

Exposing Middle School Girls to Programming via Creative Tools

Gahgene Gweon, Jane Ngai, and Jenica Rangos

Carnegie Mellon University, Pittsburgh, PA USA
{gkg, jngai, jrangos}@cmu.edu

Abstract. This paper explores design concepts and principles to engage middle school girls in learning preliminary programming concepts through different media and interaction techniques. Creating a greeting card and creating a personal avatar for an Instant Messenger (IM) were two approaches that were examined. Findings suggest that an IM avatar creation tool, with guiding principles including partial manipulation of code, immediate feedback, engaging content, reinforcement exercises, and transition from concrete to abstract examples, may interest girls to start learning programming concepts.

1 Introduction

Gender imbalance with male domination in the professional IT workforce has long been observed, as the number of female graduates in Computer Science majors or similar disciplines remains low. Efforts have been made to promote Computer Science in high schools. Yet, statistics reveal that the number of female computer science graduates has been declining each year from 32.5% in 1980[15] to 27.6% in 2002 [8]. The root of this phenomenon traces back to a low enrollment rate in High School, where programming classes have been traditionally dominated by boys [15].

There has been a lot of research towards investigating reasons for a lack of female interest in Computer Science. Currently, promotions and educational campaigns are trying to correct girls' perception of Computer Science as a "male" domain, and the stereotype that programming is a "nerdy" activity. Prior research indicates that girls prefer collaboration more than competition. Furthermore, girls view the computer as a tool used to achieve an end result, while boys see the computer as a toy for enjoyment [4, 15]. As of 2000, girls seem to be more familiar with technology. They are more "computer literate"; many understand common widgets in software applications. They use computers frequently as a social tool, as they use email and instant messengers. They also use computers to perform tasks such as doing research, writing papers, handling graphics, and drawing with visual tools [5].

So, what distinguishes girls from boys? Programming is difficult for all beginners and although girls are computer literate, they still don't go into computer professions as much as boys. Since girls envision the computer as a tool, the materials covered in traditional introductory programming classes become mundane and impractical [6]. For example, typical assignments in beginner programming classes may include an exercise to calculate the sum of numbers using loops and conditions. Summing up

numbers may seem irrelevant to an interesting end result. Even if gender differences are disregarded, programming is difficult for all novice users. This is shown by a study done at the University of Michigan that estimates the percentage of novices that will continue to program after taking a programming class is less than 1 percent [11]. One reason could be that the distance between the programming world and the problem world is too large [10]. Programming languages are not presented in a way that matches the users' mental model. Creating an interesting end result requires more advanced programming skills, and girls may decide that programming is not interesting before they can learn sufficient programming skills to apply to a problem with a practical and "interesting end-goal".

The purpose of this study is to expose middle school girls to some preliminary programming concepts through some attractive means. Although many studies have been done in exploring reasons for the low female involvement rates in technology, work in exploring ways to motivate young girls to programming is still at a relatively pioneer stage. Middle school girls from 5th to 8th grade were chosen to be the target audience for this study. At this point of their education, they are not settled on the path they will pursue in High School and later in college. These years are the critical stage when girls are introduced a variety of subject areas, and likely the period when many girls decide not to pursue computer science [6].

A tool to entice girls to learn programming concepts would require compelling, engaging content that would produce a concrete end result. Over the course of this study, two approaches were considered and carried out in this study to address the needs of teenage group – a Greeting Card tool and an avatar for Instant Messaging.

2 Phase I – Greeting Card Tool

The greeting card tool would provide means of creating a greeting card with simple animations and text while exposing girls to introductory programming concepts. Greeting cards have a large capacity in fulfilling girls' desire to construct an interesting product.

2.1 Our Hypothesis and Design Realization

In order to meet the main goal of introducing girls to programming and sparking their interest in computer science, a greeting card application seemed to hold potential as an excellent introduction. We hypothesized that greeting cards would provide social connection between people and a motivating factor in making something for specific recipient(s). Our tool consists of 3 sections: step-by-step wizard, code, greeting card stage.

