

# Measuring Player Skill Using Dynamic Difficulty Adjustment

Simon Demediuk  
RMIT University  
Melbourne, Australia  
simon.demediuk@rmit.edu.au

Marco Tamassia  
RMIT University  
Melbourne, Australia  
marco.tamassia@rmit.edu.au

William L. Raffe  
University of Technology (UTS)  
Sydney, Australia  
william.raffe@uts.edu.au

Fabio Zambetta  
RMIT University  
Melbourne, Australia  
fabio.zambetta@rmit.edu.au

Florian “Floyd” Mueller  
RMIT University  
Melbourne, Australia  
florian.mueller@rmit.edu.au

Xiaodong Li  
RMIT University  
Melbourne, Australia  
xiaodong.li@rmit.edu.au

## ABSTRACT

Video games have a long history of use for educational and training purposes, as they provided increased motivation and learning for players. One of the limitations of using video games in this manner is, players still need to be tested outside of the game environment to test their learning outcomes. Traditionally, determining a player’s skill level in a competitive game, requires players to compete directly with each other. Through the application of the Adaptive Training Framework, this work presents a novel method to determine the skill level of the player after each interaction with the video game. This is done by measuring the effort of a Dynamic Difficult Adjustment agent, without the need for direct competition between players. The experiments conducted in this research show that by measuring the players Heuristic Value Average, we can obtain the same ranking of players as state-of-the-art ranking systems, without the need for direct competition.

## CCS CONCEPTS

• **Applied computing** → **Computer games**; *Interactive learning environments*; • **Computing methodologies** → Game tree search;

## KEYWORDS

Dynamic Difficulty Adjustment, Monte Carlo Tree Search, Video Games, Player Skill Ranking

## ACM Reference format:

Simon Demediuk, Marco Tamassia, William L. Raffe, Fabio Zambetta, Florian “Floyd” Mueller, and Xiaodong Li. 2018. Measuring Player Skill Using Dynamic Difficulty Adjustment. In *Proceedings of Australasian Computer Science Week 2018, Brisbane, QLD, Australia, January 29-February 2, 2018 (ACSW 2018)*, 7 pages.  
<https://doi.org/10.1145/3167918.3167939>

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).

ACSW 2018, January 29-February 2, 2018, Brisbane, QLD, Australia

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5436-3/18/01...\$15.00

<https://doi.org/10.1145/3167918.3167939>

## 1 INTRODUCTION

There are a number of reasons why a player would want to improve their skill at a particular video game; to get better at the game itself (self-improvement), or the game is being used as an educational or training tool. While video games have been shown to be a useful tool for education and training purposes [27], there has not been a reliable method to measure a player’s progress or skill level, without the need to play the participants against one another or further external testing. In the current literature player skill level is determined through multiple competitive games with other players [8, 13]. This research leverages the advantages of Dynamic Difficulty Adjustment [20] systems that improve player enjoyment and engagement in videos and presents a novel approach to measuring an individual player’s skill level.

Previous research has shown that presenting information through the use of video games is often a more successful means of education than traditional classroom-style teaching [26]. For this reason, the military has a long history of incorporating games and simulations into training to simulate combat environments [29], an approach which has proved greatly beneficial [14, 15, 17, 18, 22, 23, 34].

One of the most significant contributing factors to the success of video games as an educational or training tool is that they increase the player’s motivation and interest in the subject matter [12, 24]. This is important since player motivation has been shown to be essential to successful learning, as highly motivated players engage, persist longer and apply more effort when they approach a task [25]. This enables players to learn and progress at a much faster pace.

To increase the enjoyment and engaging aspects of video games, Dynamic Difficulty Adjustment systems have been employed in educational games with some success [28]. The goal of a Dynamic Difficulty Adjustment system is to tailor the level of challenge of the video game to the individual player’s skill level. They achieve this by aiming to produce a game outcome where the player has a 50% chance of winning, which has been established as a challenging opponent [19]. Yannakakis et al. [35] found empirically that players found that a game’s entertainment factor was at the highest when the game was set at an appropriate level of challenge for the player. Building on Csikszentmihalyi’s *Flow theory* [7, 31] Dynamic Difficulty Adjustment systems have been employed through adaptive artificial intelligence opponents [1, 4, 5, 36] and adaptive game environments [20, 37].

