**INST799 Capstone Project Proposal - Panagis Papadatos**

Advisors: Mona Leigh Guha, Tamara Clegg

January 20, 2013

## Introduction

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

Contents

[Introduction 1](#_Toc342212921)

[Objective 2](#_Toc342212922)

[Motivation 2](#_Toc342212923)

[Related Research 3](#_Toc342212924)

[Related Products 3](#_Toc342212925)

[Limitations of the field 3](#_Toc342212926)

[Summary of the methods used 4](#_Toc342212927)

[Kidsteam 4](#_Toc342212928)

[Completed Work 4](#_Toc342212929)

[Description of Kidsteam Session 1 4](#_Toc342212930)

[Analysis of Kidsteam Session 1 4](#_Toc342212931)

[Results of Kidsteam Session 1 4](#_Toc342212932)

[How did the literature inform the design 5](#_Toc342212933)

[How did the Results inform the prototype 5](#_Toc342212934)

[Iteration No.1 5](#_Toc342212935)

[Description of Kidsteam Session 2 5](#_Toc342212936)

[Analysis of Kidsteam Session 2 6](#_Toc342212937)

[Results of Kidsteam Session 2 6](#_Toc342212938)

[Iteration No. 2 6](#_Toc342212939)

[Details and explanation of the design 6](#_Toc342212940)

[Proposed Work 7](#_Toc342212949)

[Timeline 8](#_Toc342212950)

[References 8](#_Toc342212951)

[Appendix - Design 1 9](#_Toc342212952)

[Appendix - Design 2 10](#_Toc342212953)

## Objective

The objective of this project is to design and implement a touch screen application that will help young children learn skills that are integral to programming.

## Motivation

A plethora of factors led us to work on this problem. First and foremost, younger generations’ comfort with computers has made children more independent of their parents in their exploration processes (Lin and Liu 2012), which suggests a greater learning need. By giving children the tools to build their own interactive environments, they can begin to experience a level of creative autonomy previously limited to adults [1]. Furthermore, acquiring debugging skill sets is influential to the development of logical thinking, problem articulation, team working, persistency, problem solving, and social interaction skills. Moreover, providing reading opportunities to young children is important during a significant time period in their development of reading skill and comprehension [2].

Both the study of Computer Science (CS) and its relevant industries have recognized the importance of fostering the development of computational thinking and skills earlier in students’ education [3]. This is especially true in light of current market conditions, lacking in computer science majors and the lack of diversity amongst its practitioners. More specifically, many students never investigate computer science as an academic option, because they haven’t been sufficiently exposed to computer science or because they feel that their identities do not mesh 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]. Mobile technology will soon be prevalent in classrooms, therefore the need for children to understand and use mobile technology is rising significantly [2].Mobile devices can 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]. The research above suggests that it is beneficial to create a touch screen app that will allow young children to develop their computational thinking skills.

Research shows that children have the ability to control their interactive environments [1]. Furthermore, computer programming is a developmentally appropriate practice in preschools [9] since children are able to learn aspects of programming by performing specific tasks [10] and can reason logically as long as principles are applied to concrete examples [10]. Last, the desire to program exists, and is strengthened especially when the environment allows children the ability to create dynamic and interactive worlds or games [11].

## Related Research

The following authors provide useful knowledge on challenges, design ideas, do’s and don’ts in regards to exposing children to learning programming concepts, touch screens and other significant aspect that relate to the design of this app. Druin [12] provides an extensive list of challenges, guidelines and ideas that relate to designing mobile technology for children, especially in regards to learning. Kahn [5] demonstrates how programming by example works and suggests animating concepts. Kindborg and Sökjer [13] demonstrate behavior-based programming and suggest the use of voice and text for instructions. Lee et al. [2] stress that it is important to aid learners in thinking in an abstract and generalized way, while trying to minimize the effect of split attention. Lin and Liu [14] provide guidelines on child-parent collaboration and demonstrate the positive and negative effects of the parent’s involvement in the learning process. McKnight and Fitton [15] provide 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. Morgado et al. [16] address several of the issues that relate to ways of teaching programming to children through software. They provide a possible taxonomy of terms to be taught, but also summarize the research and crucial points to be considered. Wyeth and Purchase [17] also suggest design criteria for activities that encourage the learning of computational thinking. Wyeth provides a taxonomy of programming skills that can be taught [10]. Revelle and Reardon [18] provide a set of guidelines for designing touch screen applications for young children. Sheehan et al. [19] suggest ways in which to take advantage of physics laws which are intuitive to children. Sipitakiat and Nusen [20] provide different manners of demonstrating and learning debugging concepts.

