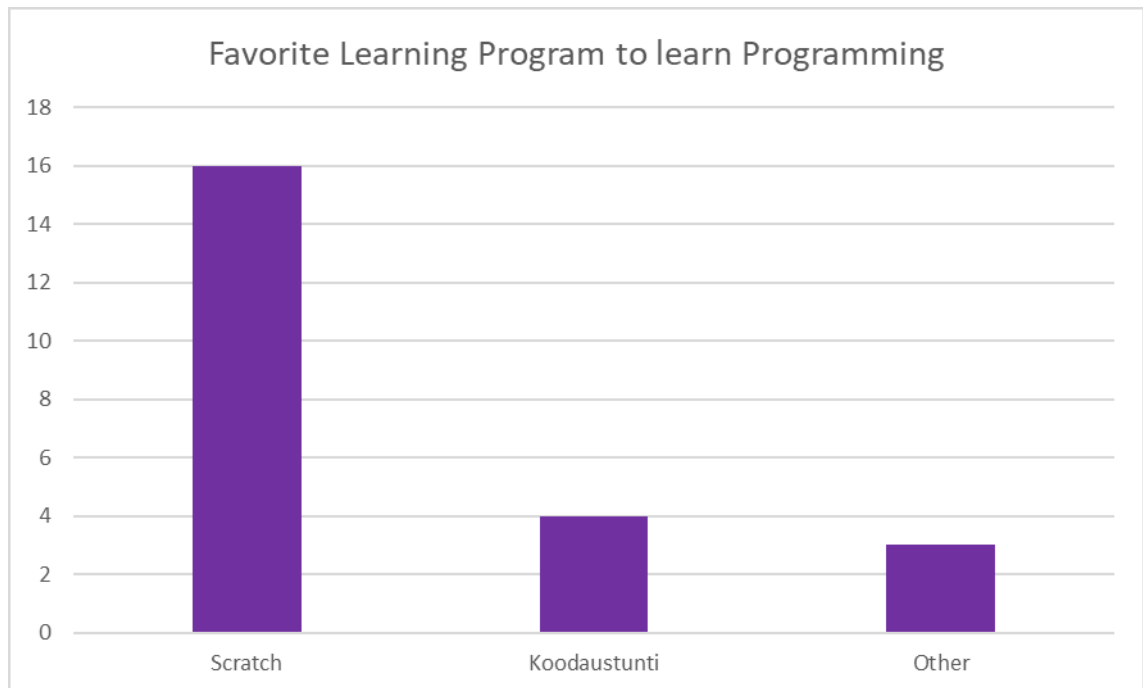


Figure 12 Students’ favorite Learning Program to learn Programming

As shown in Figure 12, the students use both Scratch and Koodaustunti in the international school in Oulu to learn programming principles. Some students had also used other programs outside of school. Scratch was by far the most common answer with sixteen students selecting it as their favorite. Koodaustunti had four students select it as their favorite, a couple students selected both Scratch and Koodaustunti. Only three students picked other programs as their favorite for learning programming they included: Code.org, Code Monster, robotics, and Minecraft.

