Making Serious Programming Games Adaptive

Michael A. Miljanovic and Jeremy S. Bradbury

University of Ontario Institute of Technology
2000 Simcoe St. N., Oshawa, ON L1H 7K4, Canada
{michael.miljanovic,jeremy.bradbury}@uoit.ca

Abstract. One of the challenges with Computer Science serious games is ensuring they are suitable for learners of different levels of ability and knowledge. To address this challenge, we propose a new methodology for incorporating adaptive gameplay and content into existing non-adaptive serious programming games. Our methodology includes four phases: (1) Identifying an existing game that is suitable for adaptation; (2) Modeling the gameplay tasks and the in-game assessment of learning; (3) Building the adaptation into the existing code base; (4) Evaluating the new adaptive serious game in comparison to the original game with respect to learning and engagement.

Keywords: Computer Science · Education · Serious Games · Adaptive Methods · Machine Learning · Software Evolution.

1 Introduction

The use of serious games is one approach that has shown effectiveness in engaging students to learn a variety of skills [11]. The potential for serious games to increase motivation and engagement among learners is particularly important for the field of Computer Science (CS), where low engagement levels give cause for concern [4]. Furthermore, the widespread interest in understanding the fundamentals of programming has led to CS being a heavily targeted field of study for serious games researchers [5].

While serious games have considerable promise, challenges still exist with respect to their design and evaluation. One of the open challenges is customizing serious games to suit learners of different levels of ability and knowledge. Existing solutions to this challenge can require substantial human effort, such as the monitoring and customization of gameplay by human experts and the creation of large, diverse problem sets. One drawback of these approaches is that they are not always practical when working with increasingly complex serious games [10]. In this work, we consider the use of adaptive serious games to make serious games suitable for learners with different skills. Adaptive serious games do not have the same drawbacks as the above mentioned approaches as they can automatically modify game elements and content to directly impact learner performance [9].

The main contribution of our paper is a new methodology for incorporating adaptive gameplay and content into existing non-adaptive serious games. We

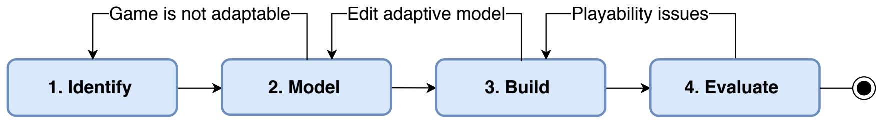

Fig. 1. An Overview of our Methodology for Making Adaptive Serious Games.

have chosen to focus our methodology on CS serious games because many of the existing serious games for learning programming have widespread adoption and have empirical research to support their educational value (e.g., Code Hunt [3]). We believe modifying existing serious games that have already been adopted and evaluated is a more desirable approach than building new games from scratch.

2 Methodology

Adaptive games can be fully or semi-autonomous, allowing for game content to be included after the game's release. An autonomous serious game can promote instructive gameplay, manage the challenge of the user experience, provide scaffolding where needed, and support learners [6]. A common approach to adaptive serious games is to use Competence-based Knowledge Space Theory (CbKST) [2] in combination with a probabilistic approach. Our methodology leverages CbKST for making adaptive serious programming games from non-adaptive games and includes four key phases: identification, modelling, building and evaluation (see Fig. 1). To assist in explaining our methodology, we use the example of creating an adaptive version of Gidget [7]. Gidget is a non-adaptive serious game where players complete missions by repairing faulty programs.

2.1 Identifying a Potential Adaptive Game

Both technical and learning factors should be considered when deciding if an existing serious game is an appropriate candidate for adaptive methods.

Technical Factors. In order to adapt a game, the source code will need to be publicly available and extendable. Thus, it is necessary to ensure that for third-party games, the software license for the chosen game allows for modification. The quality and robustness of the source code should be examined when identifying a serious game for adaptation as both of these factors can impact the modification of the source code. Also the playability of the game and the existence of playability studies should be considered. Gidget is an ideal choice technically because it is open source, includes documented source code, and has been evaluated indirectly with respect to playability.

Learning Factors. First, adapting the learning content of a game requires a clear understanding of the required knowledge, topics, and learning outcomes that are present in the original game [1].Second, making informed decisions about adapting a game requires detailed knowledge about the learners who will play

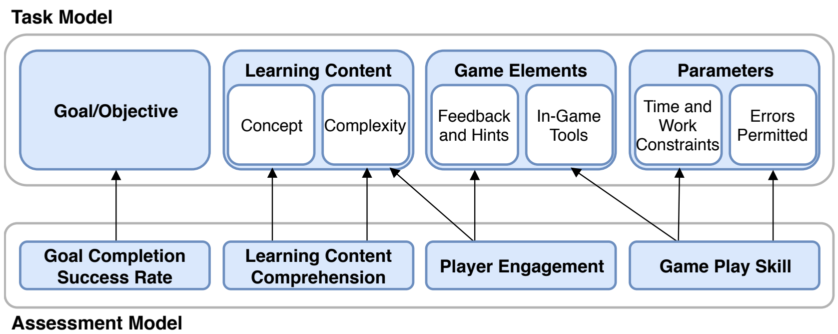

Fig. 2. Task and Assessment Models for Adaptive Games.

