Alternative Pathways into Computer Science: Investigation of Narrative-Bound Interactive Learning Environments for Teaching Girls Programming

Objectives and Purposes

Studies conducted over the past fifteen years have documented how underrepresented women and girls are in computer science- and IT-related fields involving programming and design work. Women continue to lag well behind men in information technology jobs, particularly at the design and programming levels, which offer higher salaries. Overall, only 27% of IT workers are women, even though they constitute roughly 50% of the labor force (Carver, 2000; U.S. Department of Labor, 2003). Even among those women who entered the IT pipeline and earned doctoral degrees in computer science, few have ended up in technical careers (Long, 2001). These real world inequities are mirrored in our pre-college educational system. Girls represent only 16% of the computer science Advanced Placement test takers and 10% of the higher-level computer science test takers (College Board, 2003). Computer science (understanding how information technologies work) is a gatekeeper to full participation in technology, research and development. For girls to have a hand in the directions technology takes in the future they need to develop skills and conceptual knowledge about what underpins information technology, namely programming. To facilitate this development, we need to better understand how to motivate and interest girls in learning the fundamentals of programming.

In this poster/demonstration session, researchers will share formative research findings and demonstrate features of an interactive learning environment developed for a two-year experimental study. The study is systematically investigating the power of narrative in enabling high school girls to learn key programming concepts and skills. The key hypothesis is:

> When girls are given the opportunity to learn programming situated in a meaningful narrative context their understanding of key computational concepts and skills is enhanced.

The experimental study has involved developing, testing, and teaching an introductory programming curricular unit to a treatment and control group of ninth grade students attending introductory high school computer application classes. The curricular unit engages students with one of two interactive learning environments built in Flash/ActionScript. Both interactive learning environments enable users to combine and modify various move elements (e.g., jumps/springs, turns, etc.) into executable, cohesive programs. Students in the treatment group use the metaphor of choreographing ice-skating routines to facilitate the learning of computer programming concepts and skills and to make the learning more accessible to girls. In the control group, students use identical move elements (e.g., jumps/springs, turns, etc.) to assemble routines as the treatment group, but instead of a graphic interface that shows a skater skating on a rink, the interactive learning environment has been customized to enable students to work with an abstract shape moving across the screen (as in a screen saver). The skating metaphor is absent from this version.

Modes of Inquiry and Data Sources

During the first half of year one, we worked closely with fourteen middle school students (6 girls, 8 boys) at Trevor Day School in New York City to inform the preliminary development of the curriculum and the interactive research tool that would serve as the basis for the experimental study. This was to ensure that individual class activities and the interactive tool worked from a functional perspective. Trevor Day School is a private school in New York City whose goals are to build an activity-based setting in a diverse coeducational community that emphasizes responsibility, collaboration, and mutual respect.

During the second half of year one, the refined interactive research tool, curriculum, and assessment instruments were pilot tested with one ninth grade class in Union City, New Jersey (representative of our target grade and population). The goal of the pilot test was to observe the effectiveness of revisions made to the curriculum and interactive research tool from the first trial. Sixteen students (7 girls, 9 boys) from an introductory computer applications class in a high school career academy participated in this pilot test. Union City is an urban, low-income, predominantly Latino community that has a long history and successful track record of SMET reforms. Union City's student performance scores on state tests are comparable to New Jersey's suburban districts.

This proposed poster/demonstration session aims to share insights gained from developing and testing the interactive learning environments designed to systematically study how to interest girls in programming and build their knowledge of core programming skills. After offering the audience a brief exploration of the learning environments, this session will draw on data gathered from students' structured journals, ethnographic observations, performance tasks, and transfer tests, to discuss:

- Challenges and successes in developing interactive learning activities that support and correspond to core foundational programming skills and concepts
- Developmental and situational differences between middle and high school students' approaches to exploring programming
- Issues of balancing free exploration of a graphical environment and directed skill-based activities
- Modifications to the learning environment itself that were required to effectively scaffold the learning of core programming concepts
- Issues of providing enough awareness of what is "under the hood" of programming, without requiring extensive prior knowledge and skills to actually do full programming

Theoretical Framework

Two decades of research has documented women's and girls' persistent lack of participation and interest in computer programming and computing (American Association of University Women (AAUW), 2000; Bruckman, Jensen, & Debonte, 2002; Linn, 1985; Margolis & Fisher, 2002). Fifteen years ago, Linn found in a study of middle school programming classes that girls and boys have similar levels of programming achievement once they enroll in classes, but few girls choose to enroll (Linn, 1985). Similarly, Schofield (1995) documented the complex social factors that explain how computer labs easily become clubs for young white boys at the high school level. Despite advances in programming itself, little has changed in the computer classroom. In one study of an innovative programming environment that makes use of visual interfaces, girls who enrolled in the classes spent less time programming in and out of class and less time on task (Bruckman, et al., 2002). Current thinking on girls' lack of participation in programming has been characterized by a commission of the AAUW as the "I can, but I don't want to" conundrum; it isn't that girls are not capable of doing programming but that they choose not to for a complex web of reasons (AAUW, 2000). One prominent cause pointed to by researchers and educators alike is that traditional computing and programming courses do not reflect multidisciplinary interests and the ways of working of many young women.

