

Game Design: Whose game works at the end of the day?

Susan B. Miller
University of Colorado - Boulder
Susan.Miller@colorado.edu

David C. Webb
University of Colorado - Boulder
DCWebb@colorado.edu

ABSTRACT

A study of 48 completed Frogger games created by middle school students using the Scalable Game Design curriculum were studied to determine how completion rates varied based on race and gender. Results to a student survey administered to the same sample was also analyzed to compare those who completed fully functioning games to those whose games were not fully functioning. Overall, 67% of the girls, and 50% of minority students submitted fully functioning games; however, African American and Hispanic/Latino boys had lower completion rates. Survey results provide some insight to factors that may influence who submits a fully functioning game. Students' whose games were not fully functioning were less likely to see themselves as computer problem solvers, and were less likely to they see themselves pursuing computer classes in the future.

Categories and Subject Descriptors

• **Social and professional topics~K-12 education** • *Social and professional topics~Race and ethnicity* • *Social and professional topics~Gender* • **Social and professional topics~Adolescents** • *Computing methodologies~Active learning settings* • *Computing methodologies~Simulation by animation* • *Software and its engineering~Interactive games* • *Software and its engineering~Visual language*

KEYWORDS

Game design; middle school; broadening participation; computer science

1. INTRODUCTION

In the past year, diversity within IT companies has become a significant issue, with first Apple, and then Google sharing diversity numbers for both the technical and non-technical sides of their businesses [5, 7]. Both companies have agreed that broadening diversity within their businesses is a key goal, yet they also recognize that there are substantial hurdles when few women or minorities are pursuing degrees in computer science. The challenge, then, is to find ways that encourage female and minority students to explore computer science venues, and then persist in these venues so that they are able experience personal success with the skills they have learned.

2. LITERATURE REVIEW

Although schools and corporations are both working toward increased diversity in Computer Science, recent studies show that the students taking the Advanced Placement test for computer science are still largely White and Asian males [8]. Women and students of color are significantly underrepresented in the pool of test takers. In trying to determine what prompts students to pursue an interest in computer science, researchers [6] have found that when looking at college students, prior experience with programming, as well as being frequent gamers were indicators for men pursuing Computer Science (CS) degrees. For women, the indicators for pursuit of CS varied a bit, including expressing an interest in computing as well as interest in solving problems. Studies [1] found similar results for high school level female students, suggesting that an interest in the field significantly influenced whether students would pursue computer science.

Additionally, studies [3] show that there are racial differences as well, in why students may or may not choose to pursue an interest in computer science. Specifically, a lack of prior computer programming experience, as well as a general lack of experience with computers led to students choosing not to pursue CS as a field of interest. Finally, they also found that perceptions of those who pursued CS as being seen as a 'geek' also negatively affected their interest.

3. SCALABLE GAME DESIGN

Scalable Game Design (SGD), a research project supported by the University of Colorado, and funded through the National Science Foundation, endeavors to expose middle school students to programming by teaching them to design and create their own arcade-style game. In this project, students use AgentSheets software to first design and build their own version of Frogger, and later create additional games such as PacMan, Space Invaders, and Sokoban.

4. PROFESSIONAL DEVELOPMENT

Participating teachers complete a summer institute training session, learning how to build their own games and how to teach programing using a guided discovery approach [4], which studies suggest may be particularly motivating for girls [9]. Teachers also learn ways to focus students' attention toward computational thinking, using key computational thinking patterns (actions between agents that occur commonly in games and in simulations) [2]. This enables students to move from creating known games to designing their own games and later designing math and science simulations.

4.1 Software tool

The software tool used to support SGD, AgentSheets, plays a key role in this project's success. Students begin by describing their game. Nouns become the agents in the game, while action

verbs become the agents' behaviors. Drag and drop capability enables those behaviors to be placed into if-then rules to control the agents within the game. By combining rules, students are then able to form computational thinking patterns (such as generating a new agent, or a collision between two agents) to build the game incrementally.