the game. Learners of various demographics including age groups may respond differently to in-game adaptations. Additionally, knowledge about the level of programming experience of the game's intended audience is needed in order to make good decisions on how to adjust learning content.Special consideration should also be given to adapting for learners of diverse educational backgrounds outside of CS and choosing games that are inclusive. Third, in order to properly evaluate the new adaptive serious game at the end of the process, it is best to choose an existing game that has already been evaluated with respect to learning as the existing evaluation can serve as a baseline in assessing the adaptive version. Our example, Gidget, focuses on learning debugging and has a target audience of general learners with no previous programming experience. Gidget has also been previously evaluated with respect to learning [8].

2.2 Modelling the Gameplay Tasks and Learning Assessment

Before implementing adaption into a serious game it is necessary to understand and model the gameplay tasks as well as the learning assessment (see Fig. 2).

Task Model. A typical serious programming game includes a sequence of increasingly difficult tasks that pertain to learning content. Often, serious games are designed such that a player's success in the game is dependent on the completed and failed task objectives. Although the criteria for determining whether an objective is failed varies from game to game, failure is often accompanied by feedback or hints, as well as a reset of parameters such as time or error limits. The existing tasks in the game can be modelled and used as a template for adaptation. The most important task properties that should be included in this model are objectives, learning content, game elements, and parameters. These properties can be extracted from documentation as well as the structure and content of the game's source code. In Gidget, each level is a task with one or more objectives, each of which is completed when a given physical object on a grid is moved to a specified location. The primary learning concept in Gidget is debugging, and each level presents increasingly complex objectives, with

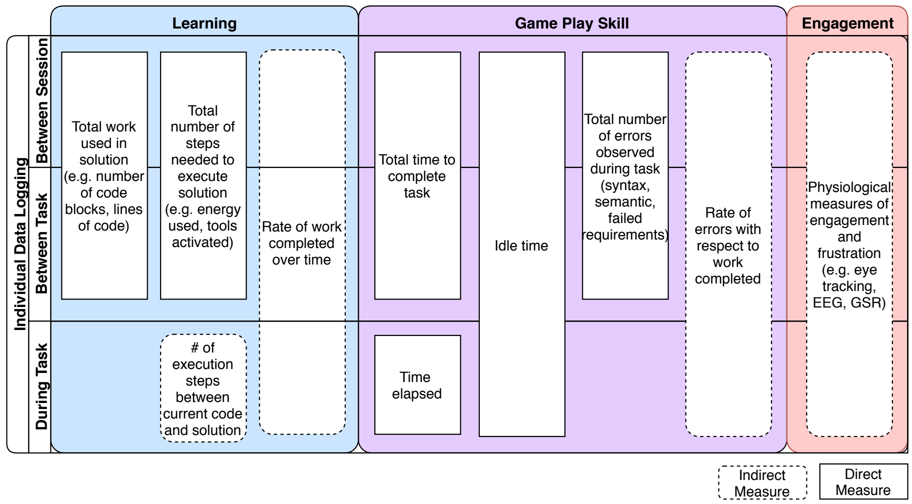

Fig. 3. Data Logging for Adaptation.

partially incorrect code for completing those objectives. In addition to the debugging levels, newer versions of Gidget include levels that introduce concepts such as conditionals, functions, and arrays. Gidget provides substantial feedback to the player by visualizing every step of the code on the grid, and allows players to choose the number of steps to process at a given time. In order to encourage efficient programs, Gidget has an 'energy' limit that restricts the number of moves that can be taken during a level, but does not limit the gameplay time or number of errors permitted.

Assessment Model. Our model of assessment is based on CbKST and a probabilistic evaluation of the learner's competence in the learning content. The use of CbKST necessitates the inclusion of goal completion success rate and learning content comprehension in our model as predictors of a learner's competence. Since Gidget allows players to repeat a task until it is correctly solved, players must be assessed based on the efficiency of their code solutions. This includes measuring the error rate in each level, the number of lines of code in each solution, and how much energy is expended per level. The model also needs to consider player engagement and how it is assessed. Maintaining player engagement in serious games is often achieved by varying the complexity of the learning content to challenge skilled learners or to aid learners who are frequently experiencing difficulty. Finally, we include gameplay skill assessment in our model as it is important to distinguish between skilled video game players and players with high competence of learning content. Gameplay skill assessment may be useful in determining if a player's in-game behavior is related to learning content competence, or due to issues with the game's mechanics. Gidget does not include many features related to gameplay skill assessment (e.g., time limit, score tabulation).

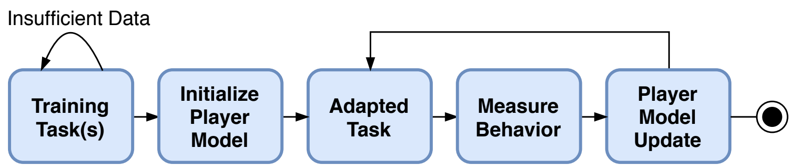

Fig. 4. Adaptive Gameplay Sequence.