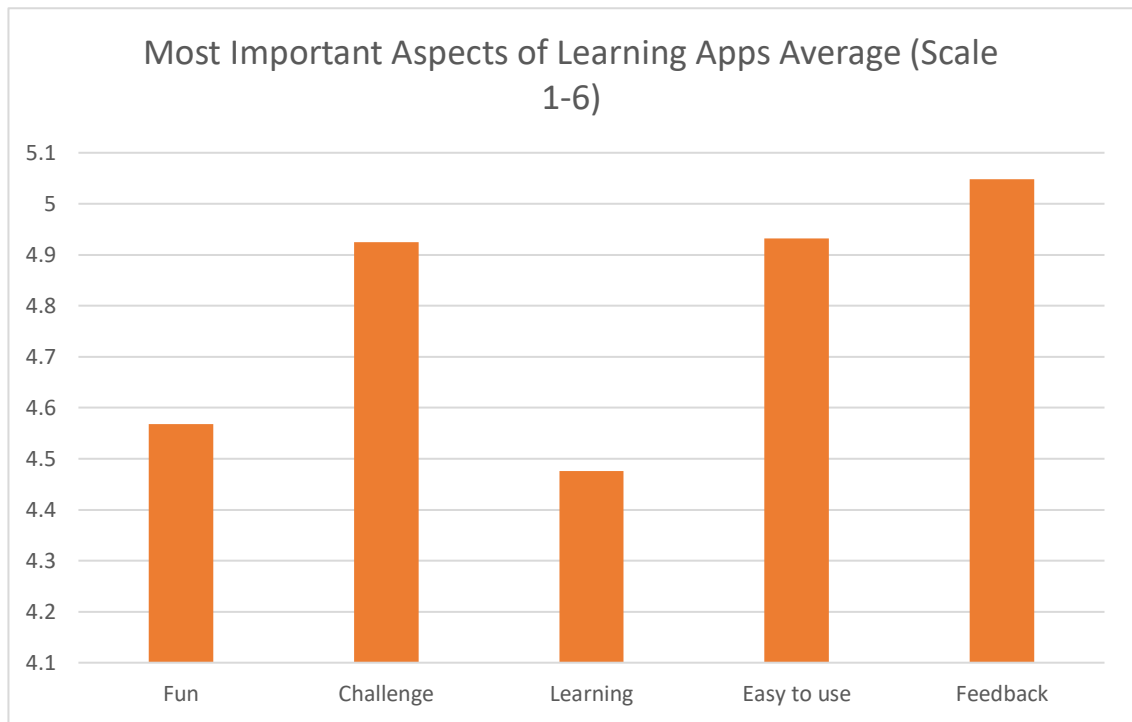


Figure 13 Students' views on most important aspects of learning apps (average).

As shown in Figure 13, we find that most students found all aspects of learning apps listed to be fairly important with lowest score being amount of learning with 4.48 out of 6. Feedback from the application was rated the highest by the children when an average of 5.05 out of 6.

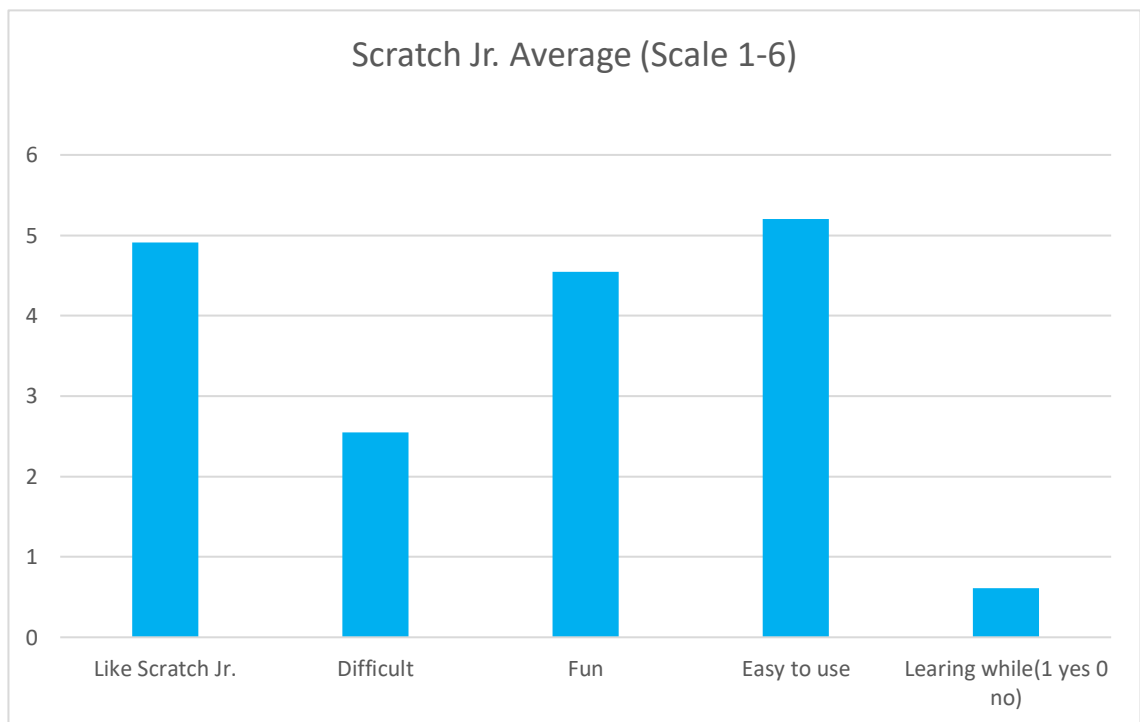


Figure 14 Students' ratings of Scratch Jr. (Average)

As shown in Figure 14, overall the students seemed to like scratch (4.91) and found it fairly fun (4.55) and easy to use. Most students did not find it difficult (2.55) and the answer to learning while using Scratch was mixed with an average of .61 (1 = yes, 0 = no) with some students answering both yes and no to this question, which was taken as .5 in the evaluation of their answer.

