

# Transforming Game Difficulty Curves using Function Composition

Anurag Sarkar

Northeastern University  
sarkar.an@husky.neu.edu

Seth Cooper

Northeastern University  
scooper@ccs.neu.edu

## ABSTRACT

Player engagement within a game is often influenced by its difficulty curve: the pace at which in-game challenges become harder. Thus, finding an optimal difficulty curve is important. In this paper, we present a flexible and formal approach to transforming game difficulty curves by leveraging function composition. This allows us to describe changes to difficulty curves, such as making them “smoother”, in a more precise way. In an experiment with 400 players, we used function composition to modify the existing difficulty curve of the puzzle game *Paradox* to generate new curves. We found that transforming difficulty curves in this way impacted player engagement, including the number of levels completed and the estimated skill needed to complete those levels, as well as perceived competence. Further, we found some transformed curves dominated others with respect to engagement, indicating that different design goals can be traded-off by considering a subset of curves.

## CCS CONCEPTS

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

## KEYWORDS

games; difficulty; adaptivity; difficulty curve

## ACM Reference Format:

Anurag Sarkar and Seth Cooper. 2019. Transforming Game Difficulty Curves using Function Composition. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019)*, May 4–9, 2019, Glasgow, Scotland UK. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3290605.3300781>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-5970-2/19/05...\$15.00

<https://doi.org/10.1145/3290605.3300781>

## 1 INTRODUCTION

The difficulty curve of a game defines how the game’s difficulty changes over the course of gameplay. In most situations, the in-game difficulty should ideally be compatible with the player’s current level of skill so that the player is neither overwhelmed by challenges beyond their capabilities, nor bored by challenges too simple to overcome [4]. Thus, defining an optimal curve that presents challenges having difficulties commensurate with player skill is an important game design problem. Traditional methods of finding such a curve involve the designer creating an initial curve which is then refined using an iterative process of playtesting [1].

In this paper, we present a method of such curve refinement by modeling transformations to a game’s difficulty curve as function composition. Function composition allows multiple functions to be composed into one by applying one function to the output of another. Given two functions  $f(x)$  and  $g(x)$ , the composition of the functions  $f \circ g$  is  $f(g(x))$ . This allows us to treat the difficulty curve largely as a black box, only needing to know a few parameters, and use function composition to adjust its inputs and outputs. Doing so also allows us to more precisely describe relative difficulty curves and formalize what it means to, for example, “steepen” or “smooth” a game’s difficulty curve. This is similar in spirit to Swink’s approach to formalizing “game feel” [20] (e.g. what does it mean for controls to feel “floaty”?). Moreover, using functions to transform curves, as opposed to manual refinement, helps capture a potential space of possible difficulty curves that can be explored by applying transformations. Such transformations could give us an approach to considering changes to difficulty curves in a game-independent manner. In addition to moving toward formalizing curve transformations and giving them more precise definitions, the use of functions opens up the possibility of combining such precisely defined curves to generate new curves with precise definitions.

Additionally, we conducted an experiment to test the effects that such curve transformations have on player engagement. Specifically, we applied function composition to transform the difficulty curve of the human computation puzzle game *Paradox* (a screenshot of the game is given in Figure 1) to see what changes in engagement would be brought

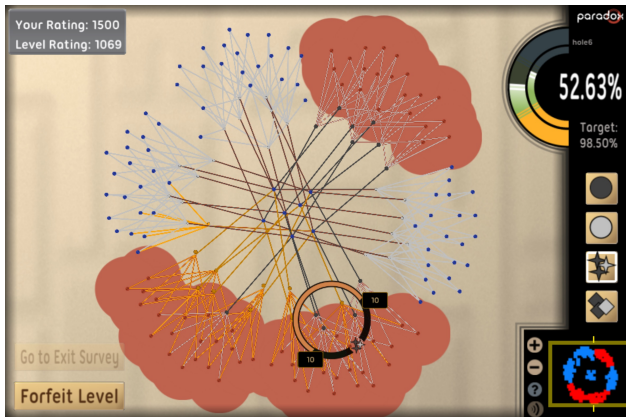


Figure 1: A screenshot of *Paradox*.

forth by different transformations and if any such transformation would further improve the engagement benefits that the heuristically-defined existing curve had demonstrated in previous work [17]. We found that different curve transformations do in fact impact gameplay and engagement. We also identified a set of Pareto efficient [13] transformations for some measures of engagement, i.e. transformed curves for which it is not possible to increase one engagement measure without decreasing another.

