RAPUNSEL: How a computer game design based on educational theory can improve girls' self-efficacy and self-esteem

Jan L. Plass*, Ricki Goldman*, Mary Flanagan**, James P. Diamond*, Chaoyan Dong*, Suyin Looui**, Christine Rosalia*, Hyuksoon Song*, Ken Perlin*

*New York University, **Hunter College

There is a critical shortage of women in computer science (CS) careers and higher education degree programs. In their landmark study of gender and CS, Margolis and Fisher (2001) noted that male dominance in information technology can be linked to social, cultural, and educational influences and patterns formed in childhood. Research shows that although girls are as talented as boys in math and science, and although most girls are excited about science during childhood, these same girls begin to lose interest in math and science in middle school (Catsambis, 1994; Chu Clewell, 2002). By the eighth grade, twice as many boys as girls show an interest in science, engineering, and mathematics careers (CAWMSET, 2000). Although opportunities for CS-related careers are broadening, and programming skills are required in many diverse fields, fewer and fewer girls are attracted to CS related activities; fewer than 33% of participants in computer courses and related activities are girls (CAWMSET). New approaches and resources are clearly needed to engage girls in computer programming activities.

The RAPUNSEL project

The goal of this three-year, NSF-funded research project was to design a software environment for real-time, applied programming for underrepresented students' early literacy (RAPUNSEL). The objective was to develop an engaging environment in which to teach computer programming to middle school girls in a scalable, approachable manner that appeals to girls' senses of curiosity and play, as well as their desire to communicate with one another. Game environments have the potential to engage players in interesting variations of constructive play (Gee, 2003; Rieber, 1996; Squire, 2003). Players typically do not learn extensive technical skills by playing commercial video games, however. Criteria for truly equitable software must go beyond representation and game scenarios and allow for models that empower students to become software designers and have technical and creative control over their own environments. The RAPUNSEL project developed a game world called *Peeps*, which is designed to provide girls with opportunities to design parts of the game and, in the process, develop computer programming skills.

Insert Figure 1 about here

The coding interface is implemented in a way that makes syntax errors impossible and provides immediate visual feedback on the code that they have developed (Figure 1). This makes *Peeps* a safe way to learn through "tinkering," as Seymour Papert would say, enabling students to build self-confidence and self-efficacy in areas related to computer programming (Schunk & Zimmerman, 1998).

Theoretical overview

Constructionism. The theory that most closely connects to this project dedicated to girls becoming computer programmers is constructionism, developed by Papert (1980) in his book *Mindstorms: Children, Computers, and Powerful Ideas*. Constructionism focuses not on Piagetian stages of development and the constructivist nature of the individual mind, but rather on the knowledge that is created by children when designing, building, making, and doing. Engaged in constructionist activities, children create games to play or characters that can dance and move around virtual spaces. Papert and his team of ten researchers were probably among the first groups to explore self-efficacy and self-esteem in girls while making computer programs that were interactive and self-motivating (Harel, 1991). Harel, for example, found that Ebby, a young girl who was struggling with self-esteem, became one of the leaders in her school community when she began to learn how to program and could share her knowledge with her classmates.

What is critical in Papert's theory of constructionism is that girls and boys need to be challenged, not only entertained, when using computers to learn. In the right kind of environment, they become epistemologists learning powerful ideas. Powerful discussions about playing and making occur among boys and girls when they have been released from the busywork of many of their classroom activities. A powerful example is the story of two boys who programmed games together. The setting was Papert's research project in a Boston inner-city school. Two 9-year old boys, while programming their game where cars maneuver around obstacles, discussed whether it is better to make the game or play the game that you made with researcher Goldman (Goldman-Segall, 1998) [At http://www.pointsofviewing.com, click on video clip to view this interaction]. The point is that they were learning by doing, not only learning about programming, but learning how to think, to tinker, to putter, to make mistakes and to learn from them.