Student motivation is also enhanced when students are able to gain some ownership in the process [9]. Through the SGD approach, students are able to design and create their own agents, and their own world, and then bring them to life through programming.

Finally, student frustrations can be reduced through the use of conversational programming, which is a set of tools that enables students to 'read' their code by turning it back into English sentences. It also uses real time color-coding (green/red) of true and false conditions to show which rules will 'fire' and which will not.

5. METHODS

As part of the SGD research project, students can upload their projects to the Scalable Game Design Arcade, enabling the game to be played by others. The SGD Arcade also allows the research team to see the completed projects. In this study, forty-eight projects were analyzed to determine which of the projects worked as designed, and to determine if there were differences in the functionality of the game based on gender and race. To minimize any variances resulting from different instructional styles, the projects selected included all projects submitted to the arcade during three semester-long classes conducted by the same teacher where the goal of the class was to create a functional Frogger game. All of the students involved in the creation of these projects were middle school students (Grades 6, 7, and 8). Table 1 shows the breakdown of the students' self-identified gender and race.

Table 1: Games by race and gender

	White	African American	Hispanic/Latino	Asian	Mixed	Total
Male	19	2	6	1	2	30
Female	4	4	3	5	2	18
Total	24	6	9	6	5	48

As the class is an elective class for middle school students, the students are allowed to choose whether they take this class. As other studies have found, this self-selection into the class left the population with a higher percentage male enrollment than female, with 50% of the students self-reporting as White.

Each game included in this research was a version of Frogger, typically the first game created in the SGD curriculum (See Figure 1 for an example). Games were analyzed to determine if they were fully functional. A fully functional game was one in which there were two parts of the game – the frog needed to be able to move across the road (avoiding cars/trucks) and the frog needed to be able to move across the river on the backs of the turtles and logs.

This phase of the Frogger game is particularly interesting to study as the lesson sequence includes beginning with a guided discovery pedagogical style to enable students to understand how to use the software and create the first part of the game (the

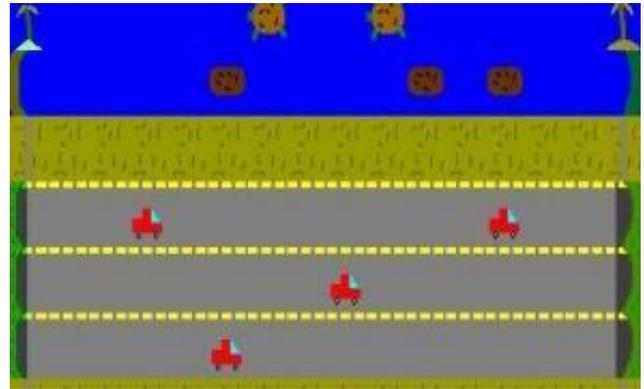


Figure 1: More traditional Frogger game

frog can cross the road). Once that is completed, students then work more independently to recreate that programming process for the second part of the game (the frog can cross the river). The ability to successfully build the second half of the game was the measure used in this study to determine if the student had mastered the programming requirements for this game.

In both portions of the game, the moving objects (cars/trucks and turtles/logs) needed to generate new agents on one side of the screen, and absorb them on the other side. The frog needed to be cursor controlled and move in all four directions (up, down, left and right). A functional game also meant that that frog would 'die' when hit by the vehicle, or when it landed in the water (See Figure 2 for example of typical code). The frog won when s/he arrived at the grotto.

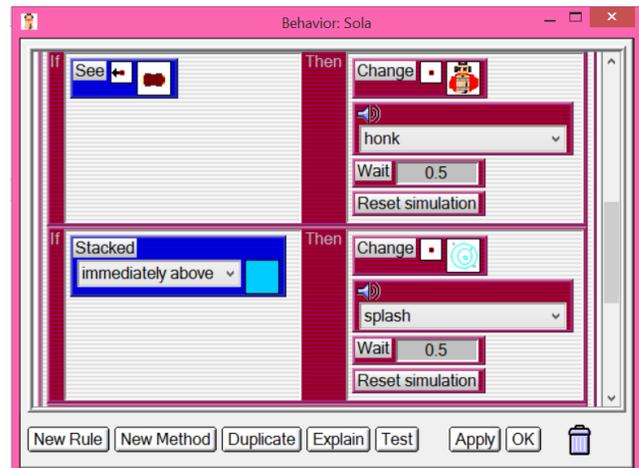


Figure 2: View of code from typical game indicating a collision between two agents.