## Related Products

In the past 10 years, there have been many attempts to create games, or otherwise educational environments that encourage children to acquire computational thinking skills. The most famous example is Logo [21], 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 technologies or research. Creator [21] uses programming by demonstration to encourage program construction and targets children around the age of 12. Toontalk [5, 13] is an animated and action-based programming language that has been tested with preschoolers. Alice [21] is a 3D programming environment that allows children to create narratives, play games and create videos. Scratch [22] 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. Magic Words [13] is a behavior-based visual programming toolkit that allows children (close to 5 years old) to create their own interactive worlds and games. Fizz [19] 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. CTArcade [2] allows learners to train their own virtual characters while playing games with it. Many of studied activities also involve physical programming. 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 Robo-Blocks system [20] allows children to connect physical command blocks (ages 8-9). Electronic blocks [9, 23, 17] is a programming environment designed specifically for children aged between 3 and 8 years. Storyroom [16] uses a set of tangible tools and metaphors in a room sized environment. Children as young as 4-6 can benefit from this system.

## Limitations of the field

Even though a wealth of research exists on teaching computational thinking to children, a large percentage of it is often about older children. Since children of this age develop very quickly, different software exist for different age groups, and knowledge about older children can’t directly be applied to younger children. As an example, Druin [12] mentions that portable device applications are categorized into 11 separate age group for the age range of 0 to 15. Furthermore, to this date, no products could be found that attempted to help young children learn computational thinking skills through the use of touch screens.

## 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. During the iterative design phase, the main method used was participatory design with slightly older children (ages 7-11) in two separate sessions. This provided a chance for different aspects of the design to be examined and included children whose age is close enough to our target group. Hence, they were able to provide us with valuable input regarding “what they would have wanted 3 years ago”. Furthermore, they were old enough to provide us with useful insights.

However, a great amount of the knowledge mentioned in the related research section of this proposal was also used, not only to improve the interface, but also to ensure proper learning methods are being applied. To further the iterative design process, we will be working with the University of Maryland Center for Young Children (CYC). We will initially show our design to the teachers and get feedback on whether they find it age appropriate, as well as whether they consider it to be an effective learning resource. Since the teachers are experts in educating children of this age, they will be able to provide us with a very valuable type of input, which would be difficult to acquire otherwise. After implementing the app, it will be evaluated with potential users, i.e. the children at the CYC, in order to assess issues that did not appear in previous stages of the iterative process that did not include real 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.

### 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 8 Children and 7 adults. During the course of this project, we performed 2 sessions with Kidsteam to inform our design using cooperative inquiry methods and techniques [6, 24].

## Completed Work

### Description of Kidsteam Session 1

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 simple robot could only take very simple commands. We asked the children to give her commands while 2 researchers wrote the different commands that they 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 a list of commands that derived from the activity, that they had to put in the right order, 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. Lastly, the children were split into 4 groups of 2 (with accompanying adults) and were asked to draw an iPad app to teach younger children how to do something similar to what they just did.

### Analysis of Kidsteam Session 1

The analysis session was done in 3 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, the researchers and the children talked about overarching themes and “tagged” the whiteboard using symbols to represent similarity. Second, shortly after the session was complete, the researchers debriefed while analyzing their observations (written or otherwise) and talked about the ideas and how they relate to each other and the process. Last, the artefacts collected (notes from the researchers, drawings and pictures) were analyzed and coded using open coding principles [25, 26].

### Results of Kidsteam Session 1

When answering the question “what does the word Programming make you think about,” it became clear that they all knew it related to technology and/or video games. Most of them did not have a good understanding of why programming exists. The robot activity indicated that the children may lack 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 very quickly acquainted) with the concepts of syntax, semantics, iteration, parameter passing and compound procedures [16] after they were shortly explained to them. Even though the children were asked to let their imaginations run rampant during the drawing phase, the prototypes that they designed revolved around robots. This could imply that they probably weren’t able to make the abstraction on what programming truly is about, but also that they associate it with technology and machines and that they liked robots.

As far as game mechanics were concerned, separate levels and modules were present in all of the 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 the teams 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). Lastly, castles, mazes, and teleporting were mentioned often. Other ideas included dodgeball and racing. An interesting observation was that all of the teams had draggable interface elements. Keeping in mind that they were designing an application for children that can’t read, most of the teams focused on large pictures on buttons and/or reading things out loud multiple times for the user.