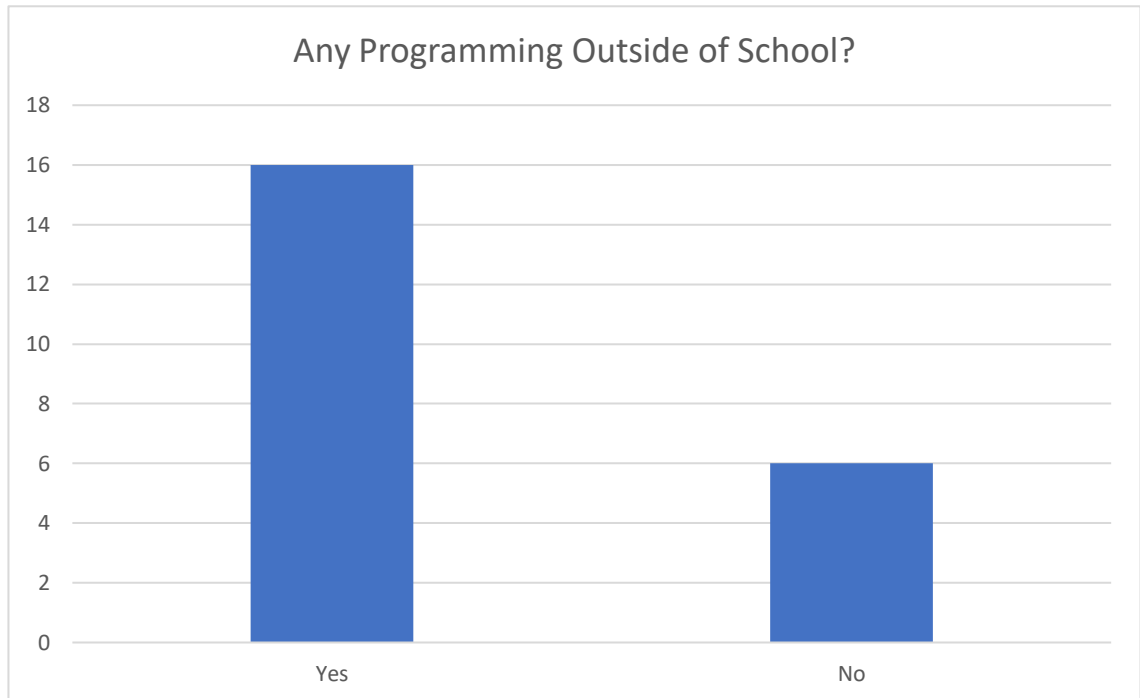


Figure 15 Students' answer to "Have you done any programming outside of school."

As shown in Figure 15, many of the students have tried some programming outside of school, 16 out 22, while 6 out 22 had not tried programming outside of school. It appears that a majority of the students are interested enough in programming to be self-motivated in their learning. However, I think that this can make designing programming apps more difficult if they students have very different levels of programming knowledge. It can be hard to find the balance to make an application not too challenging for student with little to no programming experience and not too boring for those who are very interested in programming and have done some programming in their free time.

