Advances in Gender and Education, **1** (2009), 5-10. Printed in the USA.

Barriers to Programming Engagement

Caitlin Kelleher
Washington University in St. Louis

In this article, we describe the results of a study comparing middle school aged girls' programming behavior in two programming environments: Storytelling Alice and Generic Alice, designed to enable the creation of animated stores and 3D graphics programs, respectively. The study found that girls who used Storytelling Alice spend 42% more time programming than girls who used Generic Alice and were more than 3 times as likely to sneak extra time to keep programming. We analyze the kinds of programs that girls produce in both systems and identify barriers that keep girls from fully engaging with computer programming.

Keywords: computer science education, programming, middle school, girls, engagement

Computer programming skills are becoming a valuable tool in nearly every career field ranging from medicine and basic science to business and finance. Despite the growing importance of computing, fewer than 20% of computer science students are female (College Board, 2008; Vegso, 2006). Some research indicates that when girls and boys have similar comparable programming experience, they show similar interest in and success at learning basic programming concepts (Harel, 1991; Kafai, 1995; Linn, 1985). In one study of gender and programming achievement within MOOSE Crossing (Bruckman, 1997), a programming environment designed for young students, Bruckman, Jenson, and DeBonte (2002) found that programming performance correlated with users' prior programming experience and the amount of time users spent programming. The study found no significant correlation between gender and programming performance. However, boys who used MOOSE Crossing spent significantly more time programming than girls (Bruckman et al., 2002). One of the keys to increasing the participation of girls in computer science may lie in motivating girls to program.

The middle school years are a critical time during which many girls decide whether or not to seriously pursue the study of math and science based disciplines (Gill, 1994; Zimmer, 1987). While many girls have a strong interest in science during elementary school, their opinions of math and science courses and careers become increasingly negative during middle and high school (AAUW, 1990; Zimmer, 1987). During middle school, girls decreasing interest in math and science is coupled with decreasing confidence (AAUW, 1990; Dossey, Mullis, et al., 2000) and achievement (Fennema & Sherman, 1977) in these subjects. In this article, we describe the results of a study comparing middle school aged girls' programming behavior in two programming environments. In Storytelling Alice, programming is presented as a means to the end of storytelling. The environment was designed to make the kinds of stories that middle school girls envision telling approachable (Kelleher & Pausch, 2006). Generic Alice presents programming as the end goal and enables users to write programs that control the motions of objects in a 3D virtual environment. The study found that girls who used Storytelling Alice spend 42% more time programming than girls who used Generic Alice and are more than 3 times as likely to sneak extra time to keep programming. To provide additional insight into girls' experiences with both systems, we describe the kinds of programs that girls produce in both systems and identify barriers that keep girls from fully engaging.

Address correspondence to: Caitlin Kelleher, Department of Computer Science and Engineering, 1 Brookings Dr., St Louis, MO 63130.
E-mail: ckelleher@cse.wustl.edu.

Two other programming environments for girls are designed to provide a motivating context for learning computer programming: In Virtual Family (Duplantis, MacGregor, Klawe, & Ng, 2002) programming enables users to build comic strips about a family. In RAPUNSEL (Flanagan, Howe, & Nissenbaum, 2005) programming enables girls to create dance animations. To the best of our knowledge, no formal studies demonstrate that either system motivates girls to program.

Methods

The goal of this study was to understand the potential impact of a storytelling focus on middle school girls' interest in and success at learning to program. To investigate the impact of storytelling, we conducted a between-subjects study comparing middle school girls' behavior in two novice programming environments: Storytelling Alice and Generic Alice. Both Storytelling Alice and Generic Alice enable users to construct programs using drag and drop (Kelleher et al., 2002). The drag and drop method of program construction prevents users from making syntax errors, a common source of frustration for beginning programmers (Kelleher et al., 2002).

Storytelling Alice and Generic Alice differ in three ways:

1. Programming Actions: Storytelling Alice provides high-level animations inspired by girls' storytelling goals. In Storytelling Alice, human characters walk, speak, and interact with objects in their environment. Generic Alice provides animations inspired by 3D graphics transformations (Conway et al., 2000). Users can combine basic actions like changing position, rotating or resizing to create complex actions like walking, but this can be time consuming.
2. Tutorial: The tutorials in both Storytelling Alice and Generic Alice introduce the same programming concepts in the same order. In Storytelling Alice, the tutorial users build simple stories. In Generic Alice, tutorial users build programs that move, turn, and resize 3D objects.
3. Gallery of 3D Objects: The characters included with the Storytelling Alice gallery include character-specific animations designed to help users generate story ideas (Kelleher and Pausch, 2006). 3D objects in the Generic Alice gallery do not include custom animations.