Case-based learning. In the Rapunsel study, we used a mixed-initiative framework in our pedagogical design to offer several ways to approach the domain tasks designed in the research (Lester, Stone, & Stelling, 1999). The staged "missions" in *Peeps* scaffold learning through appropriate tasks, encouragement, and quality tools that offer players formative feedback. Scenario design, character interaction, and environment design are our primary approaches to support learner growth and self-instruction. We designed and implemented learning strategies that can be acquired in real contexts and learned in such a way as to encourage transfer (Nisbet & Schucksmith, 1986). The learning environment monitors students' progress, and computer agents present curious and motivating clues for players to encourage coding and spark trial and error experiments (Moreno & Mayer, 2005). Such interactions are particularly important for our constructionist learning approach and encourage learners to participate in active problem solving (Schank, Fano, Bell, & Jona, 1993–1994).

Schank and Cleary's (1995) *teaching architectures* combine several models that can be used to attract middle school minority girls to math and science by enabling them to experience authentic problems in an engaging environment. Specifically, *Peeps* employs elements of simulation-based learning-by-doing and case-based learning. Learning-by-doing, as employed in this game, necessitates students' active participation in authentic, real-world tasks in order to solve the problem at hand. The game requires players to conceptualize and execute programming code in order to customize their own character (or avatar) and to function successfully in the game world. *Peeps* uses case-based scenarios that require students to solve problems by using knowledge that they have developed in previous experiences, or by acquiring new knowledge. The game is designed in a way that simple problems can be solved without programming, and intermediate-level problems with simple programming, such as the adjustment of parameters. To solve more difficult problems, the game uses a *code editor* in which learners can use a Java-like programming language to solve a problem or customize their character in the way they wish. The code editor uses pull-down menus for input, which means that there cannot be any syntax errors. A series of tutorials and challenges scaffold students as they develop programming skills and provide them with a set of cases from which to draw when encountering new problems.

Video game-based learning. Increasingly, video games are being investigated for their instructional applications. Dickey (2005), Gee (2003), Shaffer, Squire, Halverson, and Gee (2005) and Shaffer (2006), among others, have all suggested that video games are environments that allow for "thickly authentic" (Shaffer & Resnick, 1999) learning, or learning that enables students to acquire knowledge that is personally meaningful, has real-world application, and that is associated with practice, rather than rote memorization. Shaffer, Squire, Halverson, and Gee emphasize that video games integrate "knowing" with "doing," thus facilitating explicit connections between learners' knowledge and the tasks that they perform to which that knowledge is applied. Games may mend what Brown, Collins, and Duguid (1989) refer to as the "breach between learning and use" and, through practice and reflection, enable learners to make the connection "knowing what" and "knowing how" (p. 32).

Video games frequently allow players to personalize their avatars and their environments (see, for instance, *The Sims*) and to pursue their goals within the context of a structured narrative. Cordova and Lepper (1996) found that personalization, contextualization, and choice in a simple computer game significantly increased intrinsic motivation and learning outcomes for elementary school students. Shaffer (2005) described increases in self-confidence and in content knowledge in mathematics and art for a young woman who role-played as a graphic designer in a computer-simulated design studio. Further, he suggested that her achievements in the game transferred to other areas of her academic life, including her ability to communicate confidently with teachers and her peers. In playing games that encourage them to play roles and practice skills associated with those roles, Shaffer argued that students develop "epistemic frames," or ways of knowing about the world that are informed by disciplinary habits, in this case those of the design profession. Frames inform students' identities and empower them to approach problems with a set of core skills that they have learned within a community of practice.

The present study. This study explores how a computer game in which players can use a programming language to solve problems and change their avatar and its behavior affects children's—particularly girls'—motivation to program, their acquisition of programming concepts, and studies the extent to which girls' overall levels of confidence and self-efficacy are affected by the *Peeps* environment.