Four guiding principles were used to aid in achieving the main objective. The first principle is to enable learning by direct manipulation of graphical user interface (GUI). Generating code is a difficult activity, especially for beginners who are first introduced to the subject of programming [9]. Generating code that produces an entertaining or engaging product proves more difficult. To enhance the girls' ability to learn programming, a GUI might serve as a mediating tool between the girl and the code. Participants do not need to write the code directly, but can manipulate the GUI

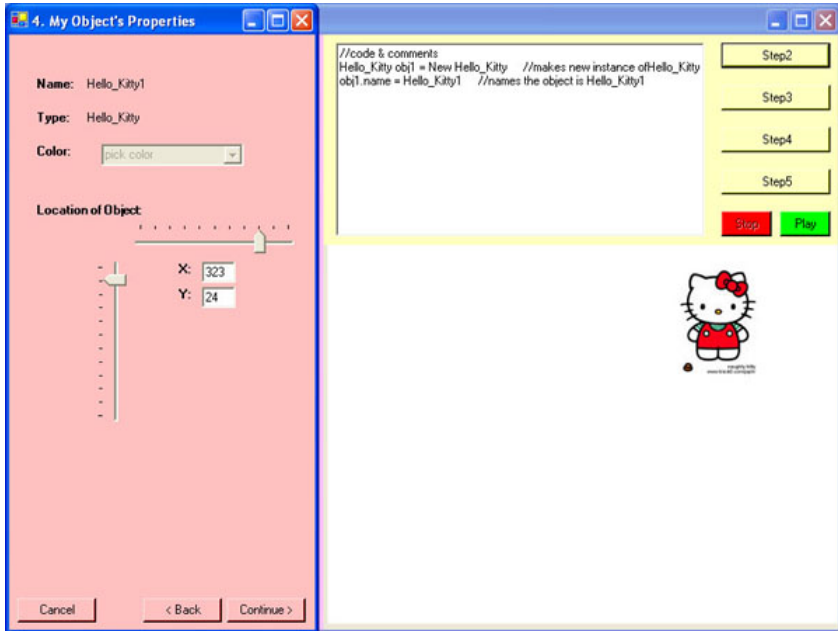


Fig. 1. Greeting Card Tool interface. The left pane is the step-by-step wizard. The right pane consists of code section on top and greeting card stage on the bottom.

in order to change the code. The aim is for users to see code, understand that it can change and produce an action. This is achieved with a step-by-step wizard in the interface. Users can select an object from a cupboard filled with popular characters, change the object's various properties, animate the object by manipulating its x and y coordinates in addition to changing the background color and type text.

The second principle is to produce practical and concrete results. Tangible results are important to girls as seen in literature [4, 5, 15]. Unlike their male counterparts, girls do not view programming or using the computer as a game; instead, they view the computer and programming as a tool used as a means to an end. Greeting cards are useful and fun end products, since girls enjoy making greeting cards for special occasions for friends and family as evidenced in our surveys.

The third principle is to provide visual feedback. Feedback provided during programming is often neither visual nor immediate. Users must first program and then compile to see the results. In most programming environments, abstract text errors provide the only source of feedback. It is difficult to understand what is going on just with the textual feedback, especially for novices. In this interface, visual feedback is provided in two ways in this interface: in both the code and greeting card stage sections when the user makes a change via the wizard section. For example, manipulating the x and y coordinates to change the character position or the background color of the card from the wizard section updates both the code and the greeting card sections instantly. Users can immediately see changes on the screen as a result of their manipulation. Traditional feedback, analogous to that of compiling is also provided. In the final wizard step, the user should press a "Play" button, placed in the code

section, to see the final animation of the card. This is an attempt to convey the correct mental model of programming - to compile programs before they are run - so that the girls will not have to unlearn concepts when they program without the aid of our tool. Moreover, comments are shown in the code section as an additional source of feedback, with the intention of encouraging good programming practice.

Our final principle is to have engaging and entertaining content. Programming greeting cards appears to be inherently more entertaining than many other first activities introduced in beginner level programming courses. Popular characters such as “Hello Kitty” and “Baby Cheese” were used in the tool. The user may choose a character of his/her choice from the repository.

2.2 Methods

Research began with a structured field interviewing method based on understanding the context in which a product is used, known as Contextual Inquiries (CI) [1].

Table 1. Methods used in Phase I

Method	Number of Users	Age	Location
Contextual Inquiry 1	12 girls	11-12	Fox Chapel Middle School
Contextual Inquiry 2	20 boys & girls	6-13	Carnegie Science Center
User Test	6 girls	6-13	Carnegie Science Center
Survey	14 boys & girls	11-12	Fox Chapel Middle School

There were 12 girls interviewed in the first CI and 20 boys and girls in second CI. During both CIs, children made greeting cards with materials such as pencil, construction paper, glitter glue, and other paper products. The goal was to observe the students’ creative process, the components children believed were essential to their greeting card, the characters they chose, the quantity of text used, and the occasion for which they made the card. Subsequently, we conducted think-aloud usability testing with 6 girls. A survey with 14 boys and girls was also conducted to understand teenagers’ card making habits, their favorite holidays, and their exposure to computers and the Internet. Participants were asked to create a greeting card using the greeting card tool. During the think aloud usability testing, participants were encouraged to think aloud while creating a greeting card by animating the character object chosen, changing the properties of their choice, and adding text to the card. Due to time constraints and the process required to recruit students to participate in user testing, we performed pilot testing with fellow master students before testing it with the target population to catch main usability problems.

2.3 Findings and Lessons Learned

Direct manipulation of GUI helped the girls engage in programming activity more easily. Yet, it had an unexpected side effect. It did not help girls understand the

relationship between the manipulated objects and the code. To complete the task, participants relied on direct manipulation provided by the wizard. No participants manipulated the code, asked what the code section of the screen did, or why it changed. However, 5 of 6 girls tried to manipulate objects on the greeting card stage in relation to the change in the wizard. Of the three main portions of the interface, the code portion was the only part that the participants did not touch, aside from when they pushed the play button to animate the card. During the debriefing section, one participant said, "I don't care about that [the code], I want to just play with this [wizard]". It was also unclear if they even noticed the code change as they manipulated the object. Placing the "play" button in the code portion of the scene did not encourage the users to look at the code. None of the users remarked on any of the dynamic changes in the code during the task. Our attempt to encourage the girls to the code by placing the "play" button in the code section failed. In sum, participants did not appear to learn any programming concepts from interaction with the interface.

Regarding the second principle of producing practical and concrete results, we observed from both contextual inquiries that greeting cards are something that users would make normally. A greeting card tool was shown to be a practical and concrete medium to use. During our first CI, all of the girls spent time before starting their card thinking of an occasion for which they could make their greeting card. Most girls made greeting cards for upcoming holidays, such as Mother's day, Father's day or St. Patrick's Day. None of the girls made cards for distant holidays such as Christmas, despite the fact that it was a popular favorite holiday among the participants according to the survey. Most girls made cards for their family members, while none of the girls made cards for their peer friends. 10 out of 12 girls took the cards with them. These observations seem to support that girls preferred to create something useful, if not immediately useful. However, in our subsequent contextual inquiries, greeting cards seemed to be not compelling enough for our target audience. The second CI that took place at the Science Center is a very different environment compared to a school environment where the first CI took place. The participants of the second CI were self-selected and the greeting card project had to compete with many other entertaining activities. Surprisingly, our greeting card table attracted both boys and girls who were approximately 6 years old who were much younger than our target audience. It was possible that the materials used afforded the interest of younger children. Nevertheless, overall evidence from our user testing also supports that the audience attracted to this activity were younger than intended. Not one middle school girl chose to make a greeting card unless solicited. The two middle school girls that participated in the user test indicated that they do not create greeting cards on a regular basis. It seemed that they were not interested in making greeting cards with the provided material, and even when asked to use the prototype, the majority preferred to engage in other activities at the science center such as viewing the exhibits or going to the Omnimax theatre.

The third principle of providing visual feedback worked well. Users were amused when they saw an object appearing or moving when they clicked the "Next" button in the wizard. Providing visual feedback at each step gave the users a sense of progress and kept them in context.

For the fourth principle to provide entertaining content, the use of characters such as "Hello Kitty" was successful. All girls chose well-known characters over alternative pictures. The younger users enjoyed the experience. From the girls' smiles, it

seemed that they were reasonably impressed with the application's functionality, and were excited. Younger girls in particular had all expressed content and excitement. Some of them mentioned that the animation is "cool", "nice", or "fun". A noteworthy observation was that younger girls (age 9 or under) enjoyed the experience more. The middle school girls that participated in the study appeared to be less amused.