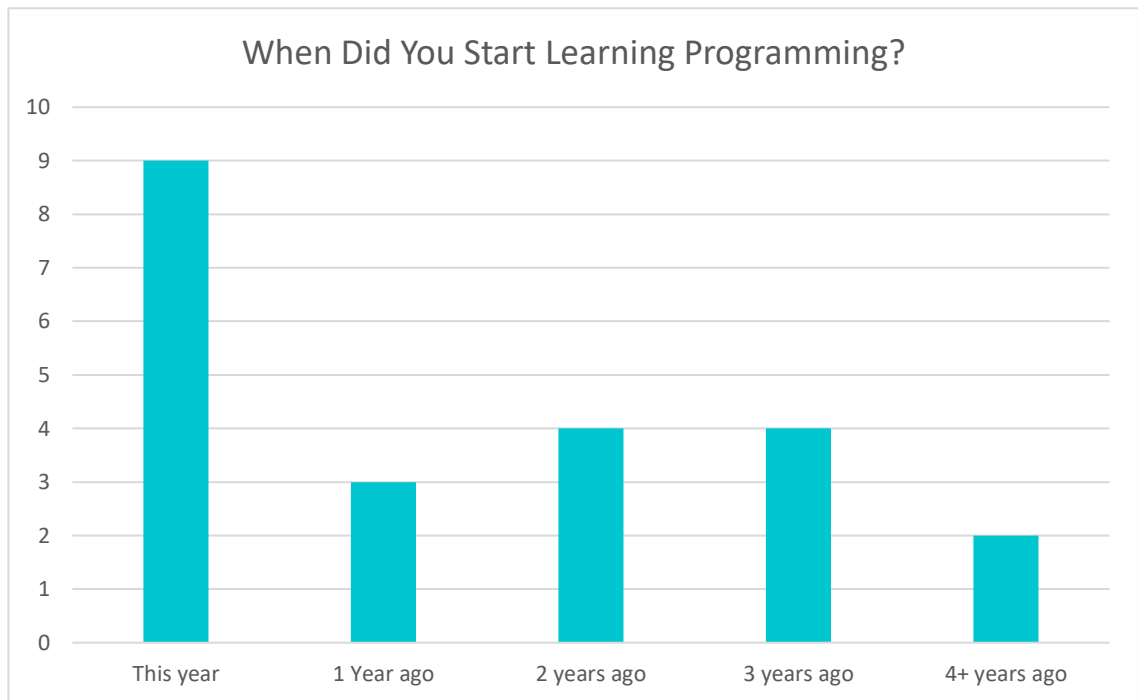


Figure 16 Students' answer to "When did you start learning programming?".

As shown in Figure 16, the most common answer to the question when did you start learning programming was this year, 9 out of 22, however, more than half, 13 out of 22 of the students stated that started at least 1 year ago. It is now required to teach some sort of programming principles to students in the Finnish curriculum and it seems that many students had not been exposed to programming before this requirement. The answer to this will change as time goes with the new requirement in the Finnish curriculum students' perceptions on programming may change as well when they are taught programming from a younger age.

5. Discussion

To revisit the research question: “What are children’s perceptions of programming and learning apps that teach programming?” we look at how the students answered the questions. To answer the question at a meta level, students find programming favorable overall and tend to like learning applications for programming, while offering suggestions on how to make learning applications better. At the level of the immediate thesis and its approach and results, the literature reviewed will have to be revisited. Two of the major themes discussed in the literature review were the usefulness of getting feedback from children when designing for children, and the effectiveness of learning applications/games. This is the reason the research method of a questionnaire was used and focused heavily on getting children’s feedback on their perspective of programming and learning applications. The children overall responded positively towards programming and learning applications. This reaffirms, as the literature has shown, that learning applications are at a base level effective at teaching children and have the potential to be even more effective in the future because of how positively children are reacting to learning applications. The following will discuss what the children like about learning apps and how this can be used when selecting and developing learning apps in the future.

Overall, the students thought learning apps/games were an effective way to learn. The students thought all the aspects listed in the questionnaire (fun, challenge, amount of learning, ease of use, feedback from the app) were almost equally important to learning applications. The students also ranked Scratch as their favorite learning application/game to learn programming. These findings again are consistent with the current literature. Many studies have stated student perceive programming as difficult. Contrary to this, few of the students wrote that programming was difficult in their responses, though this could be because of the lower level of programming the students were learning at their age. The students liked programming more than expected and more had tried programming outside of school than expected. Not all the children liked programming, but the majority seemed to like programming quite a bit for several reasons: to do something different; they like electronics; that there aren’t a lot of rules; can be fun; express creativity; can see what you have done on the computer; can use what you have done; like the feelings of writing something that works; and the challenge it presents. Understanding that many of the students liked learning apps because they offer different ways of learning than what they usually do, as well as the freedom that comes from using learning apps, and the excitement of using the devices offers key insights into how and why learning apps are effective. The themes of a good learning app for programming found in this study can be used when developing applications for children, when selecting what applications to use within the classroom, and when selecting or making suggestions for what apps to use at home.

