Engagement Effects of Player Rating System-Based Matchmaking for Level Ordering in Human Computation Games

Anurag Sarkar
Northeastern University
sarkar.an@husky.neu.edu

Michael Williams
Northeastern University
williams.mi@husky.neu.edu

Sebastian Deterding
University of York
sebastian.deterding@york.ac.uk

Seth Cooper
Northeastern University
scooper@ccs.neu.edu

ABSTRACT

Human computation games lack established ways of balancing the difficulty of tasks or levels served to players, potentially contributing to their low engagement rates. Traditional player rating systems have been suggested as a potential solution: using them to rate both players and tasks could estimate player skill and task difficulty and fuel player-task matchmaking. However, neither the effect of difficulty balancing on engagement in human computation games nor the use of player rating systems for this purpose has been empirically tested. We therefore examined the engagement effects of using the Glicko-2 player rating system to order tasks in the human computation game *Paradox*. An online experiment (n=294) found that both matchmaking-based and pure difficulty-based ordering of tasks led to significantly more attempted and completed levels than random ordering. Additionally, both matchmaking and random ordering led to significantly more difficult tasks being completed than pure difficulty-based ordering. We conclude that poor balancing contributes to poor engagement in human computation games, and that player rating system-based difficulty rating may be a viable and efficient way of improving both.

CCS CONCEPTS

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

KEYWORDS

human computation games, matchmaking, player rating systems, difficulty balancing, Glicko-2, level ordering

ACM Reference format:
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 FDG'17, Hyannis, MA, USA, August 14-17, 2017,* 10 pages.
DOI: 10.1145/3102071.3102093

FDG'17, Hyannis, MA, USA

DOI: 10.1145/3102071.3102093

1 INTRODUCTION

Human computation games (HCGs), or games with a purpose (GWAPs), are a popular way of motivating large numbers of volunteers to solve tasks that are computationally hard to automate by wrapping them into a game [32, 39, 47]. Uses range from simple image processing tasks such as providing labels (*The ESP Game* [47]) or transcribing text (*Smorball* [40]) to complex optimization tasks like finding protein foldings with desired properties (*Foldit* [13]) or solving graph-theoretic problems (*Pebble It* [15]).

In their attempt to harness the motivational qualities of games, HCGs have found mixed success. The most popular volunteer human computation platforms like *Zooniverse* or *Foldit* can boast participant numbers ranging in the tens, if not hundreds of thousands [4]. Yet closer analysis reveals that the lion's share of crowd work on such platforms—85 percent on *Zooniverse*—is performed by a small minority of "superusers" [42]. The vast majority of volunteers only visit a given HCG for a short duration, never to be seen again [45]. In short, while HCGs have succeeded in motivating large numbers of volunteers to *visit* them, they fail at *engaging and retaining* all but a small minority of visitors.

One suggested reason for this poor engagement is that most HCGs lack *difficulty balancing*, i.e. ensuring that players are faced with a sequence of challenges whose difficulty curve matches their skill growth [12]. Compared to entertainment games, HCGs are severely constrained in this regard: (1) the difficulty of the pool of to-be-solved tasks is unknown in advance; (2) designers cannot freely discard or modify tasks, as this would compromise the validity of generated solution data; and (3) manually identifying the difficulty of tasks would defeat the purpose of cost-efficiently crowdsourcing their solution [12, 34]. Many HCGs therefore end up serving tasks to volunteers either at random or to optimize informational gain [43], not volunteer engagement.

To address this problem, Cooper et al. [12] recently suggested adapting multi-player rating systems like Elo [20], Glicko/Glicko-2 [23, 24], or TrueSkill [26] to estimate level difficulty and player skill in HCGs from user solution data by treating both tasks and users as players. The resulting rating scores could then be fed into a standard multi-player matchmaking algorithm, effectively emulating difficulty balancing. However, to our knowledge, neither the effect of difficulty balancing on HCG player retention nor the effectiveness of player rating-based matchmaking for this purpose has been tested empirically.

In this work, we therefore examined the engagement effects of using the Glicko-2 player rating system to order levels served to

players in the human computation game *Paradox*. We recruited participants through Amazon Mechanical Turk (n=98) to gather play data to generate level difficulty ratings. A follow-up online experiment (n=294) compared engagement in matchmaking ordering with engagement in pure difficulty ordering and random ordering. Both matchmaking and difficulty-based ordering led to significantly more attempted and completed levels than random ordering, while matchmaking and random ordering led to significantly more difficult levels being completed. This supports the importance of difficulty balancing for engagement in HCGs and the feasibility of using player rating and matchmaking systems to estimate the difficulty of, and provide an ordering for, levels in HCGs.

The rest of this paper is organized as follows. Section 2 reviews existing work on difficulty balancing, player rating systems, and level ordering to derive our research questions and hypotheses. Section 3 describes the HCG used, how we trained our rating system on player data, and the setup of the online experiment. Sections 4 and 5 report and discuss results. Section 6 draws conclusions and outlines future work.

2 BACKGROUND

2.1 Engagement and Difficulty Balancing

Engagement is a fundamental construct in the psychology of motivation and digital game play [7, 41]. Broadly, it captures the degree and quality of a person's involvement in a task, i.e. the intensity, persistence, and focus with which they go about it. As a psychological state, engagement is often modelled and operationalized as *behavioral engagement*—how much externally observable effort and persistence a person exhibits in an activity—and *subjective engagement*—the self-reported experience of interest, enjoyment, enthusiasm; absence of distress, anger, anxiety; and resultant proactive, "deep" cognitive problem-solving and learning strategies [41, pp. 12–15]. Engagement is a desirable state and outcome of digital game play of its own, an important concern in people's decision-making to purchase or play particular games and an important mediator of the positive social and individual outcomes of digital games, such as learning or productive outputs.

Theory suggests that games engage players by presenting non-trivial challenges whose pursuit and mastery is arousing, attention-binding, and intrinsically motivating [18]. However, to optimally support engagement, challenges have to be balanced relative to player skill such that players are neither frustrated nor bored [14, 46]. There is good empirical support that balancing game challenges indeed affects behavioral and subjective measures of player engagement [1, 8, 17, 21, 22, 31, 35].

Difficulty balancing of challenges is therefore a key component of entertainment game design, either through pre-release playtesting, designing levels so that they form an ideal difficulty curve matching the skill growth curve of typical players, and/or using dynamic difficulty adjustment systems that adapt the game based on the player's live performance [3, 10].

