Using a Disjoint Skill Model for Game and Task Difficulty in Human Computation Games

Anurag Sarkar
Northeastern University
Boston, MA, USA
sarkar.an@husky.neu.edu

Seth Cooper
Northeastern University
Boston, MA, USA
se.cooper@northeastern.edu

Abstract

Prior research has used player rating systems to balance difficulty in human computation games (HCGs) without having to modify their levels by assigning ratings to levels to indicate level difficulty. Skill chains have also been used to define difficulty progressions for such games. Both these methods typically involve associating a level with a single rating or set of skills as being representative of the difficulty of both the in-game mechanics of the level and the complexity of the task that it models, taken together as a single unit. Though effective, this may not be suitable for HCGs where the game and the task being modeled require different sets of skills and abilities. To this end, we introduce a disjoint skill model that separately tracks game and task skill and difficulty in a 2D platformer HCG. We find that the disjoint model enables players to solve more difficult tasks compared to a baseline model.

Author Keywords

rating systems; matchmaking; human computation games; skill chain; dynamic difficulty adjustment; disjoint design

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI);**

CHI PLAY EA '19, October 22–25, 2019, Barcelona, Spain.
© 2019 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-6871-1/19/10.
https://doi.org/10.1145/3341215.3356310

Introduction

Human computation games (HCGs) leverage the abilities of players to help solve real-world problems by modeling them as in-game levels, and have found application in tasks such as image labeling [30], protein design [14] and software verification [5]. To overcome the barriers to difficulty balancing posed by such games on account of levels not being easily modifiable since they model problems, player rating systems like Glicko-2 [8] have been used to perform dynamic difficulty adjustment (DDA) via player-versus-level (PvL) matchmaking [24]. This involves assigning *ratings* to players and levels indicating their skill and difficulty, respectively, and using these ratings to find levels of appropriate difficulty for players. Moreover, prior work has also used *skill chains* within HCGs [9, 10] to analyze and craft level progressions that require players to acquire progressively more complex skills over the course of gameplay.

While such methods have been successful in balancing difficulty, boosting player engagement and performing progression analysis, they typically associate a level with a single difficulty rating or a single set of required skills meant to capture the challenge posed by the level both in terms of the in-game mechanics needed to complete it as well as the human computation task it is modeling. Though not problematic for HCGs where game mechanics are tightly coupled with the nature of the computation task being modeled, using such singular measures of difficulty may be more restrictive than necessary in HCGs where this coupling is more loose. In such HCGs, players may possess varying degrees of competence in executing in-game mechanics and performing the computation task. Thus, using a single rating or set of skills for levels and players may match players with levels that are suitably challenging in terms of the mechanics of the game but that model tasks that may be too difficult and vice-versa.

In this paper, we introduce a disjoint skill model that tracks ratings and skills (i.e., a Glicko-2 rating and a skill chain) separately for game levels and tasks. Under this model, each player has two sets of ratings and skills—one corresponding to their abilities with respect to the game's mechanics and the other to their competence in performing the tasks. Thus, the model consists of two parallel DDA systems, one determining which game level to serve the player and the other determining which task to serve on top of the level. By separating game and task in HCGs where game and task mechanics are not closely related, we can perform more fine-grained DDA ensuring that both levels and tasks are appropriately balanced given the player's current abilities. We demonstrate this model using the 2D platformer HCG *Iowa James: Hunter Collector Gatherer*. Our results show that compared to a baseline skill model where levels have fixed tasks, the disjoint model enabled players to exhibit better task performance while performing similarly in terms of the game mechanics.