That most of the students had tried some programming outside of school (16 out of 22) also shows that children are motivated and interested in programming currently. This bodes well for the future of programming education, as it can be much easier to teach something the students are actually interested in rather than something they are forced to learn. Many of the students had just begun to start learning programming in the current year (9 out of 22), as it recently became part of the national curriculum in Finland. Student’s perceptions may change over the years as they start learning from younger ages. It would be interesting to conduct a similar study using multiple age groups that have been exposed somewhat to programming and see if their answers are different based on

when they started to learn programming. As seen in the literature, many more and many more extensive studies will be needed to find the true effectiveness and value of learning applications and learning applications for teaching programming.

6. Conclusion

This study gives insight on children's perspective on learning apps, learning games and the learning of programming. In answer to the research question "What are children's perceptions of programming and learning apps that teach programming?", this study found that children view programming positively and learning applications favorably. This study introduces a unique first-hand look into the responses from children on this topic by asking questions directly to children using a questionnaire. Many of children the answered the questionnaire have been exposed to programming, to some degree, at young age, which gives authentic insight on how young learners view programming. This study found, from the literature review, the most important aspects of learning applications to be: fun, challenge, amount of learning, usability, and feedback from the application. The students ranked each of the aspects as important, with highest average being feedback from the application and lowest being amount of learning in the application. A common reason for the students liking programming and learning applications was because they offer something different than traditional teaching methods.

The children's feedback can be used when developing applications for children and educators can keep these themes in mind when selecting what applications to use within the classroom. This study reiterates that fun, challenge, amount of learning, usability, and feedback from the application are all important aspects of learning applications for developers to consider. Parents can use this information when selecting or making suggestions to what apps their children should use for learning purposes. Teachers can use this study as a model to ask for feedback from their students about the learning applications used within the classroom.

The biggest limitation to this study was that only one class was used in this study, making it hard to generalize the findings due to a limited sample. Time was a major constraint on this study as most of the practical research was done within two months. The study also relies on the answers from children, which some can view as unreliable. The findings would be more valuable if multiple classes were used and it would be beneficial if the classes came from different backgrounds, countries, etc. The study could also be expanded to include more schools and more age groups to get a better understanding of the questions asked in this study. This would mean that students being asked these questions would need to have some programming knowledge, or at least been taught some basic programming principles. This makes Finland a good choice, as students are now required to learning some programming in the national curriculum starting at age 7.

References

- Baauw, E., Bekker, M. M., & Barendregt, W. (2005, September). A Structured Expert Evaluation Method for the Evaluation of Children's Computer Games. In *IFIP Conference on Human-Computer Interaction-INTERACT 2005*, 457-469. doi: 101007/11555261
- Barendregt, W., & Bekker, T. M. (2011). The influence of the level of free-choice learning activities on the use of an educational computer game. *Computers & Education*, 56(1), 80-90. doi: <https://doi.org/101016/j.compedu.201008018>
- Barendregt, W., Bekker, M. M., & Speerstra, M. (2003, September). Empirical evaluation of usability and fun in computer games for children. In *Proc. IFIP Interact*, 3, 705-708.
- Barendregt, W., Bekker, M. M., Bouwhuis, D. G., & Baauw, E. (2006). Identifying Usability and Fun Problems in a Computer Game During First Use and After Some Practice. *International Journal of Human-Computer Studies*, 64(9), 830-846. doi: <https://doi.org/101016/j.ijhcs.200603004>
- Benford, S., Bederson, B. B., Åkesson, K. P., Bayon, V., Druin, A., Hansson, P., ... & Simsarian, K. T. (2000, April). Designing Storytelling Technologies to Encourage Collaboration Between Young Children. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI)*, 556-563. doi: 101145/332040332502
- Brederode, B., Markopoulos, P., Gielen, M., Vermeeren, A., & De Ridder, H. (2005, June). pOwerball: the design of a novel mixed-reality game for children with mixed abilities. In *Proceedings of the 2005 Conference on Interaction Design and Children*, 32-39. doi: 101145/11095401109545
- Bryman, A. (2006). Integrating quantitative and qualitative research: how is it done?. *Qualitative research*, 6(1), 97-113. doi: 101177/1468794106058877
- Colombo, L., Landoni, M., & Rubegni, E. (2012, June). Understanding Reading Experience to Inform the Design of eBooks for Children. In *Proceedings of the 11th International Conference on Interaction Design and Children*, 272-275. doi: 101145/23070962307143
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011, September). From Game Design Elements to Gamefulness: Defining Gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 9-15. doi:101145/21810372181040
- Donker, A., & Markopoulos, P. (2002). A comparison of think-aloud, questionnaires and interviews for testing usability with children. In *People and Computers XVI-Memorable Yet Invisible*, 305-316. Springer London. doi:101007/978-1-4471-0105-5_18