## 2 BACKGROUND

### Engagement and Flow

Difficulty curves aim to tune game difficulty so as to optimize player engagement. Engagement refers to a concept in the psychology of motivation that attempts to capture how engaged a person feels in the performance of a task. It draws from Csikszentmihalyi’s flow theory [4] which says that one is optimally engaged when in the *flow state*—a state of mind characterized by deep immersion in an activity coupled with a strong motivation to do it well. Thus, games seek to optimize player experience by pushing players toward such a flow state in order to keep them engaged and entertained. Most often, games do this via difficulty balancing, i.e. adjusting the challenge posed by in-game levels so as to match the skill level of the player. In other words, challenging players with tasks that are neither too difficult so as to not frustrate them, nor too easy so as to not bore them, thereby keeping them within the flow state. Such difficulty balancing methods can take many forms, including pre-release playtesting, dynamic adjustment during gameplay, or analysis of difficulty curves, i.e. comparing different difficulty progressions, as in this work.

### Dynamic Difficulty Adjustment

Rather than presenting every player with the same difficulty, some games tailor the difficulty presented to each player using dynamic difficulty adjustment (DDA). DDA is a general term for techniques used to dynamically modify in-game difficulty during the course of gameplay as a means to tailor the experience more toward the player’s current level of play. This has been achieved in games through various techniques such as parameter tuning [8], modifying level design [3], machine learning [10], the use of rating systems [19] and player modeling [22].

### Learning Curve Analysis

Similar to difficulty curves, learning curves focus on the rate at which skills are acquired by players rather than on the progression of difficulty. These curves are often used in analyzing the pace and challenge presented within educational or puzzle games, as done by Linehan et al [11]. Harpstead et al. [7] demonstrated the use of such learning curve analysis to analyze an educational game and discovered changes that could be made to improve the game’s design to enable it to achieve its teaching goals more effectively. Similarly, analyzing and improving upon a game’s difficulty curve would help improve its difficulty progression. While several DDA techniques have also been employed to tailor the difficulty curve appropriately as mentioned in the previous section, to the best of our knowledge, ours is the first work that explores the use of function composition to transform difficulty curves, with Jacob [9] only briefly mentioning the feasibility of doing so.

### Difficulty Balancing in Human Computation Games

For our experiment, we used the human computation game (HCG) *Paradox*. HCGs are games that aim to leverage the collective abilities of players in order to solve computationally intractable problems. Levels within HCGs, on account of modeling unsolved real world problems, are not readily amenable to traditional DDA techniques since their difficulties are unknown in advance and they cannot be easily modified without compromising their link to the underlying problem [2]. Past work [17] has shown that such barriers can be overcome by mapping player skill and level difficulty to a common rating system such as Glicko-2 and then using a matchmaking algorithm to serve levels to players in an order that enhances engagement. Through such an ordering, the rating system thus defines a skill-dependent difficulty curve for the game. However, in previous work, the curve was defined heuristically and its fitness tested against baselines that did not fully leverage the ratings system. While other work [16] demonstrated improved engagement by offering ratings-based skill feedback to players, no modifications were made

Curve Parameters	Description
$r_i = 1500$	Player’s starting rating (Glicko-2 starting rating)
$r_d = 350$	Rating scale factor (Glicko-2 initial rating deviation)
$\omega = 0.1$	Desired loss rate for starting player
Baseline Curve	Description
$f(x) = \frac{1}{1+e^{\alpha(\beta-x)}}$	Logistic curve
$\beta = 1850$	Set so $f(r_i + r_d) = 0.5$
$\alpha \approx 0.00628$	Set so $f(r_i) = \omega$
Transformation Functions	Description
$t_\delta(x) = x + \delta$	Translate by $\delta$
$s_{\sigma,c}(x) = \sigma(x - c) + c$	Scale by $\sigma$ around $c$

**Table 1: Difficulty curve-related functions and parameters.**

to the curve itself to see if orderings in which levels were served, and thereby engagement, could be improved further.

Based on prior related work, we formulated the following hypothesis for our experiment: *Transforming the difficulty curve of a game using function composition impacts player behavior and experience, with different transformations leading to different player behavior and experience.*

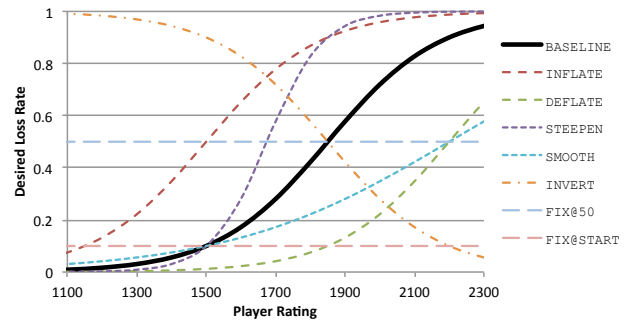
### 3 GAME DESCRIPTION

Levels in *Paradox* are puzzles derived from MAX-SAT boolean constraint satisfaction problems. Player moves within the game involve using various tools to change boolean values to satisfy constraints. The player’s score for a level is the percentage of all constraints that are satisfied; the player’s goal is to achieve a *target score* for the level. A level is considered *attempted* if the player makes at least one move and *completed* if the player is able to reach the target score for that level.

The game uses a difficulty curve-based matchmaking system to determine the next level that is served to a player as they progress through the game and uses the Glicko-2 rating system [6] for matchmaking between players and levels. Using Glicko-2, each player and level is assigned a rating. These ratings can be updated over time based on the outcomes of player attempts at levels. Ratings can be used to estimate a player’s chance of losing a level, i.e. the level’s difficulty for that player. We may consider that a level’s rating is an estimate of the player skill needed to complete the level, and the lower a player’s rating is relative to a level’s rating, the more difficult that level will be for them. Level ratings were

Curve Name	Function	Description
BASELINE	$f$	baseline curve
INFLATE	$f \circ t_{r_d}$	inflate difficulty via shifting curve left by a constant
DEFLATE	$f \circ t_{-r_d}$	deflate difficulty via shifting curve right by a constant
STEEPEN	$f \circ s_{2,r_i}$	steepen difficulty by increasing curve’s rate of change
SMOOTH	$f \circ s_{0.5,r_i}$	smooth difficulty by decreasing curve rate’s rate of change
INVERT	$s_{-1,0.5} \circ f$	invert difficulty by flipping curve upside down
FIX@50	$t_{0.5} \circ s_{0,0} \circ f$	fix difficulty at 50% loss chance
FIX@START	$t_{\omega} \circ s_{0,0} \circ f$	fix difficulty at starting difficulty

**Table 2: Overview of the difficulty curves used and their representation as function compositions.**



**Figure 2: Plot of the different difficulty curves used.**

computed in previous work and kept fixed during this experiment while player ratings were updated over the course of gameplay. Each player starts with a default rating of 1500. If a player completes a level, it is considered a win for the player and their rating goes up. If they forfeit a level (i.e. attempt it without being able to complete it), it is considered a loss for the player and their rating goes down. The player’s rating goes up or down by an amount proportional to the rating of the level. In this way, the difficulty of the game adapts to each player’s skill level as it changes over time. More detailed descriptions of both the game and the matchmaking system can be found in prior work [5, 17].