2.3 Building Adaptation into the Existing Code Base

This phase includes using the models to plan the adaption approach, logging player behavior, initializing the gameplay, and applying the adaptation strategy.

Plan Adaptation. The task and assessment models should be used to determine which game features to adapt. Once these features are chosen, an adaptive algorithm is chosen to determine when, what, and how the tasks are adapted. Example algorithms may be rules-based approach or use machine learning, but should ultimately be probabilistic and follow the principles of CbKST. In Gidget, features for adaptation include the starting code errors, the gameplay obstacles and the energy limits. The adaptive algorithm in Gidget could involve the creation of a set of rules that use the past performance of the player to determine whether or not to adjust the features.

Data Logging. Learner-specific adaptation requires constant logging and measurement of learning data, game skill data, and engagement data (see Fig. 3). Depending on the adaptation strategy, data may be gathered for assessment during a task, between a task, or between gameplay sessions. In addition to adaptation, the data logged can be used for evaluating the gameplay experience.

Initialize Gameplay. An initial sequence of the game's tasks should be designated as non-adaptive 'training tasks' in order to initially assess the learner. Following training, an adaptive game should customize each task in accordance with the individual learner's data (see Fig. 4). There are several different options that a developer might consider for initializing the training portion of the game: predefined common initialization for all players, self assessment of programming skill (e.g., expert, skilled, unskilled), or game difficulty (hard, medium, easy). As Gidget is targeted towards players with no programming experience, we chose to use a common initialization of gameplay for all players.

2.4 Evaluating the New Adaptive Game

One of the challenges with serious game development in general is the need for accurate and reliable evaluation. One benefit to evaluating adaptive versions of existing serious games is that many of the games have existing evaluation studies that can be replicated and reproduced for the adaptive versions. This allows us to evaluate the benefits of the adaptation by comparing the study results for the original and adaptive versions. If the original serious game did not have a previous evaluation, we recommend following best practices, which may include questionnaires, skill tests, interviews, and controlled experiments.

3 Conclusions & Future Work

There has already been considerable investment in the development and adoption of CS serious games. As best practices for the development of new serious games evolve, it is important that we establish practices to evolve legacy serious games to leverage new ideas and methods. With this goal in mind, we have proposed a methodology for making serious games incorporate learner-based adaptation[1]. Our approach is based on the premise that an adaptive serious game will provide a better experience for learners, and improve their achievement of learning outcomes by directly adapting to their needs. The use of automatic adaptation within a serious game can provide benefits for engagement by adjusting gameplay difficulty to the learner's abilities. We are currently applying our methodology to create and release an adaptive version of Gidget.

References

1. ACM/IEEE-CS Joint Task Force on Computing Curricula: Computer science curricula 2013. Tech. rep., ACM Press and IEEE Computer Society Press (Dec 2013)
2. Albert, D., Lukas, J.: Knowledge spaces: theories, empirical research, and applications. Psychology Press (1999)
3. Bishop, J., Horspool, R.N., Xie, T., Tillmann, N., de Halleux, J.: Code Hunt: experience with coding contests at scale. In: Proc. of ICSE'15 - Volume 2. pp. 398–407 (2015)
4. Butler, M., Sinclair, J., Morgan, M., Kalvala, S.: Comparing international indicators of student engagement for computer science. In: Proc. of the Australasian Computer Science Week Multiconference. pp. 6:1–6:10 (2016)
5. Combéfis, S., Beresnevičius, G., Dagien, V.: Learning programming through games and contests: overview, characterisation and discussion. Olympiads in Informatics **10**(1), 39–60 (2016)
6. Johnson, W.L., Vilhjálmsson, H.H., Marsella, S.: Serious games for language learning: How much game, how much AI? In: Artificial Intelligence in Education. vol. 125, pp. 306–313 (2005)
7. Lee, M.J., Ko, A.J.: Investigating the role of purposeful goals on novices' engagement in a programming game. In: Proc. of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2012). pp. 163–166 (2012)
8. Lee, M.J., Ko, A.J.: Comparing the effectiveness of online learning approaches on cs1 learning outcomes. In: Proc. of the 11th Int. Conf. on Computing Education Research (ICER'15). pp. 237–246 (2015)
9. Reichart, B., Ismailovic, D., Pagano, D., Brügge, B.: Adaptive serious games. Computer Games and Software Engineering **9**, 133–149 (2015)
10. Westra, J., Dignum, F., Dignum, V.: Scalable adaptive serious games using agent organizations. In: Proc. of the 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3. pp. 1291–1292 (2011)
11. Yue, W.S., Wan, W.L.: The effectiveness of digital game for introductory programming concepts. In: Proc. of 10th Int. Conf. for Internet Tech. and Secured Transactions (ICITST 2015). pp. 421–425 (2015)

[1] This research was partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC).