Optimizing Adaptivity in Educational Games

Erik Andersen
Center for Game Science
Department of Computer Science & Engineering, University of Washington
eland@cs.washington.edu

ABSTRACT

One of the most promising ways that games for learning can improve education is by adapting to each child individually. However, it is often difficult to instrument game mechanics so that they can be controlled to promote learning. Furthermore, even if this parameterization is possible, there is little knowledge of how to generate adaptive level progressions that optimize engagement and learning. We have taken the first step towards enabling adaptivity in an educational game for teaching fractions through the automatic generation of levels in a way that allows for multiple axes of mathematical and spatial difficulty to be controlled independently. We propose to expand on this work by developing a framework for representing conceptual knowledge. This framework will keep track of each player's knowledge, generate game levels that are tailored to the player's knowledge and skill level, and create progressions of these levels that allow players to learn new concepts through experimentation. We will compare multiple adaptive concept sequencing algorithms by evaluating their effects on player learning and engagement through multivariate tests with tens of thousands of players.

Categories and Subject Descriptors

K.8.0 [**Personal Computing**]: General – Games

Keywords

games for learning, procedural content generation, adaptivity, multivariate testing

1. INTRODUCTION

We believe that adaptivity is one of the most promising ways that games for learning can impact education. James Gee [9] has argued that learning in games occurs when problems are well-ordered and players feel like they are at the edge of their "region of competence". According to Csikszentmihalyi's theory of flow [6], players reach maximal enjoyment when a game's challenge is matched to their skill level.

FDG 2012, May 29-June 1, 2012, Raleigh, NC, USA.

Therefore, we believe that educational games will reach their maximum effectiveness when they tailor the challenge to the player's knowledge and skill.

However, making an educational game adapt to the player in a meaningful way presents several challenges. One key difficulty is determining how to map abstract educational concepts to concrete instantiations of game content. Another challenge is how to parameterize the game so that multiple axes of difficulty can be controlled independently. Since there are likely many possible combinations of learned and unlearned concepts for a complex game, another challenge is how to generate a level automatically for any possible point in this parameter space. Furthermore, it is difficult to know what to do when the player succeeds or fails and how to generate adaptive progressions that maximize learning.

Ideally, we believe that educational games should have an internal model of the concepts that players must learn and track which concepts the player knows and does not know. The game should be able to map each target concept to specific instantiations and parameterizations of game content, and should create custom-tailored challenges for any possible combination of learned and unlearned concepts. Finally, the game should be able to create sequences of such challenges, or progressions, that move the player towards mastery of all targeted concepts.

The goals of this work are to use a hierarchical concept map to drive the automatic generation of levels to teach those concepts, explore multiple potential algorithms for progressing through this map, and evaluate these algorithms by conducting multivariate experiments with tens of thousands of players.

Our work so far has focused around Refraction[1], a game we developed for teaching fractions. Multiple levels of Refraction can be seen in Figure 1. Each level of the game is played on a grid, and consists of laser sources, target spaceships, and asteroids. Each spaceship desires a target fraction of the laser, indicated by the yellow number on each spaceship. The player can satisfy the targets by manipulating pieces that change the direction of the laser and splitters that split the laser equally into two or three parts. The player must satisfy all of the spaceships at the same time with the correct amounts to complete the level.

[1] `http://www.kongregate.com/games/GameScience/refraction`

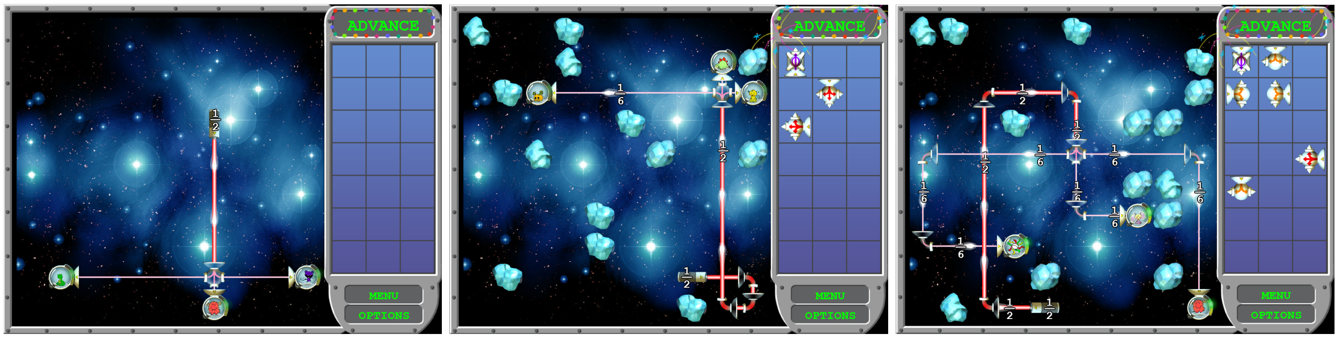

Figure 1: These three levels of Refraction were automatically generated from the same mathematical expression, $(1/2)/3 = 1/6$, with varying levels of spatial difficulty. The levels increase in difficulty from left to right. Our ability to parameterize difficulty in this way allows us to engage and retain players with a wide range of skill levels.