2.2 Difficulty Balancing in HCGs

As noted, HCGs cannot readily employ standard difficulty balancing because they operate with a pre-determined set of computation tasks (i.e. levels) that are unknown in their difficulty and not readily manipulable [12, 34]. The dominant approach to address these constraints has been to computationally order levels into a sequence that approaches a well-formed difficulty curve [12]. However, systems using this approach so far are based on rough heuristics that are often neither validated nor readily generalizable. The HCG *Xylem*, for example, used task size as a rough (but self-admittedly problematic) heuristic to manually assign difficulty levels to tasks and arrange them in rough order [34]. To estimate the score players could earn by solving a level, the HCG *Binary Fission* similarly used task size [11] as a main heuristic. In both cases, the effect of difficulty balancing on player engagement has not been rigorously validated.

Addressing a similar problem in educational games, Butler et al. [6] presented a system that automatically identifies the *solution features* of levels in an educational game—roughly, the required operations to solve a level. Based on this information and data on a player's previous success at solving levels with certain solution features, the system dynamically selected levels from a pre-existing pool in an order that grows in difficulty with the player's skill. Butler et al. found that players in the automatically generated level ordering engaged with the game for comparably long times as with a game design expert-produced level ordering. However, they did not test how play times compared to a random ordering control condition, and their approach required being able to solve all levels computationally *a priori*, which is, by definition, not the case in HCGs. Additionally, the solution features were highly tailored to the specific game used. Relatedly, Liu et al. [33] developed an educational game that adaptively creates new levels for players based on performance on preceding levels. However, while this may be a useful approach for tutorial sections in an HCG, creating new levels or tasks defeats the purpose of HCGs to solve the pre-given ones.

2.3 Player Rating Systems and Applications

In short, difficulty balancing for HCGs ideally requires a system capable of selecting and sequencing tasks/levels from a task pool in an order that matches the skill growth of individual players, with the constraints that (a) neither task difficulty nor player skill is known in advance and (b) computationally generating or solving tasks to determine their difficulty would defeat the purpose of HCGs. Existing systems have tackled this problem with rough and not easily generalizable heuristics such as using game-specific measures of task size as a stand-in for difficulty. Recent work has suggested an approach that promises to satisfy the stated requirements and constraints while being readily generalizable: matchmaking based on player rating systems such as Elo, Glicko, or TrueSkill [12].

To rate and rank competitive chess players, Arpad Elo [20] created a mathematical formula to calculate their relative competitive strength. Elo postulated that the outcome of a match between two players can be considered the outcome of a pairwise comparison. He further assumed that a player's performance in a given match is a normally distributed random variable, where the mean of all those performances is the player's "true" skill. Elo uses the Bradley-Terry model [5] to predict the likelihood of one player defeating the other based on the rating of both. This information then also feeds an

update algorithm modifying both players' rating: each player wins or loses a commensurate amount of rating depending on predicted versus actual match outcome.

Glickman expanded the traditional Elo system into Glicko [23], adding "ratings deviation" as a measure of rating reliability—effectively a standard deviation measuring the uncertainty of the rating: the more data available on a player, the more reliable the prediction, and the lower the ratings deviation. Glicko-2 [24] additionally incorporates a parameter of volatility, expressing how stable versus volatile a player's performance is from match to match.

TrueSkill, developed at Microsoft Research and notably used for player matchmaking on Xbox Live, additionally models team performance as the aggregate of the individual performance parameters on a given team [26]. One advantage of this system is that it can accommodate any number of teams and players. It performs well in free for all, team-based, and 1-versus-1 games such as chess. Together, Elo, Glicko, TrueSkill, and their derivatives are widely used today in the entertainment games industry to rate player skills and provide enjoyable matchmaking in player-versus-player games [37].

Cooper et al. [12] suggested that such player rating and matchmaking systems could be used to emulate difficulty balancing in HCGs. By equating levels with players, prior task solutions with matches, and level difficulty with player skill, player rating systems can readily calculate skill and difficulty ratings of players and levels from past task solutions, which is plausible since HCGs usually accumulate multiple solutions for each task to validate and/or optimize solutions. Reanalyzing existing game data, Cooper et al. demonstrated that the bipartiteness of the solution graph—levels are never directly compared with levels, players never with players—does not harm the quality of resultant rankings. However, they did not empirically test whether resulting level sequences empirically improved player engagement. The platform game *Jumpcraft* similarly orders user-generated levels by using TrueSkill on the outcome of player attempts [44]. Yet again, there is no data on the actual engagement effect of this ordering system.

2.4 Research Questions and Hypotheses

To summarize, we lack ready difficulty balancing systems for HCGs; player rating systems such as Elo have been suggested as a possible solution; yet we don't know empirically whether difficulty balancing actually makes a difference in HCG engagement, and if so, whether player rating systems are an effective means of achieving it. Hence, we articulated the following research questions:

- *RQ1*: How does difficulty balancing affect player engagement in HCGs?
- *RQ2*: How does player rating-system based player-level matchmaking affect player engagement in HCGs?

Since it is a more robust and immediately relevant outcome, we decided to focus on behavioral engagement. In order to capture both (a) amount and (b) "quality" of behavioral engagement, we operationalized it as (a) time spent playing, number of levels attempted and completed and (b) the highest difficulty level completed per player as well as aggregate difficulty of all levels completed. We thus posed the following hypotheses:

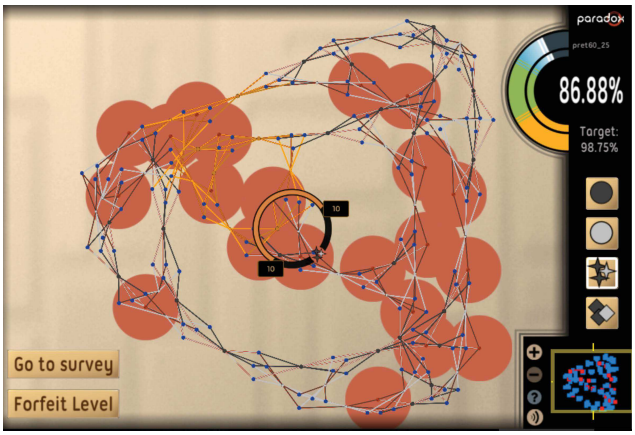

Figure 1: A screenshot of the version of *Paradox* used in this work. Using the buttons in the bottom left, the player has the option to either forfeit the current level and proceed to the next one, or bypass all remaining levels and proceed to the survey.

- *H1*: Serving levels in increasing difficulty will lead to higher behavioral engagement than serving levels in random order, as measured by the time spent playing (H1.a), number of levels attempted (H1.b), number of levels completed (H1.c), and the most difficult level completed (H1.d) by each player, as well as the aggregate difficulty of levels completed by all players (H1.e).
- *H2*: Serving levels in order of matchmaking difficulty will lead to higher behavioral engagement than serving levels in random or increasing difficulty order, as measured by the time spent playing (H2.a), number of levels attempted (H2.b), number of levels completed (H2.c), and the most difficult level completed (H2.d) by each player, as well as the aggregate difficulty of levels completed by all players (H2.e).