The greeting card interface had 2 general usability problems. First, none of the participants could add text to the card easily. In order to add text, the participant had to select the text object from the cupboard (the repository with character and text objects). The text object was not easily identified or distinguished from the other objects, because none of the participants saw this object immediately. Second, some of the objects in the cupboard seemed to afford dragging and dropping onto the stage; however, this functionality was not available. The participant could only manipulate the object by using the cupboard. Younger children accepted this functionality, but older girls who had more experience using the computer, had difficulty not trying to directly manipulate the object of the stage. These problems were previously revealed in the pilot test. The nature of these problems was thought to be mainly limitations of the prototype and did not affect the presentation of the main ideas of the interface. When participants reached the glitches and clearly demonstrated that they experienced the difficulties, the facilitators guided them through this step. These problems did not yield an unusable application; the interface also had some aspects that worked well. For example, the x and y axis scroll bars used to select the beginning and of an objects path, used in animation, seemed to be easy to use. Due to the visual feedback, most girls understood that manipulating the x or y scroll bar would move the object. Also, none of the girls had difficulty selecting a color for their card.

3 Phase II – Avatar for Instant Messaging

To address the concerns of motivating girls to programming through greeting cards, another medium was introduced in the second phase – a tool to create an avatar for Instant Messaging.

A study on what appeals to teenagers revealed that girls often identify themselves with characters; they like to be represented by characters, as if they were in a hypothetical story [4]. In addition, girls enjoy social interaction. 43% of surveyed 7th graders reported that they use instant messenger (IM) "all the time". This medium allows girls to socialize in cyberspace, while helping girls overcome shyness in talking with boys or about difficult topics [12]. Teenage girls often search for characteristics that identify themselves as a unique person, while conforming to trends and group standards [13].

An avatar for IM tool allows girls to design and customize a picture displayed on a chat window. This tool, when fully developed, could be used for various tasks in the context of an IM. Programming concepts are introduced through simple statements in changing properties and conditions. For example, using this tool, girls can create and customize a character that would represent themselves. They can use this tool to customize messages and pictures according to self-defined conditions by executing programming statements. This tool is predicted to satisfy girls' desire to identify them-

selves with characters, to see visual representations of themselves, while fitting into their social lives.

3.1 Our Hypothesis and Design Realization

For phase II, six principles are selected in total. Most of the hypotheses from phase I persist with slight modifications.

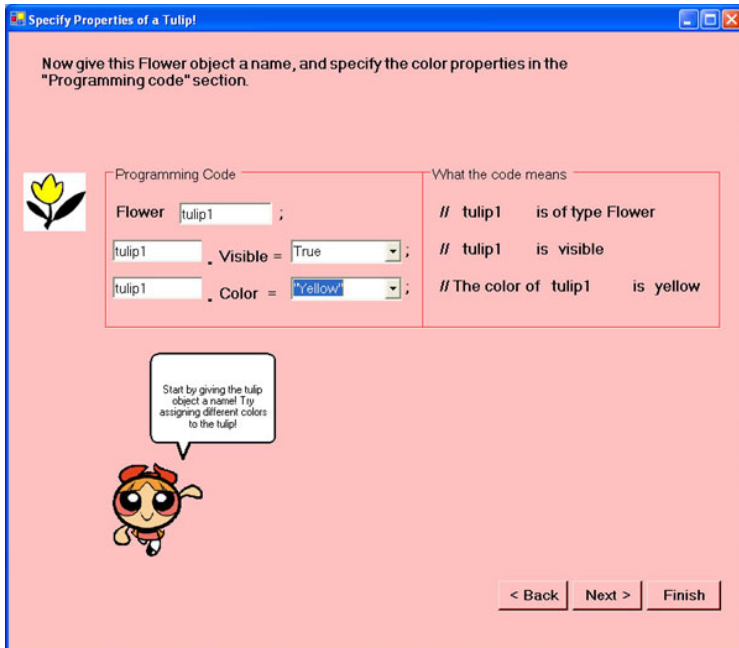


Fig. 2. IM avatar Tool Interface. The color and visibility of the tulip change as the user manipulates the dropdown menus. The line-by-line comments are changed to reflect the partial code.