#### How did the literature inform the design

The literature informed the first version of the design in multiple ways. In order to provide clarity and to aid children with their reading skills [12, 18], 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 [16] .The introductory activity (levels 1 and 2) 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. Large, visually distinct hotspots are used (e.g. very large buttons) and new interactions are introduced with demonstrations and animations [27, 5, 18, 19]. The terms used to describe touch screen gestures in the instructions (mostly auditory) are compliant to the research [15].

#### How did the Results inform the prototype

Robots are a major concept in the game. 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 very important in understanding the basics of programming and was mentioned frequently during the Kidsteam session. The robots are customizable and upgradable (with progress in the game) since each group’s designs included these elements.

### Iteration No.1

The design (See Appendix 1) is simplistic and not very colorful in order to allow for brainstorming during the next Kidsteam session, which will be centered around the method of layered elaboration [11], in which the initial design should not be too complex.

### Description of Kidsteam Session 2

The second Kidsteam session, conducted on November 6, 2012, aimed to bring back the initial wireframe for rapid iteration and evaluation within the adults and the children that participated in the session by using the technique of layered elaboration. The wireframe now contained 7 pages that were mainly black and white and not crowded with content, 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).

### Analysis of Kidsteam Session 2

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 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.

### Results of Kidsteam Session 2

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, 2 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 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 and a lot of 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.

### Iteration No. 2

#### Details and explanation of the design

The second iteration of the design (See Appendix 2) was heavily informed by the Kidsteam Session. More specifically, these additions were based almost exclusively on the elaboration of the ideas that the kids designed during the session.

##### All Levels

* Speakers were added to both sides of the robot because the robot is actually the entity that is helping the users out
  + The right speaker gives the users feedback on how they are doing
  + The left speaker is the thought process of the robot (e.g. “Gee, I seem to be stuck”)
* Speech bubbles come out the robot’s head when it talks to you
* If the player remains inactive for more than 7 seconds, the robot asks for help (through audio and speech bubble) and an animation would pop up showing what needs to be done for each specific level (e.g. for levels 1 and 2 a finger dragging a path along a path would show up and arrows pointing at the plug)
* The home button now looks like a castle
* Players can collect stars to use as currency; The game often informs players about how many stars they have picked up

##### Level 1

* Clicking on the robot changes it color and it talks to you
* If the robot hits the wall too many times, the robot complains and says “I’m dizzy”

##### Level 2

* Same as Level 1, but the stars only exist on the right path
* The “Do not touch button” creates a cool effect on the screen

##### Level 3

* The robot is now giving commands to its dog !
* The robot says “No, stop” if it’s going the wrong way
* The line turns green when you’re going the right way
* If you fail too many times, the robot starts making funny noises and there is a popcorn explosion

##### Level 4

* There are now 2 plugs! (To add a decision element)
* This level would repeat 3 times
  + Easy level: Players can move the balls and the plugs
  + Medium level: Guide the robot
  + Hard level: The balls go fast and bounce

##### Level 5

* This is no longer collecting stars, it’s about helping the dog get all the food

##### Castle

* The dog is now the king of the castle
* The speaker tells you how to draw on your castle and what to do with the shop
* Clicking the dog makes it bark

##### Shop

* There is a payment system now
* If you click a shop item it tells you its price and asks the player to click it again if they really want to buy it (and can) and also what it is

## Proposed Work

For the Spring 2013 Semester, I will be iterating on the design that will be created after interviewing the teachers at CYC. Further, I will implement the actual application, (iOS6 - iPad) and evaluate it with the children and the teachers at CYC. The evaluation with the children will be done using a simplified sticky notes technique [6], where the children will be asked to describe things that they like and dislike about the application. The teachers will evaluate the application from their perspective through another interview, providing us with feedback on whether it is age appropriate and how they perceive it to enhance the children's learning.

More Specifically,

**Design input from experts (Teachers)**

In order to acquire feedback from experts of a different field (that of young children education), we decided that we will be conducting 30 minute sessions with 2-4 teachers. The teachers will be recruited from the CYC. If the teachers are available to meet at the same time, only one session will take place. However, if they are not, we will conduct as many separate sessions as necessary. The reason for this decision is that group discussion supports the elaboration of ideas in a way that we consider more productive for this type of feedback. Each session will begin by showing the teacher(s) the latest version of the wireframe, while explaining the mechanics of the application and what feedback the children would get while using it (auditory or visual). After the teachers believe that they have gained a clear understanding of the application, they will be interviewed (in group if the option is viable, individually otherwise). The goal of the interview is to gauge the appropriateness and learning capabilities of the application. Sample questions that we will be asking include:

* Do you think this application is age appropriate (for every level, in detail)? (This question will also help us limit the target age range besides changing interface elements)
* Does it look like it would be fun for the children?
* Would they learn?
* What changes would you make to make it more appropriate / engaging?
* Would you use it in a classroom?