### 4 CURVE TRANSFORMATIONS

In this work, we consider a difficulty curve as a function that maps from player skill to difficulty. Here, player skill is represented by the player’s Glicko-2 rating and difficulty refers to the desired loss rate i.e. the desired probability of the player losing the level at their current skill rating. This rate is

a measure of the difficulty since it determines how hard the level will be for the player to complete. Thus, the difficulty curve can be given by the function  $d : \mathbb{R} \rightarrow [0, 1]$ . We can further characterize a difficulty curve using a few generic parameters:  $r_i$ , the skill rating of a starting player;  $r_d$ , a rating scale factor (derived from the rating system); and  $\omega$ , the loss rate for a starting player. For our specific baseline curve, we used a logistic function, similar to the one used in past work involving *Paradox*, that smoothly increases difficulty as a player’s rating increases. Details on the specific parameters and functions used are given in Table 1.

The baseline difficulty curve and the transformed curves obtained using function composition are listed in Table 2. The name of each new curve indicates how it transforms the difficulty progression of the game relative to the BASELINE. This is done by applying the transformation functions to either the input (i.e. the player’s current rating) to the baseline function  $f$  as in INFLATE, DEFLATE, STEEPEN and SMOOTH, or its output (i.e. the desired loss rate) as in INVERT, FIX@50 and FIX@START. Specifically, to obtain INFLATE and DEFLATE, we translate the baseline curve along the x-axis (i.e. shift the player rating to the left and right respectively). This causes the player’s desired loss rate to be higher and lower respectively for the same rating, thereby *inflating* (i.e. making the game harder) or *deflating* (i.e. making the game easier) the base curve, hence the naming. Similarly, for STEEPEN and SMOOTH, the player rating is scaled to double and half, causing the base curve to be steeper (i.e. increasing the rate at which the game gets harder) or smoother (i.e. decreasing this rate) respectively. INVERT simply flips the base curve upside down by applying scaling to the desired loss rate output by  $f$  while FIX@START and FIX@50 always serve levels that the player has a 10% ( $\omega$ ) and 50% chance of losing, respectively. A plot of the eight curves is shown in Figure 2.

## 5 DATA COLLECTION

To gather data, we ran a Human Intelligence Task (HIT) on Amazon Mechanical Turk. The HIT paid \$2 and recruited 400 players who were randomly assigned one of the 8 difficulty curves in Table 2. The version of *Paradox* used in our experiment required players to play through 8 tutorial levels. These tutorial levels did not affect player rating and were not used in our analysis. After the tutorial, players could play any number of 50 challenge levels, which they could either skip (i.e. not make any moves before moving to next level), forfeit (i.e. make at least 1 move but fail to complete the level) or complete. After failing to complete (i.e. skipping and/or forfeiting) 3 levels, players could fill out a survey based on the Intrinsic Motivation Inventory (IMI) [14] to finish the HIT. Previous work has shown that players recruited through Mechanical Turk can respond similarly to experiments as volunteer players [15].

For each player, we recorded *Time* (spent playing levels), *Levels Attempted* (those not skipped), and *Levels Completed*, as well as the *Final Player Rating* (player’s rating when they finished the HIT) and the *Highest Level Rating* (rating of the highest rated level that the player completed). The survey contained the IMI subscales for *Interest/Enjoyment* (scale of 7 to 49), *Perceived Competence* (scale of 6 to 42), and *Effort/Importance* (scale of 5 to 35). 55 players did not complete any challenge levels and were excluded from *Highest Level Rating* analysis while 4 players had errors in their surveys and were excluded from survey variable analysis.

## 6 CURVE COMPARISONS AND DISCUSSION

Due to non-normality of recorded variables, as indicated by a Shapiro-Wilk normality test, we used the non-parametric Aligned Rank Transform [21]. For each variable, we first ran an omnibus ANOVA comparison of all curves. If that was significant, we then ran a full post-hoc pairwise comparison with the Tukey method. We used  $\alpha = .05$ . Results of our analyses comparing curves are given in Table 3.