Method

Participants and Design. Ninety 6th graders at a school in a large metropolitan area in the American northeast voluntarily participated in this study. Among these participants, 59 submitted the parental consent form and completed both the pre- and post-tests. In age, they ranged from 10 to 13 years old; 29 (52%) were female (gender data was missing for 3 participants). The sample was ethnically diverse, with 20 students describing themselves as Caucasian/European-American (35.1%), 14 as Latino (24.6%), 11 as African or African-American (19.3%), 4 as Asian or Asian-American (7.0%), 1 as Native American (1.8%), 1 as Middle Eastern or Arab-American (1.8%), and 6 as "Other ethnicity" (10.5%). Of the participants with complete records (data for 7 students are missing due to technical problems), 31 used computers frequently (59.6 %) and 21 rarely used a computer (40.4 %). Twenty-one students frequently played games (58.3%), 13 rarely played games (36.1%), and 2 did not play games at all (5.6 %) (data for 2 students are missing due to technical problems). Reflecting the exploratory nature of this research, we collected data using a combination of pre-post survey and qualitative observations.

Setting. The study took place in the computer lab at a small public secondary school. The computer room had five round tables with six computer stations in each table and a teacher desk. The computer teacher was present at all times and supported the goals of the project, regularly interacting with the faculty and graduate student researchers. Often, other teachers would pop into the room to print out a document if a space was available at a computer workstation. The setting was dynamic and typical of a computer lab rather than a classroom. At least two of the faculty investigators were present at each visit. Four to five doctoral students performed a range of tasks: taking notes, giving instructions to the young people about the tasks to be done that day, ensure that the work that each student completed was saved, and trouble-shooting technical problems while students were playing the game. Similar to Papert's Hennigan Project *Headlight* (Goldman-Segall, 1998; Harel, 1991) the context was highly interactive as adults and, at times, students moved from table to table to see what others were doing as they played the game. However, unlike Project *Headlight*, these students were given tasks (missions) to accomplish and individual achievement in completing these tasks was recorded. In short, the study was a mixed method approach where researchers were not only collecting quantitative data, but were at times participant observers of the study itself (Hammersley & Atkinson, 1995).

Materials. The game introduces students to the *Peeps* world, an environment in which players must achieve their objectives by learning to program their on-screen avatars through the use of a "code window." A series of tutorials and game-related quests function as instructional scaffolds in order to assist players in learning new programming concepts as they play the game. In the *Peeps* environment, students (both boys and girls) play a female character who interacts with the inhabitants of the world by dancing with them. Students create dances by using increasingly complex computer programming skills. Players must also avoid the *Gobbler,* a non-player character designed to steal pieces of code that players have developed or acquired during the game. Players are introduced to programming concepts such as *methods*, *parameters*, and *conditional* statements as they become necessary in order to advance through the game. Similar to Papert's (1980) vision for mathematics as the natural language of *Mathland,* the *Peeps* environment encourages students to "speak" in computer programming terms by immersing them in a game environment that fosters the learning of programming concepts as a way to advance in the game. Through supporting the development of these skills in an environment that encourages exploration and play, the game should influence overall self-efficacy and self-esteem in the area of computer programming.

Instruments and Scoring. A pre- and a post-survey were used to measure whether playing the game affected the following variables: general self-efficacy, self-esteem, computer self-efficacy, programming knowledge, and self-efficacy related to programming knowledge. The survey included a general self-efficacy scale, adapted from previous research by Jinks and Morgan (1999), which gauged participants' beliefs about their capabilities in academics using 9 questions. Participants' self-esteem was assessed by asking 5 questions to determine how much a student likes, accepts, and respects herself or himself as a person (Felson & Zielinski, 1989). Computer self-efficacy was measured by asking participants' beliefs about their computer-related abilities with 11 questions (Durnell & Haag, 2002). In order to measure participants' programming knowledge, we developed four questions. Students' programming self-efficacy was measured by asking them to rate "How confident are you that your answer to above question is correct" for each question. In the last of four sessions of using *Peeps*, participants were given "Missions" in which they had to apply the programming knowledge they had acquired. The code they wrote for each mission was collected in log files, along with all other action they took in the *Peeps* game.