However, the interview will be open ended, therefore we will ask other questions based on the feedback that we will be getting and it will not be a formal evaluation of any kind. These sessions will not be recorded in any way, but the researchers present will take notes. After the sessions have been completed, the notes will be collected and analyzed through an informal frequency analysis amongst the researchers and the feedback will be used to inform the design of the implemented product.  
  
**Session with kids**  
After the implementation of the product has been completed, we will conduct sessions to evaluate it with actual users. The sessions will last 20 minutes (at most). Due to possible permission problems related to the CYC, we are considering a different option, that of recruiting children separately. Therefore, there are two possible options

1. CYC: One session with 18 children (class size), in groups of 2 or 3. This would be oversampling, but we would need to offer all children of a specific class the opportunity to participate in our research. Therefore, not all 18 children would be informing the design, since we assume that some of them would not wish to participate
2. Individual sessions with 6-8 children

The sessions (whether in groups or individually) will start with letting the children play with the application while observing them and helping them understand it. After the children get borsed of the application, or after 10-15 minutes, they will be asked to rate the application based on how fun it was using a 5-likert scale with smiley faces (smileyometer - <http://www.chici.org/references/endurability_engagement.pdf>) and they will be asked to suggest improvements.

A few pictures might be taken during the sessions but most of the collected data will consist of the researchers’ observations and written notes. More specifically, we are concerned with observing the following:

* What areas are the children having trouble with (and why)
* Do they understand it well (In order to assess this we would be observing if they’re doing what they’re supposed to do and if they are not, what they are doing)
* Do they like the application and its interactions (Do they use the application excitedly / smile/ laugh / are engaged – the latter would be assessed by establishing whether they are looking away often)

A possible extension of this session would be to ask the children to repeat some of the levels and see if they were succeeding more easily. However, this will be determined based on the time available at the venue(s).

### Timeline

**Dec 10.** Interview/Evaluation of design with the teachers from UMD Center for Young Children(CYC)  
**Dec 15.** Design Iteration: final design  
**Jan 10.** Implementation

