# Player Knowledge Modeling in Game Design Feedback and Automation

**Eric Butler**
Center for Game Science
Department of Computer Science & Engineering, University of Washington
edbutler@cs.washington.edu

### Abstract

Models that capture the knowledge of players of digital games could be used to great effect in AI-assisted tools that automate or provide feedback for game design. There are several important tasks knowledge models should perform: predicting player performance on a particular task to adjust difficulty, knowing in which order to give particular concepts for maximum learning, or understanding how the pacing of a concept impacts player engagement. While all of these have been explored individual both in games and related fields like intelligent tutoring systems, there have been no models that capture all of these effects together in a way that allows their use in design tools. We propose to expand on previous work in game authoring tools to create tools in which the designer can leverage information about how players learn their game's concepts to create better designs. We will survey the existing player modeling work to find the best representation for this task, deploy these models in adaptive games to learn from data, and then apply these models to create novel game design tools.

## Introduction

We believe that intelligent game design tools can be made more effective by integrating models that capture how players learn and master game concepts. Many have argued that ordering and pacing of concepts is critical for engagement and effective master of games (Gee 2004; Chen 2007). Some believe that skill mastery is a fundamental aspect of fun in games (Koster 2010).

Modeling different properties of players is a well-studied area, and we plan to focus specifically on a game's mechanical concepts. Cook (2007) refers to these as *skill atoms*, or the individual elements of skill that a player masters over the course of a game. Some intelligent tutoring systems use the term *knowledge components*. In this paper we will refer to them as *concepts*. In intelligent tutoring systems, cognitive models that model a student's knowledge have proven effective at increasing learning gains (Koedinger et al. 1997).

Every concept will take different players a different amount of time to master. Concepts also have complex relationships and dependencies between each other. For example, in Nintendo's *Super Mario Bros.*, players must master jumping before they can master jumping on an enemy or jumping across a gap. Knowing what these relationships are can greatly advance adaptive games and design tools. Adaptive games could wait until players are very likely to have mastered a particular concept before introducing players to dependent concepts. Design tools can integrate this information to validate a designer's game and provide valuable feedback before user testing occurs.

However, determining the interactions between these concepts is very challenging. While experts can provide a guess towards concept interactions based on user testing, such methods are both very expensive and error-prone. We believe a more effective strategy would be to learn these relationships from data. Of course, a fully data-driven model would require an exponential amount of data in the number of concepts to learn all the interactions between them. Thus, there is a need for models that can learn from fewer data points and use expert-provided knowledge as priors.

I plan to develop a framework of modeling player knowledge of game concepts and apply these models to adaptive games and design tools. The primary research questions I wish to address are:

- Which features should be captured by a player knowledge model and what representation can do this?
- What techniques can designers use to construct such player models and learn them from player data?
- How can such models be effectively deployed in AI-assisted design tools?

The primary targets of these research questions will be games developed by my research lab, the Center for Game Science, such as *Refraction*[1]. These games provide the means to perform large-scale experiments on thousands of players to train and test the models. Table 1 lists an example set of concepts from *Refraction*.

## Background on Player Modeling

Smith et al. (2011) provide a recent overview of player modeling within games. Under their taxonomy, we are interested primarily in individual, generative, empirical models. We will also draw heavily from research in the field of intelligent tutors, where student modeling has been a topic of active research for decades. Tutors such as the ACT Programming



[1] http://centerforgamescience.org/portfolio/refraction/

| Concept | Description |
|---|---|
| Bending | Must use bending pieces |
| Splitting | Must use splitting pieces |
| Adding | Must use adding pieces |
| Blockers | Must contain obstructive pieces |
| Wasted Laser | Leaves some laser beams unused |
| Crossed Laser | Laser beams are unavoidably crossed |

Table 1: A list of example *concepts* for *Refraction*, which are the fundamental element we wish to capture in our models.

Tutor (Corbett and Anderson 1994) model student knowledge towards the goal of adapting tutors to maximize learning outcomes.

The first research step will be to survey games, intelligent tutoring, cognitive psychology, and educational literature and create a taxonomy of existing student models and their applications to games. We are particularly focused on building a list of the different educational techniques that the models should attempt to capture, such as dynamically adjusting difficulty to keep the learner engaged. The end goal is the creation of a player model that subsume several existing ones by capturing many of the most salient techniques.