Procedures. The study was conducted in the school's computer lab for four 50-minute sessions over the course of one month. During Sessions 1 and 4, all students who submitted a parental consent form completed the pre-and post survey, respectively. For the end of Session 1 and the entirety of Session 2, each student was asked to complete a series of in-game tutorials designed to teach students about the game's navigation and interface and to introduce them to basic programming concepts. After finishing the tutorials, students were free to play the game on their own. In Session 3, the students were asked to complete two missions designed to assess the programming knowledge that they learned through playing in the previous sessions. The missions required students to use their programming language skills in order to create small pieces of computer code that prevented the Gobbler from stealing any code that they had previously developed.

Results and Discussion

This study employed quantitative methods in order to assess the game's impact in the areas of motivation, learning outcomes, self-esteem, general self-efficacy, and computer self-efficacy.

General self-efficacy. There was no pre-test difference in general self-efficacy for girls (M=1.85, SD=.54) and boys (M=1.98, SD=.66). A paired-samples t-test indicated significant differences in female students' general self-efficacy between the pre- and the post-test, $t(28) = -3.474$, $p<.05$, $d = .65$ while male students' general self-efficacy did not show significant differences. These results suggest that playing the game may have increased female students' general sense of self-efficacy.

Self-esteem. There was no pre-test difference in general self-esteem for girls (M=2.80, SD=.92) and boys (M=3.01, SD=1.03). A paired-samples t-test indicated significant differences between the pre- and the post-test for female, $t(28) = -3.540$, $p<.01$, $d = .66$, and male students, $t(23) = -2.370$, $p<.05$, $d = .48$. For female and male students, self-esteem increased after playing the game.

Computer self-efficacy. There was no pre-test difference in general computer self-efficacy for girls (M=1.85, SD=.78) and boys (M=1.65, SD=.99). A paired-samples t-test showed marginally significant differences in students' computer self-efficacy between the pre- and the post-test for male students, $t(26) = -1.896$, p=.069, $d = .36$; female students did not exhibit any significant changes. The results suggest that playing the game increased male students' self-efficacy in using computers.

Programming knowledge and programming self-efficacy. There were no pre-test differences in programming knowledge for girls (M=3.23, SD=.95) and boys (M=2.75, SD=1.62) and in programming self-efficacy for girls (M=2.64, SD=1.00) and boys (M=3.11, SD=1.31). Although a paired-samples t-test did not reveal a significant difference in students' programming knowledge between the pre- and the post-test, there was a significant difference in learners' level of programming self-efficacy for female students, $t(24) = 5.296$, $p<.0001$, $d = 1.06$, and a marginally significant difference for male students, $t(15) = 1.918$, p=.074, $d = .48$. This suggests that students' programming-related self-efficacy increased after playing the game.

Conclusion

The aim of the RAPUNSEL project and the *Peeps* game was to address issues relating to gender equity and the digital divide and we believe that environments such as *Peeps* may be able to increase computer literacy in our society. Computer literacy for all citizens will be imperative for the United States to maintain a diverse, internationally competitive, and globally engaged workforce of scientists, engineers, and well-prepared citizens. This literacy must include computer programming and computer science fundamentals and involve both reading (using existing computer applications) and writing (making one's own applications).

The research findings presented here indicate that computer games such as *Peeps,* which was designed with an eye to literacy and learning, may be able to influence motivation, self-efficacy, and self-esteem for populations of students that have traditionally been "turned away" from computer science-related fields. It is particularly encouraging that even with a relatively brief intervention, girls' general and programming-related self-efficacy and self-esteem increased with effect sizes that can be considered medium to large.

Literacy, however, does not simply involve technical expertise. Literacy is a widespread and socially engaged system of skills, capabilities, and creativity formed in the context of a material support (diSessa, 2000). Our project takes small steps towards big changes by addressing the unmet needs of middle school-age girls, but more research is needed to better understand the social, affective, and cognitive impact of computer games that are designed based on educational theory.

References

Brown, J.S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32–42.

Catsambis, S. (1994).The path to math: Gender racial-ethnic differences in mathematics participation from middle to high school. *Sociology of Education, 67,* 199–215.

Chu Clewell, B. (2002). Breaking the barriers: The critical middle school years. In S. M. Bailey (Ed.), *The Jossey-Bass Reader on Gender in Education* (pp. 301–313). San Francisco: Jossey-Bass.