Participants

Eighty-eight girls from local Girl Scout troops participated in the study. The participants were randomly assigned to use either Storytelling Alice (43 participants) or Generic Alice (45 participants). The average age for participants was 12.6 years. To encourage the participation of students not drawn to computers, we donated $10 to the Girl Scout troop for each participant.

Workshop Details

During the study, participants were given two hours and fifteen minutes to complete the tutorial and create a program "to show everyone" using the version of Alice to which they were assigned. Next, users took a programming quiz and completed an attitude survey. Then, participants had thirty minutes to try the other version of Alice (Generic Alice participants tried Storytelling Alice and vice versa). At the end of the workshop, participants selected either Storytelling or Generic Alice to take home and chose a program they created to share with other workshop participants.

To avoid bias, we gave the same instructions to the control and experimental groups. We referred to Generic Alice and Storytelling Alice as Alice Green and Alice Blue, respectively.

Quantitative Results

We will consider three types of quantitative data: programming behavior, motivation indicators, and learning outcomes. We provide an overview of results in this section. Additional details can be found in Kelleher, Pausch & Kiesler (2007).

Programming Behavior

There are three high-level activities available within both Generic Alice and Storytelling Alice: scene layout (e.g. adding and arranging 3D objects in the 3D scene), editing programs (e.g. adding, deleting, or modifying lines of code that control the actions of characters in the 3D scene, and running programs (e.g. viewing the animation output of the current program). Based on log data, we found that participants who used Storytelling Alice spent 42% ($p < .001$) more time editing their programs and 54% ($p > .001$) less time on scene layout than users of Generic Alice.

Motivation Indicators

At the conclusion of the workshop, we left a 5-10 minute break. The break was designed to enable us to determine how many participants would keep programming by choice. During this break time, 16% of Generic Alice users and 51% of Storytelling Alice users snuck extra time to continue working on their programs ($\chi^2 = 20.18$, d.o.f. = 2, $p < .001$). The increased tendency among Storytelling Alice users to sneak extra time suggests that the storytelling focus helped to make programming a compelling activity for middle school aged girls. This behavioral evidence is reinforced by the attitude survey: participants who used Storytelling Alice had a stronger interest in using Alice in the future than participants who used Generic Alice ($F[1,86]=3.9$, $p=.05$). Additionally, there was a strong correlation between participants' interest in future Alice use and their interest in pursuing Computer Science ($r = .54$, $p < .0001$).

Learning Outcomes

The focus on creating a more motivating programming environment creates the potential that increased motivation can come at the expense of educational value. We found no significant differences in programming quiz performance between participants who used Storytelling Alice and Generic Alice. Given the evidence that participants using Storytelling Alice spent more time actually programming, the lack of a measurable learning difference may seem initially surprising. The Storytelling Alice participants spent a larger portion of their time programming. However, because of the short duration of the programming session, the percentage difference translates to an average time difference of 12 minutes. We expect that with extended use, we would see the Storytelling Alice users show learning gains commensurate with their additional time on task.

Qualitative Results

Participants created different types of programs in Generic Alice and Storytelling Alice. The kinds of programs they created highlight four barriers to full engagement with programming.

Generic Alice Programs

One of the striking patterns within the programs created with Generic Alice was the lack of apparent intentionality. Only 38% of the Generic Alice participants produced programs which show evidence of intentional animation. We observed four general types: arbitrary motion, character motion, story-like sequences, and choreographed dance routines.

Arbitrary Motion (62%): 28 of the 45 programs participants created using Generic Alice appear to be arbitrary animation: characters and/or their body parts move around the screen without any apparent intentionality. These programs show no evidence that participants had goals they were working towards. Figure 1-1 shows a typical arbitrary motion program in which characters and their body parts rotate around different axes and fly to different positions in space.

Character Motions (16%): After some initial experimentation with Generic Alice animations and constructs, some users began to develop a mental model that may have helped them to create intentional animations. 7 of the 45 Generic Alice programs contained one or two simple character motions (e.g. a cow moving its tail or a bunny jumping up and down) but were otherwise arbitrary motion. These worlds show users beginning to transition from experimentation to building specific animations for their 3D characters. See Figures 1 and 2.