One critical technique is dynamically adjusting challenge to prevent players from becoming bored or frustrated. One way a model can do this is by predicting player performance on a particular challenge. Many intelligent tutoring systems' student models do this, such as Knowledge Tracing (KT) (Corbett and Anderson 1994). KT directly models whether a student understands a particular concept, and enables adaptive techniques such as *mastery learning*, giving students problems covering a concept until the model is 95% confident that the student has mastered it. Such techniques have been captured by game player models as well. One recent example (Zook and Riedl 2012) used collaborative filtering to predict player performance in an RPG, allowing the game to match a designer-specified target performance curve.

One weakness of models like KT is that they typically do not capture relationships between concepts. We primarily hope to capture *prerequisite* relationships, or the partial ordering of concept dependencies. That is, concept $A$ is a prerequisite of $B$ if $A$ should be introduced before $B$. KT assumes all concepts are independent of each other, and the order of concepts is typically given by experts. Some models are general enough to capture these effects, but, as the space of all possible concept orderings is very large, would require a tremendous amount of data to explore. Expert knowledge is both expensive to produce and potentially inaccurate, but we believe that designer-specified prerequisites could be used as priors to more efficiently explore the space of possible concept orderings with less data.

Another important technique we wish to capture is the optimal frequency and rate to present a particular concept. For example, how frequently should a game reuse an already-mastered game concept? For educational games, we want the student to retain mastery of all concepts by the end of the game. Even for other games, it is frequently the case that concepts build upon one another, and so mastery must be retained until dependent concepts are themselves mastered. One one hand, concepts must be revisited periodically lest the player forget them, but concepts that are overused lead to burnout, where the player becomes bored with the overuse of a repeated challenge. Models that merely predict student behavior cannot capture these effects.

## Evaluating Player Models in Adaptive Games

After surveying existing models, the next research goal is to determine a structure that captures these effects. As a method of evaluating the effectiveness of the models, we plan to deploy them in our games as adaptive policies. We are not interested in adaptive policies, per se, but rather want to use them for evaluation: if a model can be used effectively in an adaptive game, then it provides some assurance that it could likely be used effectively in design tools. As a first step, we can train our models on the large amount of existing player data we have for our games and test prediction accuracy of the model. However, since the model is generative, a more appropriate test of its effectiveness is an experiment in an adaptive game, comparing it to both an unadaptive control and existing player models.

A major research question is in how to deploy these models effectively in adaptive policies. We plan to look at existing strategies of adaptivity. Lopes and Bidarra (2011) gives a recent survey of adaptivity within games, discussing the intent, targets of adaptivity, and techniques of adaptivity. Much of the existing work targets adjusting NPC AI behavior, narratives, or game worlds. The particular area we are interested in adjusting are the concepts required in any particular challenge in the game. Previous work from our group (Smith, Butler, and Popović 2013) presents a technique to generate puzzles for *Refraction* that contain an arbitrary concept. The technique uses Answer Set Programming, a constraint-programming technique that allows designers to concisely and declaratively define what it means for a concept to be used in a particular solution. The algorithm requires, as input, the set of concepts that should be present in the level, and using an algorithm similar to SAT solvers, searches the combinatorial space for a puzzle such that *every* solution of that puzzle requires the given concepts. This generator can be applied to other discrete, puzzle-like games. Thus, we can reduce the problem of adapting game content to the problem of choosing which concepts to present.

Therefore, we believe the majority of this part of the research will be developing an adaptive policy that uses the model to select the game concepts to present to the player. We need a good structure for this policy. As previously discussed, many existing adaptive systems perform online, heuristic policies, selecting the next challenge using certain pedagogical assumptions and predicting how the student will perform. While such policies are efficient to compute, they rely on preexisting theory. Though education researchers can draw on the large body of educational theory, the theories behind game design are much less well-understood. Therefore, policies which prescribe particular theories are not very useful for evaluating the accuracy of

new models. Fully data-driven policies, such as partially-observable Markov decision processes (POMDP) do not require theory; however, they are intractable both in time for policy computation and amount of player data required. We will need to find a balance, choosing planning algorithms that make enough assumptions to be practical without prescribing untested game design theories.

One notable hurdle in creating these models is in determining the concepts (*knowledge components* in the intelligent tutor terminology) used by the model. Intelligent tutors typically operate on very well-studied domains such as high-school mathematics, where researchers and experts have been refining the knowledge components for decades. However, not only do many game mechanics have significantly less study devoted to them, but often games contain unique or very rare mechanics. Thus, an important issue that we will tackle is how to enable designers to effectively choose the knowledge components for their particular domain. The approach will likely require creating a tool that allows a designer to iteratively define and test new concepts on existing data.

We plan to explore transferring information from a model trained on one game to another, similar game. This is one of the advantages of having interpretable models that model each concept individually: we can extract the parameters about a particular concept and use them as priors for the same or a very similar concept in a different game.