Since one of the principles of SGD is to encourage students to make the game their own, many Frogger games include variations on the main agent, such as Dogger or Hogger. In each game, we looked at the overall project, and did not concern ourselves with the actual agent or worksheet design; instead, the primary focus to determine game completion was on the programming of each agent.

Most of the students who submitted these projects (45 out of 48) had also completed a survey prior to taking a class using the SGD curriculum. These survey results were reviewed after the completion of game analysis and coding to determine if student responses could inform these findings.

6. ANALYSIS

6.1 Game Analysis

Each game was coded based on the results of the analysis. Games that met all of the criteria described previously were coded as *fully-functional*. Games that did not meet all of the criteria were studied further to determine what portions of the criteria were not met. The programming code for each non-functional game was reviewed to determine whether the code was missing entirely, or if the code was completed incorrectly.

Overall, 60% of the games were determined to be fully-functional. When looking at the results by gender, 57% of the boys' games were fully-functional, while 67% of the girls' games were fully-functional. When considering race, 63% of White students, 50% of African American students, 44% of Hispanic/Latino students, 83% of Asian students, and 40% of mixed race students had fully-functional games.

Table 2: Fully-functional games by race and gender

	White	African American	Hispanic/Latino	Asian	Mixed
Male	74%	0%	17%	100%	50%
Female	25%	75%	100%	80%	50%

When looking within groups, there are some groups with significantly smaller percentages of the students successfully creating a fully-functional Frogger game. Table 2 details the percent of each group who completed a fully-functional game. For example, 74% of White males completed a fully functional game while only 25% of White females completed a fully-functional game. Similarly, 17% of the Hispanic/Latino males completed a fully-functional game, while 100% of Hispanic/Latina females completed a fully-functional game. In general, White females, and African American and Hispanic males had more issues with their games. For the White female students, the problems were minor coding issues, with two of the three issues being that the turtles/logs did not transport (or carry) the frog, and a problem with collision between the frog and the trucks.

For the male African American and Hispanic/Latino students, the problems in their games were generally more significant, such as missing the second part of the project entirely (no river scene) or problems with missing coding for the generation and absorption of agents.

6.2 Survey Analysis

When looking at the survey results for students who did not submit fully-functional games, there were a few areas where the non-functional game submitters varied from the functional game submitters.

6.2.1 Prior Experience

First, students were asked how many computer classes they had taken prior to this particular class. For both groups (those who submitted a fully-functional game and those whose game was not fully-functional) approximately 15% of the students indicated that this was their first computer class. The remaining students whose game was not fully functional indicated that they had taken one or two prior classes. Typical prior classes

identified were keyboarding and applications classes such as PowerPoint or other Microsoft Office applications. In contrast, 45% of those who submitted a fully-functional game indicated that they had previously taken one or two computer classes prior to taking this class, while another 40% had taken three to four computer classes prior to taking this class. In addition to the common answers of keyboarding and computer application courses, many in this group listed "Applied Technology" as an additional course, which is an engineering oriented class focused on problem solving. Furthermore, 50% of the group submitting a fully-functional game had previously taken a game design class, while only 28% of those who submitted a non-functional game had previously taken a game design class.

6.2.2 Self-Perception

In another survey question, students were asked to respond to the statement, "I am good at solving computer problems," with strongly agree, agree, disagree, or strongly disagree. For those with fully-functional games, 80% overall agreed or strongly agreed with that statement *before they even took the class*, but for those without fully-functional games, only 46% responded in kind. It is important to note that the 20% of those with fully functioning games that did not agree with that statement were all girls.