Our analyses revealed that transforming difficulty curves did impact player engagement in terms of the variables that we measured, thereby supporting our hypothesis that different curve transformations would affect player behavior and experience differently. We found significant omnibus differences across curves for all variables except *Player Rating* and *Effort/Importance* and significant post-hoc differences for all other variables except *Interest/Enjoyment*. The latter may speak to the game not being very enjoyable, as can be the case with HCGs in general, on account of having to balance being fun with staying true to the underlying problem. Interestingly, this highlights that though different curves led to players exhibiting different behavior and reporting varying levels of competence, these curves did not seem to affect how much players subjectively enjoyed the game.

Looking at *Time*, players spent more time playing a SMOOTH difficulty curve than a curve that had a FIX@START “easy” difficulty. This may indicate that some ramp-up in difficulty is more engaging. For *Levels Completed*, *Levels Attempted*, and *Perceived Competence*, the general trend appears to be that these were increased by making the curve “easier”. DEFLATE was always significantly greater than INFLATE. On the other hand, the trend for *Highest Level Rating* seems to be that it was increased by making the curve “harder”. These trends make some intuitive sense as in an easier game, players would be expected to attempt and complete more levels while feeling more competent, but the levels they complete would have lower ratings. This is similar to the findings of Lomas et al. [12] which suggest that players are more engaged and play for longer when the game is easier.

It is worth noting that all the curves do not start at the same difficulty. Thus, one question is if it is the starting

Time (s)		Levels Completed		Levels Attempted			Final Player Rating	
FIX@START	413 (256) <sup>a</sup>	INVERT	0.0 (0.0) <sup>a</sup>	INFLATE	4 (2) <sup>a</sup>	INVERT	1421 (77)	
INFLATE	433 (256) <sup>ab</sup>	INFLATE	2.0 (1.0) <sup>a</sup>	INVERT	4 (1) <sup>a</sup>	BASELINE	1466 (189)	
INVERT	516 (358) <sup>ab</sup>	FIX@50	3.0 (2.0) <sup>a</sup>	FIX@50	5 (2) <sup>ab</sup>	FIX@50	1545 (155)	
FIX@50	527 (372) <sup>ab</sup>	STEEPEN	4.5 (1.5) <sup>b</sup>	FIX@START	6 (4) <sup>c</sup>	INFLATE	1573 (165)	
BASELINE	610 (352) <sup>ab</sup>	FIX@START	5.0 (3.0) <sup>b</sup>	STEEPEN	7 (2) <sup>abc</sup>	SMOOTH	1573 (193)	
STEEPEN	618 (323) <sup>ab</sup>	BASELINE	7.0 (3.0) <sup>bc</sup>	BASELINE	10 (3) <sup>cd</sup>	FIX@START	1611 (154)	
DEFLATE	682 (417) <sup>ab</sup>	SMOOTH	8.0 (3.5) <sup>bc</sup>	SMOOTH	10 (4.5) <sup>cd</sup>	STEEPEN	1621 (145)	
SMOOTH	762 (544) <sup>b</sup>	DEFLATE	14.0 (9.0) <sup>c</sup>	DEFLATE	15 (9) <sup>d</sup>	DEFLATE	1642 (209)	

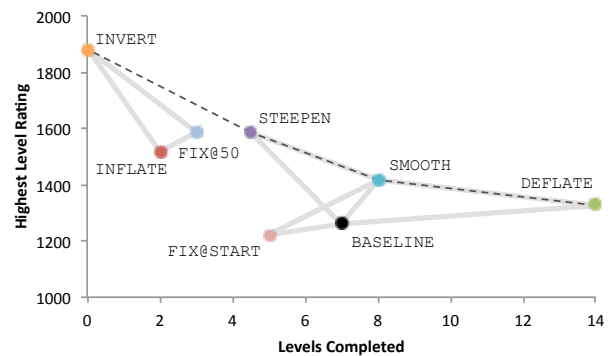
Highest Level Rating		Interest/Enjoyment		Perceived Competence			Effort/Importance	
FIX@START	1222 (153) <sup>a</sup>	FIX@50	26.5 (7) <sup>a</sup>	FIX@50	16 (7) <sup>a</sup>	INFLATE	25 (4.5)	
BASELINE	1260 (175) <sup>ab</sup>	INFLATE	28 (6.5) <sup>a</sup>	INFLATE	17 (8) <sup>ab</sup>	INVERT	25 (6)	
DEFLATE	1328 (259) <sup>a</sup>	FIX@START	29 (6) <sup>a</sup>	INVERT	23 (10) <sup>abc</sup>	FIX@START	28 (5)	
SMOOTH	1416 (108) <sup>ab</sup>	STEEPEN	31.5 (5.5) <sup>a</sup>	BASELINE	25 (5.5) <sup>abc</sup>	DEFLATE	28 (3)	
INFLATE	1517 (363) <sup>bc</sup>	INVERT	32 (11) <sup>a</sup>	STEEPEN	25 (6) <sup>c</sup>	BASELINE	28.5 (4.5)	
STEEPEN	1587 (276) <sup>bc</sup>	BASELINE	32 (6) <sup>a</sup>	FIX@START	26 (5) <sup>bc</sup>	STEEPEN	28.5 (5.5)	
FIX@50	1587 (118) <sup>c</sup>	SMOOTH	33 (7) <sup>a</sup>	SMOOTH	28 (6) <sup>c</sup>	SMOOTH	29 (4)	
INVERT	1880 (140) <sup>c</sup>	DEFLATE	33 (8) <sup>a</sup>	DEFLATE	28 (5) <sup>c</sup>	FIX@50	30 (4.5)	

**Table 3: Summary of data and analysis.** For each variable, median and median absolute deviation (in parenthesis) are shown. Letter superscripts<sup>abcd</sup> show post-hoc significance groups; that is, variables sharing a letter were not significantly different. *Final Player Rating* and *Effort/Importance* did not have a significant omnibus test and thus post-hoc tests were not run.

difficulty that accounts for these differences rather than the change in difficulty due to the curves. If we look at the three curves that start at a difficulty above  $\omega$  (INFLATE, FIX@50, and INVERT), there were no significant differences between them. Looking at the curves that start at difficulty  $\omega$  (BASELINE, SMOOTH, STEEPEN, and FIX@START), most of the differences are not significant. However, for *Time*, FIX@START was different from SMOOTH, and for *Highest Level Rating*, FIX@START was different from STEEPEN. Only one curve started with difficulty below  $\omega$  (DEFLATE). Therefore, we did see cases where the curve of difficulty change makes a difference; in these cases, it appears that increasing difficulty with skill is preferable to a fixed difficulty.

## 7 PARETO EFFICIENCY

Human computation games (HCGs) aim to enable players to maximize the number and/or quality (i.e. accuracy, usefulness) of in-game tasks that they complete [18]. Thus, HCG designers may wish to trade off between the amount of work done and its quality (in our case, level ratings, since higher rated levels are harder and thus may be less amenable to being solved by automated methods). From our analyses, comparing *Levels Completed* and *Highest Level Rating* revealed four of the difficulty curves (INVERT, STEEPEN, SMOOTH and



**Figure 3: Plot of the variables *Levels Completed* against *Highest Level Rating* for the difficulty curves.** The darker dashed line shows the Pareto frontier of curves examined for these variables. The frontier is defined by the curves INVERT, STEEPEN, SMOOTH and DEFLATE. The lighter solid lines connect curves that were not found to be significantly different for either of these two variables.

DEFLATE) to be Pareto efficient for these two variables—that is, the other curves were dominated by these four. A plot demonstrating the Pareto efficiency of the curves is shown in Figure 3. Looking at the significance groupings, we also

found four triplets of curves, namely (INVERT–INFLATE–FIX@50, STEEPEN–BASELINE–SMOOTH, SMOOTH–FIX@START–BASELINE and DEFLATE–SMOOTH–BASELINE), which were not significantly different from each other for these two variables. These perspectives suggest that different design goals may be traded off by considering only a subset of the transformed curves. Interestingly, though the differences were not significant, the original BASELINE was outperformed by SMOOTH and DEFLATE in terms of both variables, suggesting that these might be better difficulty curves for *Paradox* in the future, and speaking to the potential usefulness of this technique for exploring the space of difficulty curves.

## 8 CONCLUSION AND FUTURE WORK

In this paper, we demonstrated a formal approach to transforming a game’s difficulty curve by modeling it as function composition. This enabled us to modify the existing curve of the game *Paradox* to generate new curves for the game and give mathematically precise definitions to how each curve modified the original curve. Our experiment revealed that the transformed curves impact gameplay significantly and that some of them improve player engagement compared to the game’s existing difficulty curve. We consider a number of avenues for future work.

In our experiment, we used the difficulty curves to adapt the game on a per-level basis but this could be done in other ways such as on a per-section basis in a procedurally generated platformer. Additionally, our baseline difficulty curve was monotonic and we used simple functions during composition. In the future, it is worth exploring more complex and/or non-monotonic functions (e.g. *ln* and *sin*) as baselines and transformations. Also worth investigating in the future is combinations of different transformations (e.g. “steepen and deflate”). The parameters of the transformation functions also define a space of possible curve transformations, presenting the possibility for optimization and evolving an optimal difficulty curve for a given game. Finally, while this work transformed the difficulty curves of one game, it may be possible to determine a transformation between difficulty curves of multiple games and discuss their relative properties (e.g. “game X has an *inflated* difficulty curve compared to game Y”).

## ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under grant no. 1652537. We would like to thank the players, and the University of Washington’s Center for Game Science for initial *Paradox* development.

## REFERENCES

- [1] Maria-Virginia Aponte, Guillaume Levieux, and Stephane Natkin. 2011. Measuring the level of difficulty in single player video games. *Entertainment Computing* 2, 4 (2011), 205–213.
- [2] Seth Cooper, Sebastian Deterding, and Theo Tsapakos. 2016. Player rating systems for balancing human computation games: testing the effect of bipartiteness. In *Proceedings of the 1st International Joint Conference of DiGRA and FDG*.
- [3] Valve Corporation. 2008. *Left 4 Dead*. Game.
- [4] Mihaly Csikszentmihalyi. 1990. *Flow: The Psychology of Optimal Experience*. Harper and Row, New York.
- [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*.
- [6] Mark E. Glickman. 1999. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 48, 3 (1999), 377–394.
- [7] Erik Harpstead and Vincent Alevan. 2015. Using empirical learning curve analysis to inform design in an educational game. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*. 197–207.
- [8] 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.
- [9] João Jacob. 2016. *Estimating Player Performance and Adaptivity in Exergames and Location-Based Games*. Ph.D. Dissertation. Faculdade de Engenharia da Universidade do Porto.
- [10] 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.
- [11] Conor Linehan, George Bellord, Ben Kirman, Zachary H. Morford, and Bryan Roche. 2014. Learning curves: analysing pace and challenge in four successful puzzle games. In *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play*. 181–190.
- [12] Derek Lomas, Kishan Patel, Jodi Forlizzi, and Kenneth Koedinger. 2013. Optimizing challenge in an educational game using large-scale design experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 89–98.
- [13] Vilfredo Pareto. 1963. *The Mind and Society; A Treatise on General Sociology*. Translated by Andrew Bongiorno and Arthur Livingston. Dover, New York.
- [14] Richard M. Ryan and Edward L. Deci. 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist* 55, 1 (2000), 68–78.
- [15] 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*. 37:1–37:9.
- [16] Anurag Sarkar and Seth Cooper. 2018. Meet your match rating: providing skill information and choice in player-versus-level matchmaking. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*. 36:1–36:8.
- [17] 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*. 22:1–22:10.
- [18] 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*. 38:1–38:4.
- [19] Alex Cho Snyder and Mario Izquierdo. 2014. *Jumpcraft*. Game.

- [20] Steve Swink. 2008. *Game Feel: A Game Designer's Guide to Virtual Sensation*. CRC Press.
- [21] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for nonparametric factorial analyses using only ANOVA procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 143–146.
- [22] 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*. 93–98.