3 METHOD AND SYSTEM

3.1 Game Description

We decided to test our hypothesis with the HCG *Paradox* [16]. *Paradox* was originally developed for crowd-sourced formal verification of software, in which players would assist in producing proofs of correctness for computer programs. The game is designed as a 2D puzzle game in which each level represents a maximum satisfiability (MAX-SAT) problem. Players can use a combination of manual and automated tools to assign values to variables in the underlying MAX-SAT problem and are scored based on how many clauses they satisfy. A player "completes" a level by reaching a pre-determined target score. Some levels in the game are not fully solvable, and in general we may not know if levels are fully solvable or not (i.e. if all clauses can be satisfied). Still, even partial solutions can potentially be useful. A screenshot of the version used in this work is shown in Figure 1.

The *Paradox* levels used in this work were generated using satisfiability problems encoded in the DIMACS file format. We manually assembled a pool of 33 levels that spanned a variety of underlying

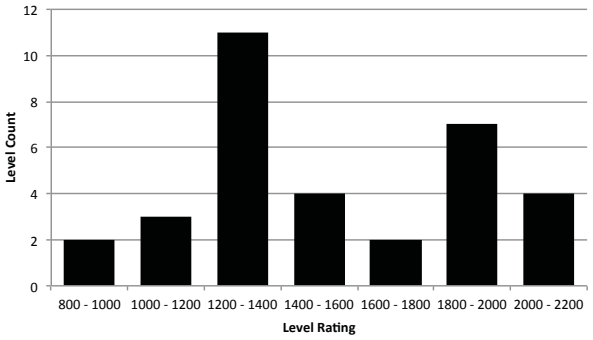

Figure 2: Histogram of the ratings of the pool of 33 challenge levels used.

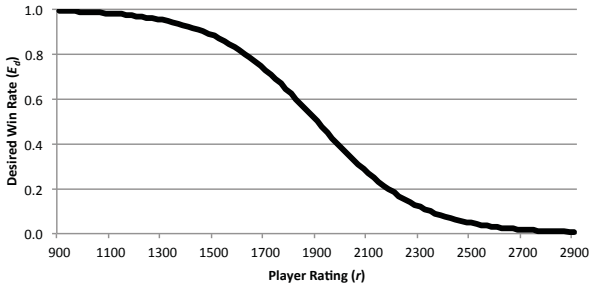

Figure 3: A plot of the "difficulty curve" used to choose a player's desired win rate based on their current rating.

problem types, connectivities, sizes, and satisfiabilities. Of these, 21 came from the set of SATLIB Benchmark Problems[1] and 12 were generated by us using a variety of randomized algorithms for SAT problem generation, such as using power law distributions [2].

3.2 Participant Recruitment

We recruited players by posting Human Intelligence Tasks (HITs) on Amazon Mechanical Turk (MTurk). Although workers recruited through MTurk may exhibit different behaviors from leisurely entertainment game players or volunteer HCG players, a growing body of work has shown that workers are also motivated by enjoyment rather than payment alone [28, 36], and that MTurk workers make decisions similarly to traditional subject pools [38]. Furthermore, MTurk has been successfully used to recruit players for evaluating game designs [29]. We posted HITs with the following details. The HIT title was:

> Human Computation Puzzle Game

The HIT description was:

> Play a puzzle game derived from a real-world problem. You would need Adobe Flash Player 10.0 or greater to proceed.

The HIT keywords were:

> survey, game, play, puzzle

Workers were paid $1.50, told that the expected time to complete the HIT was 30 minutes, and provided with the following instructions:

> There are three stages to the HIT:
> 1. Play and complete all the tutorial levels.
> 2. Try to complete as many challenge levels as you can!
> 3. Go to the survey and complete the survey.
>
> After completing the survey, you will be given the completion code. Some challenge levels may not be possible to complete. It is NOT necessary to complete all challenge levels and your submission will be approved as long as you complete the survey.

The instructions explicitly reassured players that they would be paid regardless of how many challenge levels they completed, motivated by normalizing the differing beliefs they may have had about how much they needed to do to prevent rejection of their payment, similar to the "guaranteed payment" used on MTurk by Ho et al. [27].

For our HIT, we required the players to complete nine tutorial levels to familiarize them with the mechanics of the game. Player performance in these tutorial levels was not considered for further analysis. After completing the tutorial levels, players would then move on to the challenge levels, selected from the pool of 33 described above. For challenge levels, players had access to a button that allowed them to proceed to the next level without completing the current one. This button initially said "Skip Level", but if the player attempted the level by making a move, the button text changed to "Forfeit Level". In either case, the player proceeded to the next level without completing the current one, but the outcomes were considered different (see below). After a player had skipped or forfeited three levels (either consecutively or non-consecutively), an additional button saying "Go to survey" appeared to allow the player to bypass all the remaining levels and finish the HIT. Thus, every level a player saw had three possible outcomes—*complete*, *forfeit* or *skip*. With this setup, the players were playing voluntarily and could essentially quit at any time once they reached the challenge levels—even without completing any levels—but they had to see at least four levels (skip or forfeit three levels and then go to the survey from any level after that). Once a player had seen a level, it was taken out of the pool of levels they might get next. Thus, during a single playthrough of all 33 challenge levels, a player saw each level no more than once. Although no individual player saw all the challenge levels in a HIT, if this were to happen, the levels would have been recycled and served to the player using the same ordering mechanism used for that player for the first playthrough. After finishing, players completed a short survey about their experience to assist in further development of the game, at which point the HIT was completed.

3.3 Initial Level Rating Generation

The initial set of ratings for the levels in our pool was generated using player-versus-level match data from a HIT. The HIT was completed by 98 players, each of whom was served the levels in either random order or in order of increasing size of the levels (i.e. the number of nodes in the constraint graph corresponding to the level). Players were randomly assigned one of these two level orderings, resulting in 45 receiving the levels in random order and

[1] http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html

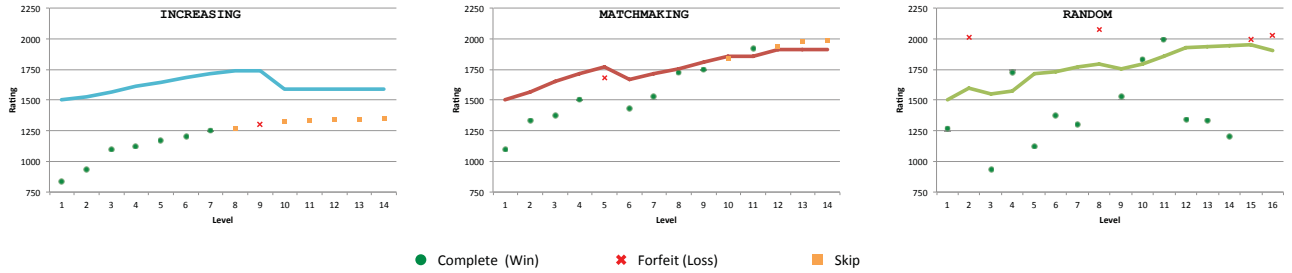

Figure 4: Example player trajectories through ratings for each condition. The player's rating is shown as a line and the ratings of the levels that were served are shown as points. The levels up to and including the level where the player finished and proceeded to the survey are shown.

the remaining 53 players being served the levels in increasing size order.