Once we have created adaptive policies that use these models, we plan to evaluate them with controlled studies. Our research group has previously conducted large-scale controlled experiments (Andersen et al. 2011; 2012) using our group's educational math games, such as *Refraction* and *Treefrog Treasure*. These games are routinely played by thousands of players on online game websites such as Kongregate[2] and BrainPOP[3]. As educational math games, they are prime targets for adaptivity, so we will use these games and their large player bases to conduct controlled experiments. We plan to evaluate the quality of the adaptive algorithm by comparing the learning gains across the different experimental groups. Learning gains could be either evaluated directly on performance in the puzzles or through pre and post assessment tests administered through the game.

## Deploying to Mixed-Initiative Design Tools

While these models have obvious applications fully-automatic generators, we believe an enticing application is the deployment of these models in mixed-initiative game authoring tools. Previous efforts in mixed-initiative game design, such as Tanagra (Smith, Whitehead, and Mateas 2011) and SketchaWorld (Smelik et al. 2010), pair generative techniques with an interactive editing interface that allows the human and machine to take turns editing a shared level design. Such tools help the designer prototype new ideas and check constraints for quality assurance while still allowing designers to craft levels with subtle properties that are difficult to formalize.

[2]http://www.kongregate.com/

[3]http://www.brainpop.com/

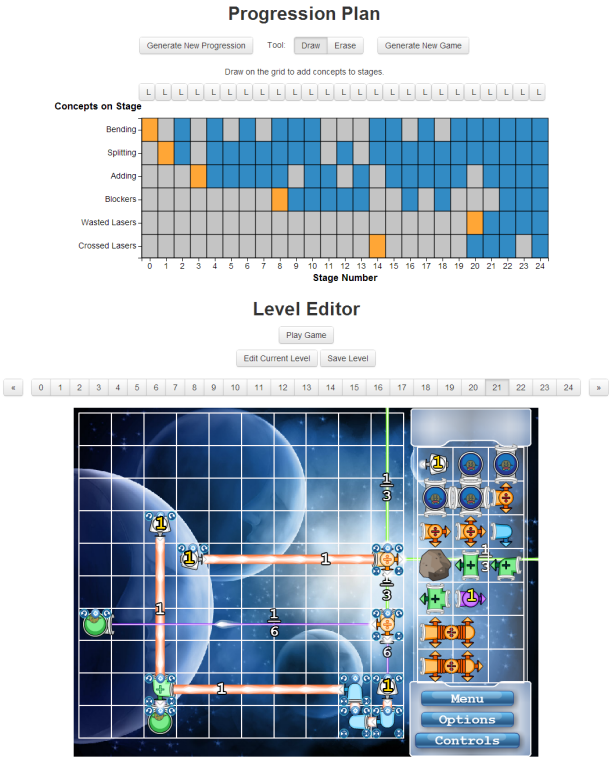


Figure 1: Screencaptures of two of the three editing environments of our game progression authoring tool, each for editing the game at a different scale. On top is the *progression plan editor*, used for viewing and manipulating which gameplay concepts appear on which levels. On the bottom is *Refraction*'s level editor. There are additional views for editing constraints and playtesting.

While most existing mixed-initiative game design tools focus on a single level or area, we previously created a mixed-initiative game authoring tool that allows the designer to work with the entire progression of a game (Butler et al. 2013). While game designers typically plan their progressions (i.e., which game concepts occur in which levels) manually with something like a whiteboard, our tool modeled the progression plan directly. The tool allows the designer to specify constraints on the progression such as a partial ordering of concepts. When the designer manually modifies a level, the tool keeps the progression plan synchronized, also checking if any constraints are violated. This enables the designer to move between broad and narrow-scale editing and allows for automatic detection of problems caused by edits to levels. We further leveraged our content generation techniques (Smith, Butler, and Popović 2013) to help the designer rapidly explore and test game progressions by generating both the plan of concepts for each and levels that use those concepts. An image of our authoring tool can be found in Figure 1.