Commission on the Advancement of Women and Minorities in Science, Engineering, and Technology Development (CAWMSET). (2000, September DATE). *Land of Plenty: Diversity as America's Competitive Edge in Science, Engineering and Technology*. Retrieved (January 12, 2001, from: http://www.nsf.gov/od/cawmset/report.htm

Cordova, D.L., & Lepper, M.R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology, 88,* 715730.

Dickey, M. (2005). Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational Technology Research and Development, 53*(2), 67–83.

diSessa, A. (2000). *Changing Minds: Computers, Learning, and Literacy.* Cambridge: MIT Press.

Durndell, A., & Haag, Z. (2002). Computer self-efficacy, computer anxiety, attitudes towards the Internet and reported experience with the Internet, by gender, in an East European sample. *Computers in Human Behavior, 18*(5*),* 521-535.

Felson, R., & Zielinski, M. A. (1989). Children's self-esteem and parental support. *Journal of Marriage & the Family, 51*(3), 727-735.

Gee, J.P. (2003). *What Video Games Have to Teach Us About Learning and Literacy.* New York: Palgrave Macmillan.

Goldman-Segall, R. (1998). *Points of Viewing Children's Thinking: A Digital Ethnographer's Journey.* Mahwah, NJ: Lawrence Erlbaum Associates.

Hammersley, M., & Atkinson, P. (1995). Ethnography: Principles in Practice. Routledge: Taylor & Francis Group.

Harel, I. (1991). *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school.* Norwood, NJ: Ablex Publishing.

Jinks, J., & Morgan, V. (1999). Children's perceived academic self-efficacy: An inventory scale. Clearing House, 72, 224-230.

Lester, J., Stone, B., & Stelling, G. (1999). Lifelike Pedagogical Agents for Mixed-Initiative Problem Solving in Constructivist Learning Environments. *User Modeling and User-Adapted Interaction, 9*(1-2), 1-44.

Margolis, J., & Fisher, A. (2001). *Unlocking the clubhouse: Women in computing*. Cambridge: MIT Press.

Moreno, R., & Mayer, R.E. (2005). Role of guidance, reflection, and interactivity in an agent-based multimedia game. *Journal of Educational Psychology, 97*(1), 117–128.

Nisbet, J. & Shucksmith, J. (1986). *Learning Strategies.* London: Routledge Education Books.

Papert, S. (1980). *Mindstorms: Children, computer, and powerful ideas.* New York: Basic Books.

Petraglia, J. (1998). *Reality by design: The rhetoric and technology of authenticity in education*. Mahwah, NJ: Erlbaum Associates.

Rieber, L. P. (1996). Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research & Development, 44*(2), 43-58.

Schank, R., Fano, A,, Bell, B., & Jona, M. (1994). The design of goal-based scenarios. *Journal of the Learning Sciences, 3*(4), 305–345.

Schank, R. C., & Cleary, C. (1995). *Engines for education.* Hillsdale, NJ: Erlbaum Associates.

Schunk, D., & Zimmerman, B. (1998). Self-Regulated Learning: From Teaching to Self-Reflective Practice. New York: The Guilford Press.

Shaffer, D.W. (2006). Epistemic frames for epistemic games. *Computers & Education, 46*(3), 223–234.

Shaffer, D.W., & Resnick, M. (1999). "Thick authenticity": new media and authentic learning. *Journal of Interactive Learning Research, 10*(2), 195–215.

Shaffer, D. W., Squire, K. D., Halverson, R., & Gee, J. P. (2005). Video games and the future of learning. *Phi Delta Kappan*, 87(2), 104-111.

Solomon, G. (1991, January). Mass Multimedia: Classrooms in Shrewsbury, Mass. Use Multimedia Technologies to Add "Real Substance" to Numerous Curriculum Areas. *Electronic Learning, 10*(4), 14.

Squire, K. (2003). Video games in education. *International Journal of Intelligent Simulations and Gaming, 2*(1), 49–62.

Figures

Figure 1. Screen shot from the *Peeps* game: The code editor