Programming is not intrinsically interesting to many girls, in part, because it is typically taught in a relatively abstract manner with an emphasis on either procedural knowledge (how to program the computer to do some pre-determined chore) or the capabilities of the machine (AAUW, 2000; Margolis, Fisher & Miller, 1999; Margolis & Fisher, 2002; Turkle, 1984, 1988). Research conducted over the past ten years tells us that girls, in particular, are more likely to be attracted to learning skills situated in a meaningful context than to the possibility of programming for programming's sake (Bennett, 1996; Brunner, Bennett, & Honey, 1998; Brunner & Bennett, 2002; Honey, 1994). For many girls, programming has to be a means to a motivating end, a way to solve a genuine social, personal, or environmental problem (Margolis & Fisher, 2002).

Sherry Turkle's and Seymour Papert's early work on gender and computing revealed that women in Harvard programming courses often exhibited a "bricoleur" or "concrete" approach to programming characterized as arranging and rearranging program elements as if they were material elements comprising a whole -- much like brushstrokes that are slowly being added to a painting with no formal end plan (Turkle & Papert, 1990, 1992; Turkle, 1984, 1988). This approach contrasted with a more hierarchical, rule-driven, "abstract" approach that was commonly exhibited by young men and supported by the programming curriculum. While some studies have taken issue with how "concrete" and "abstract" approaches are defined and whether these approaches are gender dependent (McKenna, 2001), others continue to find that programming courses fail to meet the needs of diverse learners and do not succeed in sustaining young women's interests (Margolis & Fisher, 2002).

Clancy and Linn (1999) argue that to teach programming requires the use of "case studies and scaffolded examples." They assert that in order to abstract patterns and apply them to new situations programming exercises need "rich connections to examples and multiple links to context of use" (Clancy & Linn, 1999, p. 40). Linn and others (Hoyles, Noss, Adamson, & Lowe, 1999) note that this approach is even more essential at a time when programming languages are rapidly evolving and changing. We argue further that this approach may be especially conducive to girls, given research in other areas suggesting the role narrative plays in increasing girls' interests in science and technology (Caleb, 2000; Davidson & Schofield, 2002; Margolis & Fisher, 2002; Rosser, 1990, 1995; Schofield, 1995; Turkle, 1984, 1988; Turkle & Papert, 1990, 1992).

With the advent of object-oriented programming languages (such as Java and Flash), it is possible to make programming more accessible to girls since object-oriented languages allow one to program by manipulating the relationships between objects in a narrative context rather than executing abstract, linear code. While other researchers (Kafai, 1996, 1998) have found some indication that narrative context plays a role in engaging children in software design (e.g., by providing impetus for girls to create video games), no systematic study exists to test the value of such interventions for girls in the area of programming. This poster/demonstration session will share formative findings that shed light on the role narrative context embedded in an object-oriented programming environment might play in enhancing girls' ability to learn core programming concepts and skills.

Methods and Techniques: The Interactive Learning Environment

This past year, the study has focused on the development and testing of an introductory programming curricular unit and interactive learning environment built in the object-oriented programming language, FLASH/Actionscript. The interactive learning environment, known as Soft Ice, is a drag and drop software tool that enables users to choreograph ice-skating performances by combining and modifying various move elements, such as jumps and turns, into executable, cohesive skating programs. Students interact with the interactive learning tool on three levels:

- Arranging preprogrammed objects into a sequence of movements
- Manipulating variables and parameters for each object's movement and its characteristics

- Editing the actual textual code which control the movements

Using ethnographic observations, structured journals, and transfer tasks, the aim of our formative research was to test how the affordances of object-oriented programming can be maximized through a metaphor that does not window dress or sugar coat, but that corresponds with the principles of creating and modifying code. Ice-skating is potentially an ideal metaphor for communicating programming skills and concepts because ice skating moves must be arranged and executed properly in a winning sequence to achieve results, not unlike programming itself. Thus computing elements and processes like functions, variables, loops, conditionals, debugging, etc. each have a corresponding element in the design and performance of an ice skating routine.

Results/Conclusions

Both formative trials were instrumental in shaping the design of the curricular intervention and the *SoftIce* interactive learning tool. Our formative research confirmed that ice-skating is widely accessible to children of different backgrounds and also offers direct analogues to programming concepts that are considered foundational in high school computer science (e.g., objects, properties of objects, arrays, functions/methods, and conditional statements).

Educational Importance of Study

This session will examine whether girls can learn as much or better if one reverse-engineers what a traditional programming curriculum typically does and provides a meaningful context for the investigation of core programming skills and processes (i.e., reading, using, modifying, and implementing aspects of code) advocated by the Advanced Placement Computer Science curriculum and the Association for Computing Machinery. Findings from this study will help inform the future development and reform of computer science curricula for youth, particularly as Advanced Placement computer science guidelines move toward the use of Java programming as the basis for introductory computer science at the high school level.