For the purposes of match outcomes, we considered a player completing a level as a win for the player and a player forfeiting a level (making at least one move and then not completing the level) as a loss for the player. If a player skipped a level without making any moves, it was ignored for the purposes of match outcomes. We chose Glicko-2 as a rating system as it outperformed Elo and performed on par with TrueSkill in predicting HCG player-level matching outcomes [12]. We started players and levels with the default Glicko-2 parameters: a rating of 1500, a rating deviation of 350, and a rating volatility of 0.06.

To generate ratings for each level, we treated each instance of a player seeing a new level as a match. We considered match outcomes as described above. We assigned players and levels the default Glicko-2 rating parameters and played back the match data for all player-level pairings obtained from the HIT using the Glicko-2 rating system in order to generate our initial ratings for each level. For this playback, we used the pyglicko2 Python module [30]. Resultant level ratings spanned from 831 to 2077 on the Glicko-2 rating scale; a histogram of the level ratings is shown in Figure 2.

3.4 Ordering Experiment

To test our hypotheses, we ran another HIT on MTurk. In addition to the HIT setup described above, we included the following line in the instructions to emphasize that players needed to complete all of the tutorial levels and had to skip at least 3 levels before being able to skip to the end:

> You MUST complete all the tutorial levels. The survey will not be accessible during the tutorial and will become available once you fail to complete at least 3 challenge levels.

For this experiment, we had three different conditions which differed only in the order in which levels were served to players. Our conditions were: serving levels in random order (`RANDOM`), serving levels in increasing order of difficulty (`INCREASING`), and using rating-based matchmaking to order levels (`MATCHMAKING`). Players were randomly assigned one of these three conditions.

In random order (`RANDOM`), players were served levels in a random sequence without any consideration of level ratings. In increasing order (`INCREASING`), we simply served the levels in ascending order of the level ratings generated in the previous HIT. This was a pure difficulty-based ordering that was static for all players; each player saw the exact same sequence of levels, regardless of their performance within the game.

In matchmaking order (`MATCHMAKING`), players were served levels based on their in-game performance prior to each new level. This ordering was motivated by the goal of matching up the players with levels that were appropriate for their skill. To implement this order, we performed matchmaking based on the Glicko-2 ratings of players and levels. When starting, players and levels were assigned the default Glicko-2 rating parameters, except that levels were assigned their ratings from the initial rating generation step as described previously. Prior to serving each level, we determined a desired win probability for the player based on the player's current rating using the function

$$E_d(r) = 1 - 1/(1 + e^{-k(r-r_0)}) \quad (1)$$

where r is the player's current rating, constants k and r_0 are set to 0.005 and 1900 respectively and the output $E_d(r)$ is the desired win probability for a player with a Glicko-2 rating of r. This is based on the logistic curve shown in Figure 3. Once we determined the desired win probability, we calculated the player's expected win probability against each level using the simplified winning expectancy formula

$$E_p(r, v) = 1/(1 + 10^{(v-r)/400}) \quad (2)$$

discussed by Glickman and Jones [25], where r is the player's current rating and v is the rating of the level. We then selected the levels for which the player's expected win probability was within a window ±5% of the desired win probability computed previously. From among these levels, we chose a level uniformly at random to serve to the player. In the event that no levels satisfied the above criteria, we kept increasing the size of the window around the desired win probability by increments of 5% in both directions until we found levels that did. A level was found within ±5% of the desired win probability (the initial window) for 72% of matches and ±20% for 97% of matches. After each match, we updated the Glicko-2 rating parameters of both the player and the level depending on the outcome, as defined previously, and then repeated the process for the subsequent levels until the player finished the game to go to the survey. This process was run independently for each player so, although we updated the rating parameters of the level, it did not impact that level for other players. Additionally, since each level

Variable	Omnibus	MATCHMAKING / INCREASING	INCREASING / RANDOM	RANDOM / MATCHMAKING
Challenge Time (s)*	$n.s.$, $H(2) = 1.62$	395 / 329	329 / 386	386 / 395
*Levels Attempted**	$p < .001$, $H(2) = 14.91$	7 / 7 $n.s.$, $U = 3869$	7 / 4 $p < .001$, $U = 4143$ $r_{rb} = 0.28$	4 / 7 $p = .003$, $U = 3441$ $r_{rb} = 0.25$
*Levels Completed**	$p < .001$, $H(2) = 45.80$	5 / 6 $n.s.$, $U = 3536$	6 / 2 $p < .001$, $U = 2911.5$ $r_{rb} = 0.49$	2 / 5 $p < .001$, $U = 2672$ $r_{rb} = 0.42$
*Highest Rating***	$p < .001$, $H(2) = 55.67$	1431 / 1249 $p < .001$, $U = 1631$ $r_{rb} = 0.52$	1249 / 1431 $p < .001$, $U = 1436$ $r_{rb} = 0.60$	1431 / 1431 $n.s.$, $U = 2581$
Per-level Rating†	$p < .001$, $H(2) = 224.41$	1328 / 1171 $p < .001$, $U = 88440$ $r_{rb} = 0.45$	1171 / 1328 $p < .001$, $U = 84872$ $r_{rb} = 0.43$	1328 / 1328 $n.s.$, $U = 102830$

Table 1: Summary table of variable analysis. Variables analyzed using *all players, **players who completed at least one level, and †all completed levels. Shaded cells show significant post-hoc comparisons. Medians are given.

was seen only once per playthrough, such updating had no effect on the sequence of levels the players received in the future.

As is evident from the curve in Figure 3, the desired win probability decreases as the player's rating increases. Thus, when the players have a lower rating, the higher desired win probability will lead to them being matched with levels against which they have a higher expected probability of winning, starting players off with easier levels. As the players' ratings increase, the drop in the desired win probability will lead to them being matched with levels for which they have lower expected win probabilities, i.e. harder levels. In this way, the desired win probability function serves as a kind of "difficulty curve" that shapes the change in difficulty of levels a player faces. The exact form of the curve we used was set heuristically, and is a potential area for future work. In early pilot tests, we found that a fixed desired win probability rate could result in players consistently facing easy or hard levels, thus making it difficult for them to raise their rating.

