Gidget: An Engaging Online Game for Learning Introductory Programming

Michael J. Lee
The Information School
University of Washington
Seattle, WA 98195-2840
mjslee@uw.edu

CSCW '14 Workshop on Designing Futures for Peer-to-Peer Learning
February 15, 2014, Baltimore, MD.

Abstract
Online games have the potential to reach a wide audience and teach new skills. I propose to use Gidget, an online debugging game, to teach novices computer programming concepts in an engaging way. Learners must debug faulty programs to progress through the game, which are set up in modules to teach specific computer programming concepts. Once all the levels are complete, learners are given the option to further engage in the game by creating their own levels that can be shared with their friends and family. I would like to incorporate more social and peer-to-peer collaborative capabilities into the game, exploring how these additions may affect engagement and learning within the game.

Author Keywords
Debugging; educational game; engagement

ACM Classification Keywords
K.3.2 Computer Science Education.
D.2.5 Testing and Debugging.

Introduction
As educational resources and materials move from formal settings to discretionary settings, it becomes increasingly important to create engaging and motivational materials to sustain learners' interest.

Unlike formal settings such as classrooms – where learners are typically expected to stay for a given duration and/or motivated by performance measures such as grades – learners who are unmotivated or unsatisfied with materials in a discretionary learning context may disengage from the activity, or even worse, they may lose interest completely and never come back to the activity or subject.

My vision is to use online video games as an easily accessible, motivating, and engaging medium for people all over the world to learn computer programming concepts. Learners' motivation is of critical importance and can have a major impact on their learning [1,3], and games have been shown to positively motivate players in informal learning interventions [2,4,5]. Therefore, I would like to create, test, and use a game designed specifically to teach novices computer programming concepts.

The Online Gidget Debugging Game

To carry out this vision, I have created a fully functional game called Gidget – an HTML5 web application that is playable in a browser (see Figure 1). In this game, the learner must help the game protagonist, Gidget the damaged robot, complete its mission to clean up a chemical spill and save the local animals by fixing its faulty programs. The Gidget language, which is modeled after Python, supports dynamically typed-variables, Boolean operators and expressions, conditionals, loops, mathematical operators, objects, functions, and domain-specific keywords for the game characters to interact with their world.

The game consists of 37 levels, which are divided into 7 distinct modules focused on teaching specific computer programming concepts typically taught in an introductory programming course (e.g., one module covers variable assignment and arithmetic operators, another covers Booleans and conditional statements, and another covers loops). After completing all the levels, learners are given access to a level designer where they can create their own levels using the Gidget language to share with their friends and family members. Content creation through storytelling has been shown to be engaging, especially for children learning programming concepts [6,10,11]. Levels can be shared individually, or shared as a sequence of levels that make up a cohesive story.

Many of the features in the game have been carefully made to maximize learners' engagement. We have run several controlled experiments to determine how changing certain interactions, graphics, or features of the game affect players' engagement. In short, we found that learners were significantly more engaged with the game when:

- A personified character that had a face gave feedback (e.g., instructions and error messages) using personal pronouns. This was in contrast to a terminal screen providing more traditional feedback that did not use personal pronouns [7].
- The goals of the levels were more purposeful. Purposefulness of goals was manipulated by the image and label of the objects that the game protagonist interacted with in the game [8].
- Assessment levels were included at the end of each module (i.e. set of levels) that explicitly asked the learner to answer an exam-like question and explain why they chose their particular answer [9].

Peer-to-Peer Collaboration, Play, and Level Design

I hope to learn more about the game's appeal to a younger audience (all of our prior studies with the game only included participants who were 18 years or older). During public demonstrations at local computer science related events (e.g., the computer science education week), young children and teens appear to be most attractive to the game. Interestingly, most of the players I have observed played the game in pairs, with one person "driving," and the other person giving suggestions and pointing to different elements on the screen. I would like to explore this more thoroughly, looking at how learners pair program/debug collaboratively, and how this may affect their engagement or learning through the game.

In addition, I am planning to launch the game publically sometime in 2014. Including social motivators in the