- Druin, A. (1999, May). Cooperative inquiry: developing new technologies for children with children. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 592-599. ACM. doi:101145/302979303166
- Druin, A. (2002). The Role of Children in the Design of New Technology. *Behaviour and Information technology*, 21(1), 1-25.
doi:http://dx.doi.org/101080/01449290110108659
- Druin, A. (ed.) (1999). *The design of children's technology*. San Francisco: Morgan Kaufmann Publishers.
- Druin, A., Bederson, B., Boltman, A., Miura, A., Knotts-Callaghan, D., & Platt, M. (1999). Children as our technology design partners. In A. Druin (Ed.), *The Design of Children's technology*, 51-72. San Francisco, CA: Morgan Kaufmann.
- Echeverría, R., & Jurado, F. (2015, November). Using player profiles and learning styles in the design of Educational Games. In *Computers in Education (SIIE), 2015 International Symposium on*, 141-144. IEEE. doi: 101109/SIIE.20157451664
- Egenfeldt-Nielsen, S. (2007). Making sweet music: The Educational use of computer games. *Datorspelandets Dynamik*. Studentlitteratur.
- Ferreira, S. M., Gouin-Vallerand, C., & Hotte, R. (2016, September). Game Based Learning: A Case Study on Designing an Educational Game for Children in Developing Countries. In *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2016 8th International Conference on* (pp. 1-8). IEEE. doi:101109/VS-GAMES.20167590350
- Forman, J., & Damschroder, L. (2007). Qualitative content analysis. In *Empirical methods for bioethics: A primer*, 39-62. Emerald Group Publishing Limited.
- Fullerton, T. (2014). *Game Design Workshop: a Playcentric Approach to Creating Innovative Games* (2nd ed.). Burlington, MA: CRC press.
- Gauthier, A., & Jenkinson, J. (2016, September). Woes of an RCT for Game-Based Learning Research-Past Problems and Potential Solutions. In *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2016 8th International Conference on*, 1-2. IEEE. doi:101109/VS-GAMES.20167590382
- Gee, J. P. (2003). What Video Games Have to Teach Us About Learning and Literacy. *ACM Computers in Entertainment (CIE)*, 1(1), 20-20. doi:101145/950566950595
- Girard, C., Ecalle, J., & Magnan, A. (2013). Serious Games as New Educational Tools: How Effective are They? A meta-analysis of Recent Studies. *Journal Of Computer Assisted Learning*, 29(3), 207-219. doi:101111/j.1365-2729201200489x
- Hong, J. C., Cheng, C. L., Hwang, M. Y., Lee, C. K., & Chang, H. Y. (2009). Assessing the Educational Values of Digital Games. *Journal of Computer Assisted Learning*, 25(5), 423-437. doi:101111/j.1365-2729200900319x
- Huizenga, J., Admiraal, W., Akkerman, S., & Dam, G. T. (2009). Mobile Game-Based Learning in Secondary Education: Engagement, Motivation and Learning in a Mobile City Game. *Journal of Computer Assisted Learning*, 25(4), 332-344. doi:101111/j.1365-2729200900316x