The first principle is to use manipulation of partial code. As discussed in phase I, direct manipulation did not encourage girls to look at actual code with programming syntax, for it allowed girls to complete the activity without looking at the code. Therefore, in phase II, the girls must complete code that is partially filled out. Manipulation directs girls' attention to code-like syntax, and also conveys the message that they are not using a tool with a GUI, but dealing with programming. Furthermore, "Partial" manipulation provides guidelines for beginners, so they only work within a valid set of possibilities. This introduces simple syntax and semantics, and prevents errors that discourage them.

The second principle is to produce practical and concrete results, which persists from the first phase. An IM avatar tool may be more practical to girls than a greeting card tool. The survey indicates that many girls use IM frequently, whereas they would only make greeting cards on special occasions. According to the survey, most girls make a greeting card approximately once a month and use IM daily or at least weekly.

Furthermore, the avatar for IM is predicted to be more socially compelling as it supports a two-way communication, while greeting cards only support one-way interactions.

The third principle is visual feedback. As mentioned in Phase I, visual feedback is necessary for better understanding of materials [3, 14]. In the previous interface, the code was placed above the result, which could not be seen until the last wizard step. In phase II, the code and its results are placed adjacently to establish the connection. Rollovers were used to provide immediate feedback. Users could click on the button “Check my answers” to receive visual feedback on the exercises. When the user presses a button “Refresh”, the altered code will update, showing visual feedback.

The fourth principle of having engaging and entertaining content is also carried from phase I. The team tried to keep the project less like a dry textbook. Interactive lessons were used to keep girls engaged. Cute and funny graphics were also used to entertain the girls. For example, popular characters such as “power puff girls” were used in the lessons.

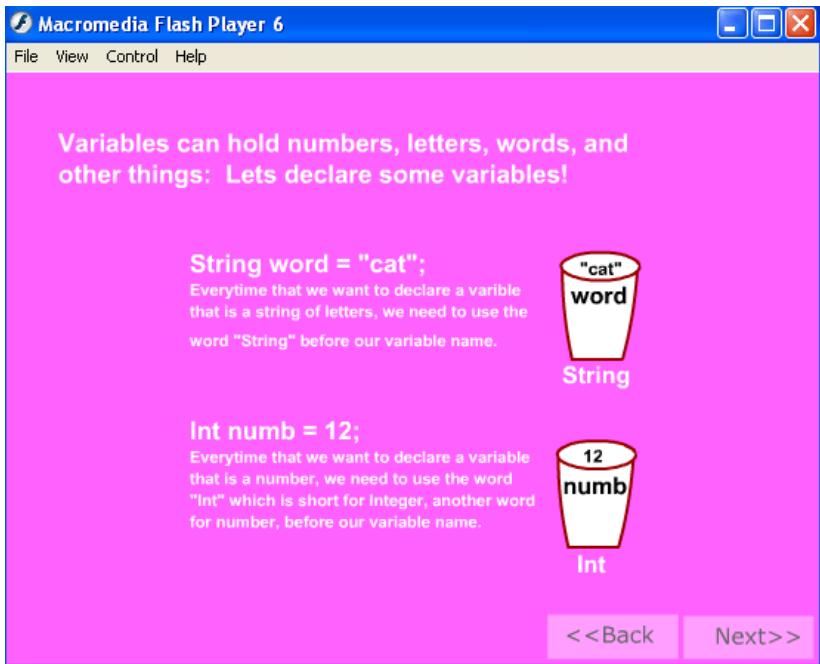


Fig. 3. Programming Concepts Lesson. In an introductory lesson, concrete examples in everyday language are used to enhance understanding of abstract concepts.

The next principle is to reinforce concepts through examples and applications. Students learn better by working through examples rather than strictly reading text [2]. The interface consisted of lessons that teach concepts in programming, and “projects” with exercises to reinforce concepts previously taught. The IM tool consists of 4 lessons and 3 projects. Lessons are used to teach the concepts in programming. Projects

that follow contain exercises that help the user to try out the lesson just learned. Since alternating lessons and projects are more efficient than having lessons and examples grouped together [3], we placed each project immediately after each lesson.