Background

Human Computation Games

The design of most HCGs is centered around the problem they are trying to solve resulting in mechanics that are tightly coupled with the task being modeled. Thus the principal focus behind the design of many such games is to enable players to solve problems and not necessarily to maximize their engagement and experience. Jamieson et al. [12] thus argues for adopting commercial and mainstream game genres to model human computation tasks by discovering isomorphic relationships between human computation problems and mechanics of popular game genres. However, Krause et al. [15] discuss *disjoint* human computation game design and demonstrate the game *OnToGalaxy* whose space shooter mechanics are unrelated to its task of populating an ontology. Other examples of such HCGs are the *Landspotting* games [28] that consist of a strategy game, a tower defense game, a tagging game and a tile-based game, all for the task of labeling land cover data. Most similar to the game that we used is *Gwario* [25], a platformer based on *Super Mario Bros.* [6], where the task is to correctly identify items given the purchasing location. Tuite [29] refers to such HCGs as having *orthogonal mechanics*, i.e. mechanics that do not directly serve the task as in the type of HCGs for which our disjoint model would be useful. Tuite warns against obfuscating the underlying purpose of the HCG with such mechanics and recommends making the game's problem statement more explicit to players, citing ethical concerns and arguing that players are likely to be more invested in the game if they better understand its underlying goals. A disjoint model, as presented here, could be useful towards this end since by tracking game and task difficulty separately, players may become more proficient at performing the task and thereby gain a better understanding of the game's underlying objectives.

Skill Chains and Rating Systems in HCGs

Cook [2] describes a skill chain model, which defines atomic game skills and the dependencies between them. Prior work has utilized skill chains to define and analyze difficulty progressions within HCGs. Horn et al. [10] used the skill chain of the 2D puzzle HCG *Foldit* to design AI agents simulating players of varying levels of competence and used them to analyze different level progressions of the game. Along with skill chains, player rating systems have also been used in HCGs for balancing difficulty specifically via using the Glicko-2 [8] system to perform player-versus-level matchmaking [24]. Prior applications of these methods within HCGs however have assumed the modeled task to be an implicit component of each level and assigned ratings and skills to levels and tasks taken together as a single unit. Our disjoint model extends existing applications of skill chains and rating systems by assigning skills and ratings to levels and tasks separately to be more suited to HCGs with a disjoint design as described previously.

Dynamic Difficulty Adjustment (DDA)

Outside of HCGs, much prior work has focused on DDA (i.e. dynamically altering in-game difficulty based on player ability) using techniques such as, among others, machine learning [13], modifying level design [4], player modeling [32] and systems such as *Hamlet* [11] which modify the game world based on evaluations of player performance. While we used skill chains and player rating systems in this work due to their previously mentioned benefits for DDA in HCGs, future work could consider incorporating elements of the above DDA approaches.

Educational Games

The player models we use have much in common with learner models used in education games and intelligent tutoring systems. In particular, knowledge tracing [3] estimates the likelihoods that a learner has mastered specific skills, and item response theory [20] estimates learner skill based on performance. Many models have proposed to extend these approaches [1, 18, 31]. Several educational games could lend themselves to such disjoint models, which allow game and educational aspects to be handled independently. For example, prior work has explored techniques for optimizing the mathematical content of a game meant to teach numberlines [16, 17]. Pelánek et al. [19] propose a learner model (among others) that uses multiple Elo ratings [7] to model both a learner's global skill and their skill at specific concepts. From this perspective, our disjoint model could be considered to have concepts for the game and the task.

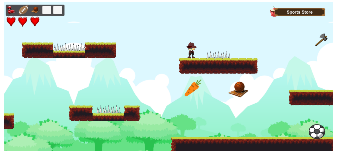
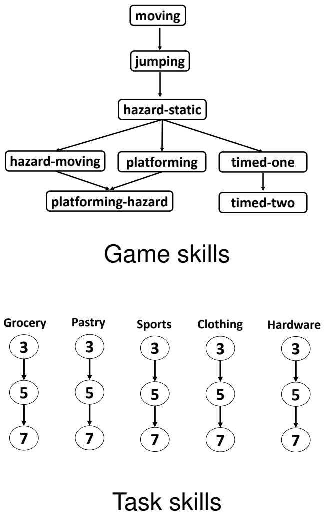


Figure 1: Iowa James screenshot.



Game skills

Task skills

Figure 2: Iowa James skill chains used in this work.

