A Demonstration of Gidget, A Debugging Game for Computing Education

Michael J. Lee & Andrew J. Ko
The Information School | DUB Group
University of Washington
Seattle, Washington, USA
{mjslee, ajko}@uw.edu

***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. Over 800 people have played the game online as part of several research studies and it will be released freely to the public in the near future.**

I. 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.

Online video games provide an easily accessible, motivating, and engaging opportunity for people all over the world to learn computer programming concepts at their own pace and discretion. Learners' motivation is of critical importance and can have a major impact on their learning [1,4], and games have been shown to positively motivate players in informal learning interventions [3,5,6]. Our work uses an online debugging game called Gidget (shown in Figure 2), which has been shown to engage novices learning programming concepts [8,9,10].

II. The Game: Gidget

Gidget is a web application that is playable in a browser and was created using seven specific design principles [7]. The game is motivated by a story: there has been a chemical spill from a factory and Gidget, a small robot capable of identifying and solving problems with programs, has been deployed to clean up the area (see Figure 1). Unfortunately, Gidget was damaged in transit, and is only able to provide code that partially, but not completely solves each level's goals. It is the player's job to help the robot through missions by diagnosing and fixing the problems in each level's code, then executing the code, so the robot can fulfill the executable mission goals.

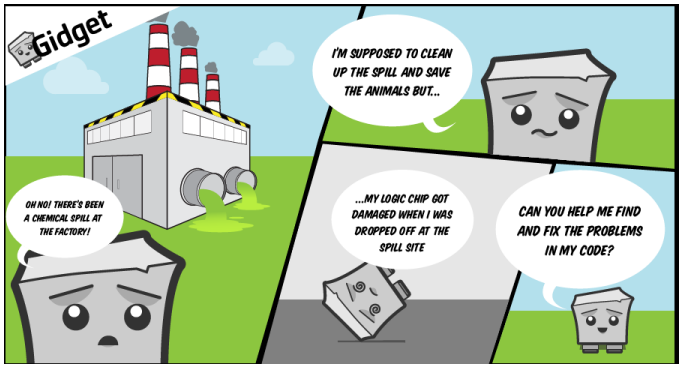

Figure 1. In the game, Gidget the robot is damaged on the way to clean up a chemical spill, so the player must help by fixing faulty code.

The game uses an imperative, Python-like programming language designed specifically for the game. The language supports dynamically typed-variables, Boolean operators and expressions, conditionals, mathematical operators, objects, functions, and domain-specific keywords for the game characters to interact with their world. These interactions primarily include finding things in the world, going to them, checking their properties, and carrying them to other places in the world. In some cases, objects have their own abilities, which Gidget can call as functions. After each execution step, the effect of these commands are shown in the 'program state' panel and explained by Gidget to reinforce the semantics of each command. Each step costs Gidget 1 unit of 'energy', which forces players to consider how to write efficient programs that can be solved using the allocated energy.

To aid the players with debugging while playing the game (see Figure 2), the game offers four execution controls: *one step, one line, to end*, and *stop*. The *one step* button evaluates one compiled instruction, displaying text explaining the execution of the step. The *one line* button evaluates all steps on one line of the code, just as a breakpoint debugger does, jumping to the final output of that line. The *to end* button evaluates the entire program and the goals, animating each step in quick succession. The *stop* button allows the player to halt the program and edit code during any part of the execution. When the learner uses *one step* or *one line*, Gidget provides a detailed explanation of the execution of each statement in the program, highlighting changes in the runtime environment. This serves as the game's primary instructional content, explicitly teaching the language syntax and semantics.

978-1-4799-4035-6/14/$31.00 ©2014 IEEE

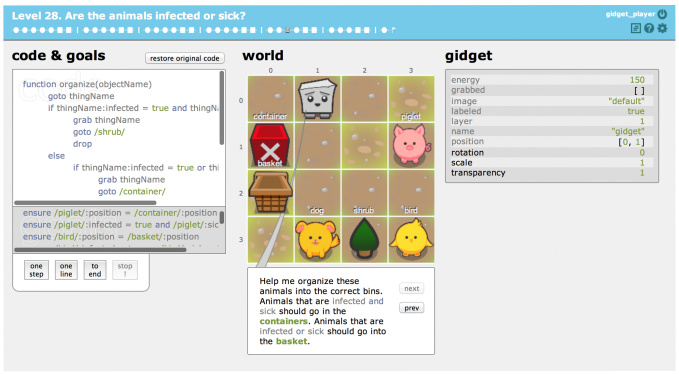

Figure 2. The Gidget game, shown here in its puzzle game mode, where learners have to debug Gidget's faulty code to complete mission goals.