The final principle is the transition from concrete to abstract. To lessen the intimidation to learn programming, efforts have been made “to close the gap” between programming languages and natural languages for beginners [7]. Most lessons have concrete examples, often making analogies between code and everyday items. For example, when introducing the concept of a variable, metaphors of cups holding different items were used. To introduce the concept of objects and property, an example of a girl and her dress color was used. Eventually, the tool advanced into the concepts that are more abstract and mathematical in nature, such as “for” loops are taught with Java-like syntax.

3.2 Methods

Questions about Instant Messenger were added to the survey. Phase II included two iterations with 3 sets of user tests using the most recent version of interface incorporating the concept of IM system. The number of students in each user testing was 3, 2, and 3 respectively. After hands-on experience using the prototype, the girls were also asked 4 questions to test their understanding of the concepts learned.

Table 2. Methods used in Phase II

Method	Number of Users	Age	Location
User Test	3girls	11-12	Fox Chapel Middle School
User Test	2 girls	11, 14	Fox Chapel Middle School
User Test	3 girls	11-12	Fox Chapel Middle School

3.3 Results and Discussions

Testing with the updated prototype that used manipulation of partial code proved to be more successful than the version made in phase I in introducing programming concepts. Users had to look at the code with the IM tool in order to complete the task. In the quiz that followed the user test, most questions testing the programming concept were answered correctly. In addition, 75% (6/8) of phase II users completed the tasks while only 50% (3/6) of the Phase I users were successful. Some users commented “This is easier than I thought”, indicating that we were approaching a better way to communicate the programming concepts.

The second principle of producing practical and concrete results worked better as well. Girls were positive about the concept of making an avatar for IM. In the post interview, three girls commented that “the idea is cool” and they would use it if such a tool was implemented. Based on such observations and the survey revealing the girls’ frequent use of IM, it seems that the tool used in phase II is more practical than the greeting card tool used in phase I.

Better visual feedback also improved the prototype. The wizard metaphor provided a stronger path cue that helped the participants stay on the correct path to complete the task in phase II. Additionally, various forms of visual feedback informed participants when they had completed the task. For example, the participants were given the task of changing the type and property of a flower. Real-time feedback showing the effects of their code manipulations help the participants determine when they had selected the proper type of flower and its color.

Users were also engaged when they saw the results changing when they changed colors and properties of an object. 6/8 participants said “this is neat”, when they saw the color of the tulip or the look of the Power Puff Girl changing. The if/else lesson seemed particularly interesting, 3/8 girls went back to try different mood combinations, to see how the power puff girl would dress depending on her mood.

The principle of reinforcing concepts with examples and applications seemed to improve understanding. Users indicated they understood concepts better. In the quiz, 71% (5/7) of users could verbally explain what the code means, and how they would change a “For” loop. In addition, 2 girls wrote down “comments” with “//” on the quiz. This showed attempts to apply knowledge out of initiative. However, it appeared that some girls understood what simple code would do and how to do tasks according to examples, yet they did not understand the concept of objects clearly. Furthermore, it seemed that some girls could still finish lessons without learning the concepts completely. This is a design flaw. When creating a GUI, participants should not be able to easily bypass the lesson without understanding the concepts. “Step-by-step” error messages can be used when participants do not complete a field, to help them make better decisions.

Table 3. To assess how much concepts were retained by the end of all lessons, a post user test quiz was given to the girls during the debriefing session. 7 of 8 participants performed the quiz, while one participant did not have time to do it.

Question Description	# Correct Responses	% Correct
1a. Explain verbally what a small section of code means. (Naming an object and assigning a property)	5/7	71%
1b. A small section of code was given, and the girl is asked to modify a For loop to change the number of executions.	5/7	71%
2. The girl is asked to distinguish statement(s) that are not “If” Statements.	6/7	86%
3. True or False question about whether an object has to have a name.	2/7	29%
4. True or False question about identifying different kinds of loops.	5/7	71%

Lastly, in applying the principle of transition from concrete to abstract, some of the ideas worked well, such as introducing familiar objects before mathematical concepts. Only 1/8 girls were hesitant about starting the activity. They might be less intimidated by the code, for they were first introduced with familiar and concrete concepts. Girls were quicker in reading the beginning lessons compared to the later, more abstract lessons. The learning curve from the everyday pseudo-code to “Java-like” code was steeper than expected, as reflected in the quiz results. Three girls expressed that the curly brackets were “confusing” during debriefing. They said that they had to think twice to understand when a curly bracket appeared. Some girls did not want to read the screen, which contained such programming symbols.