However, despite the benefits of using video games as a more engaging method of transferring knowledge to players, the problems

that are experienced by traditional teaching can still occur [27]. In particular, players' concentration can still wane, especially when the training lasts for a long period of time. This trend is supported by Dekkers et al. [9] who found that the effectiveness of training was inversely proportional to its duration. The longer the game was played, the more likely the trainee would become bored and thus unreceptive to the information being presented.

This research aims to address this issue by presenting a method by which to measure the skill level of the player after each engagement with the game, by investigating the performance of the player against a Dynamic Difficulty Adjustment artificial opponent. The strength of the actions that the artificial opponent chooses to employ against the player are recorded and a mean strength of the artificial opponent is calculated. As the artificial opponent is tailoring its difficulty based on the skill level of the player, the player skill level can be inferred from the strength of the opponent. This method is useful in providing a way in which to effectively compare players to one another and also provide a metric by which player progress can be monitored over time. Training can then be terminated once the player reaches the desired skill level, to over-training.

While this research presents an example implementation in a simple adversarial game (a Fight Game) in which a player would just be attempting to increase their skill level. This work could be easily applied to an adversarial game for education or military training, where the desired outcome is for the player to increase their skill level at that particular game. Additionally, Dynamic Difficulty Adjustment can be applied in a variety of different forms, not just in control of a adversarial agent. It can also be applied to content generation [33] and controlling in-game environment and intractable items [20]. The Dynamic Difficulty Adjustment and measuring techniques presented can be readily applied to these other applications. This leads to a large variety of game types and genres that these techniques can be applied to.

This paper is structured as follows: Section 2 gives a brief background on the video game used for testing as well as the Dynamic Difficulty Adjustment agent that was developed using Monte Carlo Tree Search (MCTS) [6]. Section 4 describes the experiments that were conducted with followed by a discussion of the results obtained in Section 5. While Section 6 gives a summary of the findings and their implications for future work.

## 2 BACKGROUND

In this section we will cover the Adaptive Training Framework that we build on for this research. As well as how the Dynamic Difficulty Adjustment artificial opponent was designed and implemented into a 2D-Fighting video game.

### 2.1 Adaptive Training Framework

The Adaptive Training Framework [10] is a framework that provides a mechanism by which a player's skill level and progress can be monitored when learning by playing a video game. The steps of the adaptive training framework are shown in Figure 1.

Step 1 involves the player, playing against a Dynamic Difficulty Adjustment artificial opponent. After the game is finished the performance of the artificial opponent is compared to fixed benchmarks and its rank determined. As the artificial opponents aim is

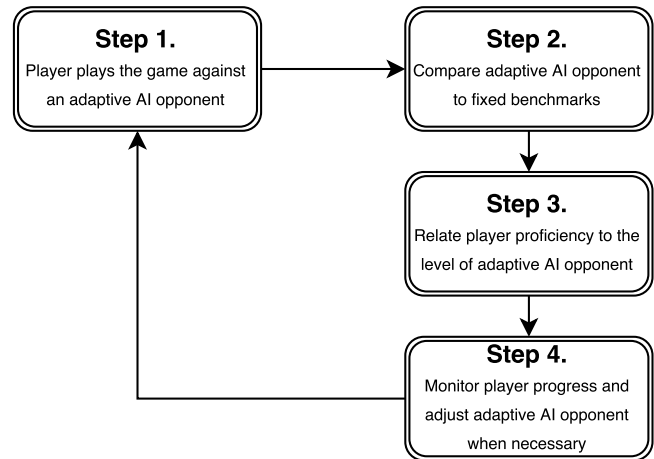


Figure 1: The steps of the adaptive training framework

to produce an even game (i.e 50% chance of the player winning) the player and the artificial opponent are evenly matched, so in Step 3 the player's skill level can be inferred from Step 2. Step 4 involves monitoring the player's skill level from game to game and ensuring the player's skill increases. Step 4 also provides a mechanism that enables the player to stop playing once a desired skill level is reached.