Figure 1 (left): Examples of programs participants created in Generic Alice: 1) an arbitrary motion program; objects and their parts move around in space, 2) a program containing a character motion; a girl waves hello, 3) a choreographed dance routine for penguins, and 4) a story-like program in which a knight kills a dragon.
Figure 2 (right): Examples of programs users created in Storytelling Alice: 1) a romantic relationship story about a boy who is involved with three girls and gets caught, 2) a familial relationship story about a father taking his children on vacation and getting lost, 3) a good vs. evil story about the big bad wolf trying to befriend the three pigs so he can eat them later, and 4) a choreographed cheerleading routine.

The final two groups show evidence that users progressed from experimentation to developed enough Generic Alice animation skill to create fully or nearly fully intentional programs.

Choreographed Dance Routines (7%): 3 of the 45 users created choreographed dance routines for a group of characters. These dance routines consisted primarily of characters performing move and turn animations either in sequence or simultaneously. See Figure 1-3.

Story-Like Sequences (16%): 7 of the 45 users created short story-like sequences. These stories often incorporated simple motions designed to help communicate the action of the story. For example, a character might raise his arms in fear before sliding off the screen or an injured dragon might turn onto its side to suggest a fall. See Figure 1-4.

It is notable that less than a quarter of the Generic Alice users wrote fully intentional programs and more than a half of them wrote programs that demonstrate no intentionality. Users who successfully moved from exploration to intentional control performed better on the programming quiz: the average quiz score for users who built unintentional programs was 3.53 as compared to 4.7 for users who created story-like sequences or choreographed dance routines. There is a positive correlation ($r=0.270$, $p<.1$) between intentionality and quiz performance.

Two barriers may prevent users from achieving full intentional control: 1) users' lack of interest in programming and 2) users' failure to develop a sense of control.

Barrier: Lack of Interest in Programming: As programming environments, Generic Alice and Storytelling Alice only succeed if users actually spend time programming. 11 of the 12 Generic Alice participants who spent more than 50% of their time on scene layout created arbitrary motion programs. Not surprisingly, these users tended to learn very little programming (the average quiz score was 2.8). Based on observations within the study workshops, these participants found laying out 3D scenes significantly more rewarding than programming.

Barrier: Failure to Develop a Sense of Control: When users begin to program in Generic Alice, they frequently have goals (ranging from very simple to quite complex) that they want to pursue. As users begin working towards their goals, they often carry out a series of exploratory experiments. If users' initial experiments and the affordances within the interface do not help users develop an idea about how to accomplish their goals, many of them simply give up. Rather than forming a new more approachable goal, users often stopped trying to explain the behavior of their program and began to add animations and programming constructs at random.

Storytelling Alice Programs

We observed that Storytelling Alice encouraged users to identify a story goal quickly and begin working towards that goal. The programs created with Storytelling Alice were of three general types: relationship stories, good vs. evil stories, and miscellaneous programs.

Relationship Stories (51%): 22 of the 43 users created stories about relationships, including romantic relationships, peer relationships, and familial relationships. Users used relationship stories to explore issues that were potentially relevant in their own lives. See Figures 2-1 and 2-2.

Relationship stories dealt with a range of issues including jealousy between two girls who liked the same boy, struggling to fit into a social group, and divorce. These topics may indicate that girls used the story creation in Alice as a way to think through issues in their lives.

Good vs. Evil Stories (21%): 9 of the 43 users created stories depicting conflicts between good and evil. See Figure 2-2. In the good vs. evil stories created by girls using Storytelling Alice, violence or the threat of violence were often (but not always) employed as a way to resolve conflicts. For example, in one story, an evil samurai attacks an innocent pig. A good magical tree resurrects the pig, enabling the pig to attack the samurai in retaliation.

Other Programs (28%): 12 of the 43 programs created with Storytelling Alice do not fall neatly into a single category. These miscellaneous worlds include two stories about finding lost dogs, two stories depicting running and swimming races, and three choreographed routines (circus and cheerleading) similar in nature to the dance routines created by Generic Alice users.

Nearly all of the users of Storytelling Alice made stories (with the exception of the 3 choreographed routines). Further, all of the users of Storytelling Alice (as compared to 38% of Generic Alice users) moved from experimental programming into intentional programming. Storytelling Alice helps to minimize the time to identify and begin working towards a goal.

While all of the users successfully created intentional programs, some were more complex than others. For example, one user created a crying animation which required her to create a new method for her character, learn how to use `loops`, `dotogethers`, and control the character's hands. Other users focused most of their attention on dialog and used existing animations such as walking. There are two potential reasons for these users' focus on a small subset of the system's functionality: 1) users do not know what is possible within the system 2) users cannot map programming tools to their story goals.