Selected example trajectories of players through each of the ordering conditions are shown in Figure 4.

4 RESULTS

The ordering HIT was initially accepted by 393 players. Of these, 294 (75%) completed the HIT. In the context of existing MTurk research [9, 19, 38], we consider this dropout rate normal. We randomly assigned 79 players into MATCHMAKING, 99 into INCREASING, and 116 into RANDOM. A chi-squared test did not find that completion of the HIT varied significantly by condition. Thus, for our analysis, we only considered data from those players who completed the HIT. For each of those players, we examined the following variables:

- *Challenge Time*: The total time spent by a player in the levels, in seconds.
- *Levels Attempted*: The number of levels *attempted* by a player, where they made at least one move.
- *Levels Completed*: The number of levels *completed* by a player, where they reached the target score.

For each of the players who completed at least one level (n=244), we examined:

- *Highest Rating*: The highest rating of any level completed by a player.

Additionally, for each of the completed levels (n=1591), we also analyzed:

- *Per-level Rating*: The rating of each completed level. This gives an indication of the aggregate difficulty of all levels completed in each condition.

As data were not normally distributed, we used non-parametric tests for our analysis. We first performed an omnibus Kruskal-Wallis test to check for differences among all conditions. If found, we then performed three post-hoc Wilcoxon Rank-Sum tests to check for pairwise differences between conditions. For significant pairwise differences, we computed the effect size using rank-biserial correlation (r_{rb}). A summary of all comparisons is shown in Table 1, and plots of variables with significant differences are shown in Figure 5. Histograms of completed level ratings by condition are given in Figure 6.

For both *Levels Attempted* and *Levels Completed*, we found significant differences among all conditions. We found no pairwise difference between MATCHMAKING and INCREASING, but MATCHMAKING and INCREASING both significantly outperformed RANDOM. Further detail on the progress of players at completing levels in each condition is given in Figure 7.

We found no significant differences among the conditions for *Challenge Time*.

For *Highest Rating* and *Per-level Rating*, we found a significant difference among all conditions. We found no pairwise difference between MATCHMAKING and RANDOM, but a difference between MATCHMAKING and INCREASING as well as RANDOM and INCREASING, such that MATCHMAKING and RANDOM outperformed INCREASING.

5 DISCUSSION

Regarding our original hypotheses, we conclude that *H1 is partially supported*. For all quantity measures of behavioral engagement but time spent, we observed significantly higher measures when serving levels in increasing rather than random order (supporting hypothesis H1.b and H1.c, but not H1.a), However, when it comes

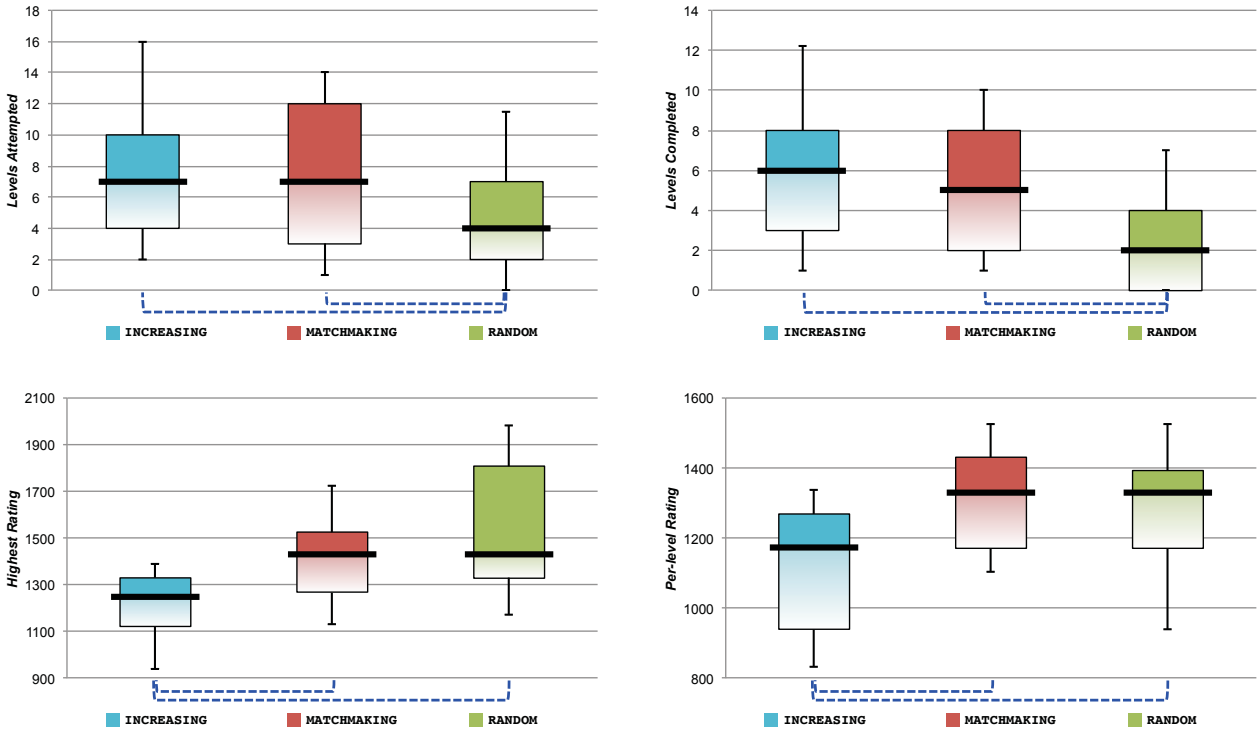

Figure 5: Summary plots from analysis of variables. Box-and-whisker plots showing the 10th, 25th, 50th (median), 75th, and 90th percentiles of variables with dashed lines noting significant pairwise comparisons.

to quality measures, we found that players, both individually and in aggregate, completed levels with significantly higher difficulty when levels were served in random order as compared to increasing order (rejecting H1.d and H1.e).

This finding sheds interesting light on Butler et al.'s [6] observation that automatic and expert difficulty ordering of levels result in comparable play times: as in our case, it might be that playtime was chiefly affected by factors beyond level ordering and is hence not a good measure of player engagement. For instance, it may be that regardless of level difficulty, players on MTurk have norms of how much time they feel they "ought" to spend on a HIT, or that particular games hold interest for a certain amount of time regardless of the level ordering. A more reliable and relevant time measure (requiring different experimental setups) would be the number and duration of re-engagements with the HCG.