For this research we are focused primarily on developing generalized method for implementing Steps 1-3. This is done by using a MCTS-based adaptive agents for Dynamic Difficulty Adjustment (see Section 2.2), as opposed to the naive adaption method employed in [10], and a concise way to measure the performance of the artificial opponent (see Section 3). However, once the players skill level is established it can be easily monitored in between game rounds based on the users requirements.

### 2.2 Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) [6] has been used in previous work the as backbone for building Dynamic Difficulty Adjustment artificial opponents [11, 16]. MCTS is a tree-based search algorithm that builds a search tree through "playouts". In the search tree, each edge represents an action available to an agent and the leaf nodes are an encoded sequence of actions stemming from the root node to the leaf node. A playout is a simulation of the game where actions are performed according to the search tree starting from the root node to a leaf. The game is then simulated further through a sequence of random actions performed by the agent and the player. The result of all the simulations passing through a node is used to evaluate the suitability of said node.

During the time allocated to MCTS to make a decision, the algorithm iterates through the 4 phases shown in Figure 2. In the selection phase the tree is descended to the most promising leaf node from the root node. The leaf node is then expanded by adding children to the leaf node, if this is within a predefined depth limit. The playout is then performed in the simulation phase, where root-leaf actions first and random actions following are executed by the simulator. During the back-propagation phase, the outcome of the

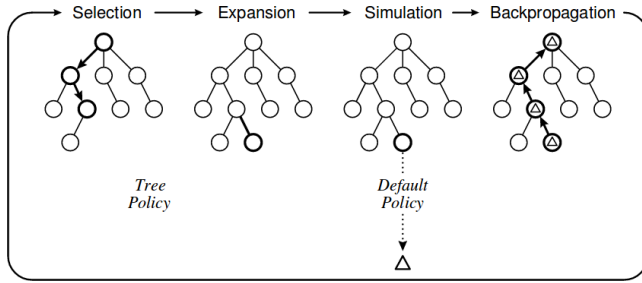


Figure 2: One iteration of the general MCTS approach

playout (determined by a heuristic or by the outcome if the game finished during the simulation) is used to update all the nodes in the path root-leaf. Once the time allocation has expired, a child action of the root node is selected. In the current literature the most common technique for action selection is to choose the most visited node.

**2.2.1 Adaptive AI.** The Dynamic Difficulty Adjustment artificial opponent that was used in this work was Outcome-Sensitive Action Selection (ROSAS) [11]. ROSAS is a Monte Carlo Tree Search (MCTS) algorithm with Upper Confidence Bound 1 applied to Trees (UCT), with a variation on the action selection criteria.

The goal of ROSAS is to closely match the skill level of the current opponent. To achieve this, ROSAS builds a search tree in the same manner as a regular MCTS algorithm, where nodes are evaluated according to a heuristic whose goal is to win. Once the search tree has been built, the algorithm selects the outcome that will mostly likely lead to a drawn game rather than a win. This is done through the application of this formula.

$$\text{action} = \arg \max_a - |r[a].\text{score}|, \quad (1)$$

where  $r$  is the root node of the tree,  $r[a]$  is the child of  $r$  corresponding to action  $a$  and score measures the advantage of the adaptive agent over its opponent, which can be positive (if the adaptive is winning) or negative.

### 2.3 FightingICE

The game engine used for this research is the 2D fighting game [21] called FightingICE. Similar to Street Fighter [32], FightingICE is a real-time fighting game, where two players face off in a 2-dimensional arena shown in Figure 3. The objective of the game is to reduce the opposing player’s health points to zero, using a combination of attacks and movement actions (e.g. punches, kicks and jumping). Typically the game is timed and played over multiple rounds; rounds end early if one player’s health points reaches zero.