**Feb 10.** Evaluation with CYC kids and teachers

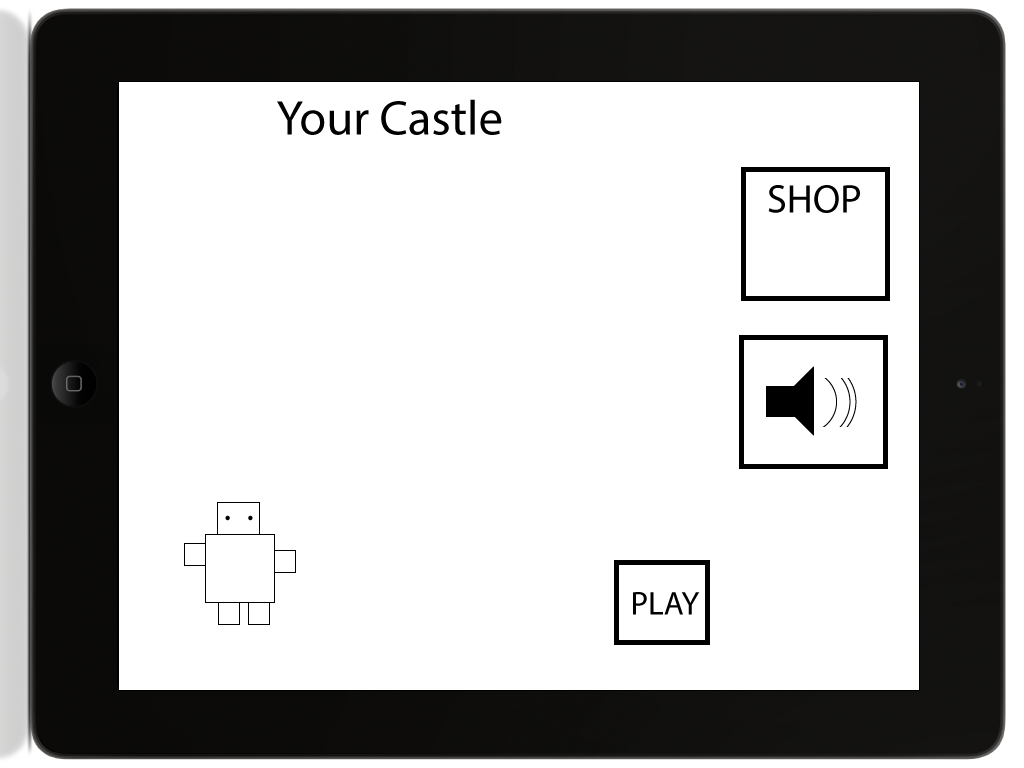
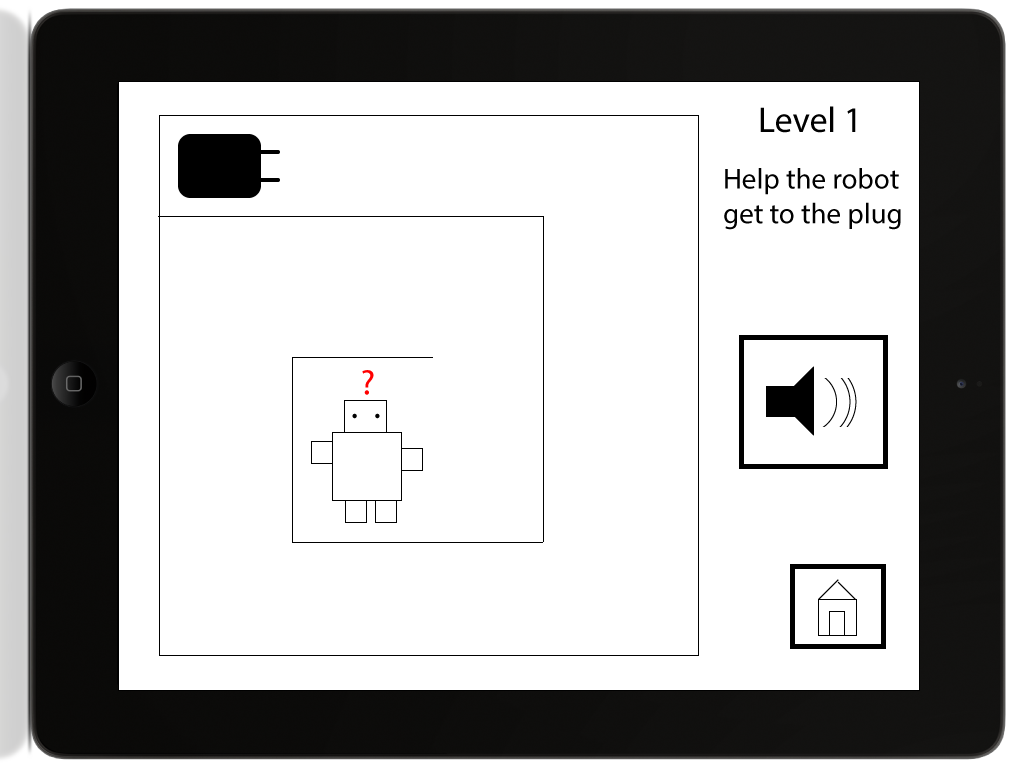
**March 10** Analysis of data, improvements and paper writing