Game and Task

For this work, the game used was *Iowa James: Hunter Collector Gatherer*, a 2D platformer HCG similar to *Gwario* [25] and based on *Iowa James: Treasure Hunter* [23]. Players have to collect items that are relevant to a given scenario as indicated by the UI. A screenshot of the game is shown in Figure 1. Levels feature items that players need to collect and hazards that they must avoid in order to reach a treasure chest at the end. The goal of each level is to unlock the treasure chest by collecting all items relevant to the scenario and then reach the opened chest to move to the next level. Players have three lives per level and lose a life either when they collect an irrelevant item or come in contact with a hazard. The skill chain for the game comprises of typical platformer mechanics and include:

- *jumping*, *moving*: typical platformer movement
- *hazard-static*, *hazard-moving*: jumping over static and moving hazards respectively
- *platforming*: traversing platforms via jumping/moving
- *platforming-hazard*: traversing platforms with hazards
- *timed-one, timed-two*: traversing timed hazards with short and long lengths respectively

The task for this work was that of collecting all items relevant to a presented scenario while avoiding the irrelevant ones. Such a task, which can be viewed as an application of the broader task of item classification or tagging, is one for which the ground truth is known and is thus suitable as a test task for HCG research [25, 26, 27]. We used 5 possible scenarios - *Grocery Store*, *Pastry Shop*, *Clothing Store*, *Sports Store* and *Hardware Store* with players required to collect either 3, 5 or 7 relevant items while avoiding 3, 5 or 7 irrelevant items respectively. Each skill in the task skill chain corresponds to collecting a certain number of items relevant to a specific scenario with collecting 7 items of a scenario being dependant on collecting 5 of that scenario, which in turn depends on collecting 3 of that scenario. Both game and task skill chains are depicted in Figure 2.

Both the player's game and task scores ranged from 0 to 1 and consisted of two parts. For game score, the player earned 0.5 for reaching the treasure chest at the end, with or without having opened it, while the other 0.5 was proportional to the number of times they died due to a hazard. For task score, the player received 0.5 for opening the chest by collecting all relevant items with the other 0.5 being proportional to the ratio of relevant to irrelevant items collected.

Method

In this section we describe our disjoint skill model. This involves 3 stages: 1) Skill chain definition for both game and task 2) Annotation and initialization of game levels and tasks and 3) Matchmaking to determine which level and task to serve the player. Each of these is described below.

Game and Task Skill Chain Definition

First, we define separate skill chains for the game and the task. Skill chains can be represented as directed graphs where each node corresponds to a skill and edges between nodes correspond to dependencies between skills. If there exists an edge from node A to B, then the skill represented by node B depends on that represented by A. For this work, we manually defined both the game and the task skill chains which are depicted in Figure 2.

Annotation and Initialization

After defining the skill chains, the model must be informed about the game levels and tasks for matchmaking. This involves annotating levels with the game skills needed to solve them and initializing each level and task with a default Glicko-2 rating of 1500. Note that each skill in the task skill chain corresponds to a separate task whereas a level may require any number of skills from the game skill chain.

Level and Task Matchmaking

Once levels and tasks have been initialized, they can be used for matchmaking. In the disjoint model, matchmaking involves two DDA systems running in parallel—one to determine which level to serve the player and the other to determine which task to serve. Each player is thus assigned two ratings and two sets of skills, one each for the game DDA system and the task DDA system. Both these ratings are initialized to 1500 and both sets of skills are initially empty.

To determine the levels eligible for serving, the system first filters those levels that the player has already beaten and the immediately previous level they played. From those left, levels requiring exactly one additional game skill not in the player's current set of acquired skills are deemed eligible to be served. If such a level is not found, the system looks for levels that require two additional skills and so on, until an eligible level is found. If the player has acquired all game skills, then all unbeaten levels are eligible.

Task eligibility is determined similarly except that instead of the game skill chain, the task skill chain is used and we don't filter out tasks based on if they've been completed previously or if it was the last task that the player encountered. Thus task eligibility was determined solely based on task skills that the player had acquired at that point.