An ROSAS agent was built within the FightingICE game, as a Dynamic Difficulty Adjustment artificial opponent. The ROSAS agent builds the MCTS search tree according to a heuristic based on the gap between the agents health points and the opposing player’s health points; the larger the gap in favour of the agent, the more likely the agent will reach the goal of a win. During action selection however, ROSAS bases its decision using Formula 1. This is motivated by the current literature for real-time fighting games,

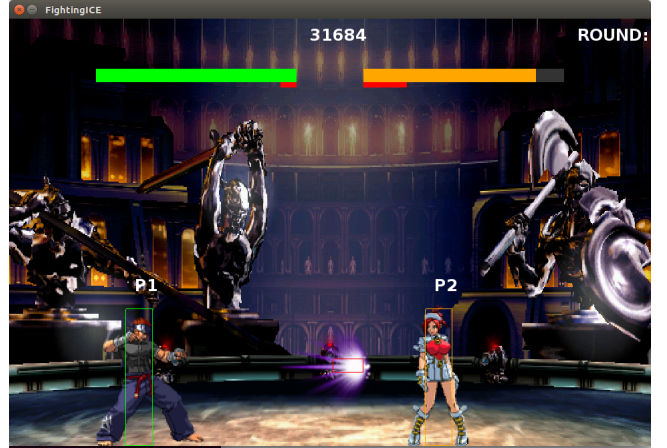


Figure 3: Screen capture of the FightingICE game.

Trial Name	Mean	Median	St. dev.
Bot	-1	-1	17
Human	-9	-15	34

Table 1: Aggregated mean and median final health difference, standard deviation, of the ROSAS agent in trials against bots and human players.

where it has been suggested that a challenging opponent is one that plays at the same level as the player [2]; this work defines this as a zero health point difference between the agent and the player throughout the game. So the agent action selection policy is based on reducing the current state’s health point difference to zero. For example, if the agent’s health points were behind those of the player by 10, the the agent would chose an action with a net outcome as close as possible to 10 health points.

The ROSAS agent was able to closely adapt to the level of both players and bots, by keeping the health point difference as close to zero as possible. Table 1 [11] shows the results of how close ROSAS was able to adapt to both bots and human players. In Table 1, the negative values indicate the agent final health was less then the health its opponent.

## 3 PLAYER SKILL MEASUREMENT

To determine the skill level of the player, we implemented the first 3 steps in the Adaptive Training Framework described in Section 2.1. The first step in the framework is for the player to play against an adaptive opponent that was implemented in this work as ROSAS, however this could be easily replaced by any state-action based method, such as the Reinforcement Learning [30] based algorithm Challenge-Sensitive Action Selection [3]. During this Step all the actions available at each decision point of the adaptive agent are recorded, along with their heuristic value. It is important to remember here that the heuristic value is based on the action’s projected health point differential.

Once Step 1 has been completed, each decision made by the agent is iterated over. There are two possible metrics to indicate the level of challenge perceived by the adaptive agent: the heuristic value of the selected action and the rank of the selected action in the ranking induced by the heuristic value. However, since the same rank is assigned to actions that result in the same heuristic value, the rank is not the best indicator of performance. We chose, instead, the heuristic value. The heuristic value action is averaged over the entire set of actions selected during the match: this results in a single metric, which we are calling Heuristic Value Average (HVA). The HVA is an estimate of the level at which the agent is performing in order to adapt to the player. The larger the magnitude of this value the more likely the agent is choosing actions closer to the best action (towards victory) at each decision point.

This metric can also be initially calculated for fixed AI players (if they are available) in the same manner; by playing the fixed AI player (e.g an easy, medium or hard AI opponent) against the adaptive agent. This then enables the comparison between the human player and fixed AIs without the need for them to actually play against each other; this can be used to gauge where on the difficulty scale the human player is at. Furthermore, different human players can be ranked against each other, again, without the need for them to directly play versus each other (which, instead, is how players are normally ranked in competitive games). In an educational and training system, this is beneficial as it allows the instructor to gauge the skill level of each player without the players needing to compete directly, in a similar fashion to how traditional classroom checkpoint tests work. Additionally, this metric can be re-evaluated after each interaction with the game to monitor how the players skill level progresses over time. Once the player reaches the desired skill level, which may be a different number of iterations for each player, the Adaptive Training Framework can stop.

## 4 EXPERIMENTAL RESULTS

Two different experiments are conducted in order to determine if the method for measuring player skill through adaptive AI investigation. The first trial involves the use of bots in playing against the ROSAS agent, with the bots rank computed and the results then compared to their Trueskill [19] rank. The second trial, is a small scale trial of human players playing against the ROSAS agent to investigate if this method can be used for ranking human players.