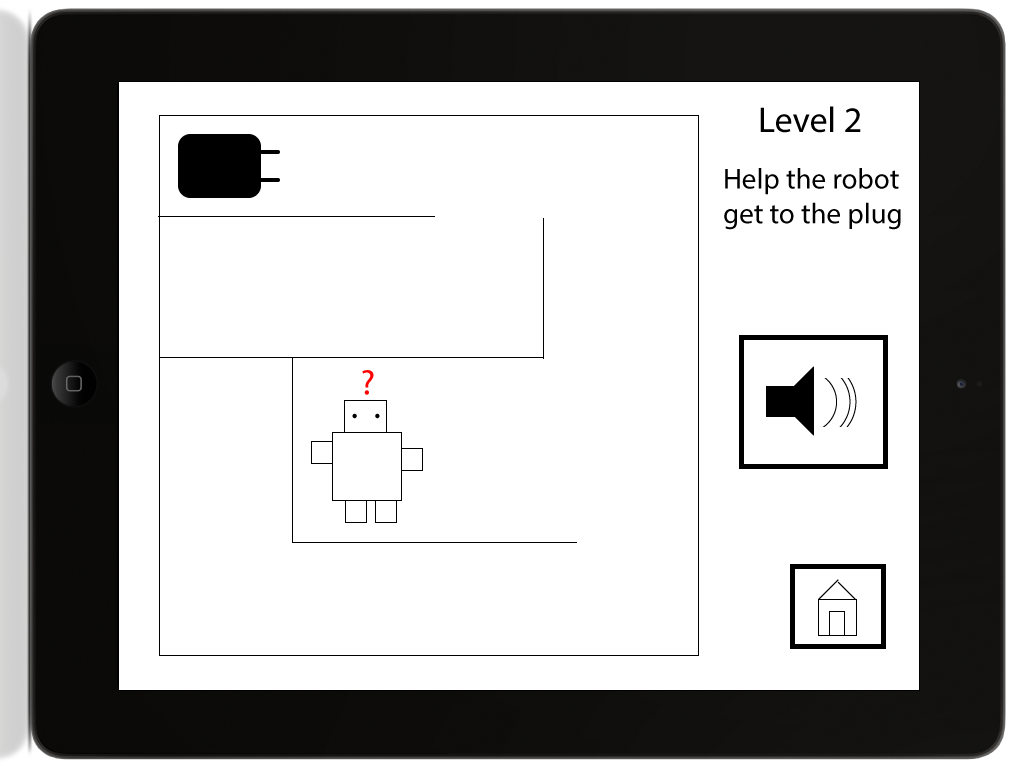
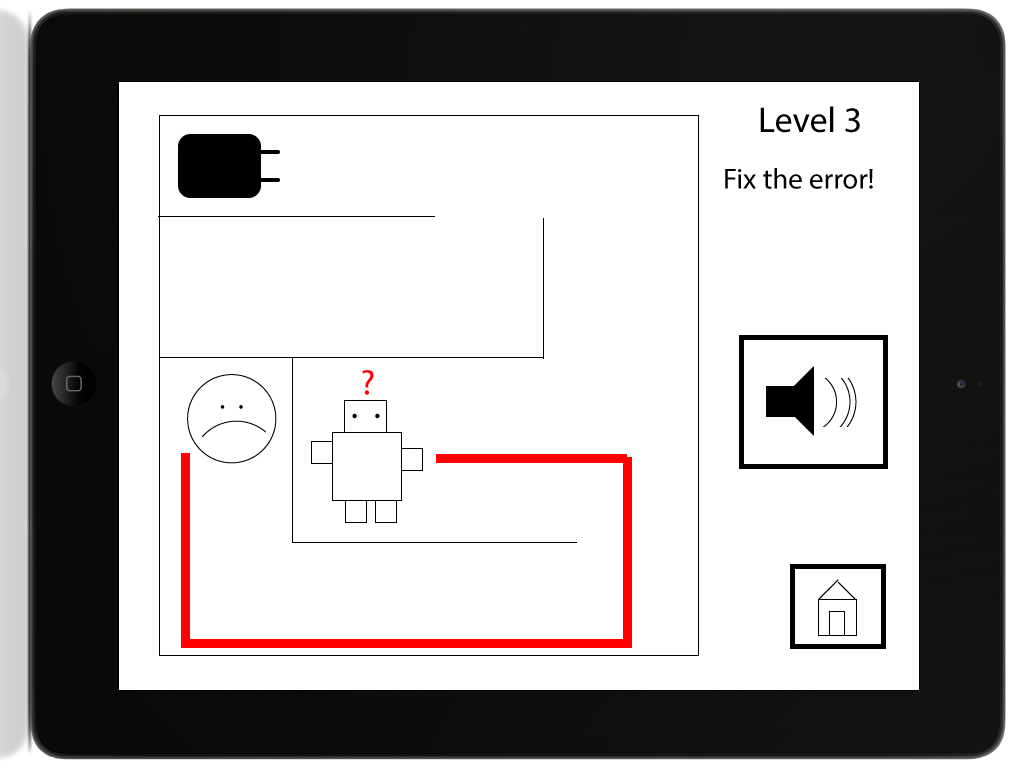
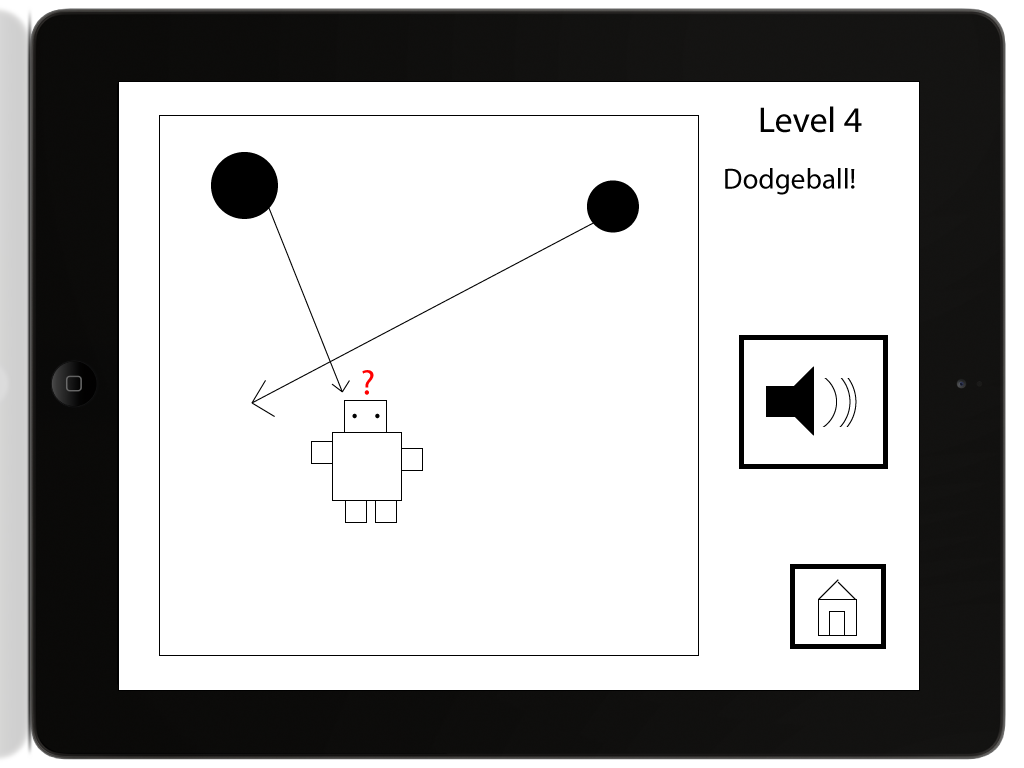
