

10-2013

Adaptive Gameplay for Programming Practice

Chris BOESCH

Singapore Management University, cboesch@smu.edu.sg

Sandra BOESCH

Pivotal Expert Pte Ltd, sandracboesch@gmail.com

DOI: https://doi.org/10.5176/2251-2195_CSEIT13.22

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Education Commons](#), and the [Software Engineering Commons](#)

Citation

BOESCH, Chris and BOESCH, Sandra. Adaptive Gameplay for Programming Practice. (2013). *4th Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2013): Proceedings: October 28-29, Phuket, Thailand*. 36-38. Research Collection School Of Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/2045

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Adaptive Gameplay for Programming Practice

Chris Boesch & Sandra Boesch

School of Information Systems (SIS) Singapore
 Management University (SMU) Singapore
 and Pivotal Expert, Pte. Ltd. Singapore

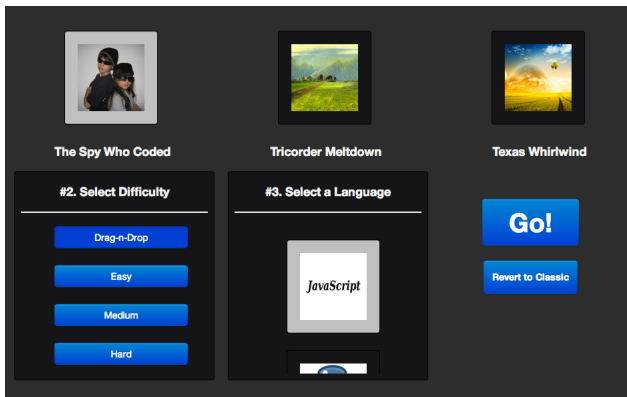
Abstract— Over the past four years, we have collaborated to develop a set of online games to enable users to practice software languages in a self-directed manner and as part of a class. Recently we introduced a new adaptive difficulty feature that enables players to self-regulate the difficulty of the games they are playing to practice. These new features also provide additional information to further adapt the problem content to better meet the needs of the users.

Keywords-education, programming, game-based learning

I. INTRODUCTION

When setting out to develop a more effective method to teach basic computer science, the authors were looking for innovative ways to provide additional, individualized feedback to students learning software languages such as Python, JavaScript, and Java for undergrad university courses. The authors took the approach to enable students to practice software languages on their own by having them solve short programming problems in an online game (see Figure 1) in a variety of software languages.

Figure 1. SingPath Difficulty Selection Screen



Students were able to practice solving these problems on their own time, from their own systems, wherever they had Internet access. This method enabled the authors to provide additional feedback to students in a more real-time manner than had been previously possible with live, in-class quizzes [1] and weekly problem sets turned in as homework. Students were still assigned problems to solve as in previous terms, but by requiring students to solve all problems in an online system, the authors were able to provide students with real-time

feedback on their progress and at the same automatically track which students were on pace to solve all required problem prior to weekly deadlines.

As more people around the world began to solve problems on the online system, two consistent categories of feedback were received. A portion of the users continued to make comments such as “These problems are too difficult” while other users would provide comments such as “I am bored. Please let me skip the easy problems.” To address this feedback, the authors created an adaptive difficulty mode. This enabled users to adjust the difficulty of problems to reduce the amount of boredom or frustration users might be encountering. The authors hypothesize that by enabling users to adjust the difficulty of the games they are playing, the users will be able to better balance their own boredom and frustration and increase the likelihood that they will be able to enter in to and stay in a state of learning flow [2] while practicing. Four difficulty modes were introduced: Easy, Medium, Hard, and Drag-n-drop. When users play on easy mode, they are provided step-by-step problems suitable for beginners. These problems often contain skeleton code to prompt the users and point them in the right direction for developing solutions. The hard setting presents the user with the hardest problems that have been loaded into the system for a given level. The relative difficulty of problems is determined by keeping track of how many attempts and how much time it takes all players to solve the problems. This enables the development of a ranking of relative problem difficulty. The medium difficulty level was designed to be adaptive to each user’s individual skill level. This was accomplished by attempting to forecast how much time and how many attempts a user would require to solve problems in a given level. The problems that were considered to be easy or difficult were excluded. The remaining problems where considered medium difficulty problems. The initial settings for easy problems, based on past problem solving results, were set to be one or two attempts and less than sixty seconds. This meant that any problem that a user could solve with only one or two attempts within less than sixty seconds was considered an easy problem for the player. Similarly, hard problems were set to be any problems that required more than five attempts or more than five minutes to solve. This left the range for medium problems to be any problems that a user was able to solve in three to five attempts and in between one and five minutes. When there are insufficient problems for the user available that fall within the difficulty range the user is playing in, the remaining problems are selected from the next problem difficulty level(s). When playing in easy, if there are no

ACKNOWLEDGMENTS

We are grateful to the Singapore Management University School of Information Systems for enabling us to conduct tournaments in live classroom settings. We are also grateful to the staff of Pivotal Expert for maintaining and enhancing the SingPath platform and for making it free to students and faculty around the world.

REFERENCES

- [1] Boesch, Chris; Boesch, Sandra. Tournament-based Teaching. 4th Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2012).
- [2] Csikszentmihalyi, M. *Finding flow: The psychology of engagement with everyday life*. 1st edition. Basic Books: New York, NY;1997.
- [3] Michaelsen LK, Knight AB, Fink LD. *Team-Based Learning, A Transformative Use of Small Groups in College Teaching*. 1st edition. Sterling, Virginia: Stylus Publishing, LLC; 2002.
- [4] Boesch, Chris; Steppe, Kevin. *Case Study on Using a Programming Practice Tool for Evaluating University Applicants*. 3rd Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2011).