Capstone Project – Designing a touch screen application to help young children develop programming skills

Panagis Papadatos, Advisors: Tamara Clegg, Mona Leigh Guha

Table of Contents

1 Thesis Statement

The iPad application that was designed and built during this capstone project helps young children learn skills that are integral to programming.

2 Abstract

Acquiring skills that are fundamental to programming early on in one’s life is a valuable skill. Not only does it help with the development of basic logic skills, but it also encourages children to learn more about computer science and understand technology at a more fundamental level. The goal of this capstone project was to design and develop an iPad application that would scaffold young children (3-5 years old) in developing skills that will later support them in programming. In attempting to achieve this goal, we conducted research with children and expert educators, while aiming to develop guidelines for developing the application on a touch screen device.

3 Introduction

Very young children are often left out of consideration in Computer Science learning. However, learning these skills early on is possible and extremely beneficial. Our contribution to the field of Human Computer Interaction is a touch screen application that helps young children (ages 3-5) be ready to face a world that requires thinking like a programmer. Mobile touch screen technologies are currently dominating our lives, and especially the lives of children, especially since they provide a more fluid and pleasurable experience compared to most other types of interfaces. The objective of this project was to design and implement a touch screen application that will help young children learn skills that are integral to programming. This application will also help children in developing computational thinking (CT) skills, which are widely considered to be a valuable asset. Furthermore, the process of designing this application provides a foundation for other designers and perhaps knowledge useful for researchers trying to create similar products or study similar subjects.

A plethora of factors led us to work on this problem. First and foremost, the comfort of younger generations with computers has made children more independent of their parents in their exploration processes, which suggests a greater learning need. By giving children the tools to build their own interactive environments, they can experience a level of creative autonomy previously limited to adults [1]. Furthermore, acquiring debugging skill sets enhances the development of logical thinking, problem articulation, team working, persistency, problem solving, and social interaction skills. Finally, providing reading opportunities to young children is important during a significant time period in their development of reading skill and comprehension [2].

Our motivation was also drawn from a different perspective; both the study of Computer Science (CS) and its related industries have recognized the importance of fostering the development of CT and skills earlier in students’ education [3]. This is especially true in light of current market conditions, where there is huge demand for computer scientists and an intense lack of diversity amongst its practitioners. More specifically, many students never investigate computer science as an academic option, because they have not been sufficiently exposed to computer science or because they feel that their identities do not match well with the stereotype of programmers [4, 5]. The latter is particularly true for African American populations, Latino populations and women [5]. These problems could be tackled by educating people from a younger age, thus encouraging them to be more interested in CS [6, 7, 8].

Before beginning our research, we established that computer programming and CT are age appropriate materials for preschoolers. Research shows that children have the ability to control their interactive environments and that ability is stronger in the modern era [1]. Computer programming is a developmentally appropriate practice in preschools [9] since children are able to learn aspects of programming by performing specific tasks and can reason logically as long as principles are applied to concrete examples [10]. Most importantly, preschoolers enjoy programming-like activities and have the desire to do perform them; this desire is strengthened when the environment allows for the ability to create dynamic and interactive worlds or games [11].

4 Background

Two different types of research proved valuable to the process of conducting our research and developing our application. The first type, past attempts to help young children learn concepts related to Computer Science, helped define the space in the field where our application would fit and determine what types of concepts there are to be learned. It also taught us useful lessons, not only about how to conduct the research, but also about how to design the application and its interactions. The second type of research, concerned with designing touch screen and mobile applications for children, provided useful guidelines and lessons that helped us determine the structure of the application. The research also posited types of interactions that are most appropriate, as well as the types of feedback and instructions required. Researchers have often attempted to scaffold the learning of programming and CT concepts; therefore, the literature provides a lot of insight into challenges, design ideas and suggestions that proved valuable in understanding the field and expanding it.

In the past 10 years, there have been several attempts to create games, or otherwise educational environments that encourage children to acquire CT skills. The most famous example is Logo [12], a mostly graphic computer programming language developed in 1967 that has been employed and adapted multiple times [2]. Logo, despite being used in a wide variety of computer science learning for young children, does not take advantage of modern research or new technologies and their affordances, which could be extremely valuable.

Researchers have also employed different techniques with widely different goals and target groups. Kahn [5], for instance, demonstrated how programming by example works and suggests animating concepts, whereas Kindborg and Sökjer [13] demonstrated behavior-based programming. Another example would be the software Creator [12], which uses programming by demonstration to encourage program construction and targets children around the age of 12, or Magic Words [13], which is a behavior-based visual programming toolkit that allows children (close to 5 years old) to create their own interactive worlds and games. Similarly, Toontalk [5, 13] is an animated and action-based programming language that has been tested with preschoolers. Alice [12] is a 3D programming environment that allows children to create narratives, play games and create videos. One of the most popular examples, Scratch [14], is a visual programming environment that allows users (primarily ages 8 to 16) to learn computer programming while working on personally meaningful projects such as animated stories and games. Last, Fizz [15] is a physics-based programming system (ages 6 to 12) that allows the production of games and simulations using events and drag and drop programming whereas CTArcade [2] allows learners to train their own virtual characters while playing games with it. These examples provided us with interesting ideas and methods of modelling programming concepts that affected the design of our application.

Many of the activities that were studied used a different perspective on this topic, that of "physical programming": The Robo-Blocks system [16] allows children to connect physical command blocks (ages 8-9) while Electronic Blocks [9, 17, 18] is another physical programming environment designed specifically for children aged between 3 and 8 years. Last, Storyroom [19] uses a set of tangible tools and metaphors in a room sized environment that allows for children as young as 4-6 to benefit from it. Since physical programming admittedly has qualities that are beneficial to scaffolding the learning, we believe that the iPad, a device whose interface and interactions are more physical by nature, would provide affordances similar to those mentioned in the above research.

Some researchers were focused on providing guidelines and examining proper ways of conveying this type of information; their conclusions and guidelines provided significant insights concerning the design of this application. More specifically, Lee et al. [2] stressed that it is important to aid learners in thinking in an abstract and generalized way while trying to minimize the effect of split attention, whereas Lin and Liu [20] provided guidelines on child-parent collaboration and demonstrated the positive and negative effects of the parent's involvement in the learning process. One of the most closely related papers, written by Morgado et al., [19] addressed several issues that related to ways of teaching programming to preschoolers through software and summarized the research and crucial points to be considered. Finally, Sipitakiat and Nusen [16] provided different manners of demonstrating and learning debugging concepts, while Wyeth and Purchase [18] suggested design criteria for activities that encourage the learning of CT.

Regarding the design of touch screen technology for children, McKnight and Fitton [21] provided guidelines on the terminology that needs to be used when giving instructions to young children in regards to touch screens, as well as some guidelines for the interaction methods. These guidelines proved extremely useful and significantly affected every iteration of the design process. Similarly, Revelle and Reardon [22] provided a set of guidelines for designing touch screen applications for young children while Sheehan et al. [15] suggested ways in which to take advantage of physics laws which are intuitive to children. Lastly, Kindborg and Sökjer suggested the use of voice and text for instructions while Druin et al. [23] provided an extensive list of challenges, guidelines, and ideas that relate to designing mobile technology for children, especially in regards to learning.

4.1 Limitations of the field

Though a wealth of research exists on teaching CT to children, a large percentage of it is often about older children. Since young children grow and change with a rapid pace, different software exists for different age groups, and knowledge about older children cannot directly be applied to younger children. As an example, Druin [23] mentioned that portable device applications are categorized into 11 separate age groups for ranging from 0 to 15. Furthermore, to this date, we could only find one product that attempted to help young children learn programming skills through the use of touch screens: Move the Turtle [24] is an iPad application that targets children five and up; however, parents have criticized it for not being designed in a way that is age appropriate, since it includes too much text and no auditory instructions. This project attempts to expand the field in two ways: by developing such an application for children younger than 5 years old, and by examining the implications of implementing it on a touch screen.

4.2 Related Concepts

In the process of creating this application, we sought to incorporate several concepts that are key to acquiring programming skills and developing CT. More specifically, elements from the following taxonomies were used:

Morgado et al. [19] provided the initial taxonomy for the design of the application:

1. Syntax and Semantics
2. Compound Procedures
3. Parameter Passing
4. Parallel/Concurrent Execution
5. Message Passing / Communication Channels
6. Input Guards
7. Clients and Servers

Of the above, the application attempts to assist the learning of 1, 2, 4 and 6. However, two more taxonomies were considered while designing the application, though not as strictly. Wyeth provides a taxonomy of programming topics [9]:

1. Syntax and functionality
2. Achieving specific outcomes through programming
3. Reusing parts
4. Debugging
5. Planning
6. How often children built and compared alternative solutions to programming tasks

Barr and Stephenson provide a useful list of CT skills [25]:

1. Design solutions to problems (using abstraction, automation, creating algorithms, data collection, and analysis)
2. Implement designs (programming as appropriate)
3. Test and debug
4. Model, run simulations, do systems analysis

5. Reflect on practice and communicating
6. Use the vocabulary
7. Recognize abstractions and move between levels of abstractions
8. Innovation, exploration, and creativity across disciplines
9. Group problem solving
10. Employ diverse learning strategies

Since some of these concepts are outside the range of what a 3-5 year old child can fathom, we attempted to design the application so that – for these concepts - it provides a precursory scaffold for their learning, so as to allow for a more fluid transition when they grow older.