To help learners start playing Gidget, the game presents a ten-slide tutorial to every player upon game start. The game also features an in-game reference guide, providing explanations and examples of each command in the language, along with information about programming concepts such as variables, functions, the stack, and loops. The reference guide is available as a standalone help guide or as tooltips that appeared when hovering over tokens in the code editor. Finally, the game's code editor provides keystroke-level feedback about syntax and semantics errors, highlighting erroneous code in red and explaining the problem in Gidget's speech bubble.

III. VL/HCC Community

The VL/HCC community has been strong advocates for making programming easier to understand and evaluating new ways of teaching programming concepts effectively. Several controlled experiments with online users have shown that is is possible to translate debugging into engaging puzzle game mechanics that is appealing to a broad demographic [8,9,10]. More than 800 people between the ages of 18 and 66 years and of various genders, ethnicities, income-level, and education were recruited online and played the game. In addition, we had a total of 44 teens (between the ages of 13-18 years) play the game in a lab study and two summer camps [7].

Figure 3. The Gidget game, shown here in its level design mode, where players can create their own levels and share them with other players.

Our work has demonstrated that novices can be highly engaged in learning programming concepts through a debugging game [8,9,10], that their initially negative attitudes towards programming can be changed [2], and that they can create their own programs after playing through the puzzles [7]. Across our studies, we have found that novices playing through our online game struggle largely with the same programming concepts that others have difficulties with in classroom settings [7,8]. However, we also observed that these novices were able to create novel, complex programs (see Figure 3) on their own by the time they completed the game [7].

We plan to release the game freely to the public in the near future. We would benefit from demoing the game during the VL/HCC showpiece presentation session by allowing a community of experts to play through the game and give us feedback to improve it before its release.

IV. Showpiece Presentation

Gidget will be showcased as a fully-playable game during the conference. We will provide at least two laptops during the showpiece session where attendees can play through the game normally in sequence, or via a freeplay-mode that will allow them to skip to any level and have immediate access to the level designer.

V. Acknowledgements

This work was supported in part by the National Science Foundation (NSF) under Grants CNS- 1240786, CNS-1240957, CNS-1339131, CCF-0952733, CCF- 1339131, IIS-1314356, IIS-1314384, and OISE-1210205. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of NSF.

References

1. S. Armstrong, S. Brown, & G. Thompson, "Motivating Students," Routledge. 1998.
2. P. Charters, M.J. Lee, A.J. Ko, D. Loksa, "Challenging stereotypes and changing attitudes: the effect of a brief programming encounter on adults' attitudes toward programming," ACM SIGCSE 2014, 653-658.
3. D.C. Cliburn, "The effectiveness of games as assignments in an introductory programming course,"Frontiers in Education Conference, 6-10, 2006.
4. D.W. Farthing, "The three Ms of open learning: Medium, Material and Motivation," Conference on the Teaching of Computing 1997.
5. R. Garris, R. Ahlers, & J.E. Driskell, "Games, motivation, and learning: A research and practice model," Simulation & gaming 33(4), 441–467, 2002.
6. J.P. Gee, "What video games have to teach us about learning and literacy," Computers in Entertainment, 1(1), 2003.
7. M.J. Lee, F. Bahmani, I. Kwan, J. Laferte, P. Charters, A. Horvath, F. Luor, J. Cao, C. Law, M. Beswetherick, S. Long, M. Burnett, A.J. Ko, "Principles of a Debugging-First Puzzle Game for Computing Education," IEEE VL/HCC 2014.
8. M.J. Lee, A.J. Ko, "Personifying programming tool feedback improves novice programmers' learning," ACM ICER 2011, 109-116.
9. M.J. Lee, A.J. Ko, "Investigating the role of purposeful goals on novices' engagement in a programming game," IEEE VL/HCC 2012.
10. M.J. Lee, A.J. Ko, I. Kwan, "In-game assessments increase novice programmers' engagement and level completion speed," ACM ICER 2013, 153-160.