2. RELATED WORK

We are aware of a commercial educational game, Dreambox Learning [8], that features minigames that can be parameterized for difficulty and can adapt to the player. Peirce et al. proposed a framework for adaptivity in educational games and studied it in ELEKTRA [12]. The SIREN project used procedural content generation to adaptively generate scenarios for conflict resolution [10]. Bellotti et al. used experience management to sequence tasks in order to maximize learning objectives while trying to match a specified learning curve [4]. We build off of this work by using a hierarchical concept map to drive the generation of highly parameterized levels for a complex game and experimentally comparing potential approaches for generating adaptive level progressions.

3. ADAPTIVITY THROUGH PROCEDURAL CONTENT GENERATION

Adaptivity is common in intelligent tutoring systems. Intelligent tutors such as the ACT Programming Tutor [5] and the PUMP Algebra Tutor [11] have large lists of production rules that represent procedural knowledge. Each of these production rules represents a transformation of the problem state in order to accomplish some goal. For example, a production rule for an algebra tutor might say "for an equation of the form $x+a = b$, in order to get x on a side by itself, subtract a from both sides." Intelligent tutors can adapt to each student individually because the system always knows what the student should do in each situation and can respond appropriately to any action taken by the student. Since there are only a few conceptually different actions that the student can take when solving a typical algebra problem, authoring all of these production rules by hand is tractable. However, making games adaptive is much harder because the mapping between concepts and game mechanics is often less clear and the state space is likely much larger.

Procedural content generation is of particular interest to game designers because it can greatly expand a game's content beyond what designers can create by hand. However, one of the key difficulties of random level generation is to balance global objectives for what the player is supposed to do in a level with local constraints that are enforced by the game mechanics. In order to provide a conceptual framework for reconciling high-level goals with local constraints, Dormans [7] recently divided content generation into two distinct conceptual components: a *mission* and a *space*. He defined a *mission* as a logical ordering of the goals the player must accomplish to complete the level. A *space* is the actual physical layout of the level, built to constrain the player's actions so that he or she must accomplish the goals specified in the mission.

Building on this idea, we have been able to map mathematical expressions directly into Refraction levels by converting these expressions into an *educational mission*. This mission determines the required properties that any solution to the level must have. After creating the educational mission, we embed it into the *space*, a 10x10 Refraction grid, using answer-set programming [13, 14].

Our approach allows us to modify mathematical difficulty and game difficulty semi-independently. Figure 1 shows three levels of Refraction that each force the player to use the available pieces to solve the equation $(1/2)/3 = 1/6$ but vary widely in difficulty. This leads to several key advantages. As students learn at different speeds and some may require more practice than others, we can quickly generate a wide variety of levels that reinforce a particular concept. Furthermore, since students may vary widely in video game skill and interest, we can take a particular expression and generate easy levels for some students and harder levels for those who want a challenge. However, in order to facilitate learning of a whole list of target concepts, we still need a system for evaluating progress and generating progressions of levels.

4. CONCEPT MAP PROGRESSIONS

We plan to represent player knowledge with a hierarchical concept map and have already implemented a basic version of one for Refraction, shown in Figure 2. Each node in the concept map represents a particular target skill or concept that the player must learn before moving on. The level generator can combine multiple concepts together to create a level that cannot be completed unless the player demonstrates knowledge of each of those concepts.

There are many possible ways to lead a player through the concept map, which we can parameterize in terms of ag-

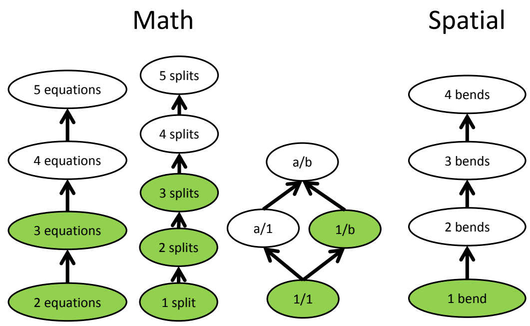

Figure 2: Example of a concept map for Refraction. Each node represents a particular skill that the player must demonstrate. The graph is hierarchical and the player must demonstrate knowledge of simpler concepts before moving on to more advanced generalizations of those concepts. The nodes marked in green indicate concepts that the game believes that the player has mastered. This concept graph features three concept progressions related to mathematical difficulty and one related to game difficulty.

gressiveness, thoroughness, and forgiveness. Aggressiveness represents how quickly the concept map advances when the player successfully completes a level and how many concepts the progression generator tries to evaluate at the same time. Thoroughness represents the number of times that the progression evaluates a particular concept before assuming that the player knows it and moving on. If the player starts failing levels, the forgiveness parameter controls the degree to which the progression "backs off" and becomes easier in order to reevaluate the player's skill level.