Some user interface problems were revealed in the user testing, such as placement and color of buttons that were not prominent enough. Some other UI issues were cluttered screen in which the user could not find an obvious place to start. Another problem was use of difficult vocabulary. Use of words such as “initiate” and “emotion” confused the users. These words changed to “set” and “feelings” respectively. After this initial phase of testing, easier ways were discovered to explain some concepts. For example, our last project has 2 types of loops combined into one exercise. This turned out to be too much information. Therefore, users suggested separating them. Visual cue of where the users can start reading was also provided to help user parse the complex text after first iteration.

4 Conclusions and Future Work

The goal of this study is to kindle girls’ interest in programming via lessons and projects that appeal to both girls’ entertainment and social needs. Overall, we have achieved the goals as seen in the girls’ expression of interest and comprehension of the concepts introduced in the lessons. Yet some girls still seemed slightly intimidated in touching the projects. They asked the experimenters questions to confirm their answers before completing a blank field although they were correct. They were reluctant in exploring different types of combinations beyond instructions. A friendlier interface, a more robust set of lessons, and a more comfortable environment may reduce this sensitivity.

One way to bring this concept of enticing girls to program into realization of concrete application is to introduce a finished product into a classroom setting, where the users would learn introductory programming skills and interaction methods. After acquiring some level of proficiency acquired in classrooms, users might be motivated to make characters in their own time outside the required school work.

Additionally, there is a great potential of this project on a broader scale. Currently, some online communities allow each member to create and personalize a character that represents them (avatar). An existing example is cyworld, which is a very popular online community in Korea. Future work can be done to develop similar communities in which the user can program to change their avatars.

Acknowledgement

We thank Ken Koedinger for his support and guidance.

References

1. Beyer H., & Holtzblatt K. *Contextual Design*. San Francisco: Morgan Kaufmann, 1998.
2. Chi, Michelene T. H., Bassok, Miriam. *Knowing, Learning & Institution; Essays in Honor of Robert Glaser*. Ch.8. Lawrence Erlbaum Assoc, 1989.
3. Clark, R. C, Mayer, R. E. *E-Learning and the Science of Institution*. Pfeiffer. San Francisco, CA, 2003.
4. Gorriz, C., Medina, C. *Engaging Girls with Computers Through Software Games*. *Communication of the ACM* (Jan 2000), 43, 1 42-49.
5. Kelleher, C., Pausch, R. *Lowering the Barriers to Programming: a survey of programming environments and languages for novice programmers*. *ACM surveys* (Oct 2004).
6. Kelleher, C. *Motivating Programming: using storytelling to make computer programming attractive to more middle school girls*. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA 2004
7. Lane, H.C, VanLehn, K. *Coached Program Planning: Dialogue-Based Support for Novice Program Design*. SIGCSE '03.
8. National Center for Education Statistics. *Digest of Education Statistics*. U.S. Department of Education, Washington, D.C.,(2003) Available at <http://nces.ed.gov//programs/digest/d03/tables/dt255.asp>
9. Norman, D, Draper, E. *Cognitive Engineering. User Centered System Design, New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates Publishers, 1986. P.31-61
10. Pane, J. F., Ratanamahatana, C. A., and Myers, B. A. *Studying the Language and Structure in Non-Programmer's Solutions to Programming Problems*. *Human-Computer Studies*, 2001, 54, 237-264.
11. Smith, D., Cypher, A., Tesler, L. *Novice Programming Comes of Age*. *Communications of the ACM*, (March 200), 43, 3, p.75-81.
12. Schiano, D., Chen, C., Ginsbery, J., Gretarsdottir, U., Huddleston, M., and Isaacs, E. *Teen use of messaging media*. In *Proc. CHI 2002*, ACM Press (2002), 594-595.
13. Talbot, M. *Girls just want to be mean*. *New York Times Magazine*, 2002, 27, 24-65.
14. Taylor, R. Cunniff, N. Uchiyama, M. *Learning, research, and the graphical representation of programming*. *IEEE* 86'.
15. Verbick, T. *Women, Technology, and Gender Bias*. *Journal of Computer Sciences in Colleges*, (Feb 2000), 17, 3, 240-250