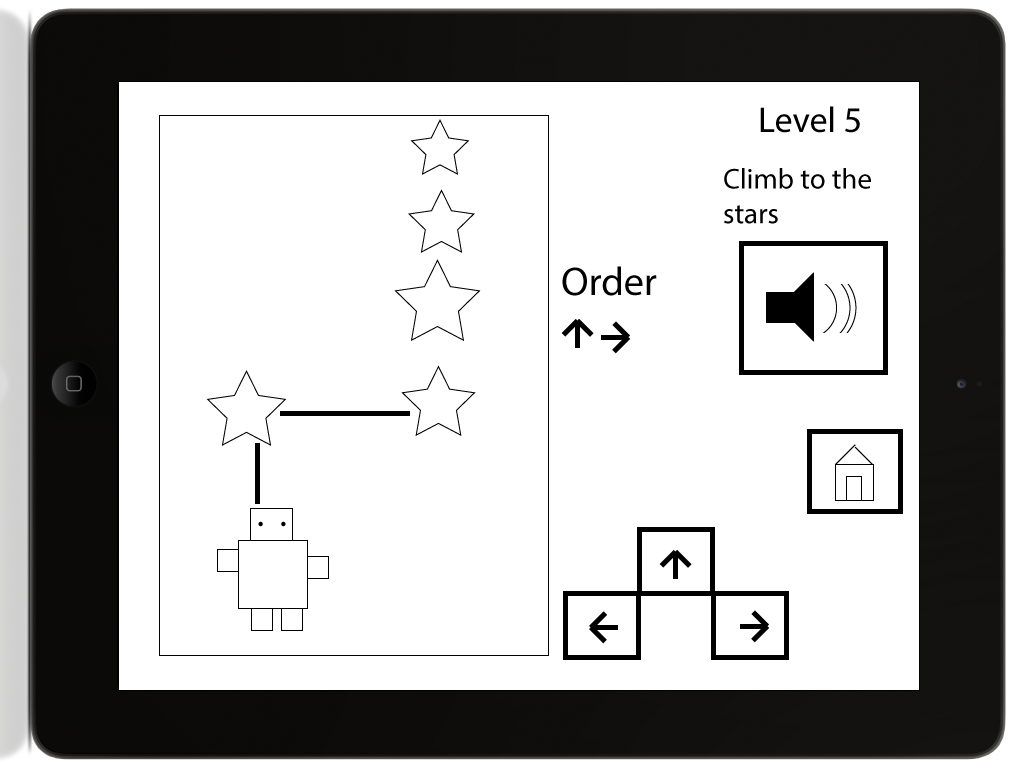
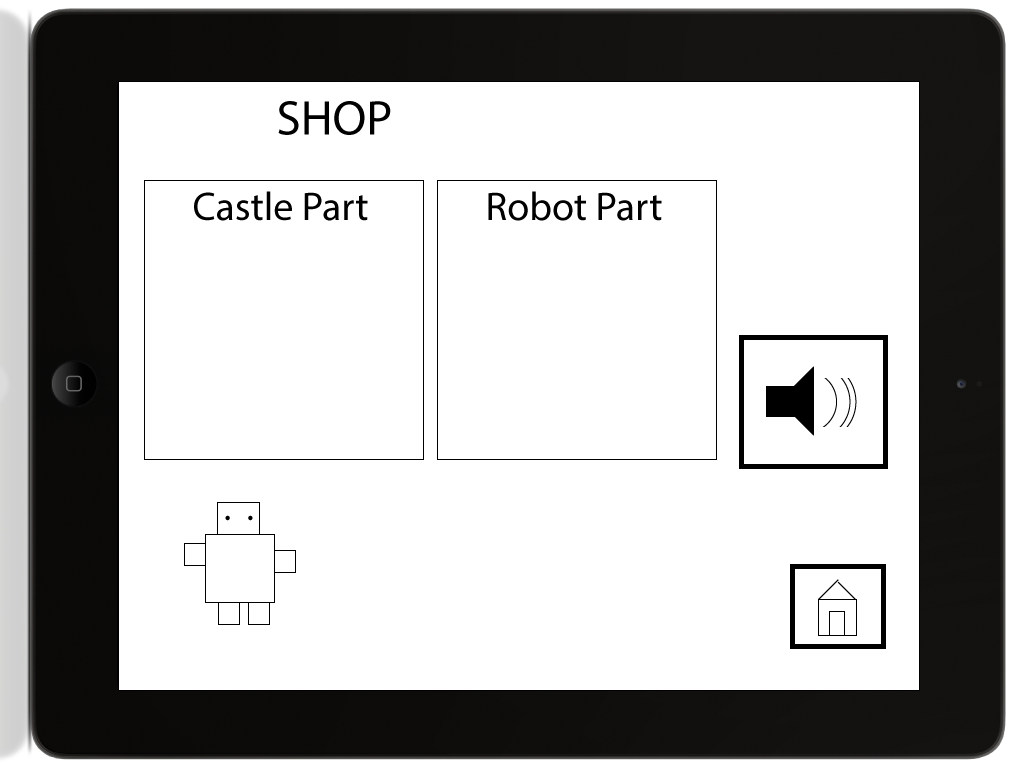
## References