But how do we explain that RANDOM order produced higher, not lower, individual and aggregate level difficulty completed than INCREASING? While not conclusive, the example player trajectories in Figure 4 and completed level ratings in Figure 6 provide an interesting entry point. Given that players in all conditions spent roughly the same time playing the game, could tackle or skip levels as they liked, and would on average attempt 4-7 levels in total, players in RANDOM ordering ended up being served and completing far more difficult (higher-rated) levels than players in INCREASING ordering. As seen in Figure 4, when players in RANDOM ordering were served a level they found too difficult to tackle, they would just skip to the next one until they reached the required minimum of 4 levels served. In comparison, players in INCREASING ordering were likely to tackle (and complete) each level they were served, which due to the strict increasing ordering of difficulty meant that they on average spent the majority of their time tackling and completing very low-difficulty tasks. In other words, while INCREASING ordering did engage players to attempt and complete more levels, RANDOM ordering made it more likely that players would encounter still-solvable levels of much higher difficulty during playtime than players in INCREASING ordering.

Moving on, *H2 has to be rejected*. We did not observe a significantly higher quantity or quality of difficulty in MATCHMAKING ordering compared to both INCREASING or RANDOM ordering (rejecting H2.a–H2.e). More specifically, MATCHMAKING did outperform RANDOM ordering on levels attempted and completed, but only on par with INCREASING, not surpassing it. From an HCG perspective, this reinforces that ordering levels by difficulty may result in *a larger quantity of work* being completed by players. Difficulty and matchmaking ordering did not necessarily keep players around longer, but players used the time they spent playing more efficiently in terms of sheer quantity of work. Notably, the level rating information generated from the initial data gathering HIT was already useful for engagement (INCREASING), even in the absence of using that information for later matchmaking ordering (MATCHMAKING).

This actually further strengthens the plausibility of using player rating systems for difficulty balancing in HCGs: as most HCG players are novice first time users and most levels are played by a few super-users, they usually feature an imbalance of rich level

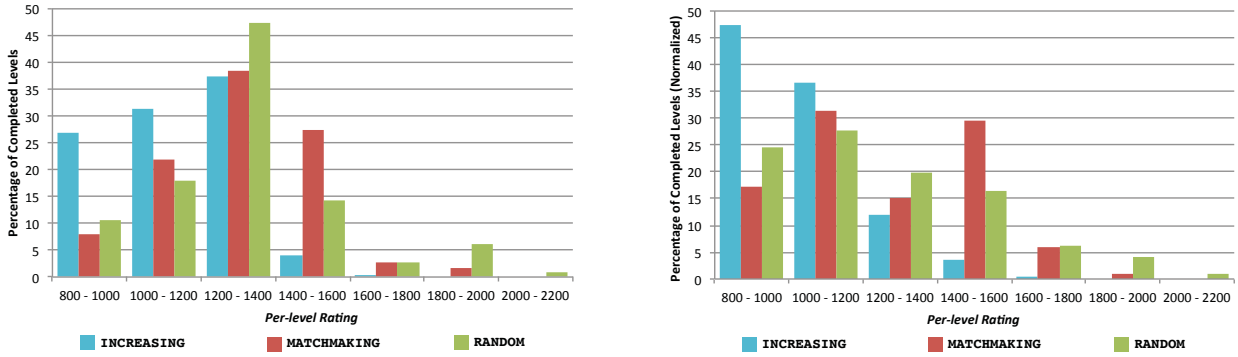

Figure 6: Histogram of the distribution of ratings of all completed levels for each condition. The x-axis shows bins of level ratings and the y-axis shows the percentage of completed levels that fell into each bin (left) unnormalized and (right) normalized by the number of levels in our pool in each bin (Figure 2). In MATCHMAKING and RANDOM, players reached and completed more higher-rated levels than in INCREASING.

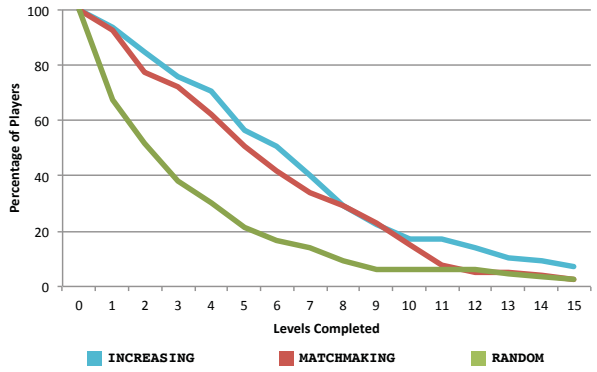

Figure 7: Chart of player progress in completing levels for each condition, up to the first 15 levels completed. The x-axis shows a count of completed levels and the y-axis shows the percentage of players who completed at least that many levels. MATCHMAKING and INCREASING are similar, while the falloff is much more rapid for RANDOM.

and sparse player rating information [12, 42]. If difficulty ordering based on level rating (against highly reliable super-users) is enough to increase engagement, this means that the lack of rating data on first time users is actually no limitation.

Qualifying our rejection of H2.d and H2.e, MATCHMAKING ordering significantly outperformed INCREASING ordering in terms of the difficulty of levels completed, both per player and in aggregate. However, it did not outperform RANDOM ordering. As before, a plausible explanation for this observation is that given roughly the same playtime and number of levels served, MATCHMAKING ordering was faster in serving players more difficult yet completable levels than INCREASING ordering, leading to players tackling and completing overall more difficult levels (see again Figures 4 and 6). Put differently, while both MATCHMAKING and INCREASING ordering were equally *effective* in engaging players to do more, MATCHMAKING ordering was more *efficient* in fully harnessing the players' skill with more difficult tasks faster. This interestingly contrasts with the work of Lomas et al. [35], who found that serving players very easy levels was most engaging.

From an HCG perspective, this makes MATCHMAKING ordering more attractive: as HCGs have a range of problems to solve, it is important to put player effort towards solving the difficult ones. INCREASING order serves players the easiest possible levels first, thus not making the best use of their work. MATCHMAKING ordering gives players more difficult levels, yet players are still able to complete a similar number of them.

Thus, overall, MATCHMAKING ordering compared favorably to both other orders, though in different measures. In terms of pure quantity of work, it outperforms RANDOM ordering and performs on par with INCREASING ordering. Hence, if task volume alone is sought after in an HCG, INCREASING may be the more efficient choice. In terms of quality or difficulty of work, MATCHMAKING ordering outperforms INCREASING ordering and performs on par with RANDOM ordering. Matchmaking may thus provide a "best of both worlds" approach that leads to players completing a larger number of levels of higher difficulty than would be possible if levels were served randomly or in order of increasing difficulty. When using matchmaking order, on average, players completed over twice as many levels as compared to random order, and the most difficult levels they completed had Glicko-2 ratings of nearly 200 more than the ratings of the hardest levels completed in increasing order.