Last, some concepts within the application are relevant to programming but were not part of the literature; they were however based on a researcher's personal experience as a programmer (e.g. the inability to alter code execution at runtime).

5 Description of the Product

We chose to implement this application on a mobile device, as mobile technology will soon be prevalent in classrooms, therefore the need for children to understand and use it is becoming more and more pressing [2]. Not only are mobile devices an emerging trend, but they can also be highly motivating for children since they can be used in the wild, thus increasing the engagement level of children in their learning processes [2]. These points made it clear to us that it is beneficial to create a touch screen application that will allow young children to develop their CT skills. More specifically, we chose to make an iPad application, since the iPad is large enough to allow for more content to be displayed on the screen without it looking crowded and confusing the users. We suspect that the slightly physical nature of touch screen devices like the iPad allows for a more immersive experience, based on the literature. The application was developed using XCode, Objective C, Phonegap, HTML, CSS, Javascript and the libraries jQuery and KineticJS.

The application is a game that consists of 5 levels (1 introductory and 4 progressive), a "Castle and Shop" and a "Replay Hall". When the game starts, an animation appears with a robot and a dog and the players are introduced to a short narrative with the goal of the game, which is to guide "Clinky the Robot" to his plug so that he can charge. A live version of the game is currently available on a browser (Google Chrome) at http://heypano.github.com/Capstone/ .

5.1 Interface

The interface (Figure 5.1) consistently consists of two panels: One square on the left where most of the interactions that relate to the game take place and one "control panel" to the right that provides information about the player's status and allows for additional instructions or switching state to "Castle and Shop", the "Replay Hall" or going back to the game if the game state is either of the latter.

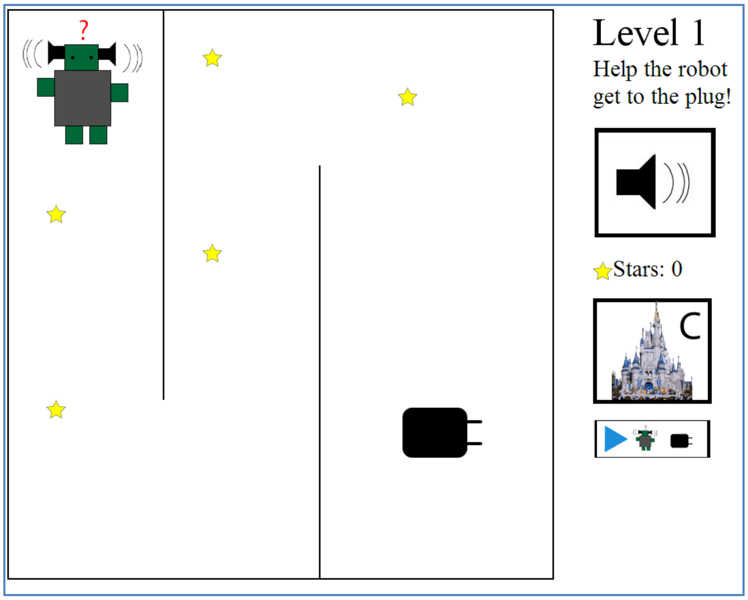


Figure 5.1 – Level 1

5.2 Level Progression

5.2.1 Introductory Level

When the starting animation is over, the narrator introduces the term “to program” and explains it in a simple way. Next, the player sees a robot and a plug and is asked to draw a line to “show the robot the path” (Figure 5.2). The player needs to then put her finger on Clinky and start dragging a line that will determine the path that the robot will start following when she takes her finger off the screen. If the robot does not reach the plug at the end of the path, it resets to its original position and the player can try again. When the robot hits the plug, it congratulates the player and moves on to the next level.

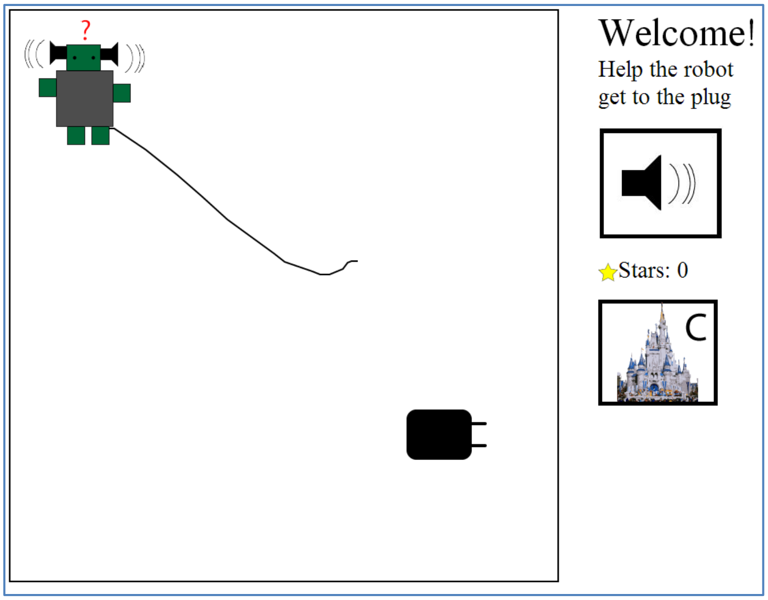


Figure 5.2 – Introductory Level

5.2.2 Level 1

Level 1 (Figure 5.1) starts with the narrator introducing the term “to execute”. This level follows exactly the same mechanics as the introductory level, with two differences; there is a simple maze that the player needs to solve and there are stars that are collected when the robot moves over them. Level 1 also introduces a new button that links players to the “Replay Hall”, which at this point allows for re-executing of the first program, i.e. it replays exactly what the player did in the introductory level

5.2.3 Level 2

Level 2 (Figure 5.3) introduces the concept of “debugging” by explaining the term and informing the player that the robot has made a mistake. After the player detects the mistake, the robot provides her with positive reinforcement and the level continues in the same way as the previous ones.

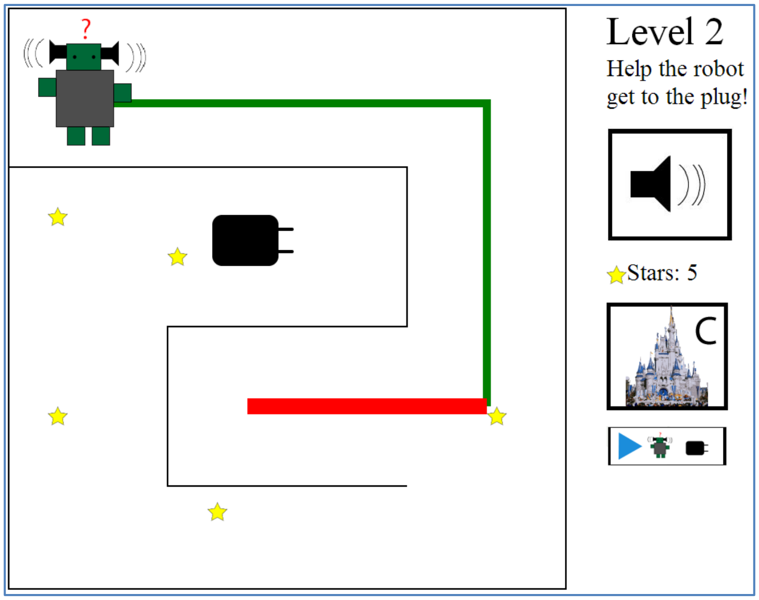


Figure 5.3 – Level 2 (Debugging)

5.2.4 Level 3

Level 3 (Figure 5.4) starts with explaining the concept of "Parallel Execution" verbally and the player is asked to execute their program "in parallel" without getting in the way of the other programs, i.e. draw a line to the plug without hitting the balls that are moving in a loop.