# Bibliography

|  |  |
| --- | --- |
| [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] | A. Druin, Mobile Technology for Children: Designing for Interaction and Learning, San Fransisco, CA, USA: Morgan Kaufmann Publishers Inc, 2009. |
| [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. 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. |
| [15] | 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. |
| [16] | L. Morgado, M. Cruz and K. Kahn, "Preschool Cookbook of Computer Programming Topics," *Australasian Journal of Educational Technology,* vol. 26, no. 3, 2010. |
| [17] | P. Wyeth and H. C. Purchase, "Using developmental theories to inform the design of technology for children," in *Proceedings IDC 03*, 2003. |
| [18] | G. Revelle and E. Reardon, "Designing and testing mobile interfaces for children," in *Proceedings of IDC 09*, Como, Italy, 2009. |
| [19] | 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. |
| [20] | 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. |
| [21] | 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. |
| [22] | 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. |
| [23] | P. Wyeth and H. C. Purchase, "Tangible programming elements for young children," in *Proceedings of CHI EA 02*, Minneapolis, MN, USA, 2002. |
| [24] | 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. |
| [25] | J. Corbin and A. Strauss, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, Sage Publications, Inc, 2007. |
| [26] | A. Strauss, "Open Coding," in *Qualitative Analysis for Social Scientists*, Cambridge University Press, 1987, pp. 28-32. |
| [27] | 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. |

## Appendix - Design 1





## Appendix - Design 2