The tool supported several progression constraints backed by theoretical (to use terminology from (Smith et al. 2011))

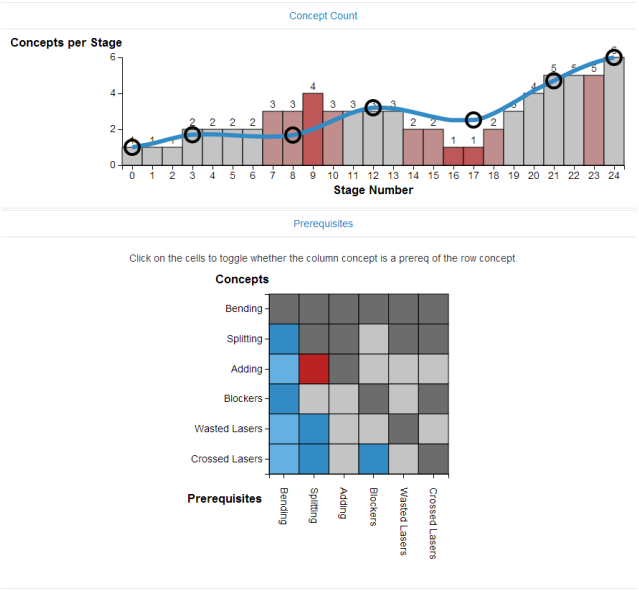


Figure 2: View of the editing interface for constraints in our mixed-initiative authoring tool. On top, the *concept count* graph shows how many concepts occur on each level. The blue curve is the target, and the bars are the values of the current plan. On bottom, the red cell of the *prerequisite* chart shows us which concepts must be introduced before others. Blue indicates an active constraint.

player models. For example, players could constrain the partial ordering of concept introductions, or control how many concepts should appear on each level. The editing interface for these constraints is shown in Figure 2. While such functions could still provide some utility to a designer, we believe and obvious and enticing avenue of future work is backing such constraints with a player knowledge model. Thus, we propose exploring how to best take advantage of our learned player models in a design tool.

We would evaluate such a system by the novel interactions it affords the designer. We propose several potential interactions. The game designer could set constraints on the expected distribution of player skill at particular game mechanics. One use case of these constraints is for feedback: for example, as the designer manually edits a *Refraction* puzzle to teach adding, the system could use the player knowledge model to report that it expects most players will have trouble mastering the concepts covered without reducing the amount of laser bending on that level. Another enticing use case is in content generation. The tool could expose editable curves (similar to Figure 2) that allow the designer to paint player skill mastery at different time points. The system could use the model to find an example game that is most likely to match this curve.

By constructing effective models that represent player knowledge of game concepts, I believe game designers can leverage such models for both more effective adaptivity and powerful design tools.

## References


Andersen, E.; Liu, Y.-E.; Snider, R.; Szeto, R.; and Popović, Z. 2011. 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: ACM.

Andersen, E.; O'Rourke, E.; Liu, Y.-E.; Snider, R.; Lowdermilk, J.; Truong, D.; Cooper, S.; and Popović, Z. 2012. The impact of tutorials on games of varying complexity. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, 59–68. New York, NY, USA: ACM.

Butler, E.; Smith, A. M.; Liu, Y.-E.; and Popović, Z. 2013. A mixed-initiative tool for designing level progressions in games. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, UIST '13. To appear.

Chen, J. 2007. Flow in games (and everything else). *Communications of the ACM* 50(4):31–34.

Cook, D. 2007. The chemistry of game design. *Gamasutra*.

Corbett, A. T., and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4(4):253–278.

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

Koedinger, K. R.; Anderson, J. R.; Hadley, W. H.; Mark, M. A.; et al. 1997. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED)* 8:30–43.

Koster, R. 2010. *Theory of fun for game design*. O'Reilly Media, Inc.

Lopes, R., and Bidarra, R. 2011. Adaptivity challenges in games and simulations: a survey. *Computational Intelligence and AI in Games, IEEE Transactions on* 3(2):85–99.

Smelik, R.; Tutenel, T.; de Kraker, K. J.; and Bidarra, R. 2010. Integrating procedural generation and manual editing of virtual worlds. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, PCGames '10, 2:1–2:8. ACM.

Smith, A. M.; Lewis, C.; Hullett, K.; Smith, G.; and Sullivan, A. 2011. An inclusive taxonomy of player modeling. *University of California, Santa Cruz, Tech. Rep. UCSC-SOE-11-13*.

Smith, A.; Butler, E.; and Popović, Z. 2013. Quantifying over play: Constraining undesirable solutions in puzzle design. In *FDG '13: Proceedings of the Eighth International Conference on the Foundations of Digital Games*.

Smith, G.; Whitehead, J.; and Mateas, M. 2011. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *Computational Intelligence and AI in Games, IEEE Transactions on* 3(3):201–215.

Zook, A., and Riedl, M. O. 2012. A temporal data-driven player model for dynamic difficulty adjustment. In *Proceedings of the Eighth Artificial Intelligence in Digital Games Conference*, AIIDE '12.