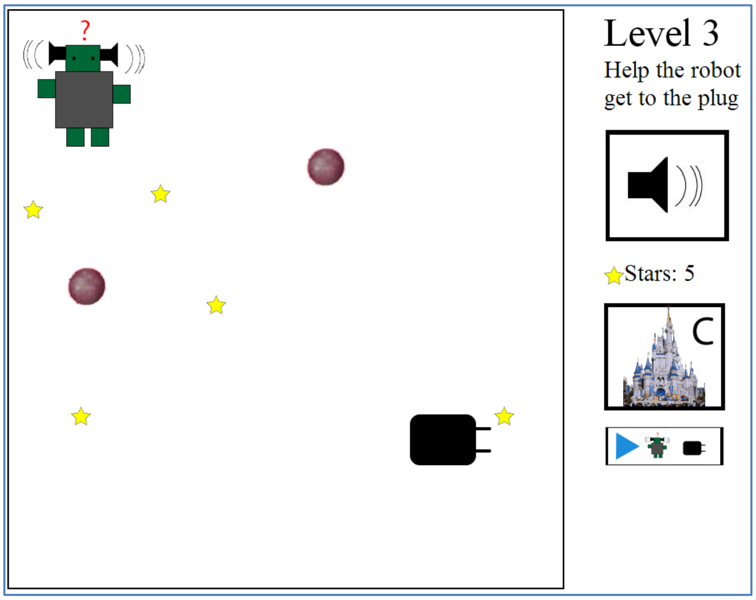
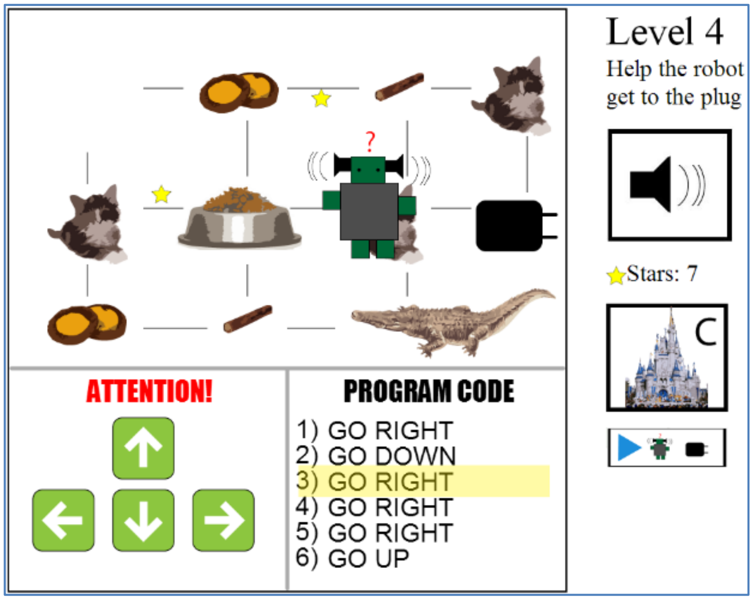


Figure 5.4 – Level 3 (Parallel Execution)

5.2.5 Level 4

Level 4 (Figure 5.5) is the last and most advanced level of the game, as it introduces a "command list" that closely resembles an actual program. The player can no longer draw a path, but has to help Clinky traverse through a grid of various items by giving him a list of 6 commands. This new type of interaction is shown to the player at the beginning of this level, through an animated image showing a simulated interaction with a drawn hand. When the player touches one of the arrow buttons, the robot says “Execute command X”, where X is left, right, up, or down. Once the list has exactly 6 commands, Clinky starts following the instructions, either by going where the current instruction says, or by saying “I can’t go that way”. The command that is currently being executed is highlighted in yellow and once the robot reaches an item in the grid, it makes a sound (e.g. cat makes a meowing sound). If the robot has not reached the plug by the end of the execution, it resets and the player can start with a new set of commands. Once the robot reaches the plug, the player is given the title of “real programmer”, is rewarded 15 stars and is redirected to the replay hall to execute all 5 programs that she wrote.



```
1) GO RIGHT
2) GO DOWN
3) GO RIGHT
4) GO RIGHT
5) GO RIGHT
6) GO UP
```

Figure 5.5 – Level 4 (Final)

5.2.6 The Castle

The castle (Figure 5.6) is available at any stage of the game and its goal it to motivate players to play the game, since it lets them buy “upgrades” for their Castle, (which is where the robot lives) or for the Robot (each upgrade costs 3 stars). If the player buys a robot upgrade, the robot will upgrade in the actual game as well (the image changes).

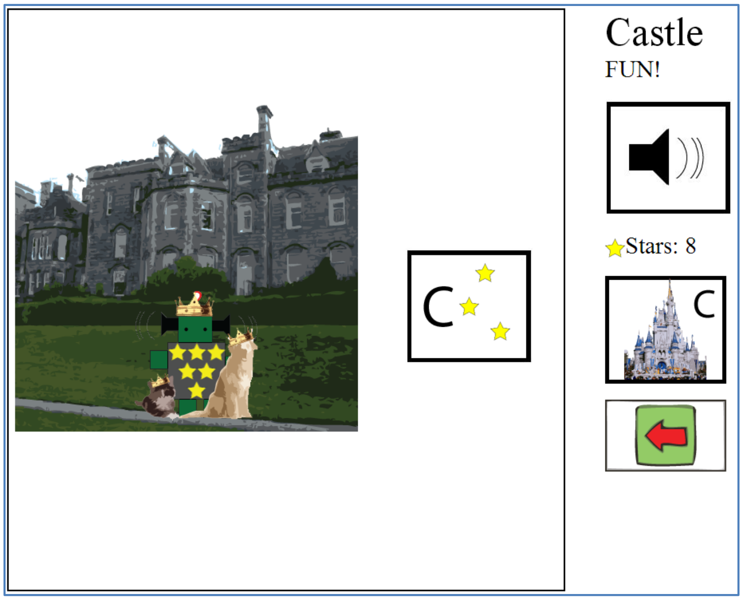


Figure 5.6 – The Castle

5.2.7 Replay Hall

The “Replay Hall” (Figure 5.7) is a space that is available to players after completing the introductory level, and it allows them to re-execute programs. Through executing the levels separately, this function aims to clarify the concept of a “compound procedure” or a “standalone program” with the aid of the terminology used in the auditory instructions.

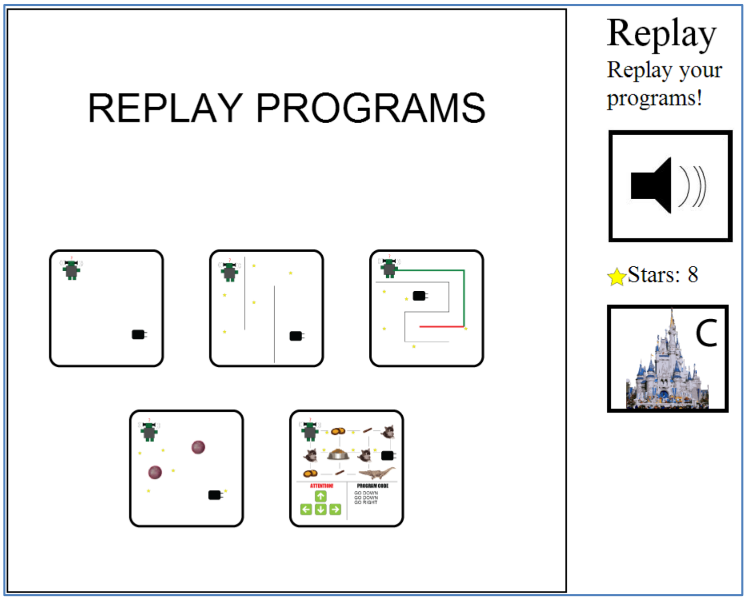


Figure 5.7 – Replay Hall

5.3 Instructions

As suggested by Ravelle and Reardon [22], there are three types of repeated instructions throughout the game, Auditory, Visual and Textual. More specifically, the auditory instructions are split into 2 categories: narrator instructions - which are triggered either by starting a new level or by clicking the speaker button, and robot instructions – which are triggered as a reaction or when a new level starts. Different voices are used for the narrator and robot in order to make the instructions from Clinky feel more personal. For example, the programming vocabulary, which is used to enhance the learning, is introduced by the narrator, whereas reactions to game events use the robot voice. Visual instructions are provided in the form of an animated image that appears after multiple times of failure or before a new type of interaction is introduced (level 4). In this way, if a child is unsuccessful at using the application, a simulated scenario is shown, where a hand is performing the task required for the specific level. Textual instructions are only used sparsely in order to provide for reading opportunities and it is never expected that the player will read them, therefore they provide no additional information.

5.4 Connection to the concepts being learned

One of the most fundamental concepts in programming is that of execution after instruction. As opposed to controlling machines in a conventional way, programming requires the entire list of instructions and possibilities to be predetermined before the execution begins. For the first 4 levels, the entire path that the robot will take is defined before the robot starts moving. In the last level, this concept is conveyed in its most elaborate form, since an actual written list of instructions with set syntax is given and written down before the robot starts moving, which closely simulates real-life programming. Another concept that is consistently present throughout all of the levels is that the program code (instructions) cannot be altered while the program is executing – during runtime. This idea is conveyed through the inability to interact with the system while the robot is moving (which includes not being able to change the path). Furthermore, puzzle solving, a developmental precursor to CT, is part of all the levels as well. Last, in terms of programming vocabulary, the first four levels introduce and use four programming terms, one at a time: “to program”, “to execute”, “to debug” and to execute “in parallel”.