From among the eligible levels and tasks, the specific level and task to serve to the player are determined by comparing the Glicko-2 ratings with the player's Glicko-2 rating, for both the game and the task. The level and task are picked independently, by calculating the player's desired loss rate (DLR) for each, which represents the desired probability the player will not successfully complete the game level or task they are matched with. The DLR for a player with rating r is given by the equation $\mathrm{DLR}(r) \approx 1/(1 + e^{0.00628(1850-r)})$, which starts players out trying to assign them levels and tasks they only have an estimated 10% chance of losing. As the player's rating, and thus ability, improves, the DLR goes up, causing the player to be matched with more difficult levels and tasks. Details of DLR-based PvL matchmaking can be found in [22]. After computing the two DLRs, we calculate the player's loss probability against each eligible game level and each eligible task using the Glicko-2 expectation function [8]. We then independently select the level and task for which their loss probability is closest to the corresponding DLR and serve these to the player.

The player then plays through the served level while performing the served task. Each such instance is treated as two matches occurring simultaneously—player-vs-level and player-vs-task. Separate game and task scores are computed for the player for each match depending on how well they navigate the level and perform the task respectively. If the game score is more than 0.5, then the player's list of acquired game skills is updated with the additional skill(s) required by that level. Similarly if the task score is more than 0.5, that task is added to the player's list of acquired task skills. The game score is also used to update the player's game rating and the level's rating while the task score is used to update the player's task rating and the task's rating.

Joint Model

In order to evaluate the disjoint model, we also defined a "joint" skill model for comparison. This model uses only the game skill chain and assigns ratings to only the levels, ignoring tracking of the skills and ratings of tasks. Consequently, players are assigned a single game rating and a single set of acquired game skills. In this joint model, the level to be served is still determined dynamically as in the disjoint model, but each level is associated with a fixed task, which were manually chosen to associate tasks requiring more skills with levels requiring more skills. Similarly, only

	Joint	Disjoint
N	143	136
Relevant items $(p = .17)$		
median	15	16
mean	28.8	34.5
Irrelevant items $(p = .89)$		
median	8	9
mean	17.3	16.1
Max task size $(p < .001)$		
median	**3.0**	**3.0**
mean	**2.9**	**4.0**
Max level skill chain magnitude $(p = .12)$		
median	3	3
mean	2.9	3.1
Levels completed $(p = .39)$		
median	3	3
mean	3.5	3.9

Table 1: Summary values for metrics along with results of Wilcoxon Rank-Sum tests. Values in bold were significantly different.

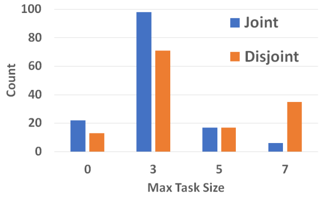


Figure 3: Distribution of *Max Task Size* for conditions.

the game score is used to update the player's rating and skills in this model.

Evaluation and Discussion

To compare the two models, we ran a Human Intelligence Task (HIT) on Amazon Mechanical Turk which recruited 300 players and randomly assigned each to one of the models. The HIT paid $1.25 but players were paid prior to playing and could choose to not play and just take the payment [21]. We ended up with data for 279 players with 136 and 143 assigned to the disjoint and joint models respectively.

For each player, we looked at *Total Relevant Items Collected*, *Total Irrelevant Items Collected*, *Max Task Size*, *Max Level Skill Chain Magnitude* and *Levels Completed*. *Max Task Size* refers to the highest number of relevant items of any scenario that the player was able to collect and could be either 0, 3, 5 or 7. Similarly, *Max Level Skill Chain Magnitude* is the highest number of skills in the skill chain of any level that the player was able to complete. Results of these comparisons are given in Table 1. For each metric, we ran a Wilcoxon Rank-Sum Test. We found significant differences between players in the two conditions in terms of *Max Task Size* $(p < .001)$. The distribution of all possible *Max Task Size* values for players in each condition is shown in Figure 3. Since *Max Task Size* is effectively a measure of how well players progress along the task skill chain, these results suggest that players under the disjoint model were better at acquiring and demonstrating task skills than those under the joint model. Note that, though not significant, the related metric of *Relevant Items Collected*, was also higher for the disjoint model. Overall, based on these results, compared to the joint model, the disjoint model seems to enable players to advance further in terms of task complexity, with players reaching and completing tasks that require a significantly greater number of relevant items to be collected.