Barrier- Determining what is possible within the system: As users interact with any software system, they build a mental model of what they believe is possible within that system. Often that model will not incorporate all of the capabilities within the system. Then, users select goals that match their beliefs about the system capabilities. In Storytelling Alice, this can lead to users exploring only a small subset of the programming tools available within the system.

Barrier- Finding appropriate programming tools to realize a story goal: While some users begin to discard goals that fall outside of their mental model for the system, other users continue to suggest and pursue goals that would require the use of unfamiliar concepts and constructs within the system. However, given the large number of possible actions within the system, it can be difficult for a new user to evaluate which programming tools are most appropriate to their goals.

Conclusion

The results of this study suggest that the storytelling focus made learning to program more engaging for middle school girls. It is clear that more work remains to develop programming environments that can engage a broad spectrum of girls in learning basic computer programming. As we continue to design and develop programming environments and curriculums, the barriers to programming engagement encountered by Generic Alice and Storytelling Alice participants represent important problems that should be considered.

References

American Association of University Women (1990). *Shortchanging Girls, Shortchanging America.* Washington, DC: American Association of University Women.

Bruckman, A. (1997) *MOOSE Crossing: Construction, Community, and Learning in a Networked Virtual World for Kids.* Unpublished doctoral dissertation, MIT Media Lab. Boston, MA. Retrieved April 14, 2009 from Web site: http://www.cc.gatech.edu/~asb/thesis/

Bruckman, A., Jenson, C., & DeBonte, A. (2002). Gender and Programming Achievement in a CSCL Environment. In *Proceedings of the 2002 Conference on Computer Supported Collaborative Learning 2002 (*119-227).

College_Board (2008). 2008 A.P. Exam National Summary Report. Retrieved April 14, 2009 from Website: http://www.collegeboard.com/student/testing/ap/exgrd_sum/2008.html

Conway, M. Audia, S, et al. (2000). Alice: Lessons Learned from Building a 3D System for Novices. In *Proceedings of the 2000 Conference on Human Factors in Computing Systems* (486-493). New York: ACM Press.

Dossey, J., Mullis, I., et al. (1988). *The mathematics report card: Are we measuring up?* Washington, DC: National Center for Educational Statistics (ERIC Document Reproduction Service No. ED 381438).

Duplantis, W., MacGregor, E., Klawe, M., and Ng, M. (2002). Virtual Family: an approach to introducing Java programming. *Proceedings of the 2002 Special Interest Group on Computer Science Education Conference* (40-43). New York: ACM Press.

Fennema, E. and Sherman, J. (1977). Sex Related Differences in Math Achievement, Spatial Visualization and Affective Factors. *American Educational Research Journal, 14*, 51-71.

Flanagan, M., Howe, D. and Nissenbaum, H. (2005). Values at play: design tradeoffs in socially-oriented game design. *Proceedings of the 2005 Conference on Human Factors in Computing Systems* (751-760). New York: ACM Press.

Gill, J. (1994). Shedding some new light on old truths: student attitudes to school in terms of year level and gender. *Proceedings of the 1994 American Educational Research Association Conference.* (ERIC Document Reproduction Service No. ED 381438).

Harel, I. (1991). *Children Designers*. Norwood, NJ: Ablex Publishing.

Kafai, Y (1995). *Minds in Play: Computer Game Design as a Context for Children's Learning.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Kelleher, C., D. Cosgrove, D. Culyba, C. Forlines, J. Pratt, and R. Pausch. (2002) Alice 2: Programming without Syntax Errors. *Proceedings of the 2002 Conference on User Interface Software and Technology*. New York: ACM Press.

Kelleher, C. and R. Pausch. (2006). Lessons Learned from Designing a Programming System to Support Middle School Girls Creating Animated Stories. *Proceedings of 2006 IEEE Symposium on Visual Languages and Human-Centric Computing* (165-172). New York: ACM Press.

Kelleher, C., R. Pausch, and S. Kiesler. (2007). Storytelling Alice Motivates Middle School Girls to Learn Computer Programming. *Proceedings of the 2007 Conference on Human Factors in Computing Systems* (1455-1464). New York, ACM Press.

Linn, M., (1985). Fostering Equitable Consequences from Computer Learning Environments. *Sex Roles 13(3-4),* 229-240.

Vegso, J. (2006). Drop in CS Bachelor's Degree Production. *Computing Research News 18(2).*

Zimmer, L. and Bennett, S. (1987). Gender Differences on the California Statewide Assessment of Attitudes and Achievement in Science. *Proceedings of the Annual Meeting of the American Educational Research Association.* Washington, DC.