Some programming concepts are also directly included in the game; level 2 introduces the concept of debugging, since it shows that execution is failing and asks the player to find the error, whereas level 3 introduces the concept of parallel execution where the player is asked to execute their program in parallel without getting in the way of the other programs, which are in a loop. Last, level 4 attempts to help the players gain a basic understanding of Syntax and Semantics, through written instructions that follow a specific “grammar”, and buttons that correspond to specific commands. The replay hall lets the player “rerun” her programs, which we believe will help create a sense of “compound procedures”, as the player has the ability to execute the different functions that she has created.

6 Design and Evaluation

6.1 Summary of the methods used

An iterative process was used for the design of the application, to allow for major changes to be made based on the various types of input acquired. All of the methods used followed formative research techniques given that the product would not be finalized at the end of our research. During the iterative design phase, we mainly used participatory design with slightly older children (Kidsteam: ages 7-11) in two separate sessions. This provided a chance for different aspects of the design to be examined and it included children whose age is not exceedingly older than our target group but old enough to provide us with useful insights. Hence, they were able to provide us with valuable input regarding “what they would have wanted a few years ago” or “what would their younger sibling want”.

A great amount of the knowledge mentioned in the related research section of this proposal was also used, not only to improve the interactions, but also to ensure proper learning methods are being applied. To conclude the iterative design process, we conducted formative evaluation with teachers and children at the University of Maryland Center for Young Children (CYC). We initially demonstrated the application to 4 teachers and received feedback on how to make it age appropriate, as well as how to optimize scaffolding. Since the teachers are experts in educating children of this age, they provided us with a valuable type of input, which would be difficult to acquire otherwise. A formative evaluation with a few target users followed (i.e., 3 – 5 year old children at the CYC) in order to assess issues that did not appear in previous stages of the iterative process, which lacked target users. Between the various stages of the design, the prototype, created using Adobe Illustrator, was iterated on within our team by using feedback from the sessions.

6.2 How the literature informed the design

The literature informed the first (and subsequent) versions of the design in multiple ways. In order to provide clarity and to aid children with their reading skills [23, 22], this design included multiple repeated auditory instructions, as well as textual legends. The activities were carefully chosen to cover multiple aspects of programming based on the taxonomy used by Morgado and Cruz [19] .The introductory level is simple and repetitive in order to allow users to familiarize with the environment. Level 3 then introduces the idea of debugging. A dynamic world (customizations, upgrades) that allows for individual expression is created by giving children the ability to upgrade their robot and castle. Furthermore, large, visually distinct hotspots are used (e.g. very large buttons) and new interactions are introduced with demonstrations and animations [26, 5, 22, 15]. The terms used to describe touch screen gestures in the instructions (mostly auditory) are compliant to what McKnight and Fitton suggest [21].

6.3 Kidsteam

Kidsteam is a design team consisting of adults and children (ages 7-11) featuring collaborative design sessions in the Human-Computer Interaction Lab, at the University of Maryland. In October 2012, Kidsteam consisted of a team of eight children and seven adults.

During the course of this project, we performed two sessions with Kidsteam to inform our design using cooperative inquiry methods and techniques [6, 27].

6.4 Design Session 1: Robot activity & drawing of iPad application

The first session with Kidsteam was extremely open ended because our goal was to explore the general patterns of what the children wanted to see in the application we sought to design. Initially, we asked the children, and the adults, "What does the word programming make you think about?" to see how well they understood the term and the concept. After receiving their responses and displaying them on the whiteboard, we explained to them what programming is in simple terms. When they seemed to have a serviceable understanding, we initiated an activity wherein one of the researchers was a "simple robot"; the goal of this activity was to demonstrate different aspects of programming to the children in an attempt to provide them with a more holistic view. The simple robot could only take simple commands. We asked the children to give her commands while two researchers wrote the different commands that the children mentioned. Their goal was to get the 'robot' from one spot in the hallway to another, while avoiding obstacles. After they completed the task successfully, we gave them the list of commands that were derived from the activity, which they had to put in the right order so as to write a 'list of directions' that would always take any robot from the starting point to the end (a map was used for help). Their instructions were simultaneously being drawn on the map to show the results of their commands. Finally, the children were split into 4 groups of 2 (with accompanying adults) and were asked to draw an iPad application to help younger children learn how to do something similar to what they just did (Figure 6.1). The children were specifically asked to imagine what they would want if they were younger and what they would design for their younger sibling.

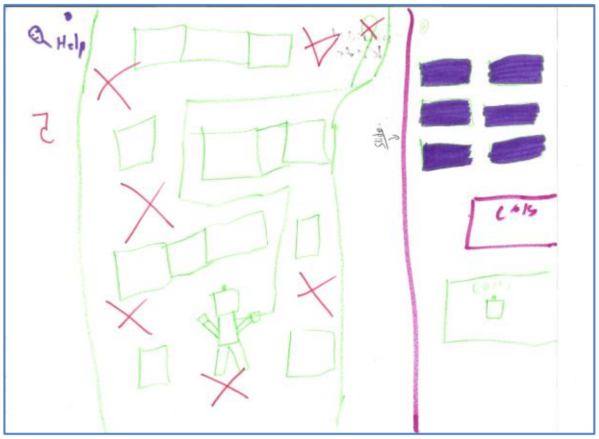


Figure 6.1 – Example of drawing from Kidsteam Session 1

The analysis session was done in three parts. First, while the design ideas were being presented, a researcher asked questions and wrote everything that was described on a whiteboard. After all the teams had presented, five researchers and the children talked about overarching themes and "tagged" the whiteboard with "big ideas" using symbols to represent similarity. Second, shortly after the session was completed, the researchers debriefed while analyzing their observations (written or otherwise) and discussed their ideas and how they relate to each other and the process. Last, the artefacts were collected (e.g., notes from the researchers, drawings and pictures) and analyzed and coded by one researcher using open coding principles [28, 29].

When answering the question "what does the word programming make you think about," it became clear that all the children knew it related to technology and/or video games. That said, most did not have a clear understanding of why programming exists. The robot activity indicated that the children may have lacked a clear understanding of what programming is, since they could not disassociate it from the robot or make any inferences about other areas to which it would apply. However, they were familiar (or became quickly acquainted) with the concepts of syntax, semantics, iteration, parameter passing, and compound procedures [19] after they were shortly explained to them. Even though the children were prompted to let their imaginations run rampant during the drawing phase, the prototypes that they designed revolved around robots. We believe that this fact implies that they associate programming with technology and machines and that they liked robots, but not that they did not grasp the concept of programming.

As far as game mechanics were concerned, separate levels and modules were present in all of the group designs (as opposed to a sandbox world or other game mechanics). The levels did not only signify progression in the game, but also difficulty (e.g., commands like turn clockwise might not be understood by all young children). All groups included an element of collecting items in their design, as well as obstacles and/or traps. Another overarching theme was development i.e., upgrading and customizing your robot (or possibly other character). Furthermore, castles, mazes, and teleporting were mentioned often, while other ideas included dodgeball and racing. An interesting observation was that all of the teams had draggable interface elements. Finally, keeping in mind that they were designing an application for children that cannot read, most of the teams focused on large pictures on buttons and/or reading things out loud multiple times for the user.

6.4.1 Effect on the design

The results from this session were used by a researcher that designed the first wireframe of the application. This wireframe was heavily based on this data in multiple ways: Robots are a major concept in the game, while a castle defines the environment of the world and different creative tasks are involved such as climbing stairs or playing dodgeball. The maze (levels 1-3) is important in understanding the basics of programming and was mentioned frequently during the Kidsteam session. Furthermore, the robots are customizable and upgradable (with progress in the game) since each group's designs included these elements. The design (See Appendix 1) is simplistic and not overly colorful in order to allow for brainstorming during the next Kidsteam session, which will be centered on the method of layered elaboration [11], in which the initial design should not be too complex.

6.5 Kidsteam Design Session 2: Layered elaboration

The second Kidsteam design session brought back the initial wireframe for rapid iteration and evaluation with the adults and the children that participated in the session by using the technique layered elaboration [11]. The goal of this session was to iterate on the initial wireframe design using feedback from a design team that includes experienced participatory designers close to the target user age. The wireframe now contained 7 pages that were mainly black and white and not crowded with content, which is suggested for this technique in order to get a high level of input from the children. The pages consisted of 5 levels of progressing difficulty, a ‘castle’ that they own and can customize, as well as a ‘shop’ where they are able to buy parts for a ‘robot’ character that belongs to them and their castle.

As in the first session, the participants were asked to answer a ‘question of the day’, which in this instance was “tell us about a time when you gave instructions to someone”. The goal of this question was to indirectly reintroduce the concept of programming to the children, and to observe once more how they think about instructions.