References

American Association of University Women. (2000). *Tech-savvy: Educating girls in the new computer age*. Washington, DC: AAUW Educational Foundation.

Bennett, D. (1996). *Voices of young women in engineering.* New York: EDC/Center for Children and Technology. Originally a paper presentation for MIT Symposium.

Bruckman, A., Jensen, C., & DeBonte, A. (2002, January). Gender and programming achievement in a CSCL environment. Long talk, *Proceedings of CSCL 2002*. Boulder, CO: Computer Support for Collaborative Learning. Available online at: http://newmedia.colorado.edu/CSCL/63.pdf.

Brunner, C., & Bennett, D.T. (2002). The feminization of technology. In N. Yelland and A. Rubin (Eds.), *Ghosts in the machine: Women study women and technology* (pp. 71-96). New York: Peter Lang.

Brunner, C., Bennett, D., & Honey, M. (1998). Girl games and technological desire. In J. Cassell & H. Jenkins (Eds.), *From Barbie to Mortal Kombat: Gender and computer games (*pp. 72–87). Cambridge, MA: MIT Press.

Caleb, L. (2000). Design technology: Learning how girls learn best. *Equity & Excellence in Education*, *33*(1), 22-25.

Carver, D. (2000). *Research foundations for improving the representation of women in the information technology workforce: Virtual workshop report*. Baton Rouge, LA: Louisiana State University , Department of Computer Science.

Clancy, M., & Linn, M. (1999). Patterns and pedagogy. *SIGCSE Bulletin, 31* (1), 37-42.

College Board. (2003). *Participation in AP: Women*. Retrieved 3/31/03, from: http://apcentral.collegeboard.com/article/0,3045,150-156-0-2060,00.html

Davidson, A.L., & Schofield, J. W. (2002). Female voices in virtual reality: Drawing young girls into an online world. In K.A. Renninger & W. Shumar (Eds.), *Building virtual communities: Learning and change in cyberspace* (pp. 34-59). New York: Cambridge University Press.

Honey, M. (1994). The maternal voice in the technological universe. In D. Basin, M. Honey, & M. Kaplan (Eds.), *Representations of motherhood* (pp. 220-239). New Haven: Yale University Press.

Hoyles, C., Noss, R., Adamson, R., & Lowe, S. (2001). Programming rules: What do children understand? In *Proceedings of the Twenty-Fifth Annual Conference of the International Group for the Psychology of Mathematics*, Utrecht: The Netherlands. Available online at: http://www.ioe.ac.uk/playground/RESEARCH/papers/programming_rules.pdf

Kafai, Y.B. (1996). Gender differences in children's constructions of video games. In P.M. Greenfield & R.R. Cocking (Eds.), *Interacting with video* (pp. 39-66). Norwood, N.J.: Ablex.

Kafai, Y.B. (1998). Video game designs by girls and boys: Variability and consistency of gender differences. In J. Cassell and H. Jenkins (Eds.), *From Barbie to Mortal Kombat: Gender and computer games* (pp. 90-114). Cambridge, MA: MIT Press.

Linn, M. C. (1985). Gender equity in computer learning environments. *Computers and the Social Sciences 1*, 19-27.

Long, J.S. (Ed.). (2001). *From scarcity to visibility: Gender differences in the careers of doctoral scientists and engineers*. Washington, DC: National Academy Press.

Margolis, J., & Fisher, A. (2002). *Unlocking the computer clubhouse*. Cambridge: MIT Press.

Margolis, J., Fisher, A., & Miller, F. (1999, December). Caring about connections: Gender and computing. *IEEE Technology and Society Magazine 13* (20). Available online at: http://www-2.cs.cmu.edu/~gendergap/papers/IEEE99.html

McKenna, P. (2001). Programmers: Concrete women and abstract men? *Journal of Computer Assisted Learning, 17* (4), 386-395.

Rosser, S. V. (1990). *Female-friendly science: Applying women's studies methods and theories to attract students*. New York: Pergamon Press.

Rosser, S.V. (1995). *Teaching the majority: Breaking the gender barrier in science, mathematics, and engineering*. New York: Teachers College Press.

Schofield, J. (1995). *Computers and classroom culture*. New York: Cambridge University Press.

Turkle, S. (1984). *The second self: Computers and the human spirit*. New York: Simon and Schuster.

Turkle, S. (1988). Computational reticence: Why women fear the intimate machine. In C. Kramarae (Ed.), *Technology and women's voices* (pp. 41-61). New York: Routledge & Kegan Paul.

Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of Women in Culture and Society, 16* (1), 128-157.

Turkle, S. & Papert, S. (1992). Epistemological pluralism and the revaluation of the concrete. In I. Harel & S. Papert (Eds.), *Constructionism* (pp.161-191). Norwood, NJ: Ablex.

U.S. Department of Labor: Bureau of Labor Statistics. (2003). *Current population survey conducted by the Bureau of Census*. Retreived 4/01/03, from: ftp://ftp.bls.gov/pub/suppl/empsit.cpseea19.txt or http://www.bls.gov/cps/home.htm