These results demonstrate the potential utility of the disjoint skill model. Though the joint model dynamically serves levels based on the player's acquired game skills and game ratings, by not tracking task skills and ratings, it may cause players to either attempt tasks beyond their capabilities, or never reach more advanced tasks that they could have completed. As a result, players fail to acquire more complex task skills despite making progress in terms of the skills related to the game's mechanics, becoming proficient in navigating levels but not in performing the tasks they model. By taking task skills and ratings into account separately, the disjoint model addresses this issue and enables players to make progress both in terms of the mechanics of the game and the computation task the HCG represents.

Conclusion and Future Work

We presented a skill model that separately tracks player skill for game and task in human computation games, particularly suited towards HCGs following a disjoint design where the particulars of the task are not directly linked to the mechanics of the game used to model it. Based on our results, a disjoint skill model enables players to exhibit better task performance than a joint model that does not take task skill specifically into account. Since this work applied the disjoint model only to one HCG and used one task, future work should look into testing its performance on other types of HCGs and tasks more complex than item collection. Additionally, the skill chains for both game and task were constructed by hand. Future work could investigate inferring such skill chains using automated methods.

Acknowledgements

We thank Pratheep Kumar Paranthaman for help with developing the game, and all the players. This material is based upon work supported by the National Science Foundation under grant no. 1652537.

REFERENCES

1. Ryan S. J. d. Baker, Albert T. Corbett, and Vincent Aleven. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian Knowledge Tracing. In *Intelligent Tutoring Systems*, Beverley P. Woolf, Esma Aïmeur, Roger Nkambou, and Susanne Lajoie (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 406–415.

2. Daniel Cook. 2007. The chemistry of game design. (2007). http://www.gamasutra.com/view/feature/1524/the_chemistry_of_game_design.php

3. Albert T. Corbett and John R. Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4, 4 (1 Dec. 1994), 253–278. DOI:http://dx.doi.org/10.1007/BF01099821

4. Valve Corporation. 2008. *Left 4 Dead*. Game. (2008).

5. Drew Dean, Sean Gaurino, Leonard Eusebi, Andrew Keplinger, Tim Pavlik, Ronald Watro, Aaron Cammarata, John Murray, Kelly McLaughlin, John Cheng, and Thomas Maddern. 2015. Lessons learned in game development for crowdsourced software formal verification. In *Proceedings of the 2015 USENIX Summit on Gaming, Games, and Gamification in Security Education*. USENIX Association, Washington, D.C.

6. Nintendo Creative Department. 1985. *Super Mario Bros.* Game [NES]. (1985). Nintendo, Kyoto, Japan.

7. Arpad E. Elo. 1978. *The rating of chessplayers, past and present*. Arco.

8. Mark E. Glickman. 2001. Dynamic paired comparison models with stochastic variances. *Journal of Applied Statistics* 28, 6 (Aug. 2001), 673–689. DOI: http://dx.doi.org/10.1080/02664760120059219

9. Britton Horn, Seth Cooper, and Sebastian Deterding. 2017. Adapting Cognitive Task Analysis to elicit the skill chain of a game. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. 277–289.

10. Britton Horn, Josh Aaron Miller, Gillian Smith, and Seth Cooper. 2018. A Monte Carlo approach to skill-based automated playtesting. In *Proceedings of the Thirteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

11. Robin Hunicke. 2005. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. 429–433. DOI: http://dx.doi.org/10.1145/1178477.1178573

12. Peter Jamieson, Lindsay Grace, and Jack Hall. 2012. Research directions for pushing harnessing human computation to mainstream video games. In *Proceedings of Meaningful Play*.