- Jiau, H. C., Chen, J. C., & Ssu, K. F. (2009). Enhancing Self-Motivation in Learning Programming Using Game-Based Simulation and Metrics. *IEEE Transactions on Education*, 52(4), 555-562. doi:10.1109/TE.2008.2010983
- Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. *The Internet and higher education*, 8(1), 13-24
doi:<https://doi.org/10.1016/j.iheduc.2004.12.001>
- Li, F. W., & Watson, C. (2011, December). Game-Based Concept Visualization for Learning Programming. In *Proceedings of the third international ACM workshop on Multimedia technologies for distance learning* (pp. 37-42). ACM.
doi:10.1145/20725982072607
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. In *Proceedings of SIGCSE '08 New York, NY: ACM Press*, 40(1), 367-371.
doi:10.1145/13523221352260
- Markopoulos, P., & Bekker, M. (2003). Interaction design and children. *Interacting with Computers*. 15, 2, 141–149. doi:[https://doi.org/10.1016/S0953-5438\(03\)00004-3](https://doi.org/10.1016/S0953-5438(03)00004-3)
- Markopoulos, P., & Bekker, M. (2003). On the assessment of usability testing methods for children. *Interacting with computers*, 15(2), 227-243.
doi:[https://doi.org/10.1016/S0953-5438\(03\)00009-2](https://doi.org/10.1016/S0953-5438(03)00009-2)
- Markopoulos, P., Read, J. C., MacFarlane, S., & Hoysniemi, J. (2008). *Evaluating Children's Interactive Products: Principles and Practices for Interaction Designers*. Burlington, MA: Morgan Kaufmann.
- Mathrani, A., Christian, S., & Ponder-Sutton, A. (2016). PlayIT: Game Based Learning Approach for Teaching Programming Concepts. *Journal Of Educational Technology & Society*, 19(2), 5-17. Retrieved from JSTOR.
- Miler, J., & Landowska, A. (2016, September). Designing effective educational games—a case study of a project management game. In *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*, 1657-1661. Retrieved from IEEE database.
- Milne, I., & Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information technologies*, 7(1), 55-66. doi:10.1023/A:1015362608943
- Minor, Jordan. (2016, March 18) The Best Apps and Services for Learning to Code. *PCMag UK*. *PCMag*. Retrieved Feb. 13, 2017, from <http://uk.pcmag.com/lyndacom/73124/guide/the-best-apps-and-services-for-learning-to-code>
- Morgan, D. L. (1998). Practical strategies for combining qualitative and quantitative methods: Applications to health research. *Qualitative health research*, 8(3), 362-376. doi:10.1177/104973239800800307
- Muijs, D. (2010). *Doing quantitative research in education with SPSS*. Thousand Oaks, CA: Sage Publications

- Myers, M. D. (2013). *Qualitative Research in Business & Management* (2nd ed.). Thousand Oaks, CA: Sage Publications
- Nazarova, M., & Galiullina, E. (2016). Game-based Learning in Teaching English. *Journal of Organizational Culture, Communications & Conflict*, 208-212. Retrieved from ProQuest Education Journals database.
- Newman, I., & Benz, C. R. (1998). *Qualitative-Quantitative Research Methodology: Exploring the Interactive Continuum*. Carbondale, IL: Southern Illinois University Press.
- Prenkys, M. (2001). Digital Natives, Digital Immigrants. *On the Horizon*, 9(5), 1-6. doi:101108/10748120110424816
- Read, J. C. (2008). Validating the Fun Toolkit: an instrument for measuring children's opinions of technology. *Cognition, Technology & Work*, 10(2), 119-128. doi:101007/s10111-007-0069-9
- Read, J. C., MacFarlane, S. J., & Casey, C. (2002, August). Endurability, engagement and expectations: Measuring children's fun. In *Interaction design and children* (Vol. 2, pp. 1-23). Shaker Publishing Eindhoven.
- Resnick, M. (2002). Rethinking Learning in the Digital Age. In G. Kirkman (Ed.), *The Global Information Technology Report: Readiness for the Networked World* (pp. 32-37). Oxford: Oxford University Press.
- Resnick, M., Maloney, J. Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52, 11, 60-67. doi: 101145/15927611592779
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172. doi: 101076/csed.13213714200
- Rosas, R., Nussbaum, M., Cumsille, P., Marianov, V., Correa, M., Flores, P., ... & Rodriguez, P. (2003). Beyond Nintendo: design and assessment of educational video games for first and second grade students. *Computers & Education*, 40(1), 71-94. doi:https://doi.org/101016/S0360-1315(02)00099-4
- Rubegni, E., & Landoni, M. (2014, June). Fiabot!: design and evaluation of a mobile storytelling application for schools. In *Proceedings of the 2014 conference on Interaction design and children* (pp. 165-174). ACM. doi:101145/25939682593979
- Shaffer, D. W., Squire, K. R., Halverson, R., & Gee, J. P. (2004). Video games and the future of learning. *Phi delta kappan*, 87(2), 105-111. doi:101177/003172170508700205
- Sim, G., MacFarlane, S., & Read, J. (2006). All work and no play: Measuring fun, usability, and learning in software for children. *Computers & Education*, 46(3), 235-248. doi: https://doi.org/101016/j.compedu.200511021
- Squire, K. (2003). Video games in education. In *International journal of intelligent simulations and gaming*, 2(1). doi:10111008500