In terms of limitations, our experiment had a relatively small pool of levels to serve, which turned out to nicely cover a range of ratings. It is likely that for many HCGs, the pool of levels would be much larger, potentially encompassing hundreds, if not thousands, of levels. Depending on the distribution of level ratings, serving such a pool in strictly increasing order might result in incoming players being mired in overly simple levels, while serving the levels randomly may result in many extremely difficult levels in a row. We would expect that matchmaking might help even more in these cases.

Player behavior in our experiment may also be impacted by the fact that the players were recruited as paid workers through MTurk. Though we believe previous work indicates that this is a reasonable recruitment method for testing a game, a comparison to

unpaid play would be useful: MTurk workers might have personal norms or limits on how much time they spend on each HIT worth a certain amount, while volunteers are capped in their playtime only in their willingness to contribute. Additionally, as in our experiment, most HCG engagements are one-time at the moment [42, 45]. Still, the HCG engagement ideal would be to not just get more (and more difficult) contributions from one-time volunteers, but to motivate them to re-engage and return to an HCG multiple times. Also, the majority of volunteer work on HCGs is done by super-users. From our results, we cannot say whether difficulty balancing in general, and matchmaking in particular, would improve engagement across multiple re-engagements or for said super-users. Though we believe that super-users wouldn't skew results in real use, as a matchmaking system would serve them levels of matching difficulty, confirming this is future work.

While we think that a total system in use should maximize engagement only and insofar as it increases total information gain per user, in this work, we focused on determining if difficulty balancing increases engagement. Comparing total information gain between our own and information-gain oriented ordering schemes such as [43] is certainly fertile ground for future work.

6 CONCLUSION

This work has explored the engagement effects of player rating-based matchmaking for level ordering in the human computation game *Paradox*. We found that using a matchmaking-based ordering for serving levels led to players attempting and completing a significantly greater number of levels than when serving levels in random order, though on par with serving levels in order of increasing difficulty. We also found that matchmaking-based ordering led players to complete levels of significantly greater difficulty than ordering levels in increasing difficulty, though on par with random ordering. Put differently, both strict increasing difficulty and matchmaking ordering engages players to do more work, but matchmaking engages them to do more difficult work. This is likely due to the fact that strict difficulty ordering spends the majority of play time on low-difficulty levels, while matchmaking and random ordering expose players faster to more difficult levels. Both increasing difficulty and matchmaking ordering outperformed random ordering on the number of levels attempted and completed, supporting that difficulty balancing in general has a significant impact on player engagement in HCGs.

Future work could explore the applicability of matchmaking-based level ordering in other HCGs similar to *Paradox* or entertainment games in general. Matchmaking-based level ordering appears particularly attractive for games with procedurally generated or user-generated levels, which can have constraints very similar to HCGs: a large pool of levels which are not readily manipulable and have unknown difficulties.

Additionally, as described previously, we performed level selection using the player's expected win probabilities against the levels, as determined by Equation 2. Though this led to meaningful data, we would still like to test the accuracy of these probabilities more thoroughly in future work, attempting to determine empirically how close the true win probability of a player rated r against a level rated v is to the expected win probability.

In a similar vein, although our method for serving levels in matchmaking order produced useful results, our choice of 0.005 and 1900 as values for the parameters k and r_0 respectively in the desired win rate function given by Equation 1 was driven mainly by intuition and the shape we expected of the "difficulty curve" given in Figure 3. In the future, we would like to make use of optimization techniques to determine and use improved values for these parameters.

Moreover, although we assigned ratings to players and levels, this information was opaque to players, who were simply served levels without knowing how their performance would impact their rating—or even that they had a rating. Exposing information such as the player's current rating or estimated difficulty of levels may be useful for players; giving players control over the next level they attempt and thus manage the difficulty of the game, which may further improve engagement.

This work required two MTurk trials in order to implement the matchmaking-based level ordering—one to generate the initial level ratings, followed by another to determine the ratings for the players. Ideally, we would like to build a more online system that implements the desired level ordering in one pass, i.e. a system that generates ratings for unrated levels as well as unrated players while the players play through the levels, instead of having to run separate trials for each. One issue is that of fully unplayed levels: player ratings can be one signal in a larger system where player solution data trains a system to learn to predict the initial ratings of such unplayed levels.

Finally, addressing our limitations, replications of this study with HCG "super-users", HCG volunteers, and a setup capturing multiple re-engagements with the HCG rather than a one-time session would be useful.

ACKNOWLEDGEMENTS

This work was supported by a Northeastern University TIER 1 grant and partly conducted in the Digital Creativity Labs (digitalcreativity.ac.uk), jointly funded by EPSRC/AHRC/InnovateUK under grant no. EP/M023265/1. This material is based upon work supported by the National Science Foundation under grant no. 1652537. We would like to thank the University of Washington's Center for Game Science for initial *Paradox* development.

REFERENCES

[1] Justin T. Alexander, John Sear, and Andreas Oikonomou. 2013. An investigation of the effects of game difficulty on player enjoyment. *Entertainment Computing* 4, 1 (Feb. 2013), 53–62.

[2] Carlos Anstegui, Mara Luisa Bonet, Jordi Levy, and Chu Min Li. 2012. Analysis and generation of pseudo-industrial MaxSAT instances. In *Proceeding of the 15th International Conference of the ACIA*. 173–184.

[3] Alexander Baldwin, Daniel Johnson, Peta Wyeth, and Penny Sweetser. 2013. A framework of Dynamic Difficulty Adjustment in competitive multiplayer video games. In *Proceedings of the 2013 IEEE International Games Innovation Conference*. 16–19.

[4] Manda Banerji, Ofer Lahav, Chris J. Lintott, Filipe B. Abdalla, Kevin Schawinski, Steven P. Bamford, Dan Andreescu, Phil Murray, M. Jordan Raddick, Anze Slosar, Alex Szalay, Daniel Thomas, and Jan Vandenberg. 2010. Galaxy Zoo: reproducing galaxy morphologies via machine learning. *Monthly Notices of the Royal Astronomical Society* 406, 1 (2010), 342–353.

[5] Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* 39, 3/4 (1952), 324–345.

[6] Eric Butler, Erik Andersen, Adam M. Smith, Sumit Gulwani, and Zoran Popovifj. 2015. Automatic game progression design through analysis of solution features.

In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, Seoul, Republic of Korea, 2407–2416.

[7] Paul Cairns. 2016. Engagement in digital games. In *Why Engagement Matters: Cross-Disciplinary Perspectives of User Engagement in Digital Media*, Heather O'Brien and Paul Cairns (Eds.). Springer International Publishing, 81–104.

[8] Jared E. Cechanowicz, Carl Gutwin, Scott Bateman, Regan Mandryk, and Ian Stavness. 2014. Improving player balancing in racing games. In *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-human Interaction in Play (CHI PLAY '14)*. ACM, New York, NY, USA, 47–56.