We then showed and explained each ‘page’ of the wireframe to the children. This step was important because a lot of the interactions that were part of the design were not visible on the wireframes. These interactions included animations that would demonstrate how to perform a certain gesture to achieve a certain task, sounds, and automated prompts that would help users understand the goal of each level. The children were split into groups and asked to draw things that they want to add or change, make suggestions on how the auditory instructions would work and how they would be phrased, as well as find their own ways to help the young children understand the goal of each ‘page’.

The layered elaboration part of this session consisted of rotations of all the designs within the groups. During each rotation, each group was given a page with a transparent sheet on top and permanent markers to draw with. The researchers participating in this session were asked to take extensive notes about the children’s ideas and quotations. The groups had 3 minutes to elaborate on each design, after which a ‘standup’ meeting took place where each group was given 1 minute to explain what they did and why. Subsequently, each group was given another design to elaborate on, often one that had already been augmented by another group, with a second transparency on top, so that the groups’ individual designs were not distorted (Figure 6.2).

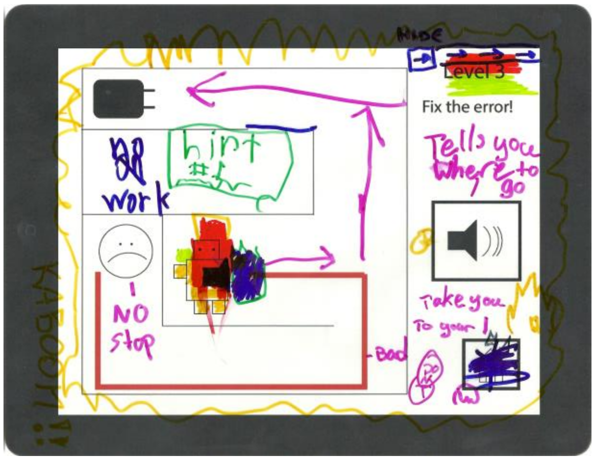


Figure 6.2 – Layered elaboration design

A debrief meeting took place a few minutes after the children had left, where each researcher presented their observations, comments, and advice. The researchers also talked about overarching themes that they observed, which were written on a whiteboard and discussed extensively. Later in the day, the designs and the notes written by the researchers were scanned, transcribed (where applicable) and analyzed by one researcher in an attempt to extract more overarching themes, as well as specific design changes and additions for each level and the overall structure of the game.

When asked to think of a time when they were given instructions, two of the children instantly mentioned the previous Kidsteam session, where they were programming the researcher acting as a robot. This suggests that the robot activity in Session 1 functioned in a desirable manner, in that the children connected the concept programming to that of giving instructions. Interestingly, two children also mentioned teaching their pets how to perform certain complicated tasks. One child mentioned that he gave his parents explicit instructions on how to help him get off his bed when he was injured.

An idea that was uniformly embraced by all children in the groups was that of including animals in the game and giving them functions. The designs also suggested that the children want surprises like bursts and explosions. The element of customizability also emerged as the children expressed desire to color their robots or adjust the game to their playing style. Furthermore, the children had a strong desire to interact with the robot on a personal level; the robot would give them instructions, praise and positive feedback. Last, they requested that limited “easy ways out” be given to them, in case they got stuck on a specific level.

6.5.1 Effect on the design

The second iteration of the design (See Appendix 2) was heavily informed by the Kidsteam Session as most additions were based almost exclusively on the elaboration of the ideas that the kids designed during the session: The children explicitly mentioned that they wanted the robot to talk to them directly, but they did not want the narrator to go away, which is why two different voices were implemented. Furthermore, if the player fails to succeed for 7 consecutive times, an animation pops up showing what needs to be done for each specific level. Some stylistic changes were also based on feedback for the children, e.g. the home button now looks like a castle with a “C”. This session introduced the concept of currency in the form of stars that are collected, and the concept of buying upgrades. Animals were also added to the game, as upgrades for the robot, but also as part of the grid in level 4.

6.6 Formative evaluation with experts (Teachers)

In order to acquire feedback from experts of a different field (that of youth education), we conducted two 30 minute semi open-ended interview sessions with a pair of teachers (recruited from the CYC). The first pair of teachers interviewed taught three and four year olds, while the second pair taught five year olds. All four teachers have been in the profession for three to five years, and they have all worked with iPads in the classroom for about six months. The goal of this session was to gain information on two levels: the quality of the interactions and the quality of the learning. The perspective of educators with experience working with touch screen devices is especially valuable, since none of the researchers had significant experience working with children of ages 3-5. We chose to interview the teachers in groups of two, in order to allow for group discussion and support elaboration on ideas in a way that we considered a more productive type of feedback. Each session began by giving the teachers one iPad with the application (after explaining its goal) and asking them to provide us with feedback on how the children would respond. We were interested in answering the following questions:

- What aspects of the application are age appropriate and what are not?
- How can we improve the usability, learning potential and fun of the application?
- What types of challenges would we expect in terms of learning?
- How would the technology support or hinder the children in their learning?
- What are the determining factors for whether they would use it in a classroom?
- How would they use it in a classroom? Would it be a guided or an independent activity?

Since the interviews were open ended, the teachers were also asked to provide us with any other type of feedback they felt would benefit us. The sessions were not recorded, but the researchers present took hand-written notes with observations, quotes and comments. After the sessions were completed, debriefing took place and the researchers observed themes on the notes through an informal frequency analysis.

One of the most interesting findings from this session was that the teachers working with 3 and 4 year olds had significantly different answers to our questions from those working with 5 year olds, which is why we are presenting the results separately.

6.6.1 Teachers of 3 and 4 year olds

In terms of age appropriateness, the teachers felt that most of the content on the application was at the right level. It would aid the children in developing hand-eye coordination, as well as with their problem solving capabilities. In terms of learning, the teachers mentioned that they would prefer a smoother transition between the levels, as well as the ability to replay the levels multiple times since repetition helps children learn and understand concepts much better. They found that Level 4 contained too many components that 3 year olds would have trouble with, like understanding spatial directions and processing more than 3 instructions at once (this level requires 6). A possible solution that they suggested for this problem was to add a duplicate of this level before it, with the only difference being that the robot moves immediately after receiving one command (instead of waiting for the whole list).

As far as the interactions are concerned, they felt that the majority of them were at the right level for the target age group. However, they mentioned that tracing a line might be difficult for some 3 year olds, although that is countered by the fact that they do not need to trace any specific line – as opposed to a letter tracing game that they are currently using in the classroom. They also believed that the children would need to be told explicitly what the buttons are and why they would click them, because they would be distracting. Finally, they suggested only allowing them to visit the castle in certain times.

Overall, the teachers felt that the application would support the children in their learning, especially if the activity is repeated multiple times. They mentioned that they are not certain the children would understand the difference between “replaying” and “running a program”, but we believe the semantic distinction between the two terms does not obscure the fundamental understanding of the concept, as the conceptual root is that the instructions they gave once will have exactly the same effect under the same conditions. The teachers strongly believed that the children would find the application engaging and fun.

When asked whether they would use it in a classroom, they said they would not because it does not look like it would fit the curriculum that they are assigned to follow; They also mentioned that they currently only use iPads for one type of activity (letter tracing) and that they cannot fully integrate them into their everyday activities, especially because they would not know how to assess them on it. The teachers proceeded to provide us with their curriculum in order to study it and determine what aspects of it might fit with adjustments. Finally, when asked how they would use the application in a classroom, they mentioned that due to the way children of ages 3 to 4 think, it is difficult for them to participate in an independent activity, since they need all expectations that the teacher, parent or application have to be predefined explicitly and elaborately. Therefore, they would use this application only in the form of a guided activity, where at each stage an adult would be present to tell them exactly what they need to do and why, what is expected of them and give them instructions (even if the instructions are just the equivalent of “Listen to the instructions that the robot is giving you”).

6.6.2 Teachers of 5 year olds

The second group of teachers also found the application to be fitting in terms of age appropriateness. They also mentioned that the problem solving capabilities of the application are worthwhile and they did not find anything that was too advanced for 5 year olds. On the contrary, they believed that it required the right amount of logic and problem solving skills. In terms of learning, this group of teachers also mentioned that they would like a smoother transition between the different levels in order to allow for repetition of similar tasks while introducing at most one new small concept at a time. Last, they suggested the same thing the first group of teachers suggested for level 4, i.e. to add a duplicate before it that allows for instructions to be executed on button press.

In terms of interaction, as with the first group of teachers, they believed that the majority of interactions were appropriate, but they felt that it would be useful to demonstrate each level to the children before asking them to do it, especially when introducing a new goal or type of interaction in order to not confuse them. They mentioned that they would like the flow of the application to be slightly faster and that they would like more graphics to be present. Furthermore, they felt that the majority of the audio in the application, while of good quality, takes too much time and might allow for distractions that could harm the learning.

Overall, they considered the application to be of high quality and mentioned that the children would particularly enjoy the voices and the pictures, as well as the reward system and the castle, and they mentioned that the choice of “charging” as an activity is appropriate for this generation of children, since it is something that directly relates to their everyday lives. They also mentioned that the way the vocabulary was being presented to the children was age appropriate, and they would like to see the terms spelled out for the children as well. They believe the children would like to see some more “extras”, like being able to visit the insides of your castle, or buy things for your garden.