Since we cannot know which algorithm for generating progressions is the best without trying them, we will generate multiple potential concept progressions, test several of these progressions on players, and evaluate their effect on learning and engagement. We have demonstrated our ability to conduct multivariate testing to optimize player engagement and retention in three games. These experiments used telemetry to collect data from over 110,000 players and showed that gameplay affects average play time more than animations and audio [2], that secondary game objectives cause many players to quit prematurely and are most effective when they support the primary objectives [1], and that text tutorials are only effective for complex and unconventional games [3]. We believe that the same techniques will help us find the best ways to adapt to the player.

We will find the best progressions by evaluating their performance though a variety of metrics. We want to find the progressions that lead to the greatest amount of learning for the largest number of players. We will measure engagement by examining how far players get in the game, how long players are willing to play the game, and how many players return to play again at a later date. We will measure learning in two key ways. First, we will use the traditional method of giving players a test before and after they play the game. We will also measure in-game learning by examining player actions and determining whether players learn over time to make fewer wrong moves and more correct moves. Since the player must understand the game mechanics in order to succeed in the game, the ability to complete levels is perhaps the best test of whether the player has learned those mechanics. Hopefully, these experiments will help us optimize adaptivity in our games, and our techniques will be applicable to a wide range of educational games.

5. REFERENCES

[1] E. Andersen, Y.-E. Liu, R. Snider, R. Szeto, S. Cooper, and Z. Popović. On the harmfulness of secondary game objectives. In *FDG '11: Proceedings of the Sixth International Conference on the Foundations of Digital Games*, New York, NY, USA, 2011. ACM.

[2] E. Andersen, Y.-E. Liu, R. Snider, R. Szeto, and Z. Popović. Placing a value on aesthetics in online casual games. In *CHI '11: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2011. ACM.

[3] E. Andersen, E. O'Rourke, Y.-E. Liu, R. Snider, J. Lowdermilk, D. Truong, S. Cooper, and Z. Popović. The impact of tutorials on games of varying complexity. In *CHI '12: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2012. ACM.

[4] F. Bellotti, R. Berta, A. D. Gloria, and L. Primavera. Adaptive experience engine for serious games. *IEEE Trans. Comput. Intellig. and AI in Games*, 1(4):264–280, 2009.

[5] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.*, 4(4):253–278, 1995.

[6] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper & Row Publishers, Inc., New York, NY, USA, 1990.

[7] J. Dormans. Adventures in level design: generating missions and spaces for action adventure games. In *PCGames '10: Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, pages 1–8, New York, NY, USA, 2010. ACM.

[8] Dreambox Learning. Dreambox Learning K-2 Math. PC Game, 2009.

[9] J. P. Gee. Learning by design: Games as learning machines. *Interactive Educational Multimedia*, 8:15–23, 2004.

[10] C. Grappiolo, Y.-G. Cheong, J. Togelius, R. Khaled, and G. N. Yannakakis. Towards player adaptivity in a serious game for conflict resolution. In *Proceedings of the 2011 Third International Conference on Games and Virtual Worlds for Serious Applications*, VS-GAMES '11, pages 192–198, Washington, DC, USA, 2011. IEEE Computer Society.

[11] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent Tutoring Goes to School in the Big City. *Proceedings of the 7th World Conference on AIED*, 1995.

[12] N. Peirce, O. Conlan, and V. Wade. Adaptive educational games: Providing non-invasive personalised learning experiences. In *Digital Games and Intelligent Toys Based Education, 2008 Second IEEE International Conference on*, pages 28 –35, nov. 2008.

[13] A. M. Smith, E. Andersen, M. Mateas, and Z. Popović. A case study of expressively constrainable level design automation tools for a puzzle game. In *FDG '12: Proceedings of the Seventh International Conference on the Foundations of Digital Games*, New York, NY, USA, 2012. ACM.

[14] A. M. Smith and M. Mateas. Answer set programming for procedural content generation: A design space approach. *IEEE Trans. Comput. Intellig. and AI in Games*, 3(3):187–200, 2011.