[9] Dana Chandler and Adam Kapelner. 2013. Breaking monotony with meaning: motivation in crowdsourcing markets. *Journal of Economic Behavior & Organization* 90 (June 2013), 123–133.

[10] David Michael Jordan Chang. 2013. Dynamic difficulty adjustment in computer games. In *Proceedings of the 11th Annual Interactive Multimedia Systems Conference*.

[11] Kate Compton, Heather Logas, Joseph C. Osborn, Chandranil Chakrabortti, Kelsey Coffman, Daniel Fava, Dylan Lederle-Ensign, Zhongpeng Lin, Jo Mazeika, Afshin Mobramaein, Johnathan Pagnutti, Husacar Sanchez, Jim Whitehead, and Brenda Laurel. 2016. Design lessons from Binary Fission: a crowd sourced game for precondition discovery. In *Proceedings of the 1st International Joint Conference of DiGRA and FDG*.

[12] 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*.

[13] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popovifj, and Foldit Players. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (Aug. 2010), 756–760.

[14] Mihaly Csikszentmihalyi. 1990. *Flow: the psychology of optimal experience*. Harper and Row, New York.

[15] Charles Cusack, Jeff Largent, Ryan Alfuth, and Kimberly Klask. 2010. Online games as social-computational systems for solving NP-complete problems. *Meaningful Play* (2010).

[16] 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.

[17] Alena Denisova and Paul Cairns. 2015. Adaptation in digital games: the effect of challenge adjustment on player performance and experience. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '15)*. ACM, New York, NY, USA, 97–101.

[18] Sebastian Deterding. 2015. The lens of intrinsic skill atoms: a method for gameful design. *Human-Computer Interaction* 30, 3-4 (2015), 294–335.

[19] Steven Dow, Anand Kulkarni, Scott Klemmer, and Bjrn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 1013–1022.

[20] Arpad E. Elo. 1978. *The rating of chessplayers, past and present*. Arco.

[21] Stefan Engeser and Falko Rheinberg. 2008. Flow, performance and moderators of challenge-skill balance. *Motivation and Emotion* 32, 3 (Sept. 2008), 158–172.

[22] Clive J. Fullagar, Patrick A. Knight, and Heather S. Sovern. 2013. Challenge/skill balance, flow, and performance anxiety. *Applied Psychology* 62, 2 (April 2013), 236–259.

[23] 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.

[24] Mark E. Glickman. 2001. Dynamic paired comparison models with stochastic variances. *Journal of Applied Statistics* 28, 6 (Aug. 2001), 673–689.

[25] Mark E. Glickman and Albyn C. Jones. 1999. Rating the chess rating system. *Chance* 12 (Jan. 1999), 21–28.

[26] Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. TrueSkill(TM): a Bayesian skill rating system. In *Advances in Neural Information Processing Systems 20*. MIT Press, 569–576.

[27] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. 2015. Incentivizing high quality crowdwork. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. ACM, New York, NY, USA, 419–429.

[28] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. 2011. More than fun and money. Worker motivation in crowdsourcing - a study on Mechanical Turk. In *Proceedings of the Americas Conference on Information Systems*.

[29] Mohammad M. Khajah, Brett D. Roads, Robert V. Lindsey, Yun-En Liu, and Michael C. Mozer. 2016. Designing engaging games using Bayesian optimization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5571–5582.

[30] Ryan Kirkman. 2010. pyglicko2: a Python Implementation of the Glicko-2 algorithm. (2010).

[31] Christoph Klimmt, Tilo Hartmann, and Andreas Frey. 2007. Effectance and control as determinants of video game enjoyment. *Cyberpsychology & Behavior* 10, 6 (Dec. 2007), 845–847.

[32] Edith Law and Luis von Ahn. 2011. *Human Computation*. Morgan & Claypool.

[33] Yun-En Liu, Christy Ballweber, Eleanor O'rourke, Eric Butler, Phonraphee Thummaphan, and Zoran Popovifj. 2015. Large-scale educational campaigns. *ACM Transactions on Computer-Human Interaction* 22, 2 (March 2015), 8:1–8:24.

[34] Heather Logas, Jim Whitehead, Michael Mateas, Richard Vallejos, Lauren Scott, Dan Shapiro, John Murray, Kate Compton, Joseph Osborn, Orlando Salvatore, Zhongpeng Lin, Huascar Sanchez, Michael Shavlovsky, Daniel Cetina, Shayne Clementi, and Chris Lewis. 2014. Software verification games: designing Xylem, The Code of Plants. In *Proceedings of the 9th International Conference on the Foundations of Digital Games*.

[35] Derek Lomas, Kishan Patel, Jodi L. Forlizzi, and Kenneth R. 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 (CHI '13)*. ACM, Paris, France, 89–98.

[36] Winter Mason and Duncan J. Watts. 2009. Financial incentives and the "performance of crowds". In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '09)*. ACM, Paris, France, 77–85.

[37] Josh Menke. 2016. Skill, matchmaking, and ranking systems design. Game Developers Conference. (March 2016).

[38] Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis. 2010. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making* 5, 5 (June 2010), 411–419.

[39] Ei Pa Pa Pe-Than, Dion Hoe-Lian Goh, and Chei Sian Lee. 2012. A survey and typology of human computation games. In *Proceedings of the 9th International Conference on Information Technology: New Generations*. 720–725.

[40] Patrick Randall. 2016. Purposeful gaming and the Biodiversity Heritage Library. *Journal of Agricultural & Food Information* 17, 1 (Jan. 2016), 71–76.

[41] Johnmarshall Reeve. 2015. *Understanding Motivation and Emotion (Sixth edition)*. Wiley, Hoboken, New Jersey.

[42] Henry Sauermann and Chiara Franzoni. 2015. Crowd science user contribution patterns and their implications. *Proceedings of the National Academy of Sciences* 112, 3 (Jan. 2015), 679–684.

[43] Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. 2013. Dynamic Bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection*, Tatiana V. Guy, Miroslav Karny, and David Wolpert (Eds.). Number 474 in Studies in Computational Intelligence. Springer Berlin Heidelberg, 1–35.

[44] Alex Cho Snyder and Mario Izquierdo. 2014. *Jumpcraft*. Game [PC]. (2014).

[45] Tobias Sturn, Michael Wimmer, Carl Salk, Christoph Perger, Linda See, and Steffen Fritz. 2015. Cropland Capture - a game for improving global cropland maps. In *Proceedings of the 10th International Conference on the Foundations of Digital Games*.

[46] Penelope Sweetser and Peta Wyeth. 2005. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment* 3, 3 (July 2005).

[47] Luis von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Vienna, Austria, 319–326.