In contrast with the first group of teachers, the second group explained that they would definitely use the application in the classroom. More specifically, they would use it as an independent activity since they did not find it to have content that the children would fail to understand by themselves, especially if the instructions are slightly more elaborate. Since helping 5 year olds be independent is one of the goals that the teachers have in the curriculum, they would use this application for this purpose and to provide them with problem solving and reading opportunities. Furthermore, they mentioned that they might use it in an activity where the children would work in pairs, since they could help each other fill in possible gaps in comprehension of the goals and mechanics of the application, as well as motivate them. Last, the teachers mentioned that this application, if expanded properly could be beneficial for children up to 7 years old.

6.7 Formative Evaluation with young children

After the implementation of the product had been completed, we conducted a 25 minute session with two target users, a 3 year old girl and a 5 year old boy that were part of a class at the CYC. The boy has an iPad at home that is only used for “bed time TV”, whereas the girl has only used the iPads at the CYC in the context of their activities. As the teachers suggested, we chose to conduct a session with 2 children at the same time so that they could help and motivate each other. The goal of this formative evaluation session was to examine how our actual target users would respond, as well as to get an idea about the possible differences between 3 and 5 year olds. More specifically, we were interested in answering the following questions mainly through observation, but also through direct feedback from the children:

- What areas are the children having trouble with (and why)?
- Do they understand the interface as well as the concepts well?
- Do they like the application and its interactions?
- How can we improve the usability, the learning interactions and the fun?
- What challenges might the children face?

The session started with the researcher introducing himself to the children, followed by a short explanation of what would be given to them and their right to stop at any time. Then, the children started playing with the application, while the researcher guided them through the activity by telling them what to do next or helping them if they were having trouble. Throughout the session, the researcher asked questions to clarify that the children understood what they were doing, e.g. asking them to explain what they did or by asking context specific questions like “Why did the robot move there.” Whenever the children expressed an opinion or provided the researcher with feedback, the researcher attempted to help them elaborate on it and perhaps get their opinion on how to improve it. At the end of the session, the children were asked to rate the application based on how much they liked it using a 5-likert scale with smiley faces (the Smileyometer [30]) and also to suggest improvements. The collected data consisted of the researchers' written notes and observations.

Overall, the application was very appealing to both children. Most of the feedback that we received from the two pairs of teachers was reflected clearly throughout the session. More specifically, the majority of the comments of the first pair of teachers applied to what was observed in the way the 3 year old girl was using the application whereas most of what the second pair of teachers said was reflected in what the 5 year old boy was doing. Moreover, the 3 year old girl was having some trouble tracing the line, as her finger would lose contact with the iPad and unwanted results would occur, something that was not an issue with the 5 year old boy. This was, however, not a significant source of frustration for the girl as she was able to complete all of the tasks. While the boy mostly interacted with the iPad independently, the girl inquired for instructions multiple times on each level ("What do I do now?"). In a manner similar to what the teachers predicted, the researcher's observations suggests that the girl did not attempt to understand the application independently, but she would only expect explicit instructions given to her by the adult present. The boy, on the other hand, reached the last level of the game before asking for help.

The difficulty of the content and the concepts of all levels seemed to be age appropriate for both children, with the exception of level 4 for the 3 year old girl. The boy was able to explain what was happening when asked, after the level was demonstrated to him, whereas the girl could not recreate the result or explain the cause and effect.

Neither of the children clicked any of the buttons on the side without the researcher suggesting it. Similarly, when they went to the castle, the researcher had to explain what the castle is and how it works due to insufficient instructions. However, once the children had understood how the castle and the stars worked, they constantly wanted to go back to the castle and buy more things. Both children, but mostly the 3 year old girl often ignored the verbal instructions because they were depending on the researcher to answer any question that they might have.

One of the most interesting observations was that after the two children had finished the game, which happened in less than 5 minutes, they wanted to replay it immediately and were excited to do so. This happened multiple times, which is why each child played the game a total of 6 times. Even after the 6th time, the children wanted to play the game again, but there was no time left. When asked why they wanted to replay the game, they said they wanted to "get more stars so they can buy more things for their castle". This proves that collecting stars and getting things for their castle became their motivation even though they were never explicitly told that they need to collect stars or to improve the castle and robot.

On the whole, both children seemed excited and engaged, as they were not once distracted by their surroundings. Similarly, the robot voices made them laugh and they seemed to be enjoying the visuals. Once during the session, the girl looked like she was getting bored of the application; however, she explicitly asked to keep on playing with it. When the children were asked to rate the application based on how much they liked it, they both immediately gave it a 5/5 (i.e., the face with the widest smile). Finally, as far as direct feedback and suggestions are concerned, it proved difficult to keep the girl engaged and help her express what she would want. However, the boy explicitly said that he wanted the introductory level to have stars and that he wanted more levels and the ability to replay the levels.

7 Discussion

This project was attempting to touch on the space of creating a touch screen application for very young children, with the goal of helping them acquire skills that are fundamental to programming. In achieving this goal, we believe that we proved such a task to be achievable and that the children enjoyed it. We believe that in order to design such an application children need to be involved in all stages of the design process. With proper scaffolding, children as young as three years old are not only able to perform programming-like tasks, but they also enjoy it. If such an application succeeds in helping scaffold the learning of programming skills, the children will benefit from it in terms of early development. Furthermore, we believe that if products such as these became widely available, more people would be interested in the field of computer science. Increased interest in computer science allows the field itself to evolve due to a more diverse set of people and skillsets. Finally, an interesting topic to consider is how such an application could connect to the objectives of a school curriculum, or how it might be used in the context of a family.

8 Limitations

Given the limited time and resources available, the last step in our research was a formative evaluation. In order for more concrete results to be established, fully evaluative sessions would have to be conducted and the product would have to be iterated on two or three more times. As it is, it is difficult to safely say that the application successfully helps children develop programming skills. In order to fully establish whether the application actually succeeds in scaffolding the learning, it would require a long term case study to be conducted, which was beyond the scope of this capstone project. Last, the application only delved into few concepts and provided few levels for the children. In order for the product to be complete, more levels would be required, and a more structured reward system developed, to allow for longer play-time, as well as to encourage players to return to the application.

9 Future Work

Future steps on this project would be heading towards two different directions: Improving and expanding the application itself and evaluating it on a larger scale through a long term case study.

In terms of changes to the application we believe that an important change would be that the application should adjust itself so that it behaves differently based on the age of the participant, since the sessions with the teachers, the session with the children, as well as the literature suggest that the differences between a 3 year old, a 4 year old and a 5 year old are vast and significant. More specifically, we believe that there should be four categories (3 years old, 4 years old, 5 years old and older than 5). This would involve not only adapting the interactions themselves (e.g. tracing would be made easier for 3 year olds), but also providing age appropriate concepts. Furthermore, various adjustments would need to be made to enhance the usability of the game as well as the instructions based on the feedback from the teachers and children (e.g. less delay, making level 4 simpler etc.). Last, one significant addition would be the ability to repeat levels and expand on them, since this was not only something the teachers said, but it was also obvious that they children desired to replay it. In order for all of these changes to be carefully studied and decided, more formative evaluation would have to be conducted with the actual target users that would be aiming to answer more specific usability or learning related questions. This application could also be expanded or changed towards two different directions: It could be re-designed for paired, cooperative activities, or so that it could adapt better to the curriculum of preschools.

Since the field of programming related touch screen applications for children is extremely new and unexplored, future work could also involve conducting research with the exclusive goal of providing guidelines for such a game or application. This research could be split into two different parts: The first part being studying the types of interactions that are appropriate, including instructions, game mechanics and information architecture, the second part being studying in depth which programming related concepts are appropriate for which ages, and how best to convey them through the touch screen application.

10 Conclusion

Children of ages as young as 3 years old can start developing skills that will later on support them in learning how to program and provide them with a good base for CT. In the process of conducting this research we created an application that attempts to address this void in the knowledge community. The data collected from sessions with teachers of young children, as well as from the sessions with children of ages 3-5 converges to the same conclusion: Children of that age want to program, and they certainly enjoy doing it on a touch screen application. The application that was built in the context of this research could be expanded and used in the classroom, either independently or with teacher guidance. Furthermore, the capabilities of children that are 3 years old are vastly different from those that are 5. Hence, there are design implications that need to be carefully considered when designing such an application, instructions need to be given at all times through multiple means such as audio, video, pictures and text and expectations need to be managed.

11 Acknowledgements

Special thanks to Mona Leigh Guha and Tamara Clegg for forming this research, actively participating, providing extremely valuable help, and supporting me throughout the entire process of conducting this research. I would also like to thank Leah Findlater for her valuable feedback and guidance, as well as Kidsteam, the Center for Young children faculty, staff and children, as well as my classmates and all the people that participate in the Human-Computer Interaction lab for their support and guidance.

12 References

[1] J. Montemayor, *Physical programming: tools for kindergarten children to author physical interactive environments,* University of Maryland, College Park, MD, USA: Thesis, 2003.

[2] T. Y. Lee, M. L. Mauriello, J. Ingraham, A. Sopan, J. Ahn and B. B. Bederson, "CTArcade: learning computational thinking while training virtual characters through game play," in *Proceedings of CHI EA 12*, Austin, TX, USA, 2012.

[3] S. Papert, Mindstorms: children, computers, and powerful ideas, New York, NY, USA: Basic Books, Inc, 1980.

[4] J. Goode and J. Margolis, "Exploring Computer Science: A Case Study of School Reform," *ACM Transactions on Computing Education (TOCE),* vol. 11, no. 2, 2011.

[5] K. Kahn, "Generalizing by removing detail: how any program can be created by working with examples," in *Your wish is my command*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 2001, p. 21–43.

[6] A. Druin, "Cooperative inquiry: developing new technologies for children with children," in *Proceedings of CHI 99*, Pittsburgh, PA, USA, 1999.

[7] B. J. DiSalvo and A. Bruckman, "Questioning video games' influence on CS interest," in *Proceedings of FDG 09*, Port Canaveral, Florida, USA, 2009.

[8] A. Fisher and J. Margolis, "Unlocking the clubhouse: the Carnegie Mellon experience," *ACM SIGCSE Bulletin,* vol. 34, no. 2, pp. 79-83, 2002.

[9] P. Wyeth, "How Young Children Learn to Program with Sensor, Action, and Logic Blocks," *Journal of the Learning Sciences,* vol. 17, no. 4, pp. 517-550, 2008.

[10] A. Fisher and J. Margolis, "Unlocking the clubhouse: women in computing," in *Proceedings of SIGCSE 03*, Reno, NV, USA, 2003.

[11] G. Walsh, A. Druin, M. L. Guha, E. Foss, E. Golub, L. Hatley, E. Bonsignore and S. Franckel, "Layered elaboration: a new technique for co-design with children," in *Proceedings of CHI 10*, Vancouver, CA, 2010.

[12] D. C. Smith, A. Cypher and L. Tesler, "Programming by example: novice programming comes of age," *Communications of the ACM,* vol. 43, no. 3, pp. 75-81, 2000.

[13] M. Kindborg and P. Sökjer, "How preschool children used a behaviour-based programming tool," in *Proceedings of IDC 07*, Aalborg, Denmark, 2007.

[14] J. Maloney, M. Resnick, N. Rusk, B. Silverman and E. Eastmond, "The Scratch Programming Language and Environment," *ACM Transactions on Computing Education,* vol. 10, no. 4, 2010.

[15] R. Sheehan, D. Cho and J. H. Park, "Improving on a physics-based programming system for children," in *Proceedings of IDC 12*, Bremen, Germany, 2012.

[16] A. Sipitakiat and N. Nusen, "Robo-Blocks: designing debugging abilities in a tangible programming system for early primary school children," in *Proceedings of IDC 12*, Bremen, Germany, 2012.

[17] P. Wyeth and H. C. Purchase, "Tangible programming elements for young children," in *Proceedings of CHI EA 02*, Minneapolis, MN, USA, 2002.

[18] P. Wyeth and H. C. Purchase, "Using developmental theories to inform the design of technology for children," in *Proceedings IDC 03*, 2003.

[19] L. Morgado, M. Cruz and K. Kahn, "Preschool Cookbook of Computer Programming Topics," *Australasian Journal of Educational Technology,* vol. 26, no. 3, 2010.

[20] J. M.-C. Lin and S.-F. Liu, "An Investigation into Parent-Child Collaboration in Learning Computer Programming," *Educational Technology & Society,* vol. 15, pp. 162 - 173, 2012.

[21] L. McKnight and D. Fitton, "Touch-screen technology for children: giving the right instructions and getting the right responses," in *Proceedings of IDC 10*, Barcelona, Spain, 2010.

[22] G. Revelle and E. Reardon, "Designing and testing mobile interfaces for children," in *Proceedings of IDC 09*, Como, Italy, 2009.

[23] A. Druin, Mobile Technology for Children: Designing for Interaction and Learning, San Fransisco, CA, USA: Morgan Kaufmann Publishers Inc, 2009.

[24] "Move the Turtle," Geek Kids, [Online]. Available: http://movetheturtle.com/. [Accessed 2013].

[25] V. Barr and C. Stephenson, "Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?," *ACM Inroads,* vol. 2, no. 1, pp. 48-54, 2011.

[26] A. Cypher, D. C. Halbert, D. Kurlander, H. Lieberman, D. Maulsby, B. A. Myers and A. Turransky, Watch what I do: programming by demonstration, MIT Press: Cambridge, MA, USA, 1993.

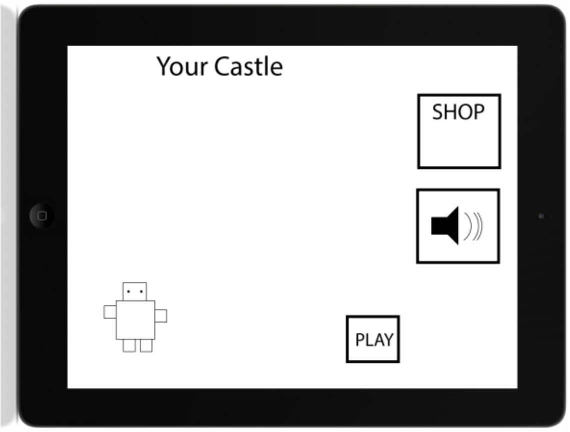
[27] M. L. Guha, A. Druin, G. Chipman, J. A. Fails, S. Simms and A. Farber, Mixing ideas: a new technique for working with young children as design partners, College Park, MD, USA: Proceedings of IDC 04, 2004.

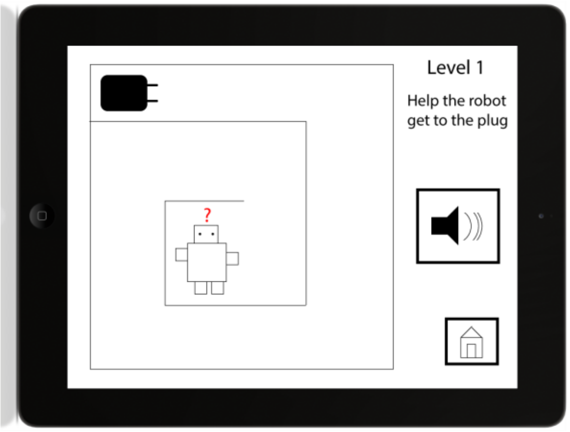
[28] J. Corbin and A. Strauss, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, Sage Publications, Inc, 2007.

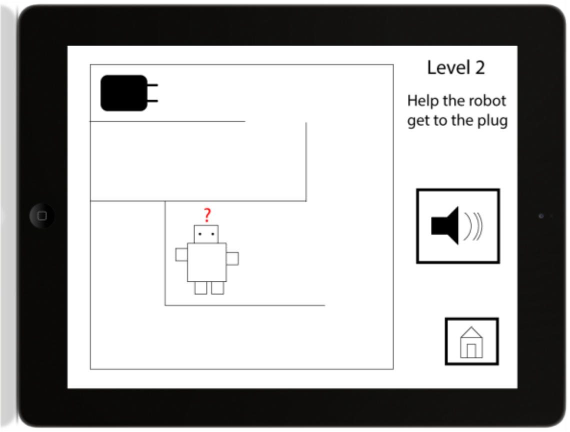
[29] A. Strauss, "Open Coding," in *Qualitative Analysis for Social Scientists*, Cambridge University Press, 1987, pp. 28-32.

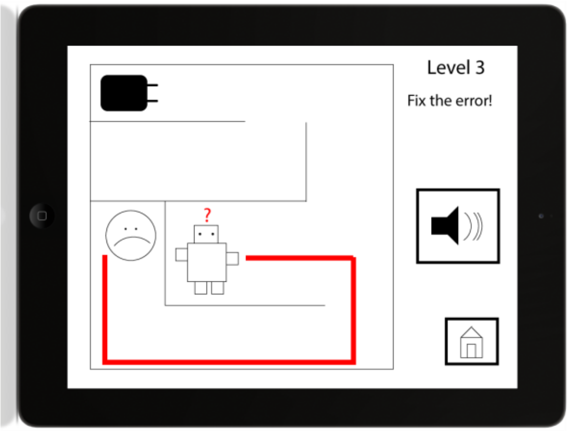
[30] J. Read, S. MacFarlane and C. Casey, "Endurability, engagement and expectations: Measuring children's fun," in *Proceedings of IDC 02*, Eindhoven, Netherlands, 2002.

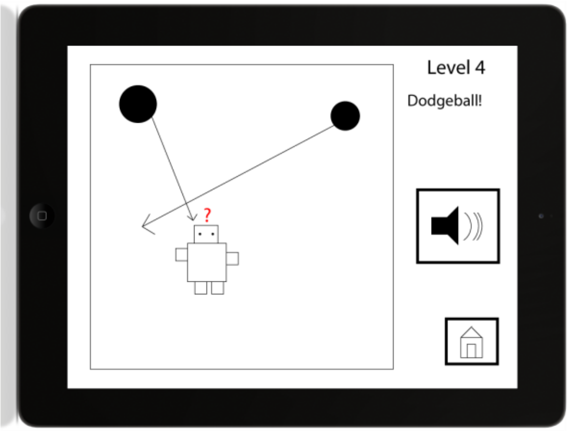
13 Appendix - Design 1

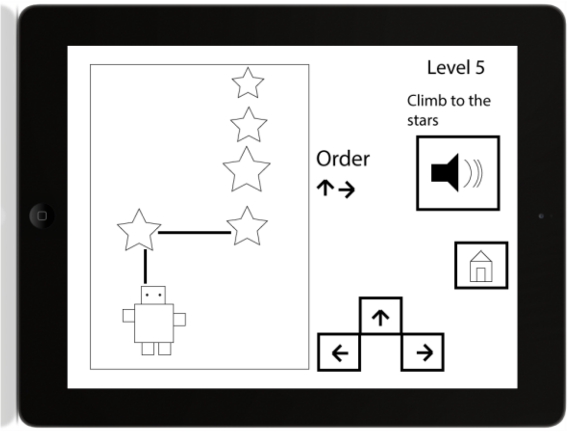


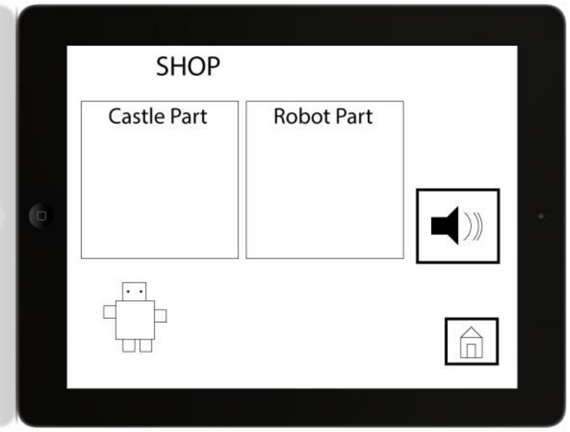




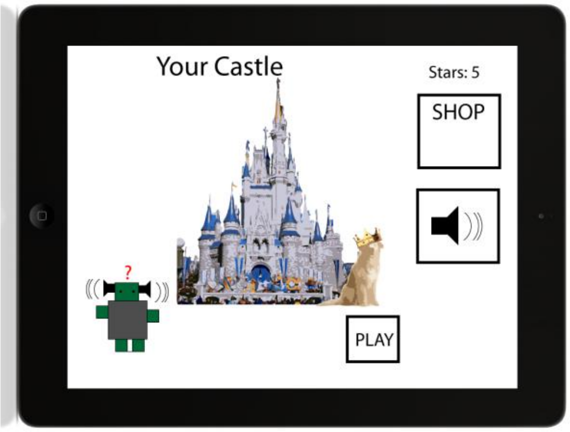


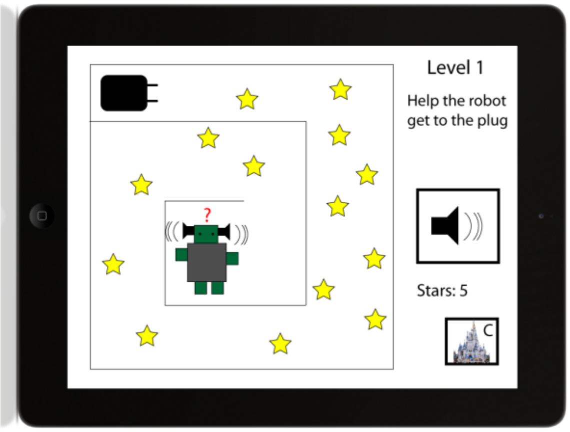


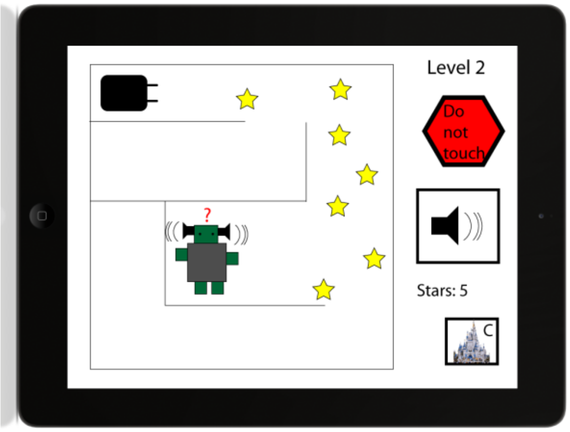


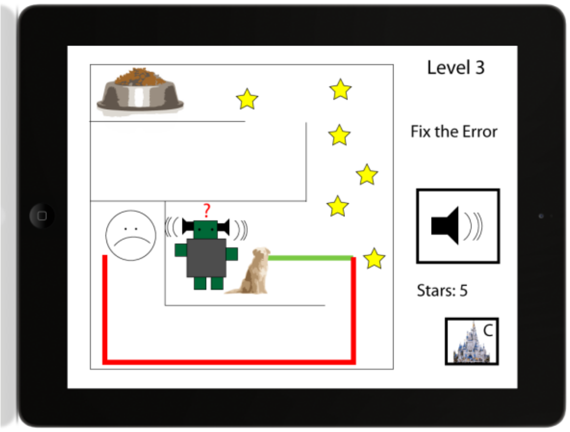


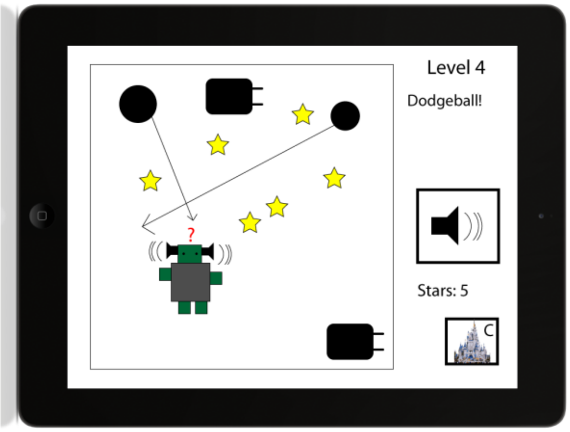
14 Appendix - Design 2

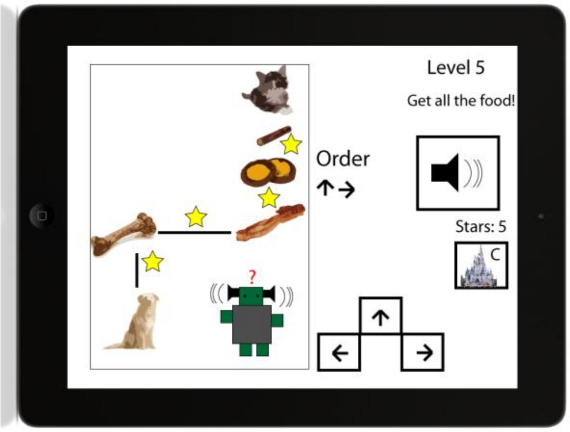

Your Castle
Stars: 5
SHOP
?
PLAY

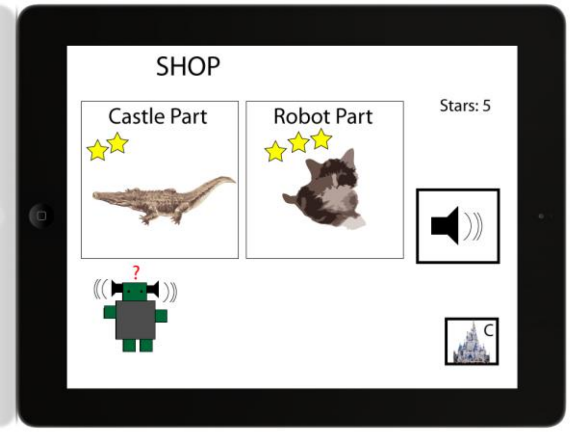

Level 1
Help the robot get to the plug
?
Stars: 5
C


Level 2
Do not touch
?
Stars: 5
C


Level 3
Fix the Error
?
Stars: 5
C


Level 4
Dodgeball!
?
Stars: 5
C


Level 5
Get all the food!
Order
↑→
Stars: 5
C
?
↑
←
→


SHOP
Castle Part
Robot Part
Stars: 5
?
C