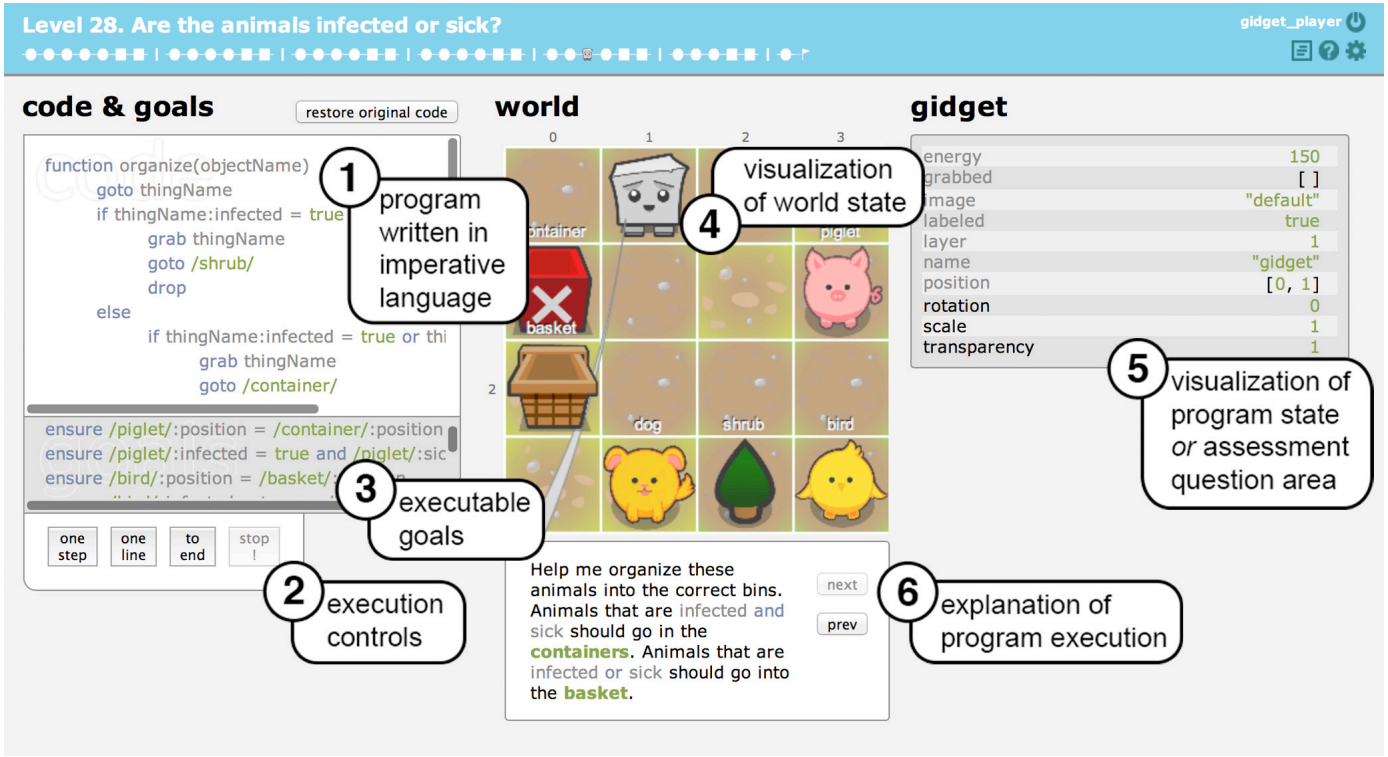

Figure 1: The Gidget game with the main components of the interface highlight in the callouts. The learner helps the robotic character, Gidget, solve its missions by debugging the faulty code it provides. The game consists of 37 levels, including assessment/exam levels, modeled after introductory computer science courses. To aid the user, the game provides help in the form of an on-demand reference guide and on-the-fly syntax error highlighting.

public release – such as the ability to post challenging levels for friends or parents to solve on social networking sites – is another avenue I would like to explore further in the game.

Conclusion

In this paper, I have briefly described my puzzle game, Gidget, which is designed to teach novices computer programming concepts. My preliminary studies have shown the efficacy of the game's design on novice learners' engagement. Next, I would like to learn more about how peers might collaborate solving the game's puzzles and creating their own levels, how they share their creations with others, and how all this affects their engagement with the game and learning activity.

Acknowledgements

This material is based upon work supported by Microsoft and the National Science Foundation (NSF) under Grants CNS 1240786, CCF 0952733, CCF 1339131, and OISE 1210205. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

Biography

Michael J. Lee is a PhD Candidate at the University of Washington Information School. His research areas are in human-computer interaction (HCI) and computing education, where he specifically focuses on examining the factors that effectively engage and teach programming skills to novices using video games. He received his Masters in Information Management and Systems from UC Berkeley, and his B.S in Cognitive Science and HCI from UC San Diego.

References

[1] Armstrong, S., Brown, S., & Thompson, G. (1998). *Motivating Students*. Routledge.

[2] Cliburn, D.C. (2006). The effectiveness of games as assignments in an introductory programming course. Frontiers in Education Conference, 6-10.

[3] Farthing, D.W. (1997). The three Ms of open learning: Medium, Material and Motivation. Conference on the Teaching of Computing.

[4] Garris, R., Ahlers, R., & Driskell, J.E. (2002). Games, motivation, and learning: A research and practice model. Simulation & gaming 33(4), 441–467.

[5] Gee, J.P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1).

[6] Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling Alice motivates middle school girls to learn computer programming. ACM CHI, 1455-1464.

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

[8] 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.

[9] Lee, M.J., Ko, A.J., & Kwan, I. (2013). In-game assessments increase novice programmers' engagement and level completion speed. ACM ICER, 153-160.

[10] Ryokai, K., Lee, M.J., and Breitbart, J.M. (2009). Children's Storytelling and Programming With Robotic Characters. ACM C&C, 19-28.

[11] Ryokai, K., Lee, M.J., and Breitbart, J.M. (2009). Multimodal Programming Environment for Kids: A "Thought Bubble" Interface for the Pleo Robotic Character. ACM CHI, 4483-4488.