### 4.1 Bot Trials

To test the measurement of player skill level we first conduct a trial which involves the use of four bots from the 2016 CIG FightingICE competition [21]. The bots selected are DragonSurvivor, paranahueBot, Poring and BANZAI. Each bot plays 10 games against ROSAS agent, and the HVA for each bot is then calculated (averaged over the 10 games). Each game consists of three 90 second rounds, with a starting health of 500 health points. The results of these games are shown in Table 2. A positive number indicates that the ROSAS agent is selecting actions with an estimated positive health point trade for the agent, in order to catch up with the player, as the ROSAS agent is consistently behind in health. Inversely a negative number indicates that the ROSAS agent has to chose actions that result in a negative health point trade for the agent. The larger

<i>Bot name</i>	<i>HVA (avg. score)</i>
Poring	0.422744
paranahueBot	0.240623
DragonSurvivor	-0.856111
BANZAI	-1.66509

**Table 2: HVA metric for each bot ranked in order from highest to lowest**

<i>Bot name</i>	<i>Average skill (<math>\mu</math>)</i>	<i>Confidence <math>\sigma</math></i>
Poring	38.438	4.195
paranahueBot	28.929	3.483
DragonSurvivor	21.192	3.471
BANZAI	11.975	4.318

**Table 3: Trueskill, order from highest rank to lowest**

the HVA value, the closer the ROSAS agent is to choosing the best action, the better the skill level of the Bot.

To determine if the ranking shown in Table 2 is a correct reflection of the skill level of the bots, the Trueskill [19] value of each bot is calculated. Trueskill is a ranking system used for competitive video games (similar to Chess' Elo Rankings [13]). Trueskill is a Bayesian inference algorithm that measures the skill level of players with a confidence level, through the results of games played by the players against each other. To get an accurate estimate of the Trueskill value for a player in a 1vs1 game, each player needs to play at least a total 12 games.

To calculate the Trueskill for the bots, each bot plays 4 games against each other bot. Each game consists of 3 rounds, each lasting 90 second; the winner of the most rounds is considered the winner of that game (drawn rounds and thus drawn games are allowed). The games were played in random order and the Trueskill of each bot was updated at the end of each game based on the results. Table 3 shows the final Trueskill value where  $\mu$  is the average skill of the bot, and  $\sigma$  is the confidence of the guessed rating. The real skill of the bot is between  $\mu \pm (2\sigma)$  with 95% confidence. Each bot started with the default Gaussian distribution for Trueskill of  $\mathcal{N}(25, \frac{25}{3})$

### 4.2 Human Trials

To test the suitability of our measurement method for human players we conduct a small internal study, in which the ROSAS agent plays against 5 different players (pseudonyms where assigned to each players result). Each player plays 1 game against the ROSAS agent. The game consists of three 90 second rounds, with the starting health total set at 500 health points. The HVA is calculated for each player, shown in Table 4. Again, the larger the HVA value, the closer the ROSAS agent is to choosing the best action, the higher the value the better the skill level of the player. During the games we recorded some anecdotal observations about the players skill level.

Player name	HVA (avg. score)
Steph	6.5351
Jessica	3.62673
Ben	3.45736
Gus	-7.09903
Simon	-8.02535

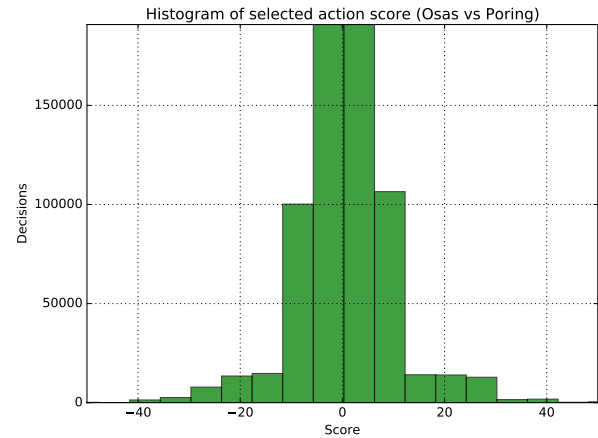
**Table 4: HVA metric for each player, ordered from highest to lowest**