- Strauss, A., & Corbin, J. (1998). *Basics of qualitative research*, 15 Newbury Park, CA: Sage.
- Susi, T., Johannesson, M., & Backlund, P. (2007). Serious games: An overview. *Technical report HIS-IKI-TR-07-001* School of Humanities and Informatics, University of Skövde, Sweden. Retrieved from DiVA.
- Tan, P. H., Ting, C. Y., & Ling, S. W. (2009, November). Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception. In *Computer Technology and Development, 2009 ICCTD'09 International Conference on* (Vol. 1, pp. 42-46). IEEE. doi:101109/ICCTD.2009188
- Tillmann, N., De Halleux, J., Xie, T., Gulwani, S., & Bishop, J. (2013, May). Teaching and learning programming and software engineering via interactive gaming. In *Proceedings of the 2013 International Conference on Software Engineering*, 1117-1126. IEEE Press. doi:101109/ICSE.20136606662
- Virvou, M., Katsionis, G., & Manos, K. (2005). Combining Software Games with Education: Evaluation of its Educational Effectiveness. *Educational Technology & Society*, 8(2), 54-65. Retrieved from JSTOR.
- Zaman, B., & Abeele, V. V. (2010, June). Laddering with Young Children in User eXperience Evaluations: Theoretical Groundings and a Practical Case. In *Proceedings of the 9th International Conference on Interaction Design and Children*, 156-165. ACM. doi:101145/18105431810561
- Zaman, B., Abeele, V. V., Markopoulos, P., & Marshall, P. (2012). Editorial: the evolving field of tangible interaction for children: the challenge of empirical validation. *Personal and Ubiquitous Computing* 16, 367-378. doi:101007/s00779-011-0409-x

Appendix A. Questionnaire used in this study

Grade: _____ Age: _____

What is your favorite subject in school? (circle)

Math Science English Finnish History
 Programming Reading Writing Gym Arts

Other: _____

What is your least favorite subject in school? (circle)

Math Science English Finnish History
 Programming Reading Writing Gym Arts

Other: _____

When did you start learning programming? (circle)

This Year 1 year ago 2 years ago 3 years ago 4+ years ago

Have you done any programming outside of school? (circle)

Yes No

How much do you like programming? (circle)

Not at all 1 2 3 4 5 6 *A lot*

Why do you like or dislike programming?

What aspects of learning applications (apps) are the most important to you?

Fun: *Not at all* 1 2 3 4 5 6 *Very*

Challenge: *Not at all* 1 2 3 4 5 6 *Very*

Learning: *Not at all* 1 2 3 4 5 6 *Very*

How easy it is to use the application: *Not at all* 1 2 3 4 5 6 *Very*

Feedback from the application: *Not at all* 1 2 3 4 5 6 *Very*

What is the best thing about learning apps?

What is the worst thing about learning apps?

How much did you like Scratch Jr., the app you used today? (circle)

Not at all 1 2 3 4 5 6 *A lot*

How difficult was Scratch Jr? (circle)

Not at all 1 2 3 4 5 6 *Very*

How fun was Scratch Jr? (circle)

Not at all 1 2 3 4 5 6 *Very*

How easy was it to use Scratch Jr? (circle)

Not at all 1 2 3 4 5 6 *Very*

Do you feel like you were learning something while playing Scratch Jr? (circle)

Yes No

What program do you like best for learning programming concepts? (circle)

Koodaustunti Scratch Other: _____

What one thing would you do to improve the learning applications you use in school?

Any other comments?
