How Can a Social Debugging Game Effectively Teach Computer Programming Concepts?

Michael J. Lee
University of Washington
Mary Gates Hall, Box 352840
Seattle, Washington 98195, USA
mjslee@uw.edu

ABSTRACT

The dramatically changing availability and sharing of information online has created new opportunities for informal, discretionary learning. This, along with the rise in online gaming across all ages and genders, gives rise to questions about how these resources can be used effectively for teaching. In my work, I examine the efficacy of an online debugging game designed to teach computer programming. More specifically, in addition to developing the game and its learning objectives, I investigate how the inclusion of new features and the manipulation of specific elements within the game affects people's motivation and learning of computer programming and debugging concepts.

Categories and Subject Descriptors

K.3.2 Computer Science Education: Introductory Programming, D.2.5 Testing and Debugging.

General Terms

Design, Human Factors.

Keywords

Programming, assessment, engagement, efficiency, debugging, serious game, educational game.

1. RESEARCH SITUATION

I am currently a PhD candidate in the Information School at the University of Washington (Seattle). With a background in cognitive science, human-computer interaction, information science, and having taught over twelve introductory programming courses, I take a multidisciplinary approach to my research. I am in my fourth year, and anticipate that I will be graduating in approximately two years. I advanced to candidacy during my third year by passing my general examination, which included the approval of a research statement, and passing of a written and oral component judged by my supervisory committee. In order to graduate, I will have to deliver a public dissertation proposal, followed by a public dissertation defense. I have several published works based on my research statement, and I am continuing with additional projects while I prepare to deliver my dissertation proposal. My research primarily consists of the development and use of an online debugging game, called "Gidget," that is designed to teach programming and debugging concepts. The ideas, past publications, and research direction described in this paper will constitute the bulk of my dissertation.

2. CONTEXT AND MOTIVATION

The availability of resources on the internet is dramatically changing the way people learn outside of formal learning settings. Discretionary learning is now less about books and more about self-paced, engaging, social experiences using digital resources. Therefore, there is an increasing need to learn more about discretionary learning resources, and how they can be used as effective tools for education. However, computing education has been slow to adapt to these trends, focusing instead on engagement, instruction, or scalability independently, but not on how to combine the three. For example, computer science distance education efforts such as Udacity.com, while scalable and instructive, can be isolating [1] and have high attrition rates in formal educational settings. Similarly, Codecademy.com, an interactive programming tutorial site, is scalable, but its effectiveness remains an open question because of its lack of evidence based instruction and assessment. Constructivist learning technologies such as Alice and Scratch, while engaging, require costly and unsustainable instruction through camps and after-school programs to promote learning [11]. Recent efforts have also effectively used game based computing puzzles [10], but these are often solitary experiences.

3. BACKGROUND & RELATED WORK

I believe that social debugging games can be used to address these limitations by being engaging, instructive, and scalable for the following reasons. Social experiences are a fundamental part of engaging individuals; this is particularly true for females, who are chronically underrepresented in computing [9]. Experiences that involve friends and avoid isolating females (particularly teens) in competitive, male dominated learning settings are far more successful than those that do not [8]. Next, debugging is a fundamental computational thinking skill and necessary for writing a correct program [4]. Our approach will be one of the first learning technologies built specifically to teach debugging explicitly before teaching programming. Finally, games are now a universal form of play for everyone: 91% of U.S. kids aged 2-17 play video games [2]. Social games are particularly popular among teen girls, who play them for an hour or more a day for fun, stress relief, mental workout, and to connect with friends [2]. Games are now also widely thought of as effective instructional tools, because they can provide concrete feedback about success and failure at reaching well-defined goals [3]. However, more research needs to examine how to design effective games for teaching specific educational topics, and how to measure success.

4. THESIS, GOALS, METHODS & STATUS

I will use an online puzzle game designed to be engaging, instructive, and scalable, to answer the question: "*how can we use a social debugging game to effectively and measurably teach programming?*" Teaching programming through debugging, especially in the context of a discretionary game, is a novel approach with unique challenges. To solve the puzzles, the game needs to teach the players skills that mirror those in debugging, including hypothesis formation and testing, mental simulation of programs, and evidence gathering. Moreover, great care needs to be taken to keep people motivated, since learning programming can be discouraging [5], and players can choose to disengage with discretionary environments at any time. In addition to carefully designing an appropriately challenging, entertaining, and educational curriculum and game, social features including pair debugging, achievement broadcasting, and puzzle sharing could entice and engage players via integration with social networking websites such as Facebook.