## 5 DISCUSSION

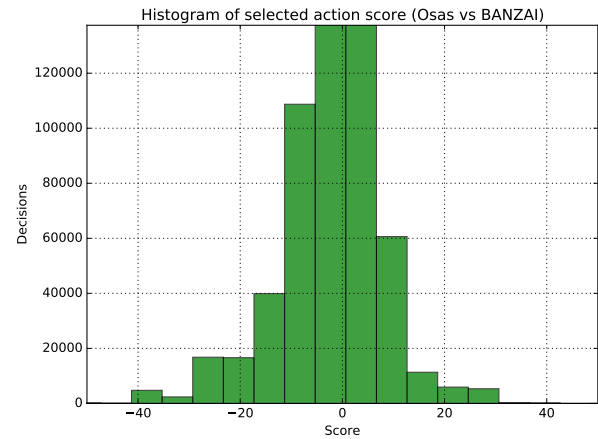
While we are not interested in the actual final rankings of the bots, the results show that the rankings of the bots obtained through the HVA metric (see Table 2), are the same as the rankings obtained via the Trueskill method (see Table 3). We can investigate this further by plotting the histogram of the health point difference of each action that was selected by the ROSAS agent for each decision. Where a positive x-value indicates actions that results in a positive health point trade for the agent (i.e the agent is behind); a negative x-value indicates an action that results in a negative health point trade for the agent (i.e the agent intentionally plays poorly as it is ahead). If we compare the histogram of the highest ranked bot, DragonSurvivor, (see Figure 4) to the histogram of the lowest ranking bot, BANZAI, (see Figure 5) we can see that BANZAI has a higher proportion of negative actions than DragonSurvivor, which also results in a lower number of positive action for BANZAI. This indicates that, in fact, when playing against BANZAI the ROSAS agent has to intentionally play more poorly, when compared to DragonSurvivor, in order to play at the appropriate challenge level for the BANZAI bot.

Although we obtained HVA rankings for a small group of human players, we were unable to play each player against each other to calculate their Trueskill value. We would expect a similar result as we know that the ROSAS algorithm adapts to bots in the same manner as it does for humans. If we look at the highest ranked player “Steph” and lowest ranked player “Simon” histogram (see Figure 6 and Figure 7 respectively), and compare them, we can see a large disparity in the scores of the actions selected by the ROSAS agent. When playing against Simon the ROSAS agent had to select a larger amount of actions that resulted in negative health point trades. Additionally, these actions also have larger negative scores associated to them compared to those selected by the agent when playing versus Steph. This result also matched the observational remarks that were recorded about the players skill level during the game play. Although we were unable to confirm the player rankings with Trueskill, HVA is a promising method for ranking human players after one interaction with the game, and can also be used as an indication of the player’s skill level.

Even though this results shows that our method can be used to rank bots and deliver the same rankings as Trueskill, it does not give the best indication of the skill difference between each bot, and subsequently each player; as the ranges are smaller than that of Trueskill and there is no concept of confidence level. However, if a suitably wide spread of different level bots can either obtained/developed a relationship between HVA and the Trueskill



**Figure 4: Histogram of the all action scores of Poring Bot vs ROSAS, for all games**



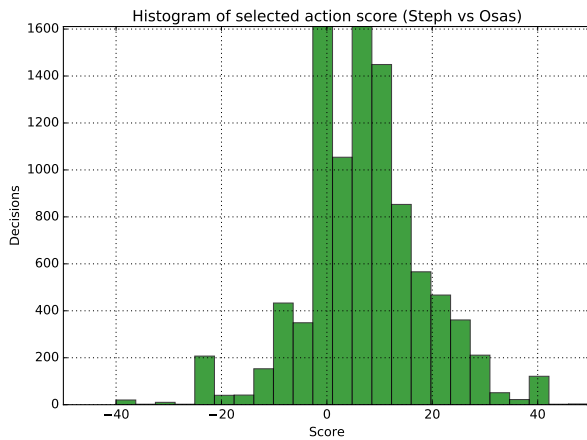
**Figure 5: Histogram of the all action scores of BANZI Bot vs ROSAS, for all games**

rating may be mapped; such that from a HVA value you can estimate a Trueskill value for a player. It is unclear at this stage what that mapping would look like, but it could provide a way to measure players Trueskill without the need for direct player competition. Overall results of this research are very promising from an educational and training perspective. Not only does a DDA agent provided a more enjoyable gaming experience, but we can also use the DDA agent to measure player skill level. Additionally, it also allows for players to be ranked against each other without direct competition which may be inappropriate or unfeasible in an educational environment.