6.2.3 Future Goals

Students were again separated by their perceptions of wanting to take additional computer classes. When asked (before they began the SGD unit) to respond to the statement, "When I get to high school, I want to take computer classes," 91% of those who submitted a fully-functional game agreed with that statement, while only 69% of those whose game was not fully-functional agreed. The gap was slightly larger when asked whether they wanted to take computer classes in college: 55% of those with a fully-functional game agreed with this statement, but only 31% of the others agreed. This variance between the two groups may be explained in part by their answers to another statement with much disagreement. Students were asked to respond to the statement, "Computer scientists have fun jobs." Of those who submitted fully-functional games, 86% agreed or strongly agreed while only 62% of those without a fully-functional game agreed with this statement.

7. CONCLUSION

While 60% of the students were able to create a fully-functional Frogger game, there were some small within-group variances that still need to be addressed. There are two areas to consider in the above research – first is the extent to which games were non-functioning. For most students, non-functioning games were the result of incorrectly using the TRANSPORT code, which not only enabled agents to move, but also enabled them to carry another agent while they moved. This code was a change from the earlier section of the game (where they had to use MOVE to enable the cars and trucks to move) and is considered a minor coding error. On the other hand, a number of students (particularly African American and Hispanic males) had games with more serious problems, such as not generating or absorbing agents, or missing part of the game entirely.

The survey results may shed some light on why some students struggled with this activity. Students whose projects were not fully functioning had taken fewer computer classes and were less likely to see themselves as good computer problem solvers.

They also were less likely to want to take computer classes either in high school or in college. Finally, they were also less likely to perceive a career in computer science as fun.

This study did find that the Scalable Game curriculum resulted in fully functional games for 60% of the students who took the class, including 67% of the female students and 50% of the students of color. Questions remain as to the extent to which other features about the teacher, the classroom, the lessons, the pedagogy, and the activity may have contributed to those who did and did not have fully functioning games at the end of the day. More research is also needed to determine whether successful completion of projects affects these students' perceptions about computer science and their ability to solve computer problems at the end of the course.

8. ACKNOWLEDGMENTS

This paper is based upon work supported by the National Science Foundation under Grant No. DRL-1312129 and CNS-1138526. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

1. Anderson, N., Lankshear, C., Timms, C., & Courtney, L. (2008). "Because it's boring, irrelevant and I don't like computers: Why high school girls avoid professionally-oriented ICT subjects. *Computers and Education*, 50, 1304–1318. doi:10.1016/j.compedu.2006.12.003
2. Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE '11*, 245. doi:10.1145/1953163.1953241
3. Bock, S. J. (2013). Women and Minorities in Computer Science Majors : Results on, 14(1), 143–152.
4. Brown, A. L., & Campione, J. C. (1994). Guided discovery in a community of learners. In K. McGilly (Ed.), *Classroom lessons: Integrating cognitive theory and classroom practice* (chap. 9) (pp. 229-270). Cambridge, MA: MIT Press.
5. Cook, T. (2014). Apple - Diversity - Inclusion inspires innovation. Retrieved March 4, 2015, from <https://www.apple.com/diversity/>
6. Denner, J., Werner, L., O'Connor, L., & Glassman, J. (2014). Community College Men and Women: A Test of Three Widely Held Beliefs About Who Pursues Computer Science. *Community College Review*, 42, 342–362. doi:10.1177/0091552114535624
7. Diversity – Google. (2014). Retrieved March 4, 2015, from <http://www.google.com/diversity/at-google.html>
8. Ericson, B., & Guzdial, M. (2014). Measuring Demographics and Performance in Computer Science Education at a Nationwide Scale Using AP CS Data. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*, 217–222. doi:10.1145/2538862.2538918
9. Webb, D. C., Repenning, A., & Koh, K. H. (2012). Toward an emergent theory of broadening participation in computer science education. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*, 173–178. doi:10.1145/2157136.2157191