The goals I am pursuing to answer my research question are to 1) develop an accessible and extendable programming puzzle game, 2) have a clear set of specific learning objectives covering introductory programming and debugging concepts, 3) verify that the game and its contents are engaging, 4) ensure that the players are satisfying the learning objectives, and 5) examine how people are sharing and collaborating within and outside the game. To satisfy these goals, I will conduct large-scale, quantitative controlled experiments online, as well as having smaller, in-person qualitative studies using the game.

I currently have three published papers that address the first four goals outlined above. The first publication reported on the differences between two versions of the game, where one version had a personified game character with a face that used personal pronouns and took the blame for players' mistakes, and the other version had a character without a face that provided more traditional compiler/interpreter feedback. Those playing the former version completed double the levels and were more likely to say they wanted to help the character than those playing the latter version of the game. The second study [6] examined how the purposefulness of the levels' goals, manipulated by the names and images of the objects the game character interacted with, affected people's motivation to play the game. Players were randomly assigned to groups using vertebrate animals (e.g., dogs, cats), invertebrates (e.g., insects), or inanimate objects (e.g., blocks), within the context of cleaning up a chemical spill. Players interacting with vertebrates completed more levels than those in the other groups, with a significant difference between the vertebrate and inanimate object groups. Finally, the most recent study [7] examined how adding assessments to a game affected people's performance and engagement with the game. We found that those who were given assessment levels (i.e., exams) not only completed significantly more levels than those who were not given assessment levels, but also completed the same set of levels faster than their counterparts. These studies indicate that people are engaged playing the game, and demonstrate that they are learning programming concepts by debugging the faulty code.

My next project, related to the fifth goal outlined above, involves testing the game in-person with teams in a high school summer camp, and also releasing the game to the public. In-person projects will provide much richer, qualitative data that we can use to learn more about the effectiveness of our system, and provide insight into building and integrating more social features into the game for public release. Social features will include achievement broadcasting, teamwork, and puzzle sharing in the game, which can sustainably engage individuals (particularly teens) and their friends in social learning of programming concepts. Feedback and discussion from the doctoral consortium will be helpful in strengthening the study design and learning about relevant works from the community.

5. EXPECTED CONTRIBUTIONS

My research findings will continue to benefit the learning science and computing education communities as learning resources become available online. The results of my research and my dissertation will shed new light on how a discretionary programming game with social features can both engage and effectively teach people programming concepts. At the minimum, we will have a thousand people who have played the game throughout the course of my research, which will allow us to develop insights into the best-practices for developing effective learning technologies for computer science education.

6. REFERENCES

[1] Brennan, K., Valverde, A., Prempeh, J., Roque, R., & Chung, M. (2011). More than code: The significance of social interactions in young people's development as interactive media creators, *ED-MEDIA*, 2147-2156.

[2] ESA. (2012). Essential facts about the computer and video game industry. Entertainment Software Association, http://www.theesa.com/facts/pdfs/ ESA_EF_2011.pdf, retrieved May 9th, 2013.

[3] Gee, J.P. (2007). What video games have to teach us about learning and literacy, *Macmillan*.

[4] Ko, A.J., & Myers, B.A. (2005). A framework and methodology for studying the causes of software errors in programming systems, *JVLC*, 16, 1-2, 41-84.

[5] Lee, M.J., & Ko, A.J. (2011). Personifying programming tool feedback improves novice programmers' learning. *ACM ICER*, 109-116.

[6] Lee, M.J., & Ko, A.J. (2012). Investigating the role of purposeful goals on novices' engagement in a programming game. *IEEE VL/HCC*, 163-166.

[7] Lee, M.J., Ko, A.J., & Kwan, I. (2013). In-Game Assessments Increase Novice Programmers' Engagement and Level Completion Speed. *ACM ICER*, *to appear*.

[8] Margolis, J., & Fisher, A. (2003) Unlocking the Clubhouse: Women in Computing, *MIT Press*, *Boston, MA*.

[9] National Center for Women & Information Technology (2012). NCWIT Scorecard: A report on the status of women in information technology. http://www.ncwit.org/pdf/ 2010_Scorecard_Web.pdf, retrieved May 9th, 2013.

[10] Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation, *Computers & Education*, 52(1), 1–12.

[11] Webb, H.C., & Rosson, M.B. (2001). Exploring careers while learning Alice 3D: A summer camp for middle school girls, *ACM SIGCSE*, 377-382.