13. Martin Jennings-Teats, Gillian Smith, and Noah Wardrip-Fruin. 2010. Polymorph: dynamic difficulty adjustment through level generation. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. 11:1–11:4. DOI: http://dx.doi.org/10.1145/1814256.1814267

14. Brian Koepnick, Jeff Flatten, Tamir Husain, Alex Ford, Daniel-Adriano Silva, Matthew J. Bick, Aaron Bauer, Gaohua Liu, Yojiro Ishida, Alexander Boykov, Roger D. Estep, Susan Kleinfelter, Toke Nørgård-Solano, Linda Wei, Foldit Players, Gaetano T. Montelione, Frank DiMaio, Zoran Popovic, Firas Khatib, Seth Cooper, and David Baker. 2019. De novo protein design by citizen scientists. *Nature* 570, 7761 (June 2019), 390–394.

15. Markus Krause, Aneta Takhtamysheva, Marion Wittstock, and Rainer Malaka. 2010. Frontiers of a paradigm - Exploring human computation with a digital game. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*.

16. Yun-En Liu, Travis Mandel, Emma Brunskill, and Zoran Popović. 2014a. Towards automatic experimentation of educational knowledge. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. 3349–3358. DOI: http://dx.doi.org/10.1145/2556288.2557392

17. Yun-En Liu, Travis Mandel, Emma Brunskill, and Zoran Popović. 2014b. Trading off scientific knowledge and user learning with multi-armed bandits. In *Proceedings of Educational Data Mining*.

18. Zachary A. Pardos and Neil T. Heffernan. 2011. KT-IDEM: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization*, Joseph A. Konstan, Ricardo Conejo, José L. Marzo, and Nuria Oliver (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 243–254.

19. Radek Pelánek, Jan Papoušek, Jiří Řihák, Vít Stanislav, and Juraj Nižnan. 2017. Elo-based learner modeling for the adaptive practice of facts. *User Modeling and User-Adapted Interaction* 27, 1 (1 March 2017), 89–118. DOI: http://dx.doi.org/10.1007/s11257-016-9185-7

20. Georg Rasch. 1960. *Probabilistic models for some intelligence and attainment tests*. Copenhagen, Denmark: Danish Institute for Educational Research.

21. Anurag Sarkar and Seth Cooper. 2018. Comparing paid and volunteer recruitment in human computation games. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*.

22. Anurag Sarkar and Seth Cooper. 2019. Transforming Game Difficulty Curves using Function Composition. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*.

23. Anurag Sarkar, Varun Sriram, Riddhi Padte, Jeffrey Cao, and Seth Cooper. 2018. Desire-path inspired procedural placement of coins in a platformer game. In *Proceedings of the Fifth Workshop on Experimental AI in Games*.

24. Anurag Sarkar, Michael Williams, Sebastian Deterding, and Seth Cooper. 2017. Engagement effects of player rating system-based matchmaking for level ordering in human computation games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*.

25. Kristin Siu, Matthew Guzdial, and Mark Riedl. 2017. Evaluating single player and multiplayer in human computation games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*.

26. Kristin Siu, Alexander Zook, and Mark Riedl. 2014. Collaboration versus competition: design and evaluation of mechanics for games with a purpose. In *Proceedings of the 9th International Conference on the Foundations of Digital Games*.

27. Kristin Siu, Alexander Zook, and Mark Riedl. 2017. A framework for exploring and evaluating mechanics in human computation games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*.

28. Tobias Sturn, Michael Wimmer, Peter Purgathofer, and Steffen Fritz. 2013. Landspotting - games for improving global land cover. In *Proceedings of the 8th International Conference on the Foundations of Digital Games*.

29. Kathleen Tuite. 2014. GWAPs: Games With A Problem. In *Proceedings of the 9th International Conference on the Foundations of Digital Games*.

30. Luis von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems*.

31. Shuhan Wang, Fang He, and Erik Andersen. 2017. A unified framework for knowledge assessment and progression analysis and design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*.

32. Alexander Zook and Mark Riedl. 2012. A temporal data-driven player model for dynamic difficulty adjustment. In *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.