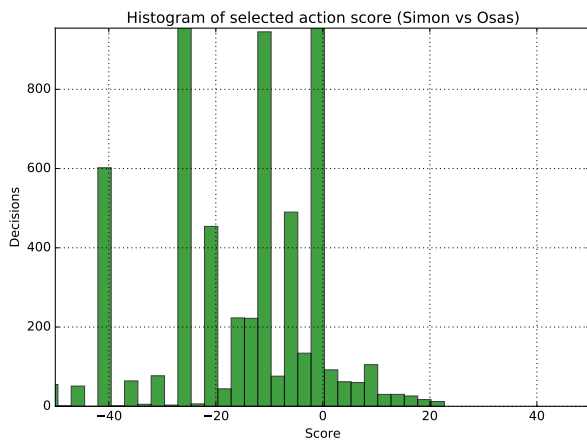
## 6 CONCLUSION AND FUTURE WORK

This paper presents a method of measuring player skill from the players interaction with a Dynamic Difficulty Adjustment agent.





**Figure 6: Histogram of the all action scores of Steph vs ROSAS**



**Figure 7: Histogram of the all action scores of Simon Bot vs ROSAS**

In contrast to current methods for measuring player skill, such as Trueskill, our method does not require players to compete directly with each other. This makes it suitable for use in educational and training environments where direct competition may not be feasible. Although this research was conducted using a non-educational or training game, measuring the Heuristic Value Average can be applied to any state-action based implementation of Dynamic Difficulty Adjustment, such as Monte Carlo Tree Search or Reinforcement Learning; which have been used in the past to produce valid Dynamic Difficulty Adjustment agents for a wide variety of video games.

While Dynamic Difficulty Adjustment agents can tailor the level of challenge of a game to the appropriate level of challenge for the player, there is a lack of research in the area of what constitutes the best level of challenge for learning in a video game. In the literature

it has been suggested that a win rate of 50% is best for matching players for a fair game. However, this may not be the best level of challenge for enjoyment or education purposes. Now that we have developed a method for measuring players skill level, we are aiming to conduct a large scale trial in which we will test the different levels of adaptive challenge and their impacts on player skill acquisition. Additionally, we will investigate a method by which the teacher or trainer can easily determine a Heuristic Value Average that represents a minimum learning outcome.

## REFERENCES

- [1] Amy L Alexander, Tad Brunyé, Jason Sidman, and Shawn A Weil. 2005. From gaming to training: A review of studies on fidelity, immersion, presence, and buy-in and their effects on transfer in pc-based simulations and games. In *The interservice/industry training, simulation, and education conference (IITSEC)*, NTSA, Orlando, Florida.
- [2] Gustavo Andrade, Geber Ramalho, Hugo Santana, and Vincent Corruble. 2005. Challenge-sensitive action selection: an application to game balancing. In *Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*. IEEE, 194–200.
- [3] Gustavo Andrade, Geber Ramalho, Hugo Santana, and Vincent Corruble. 2005. Extending reinforcement learning to provide dynamic game balancing. In *Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, 19th International Joint Conference on Artificial Intelligence (IJCAI)*. 7–12.
- [4] Phillipa M Avery and Zbigniew Michalewicz. 2010. Adapting to human gamers using coevolution. In *Advances in Machine Learning II*. Springer, 75–100.
- [5] Alexander Baldwin, Daniel Johnson, and Peta A Wyeth. 2014. The effect of multiplayer dynamic difficulty adjustment on the player experience of video games. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1489–1494.
- [6] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* 4, 1 (2012), 1–43.
- [7] Ben Cowley, Darryl Charles, Michaela Black, and Ray Hickey. 2008. Toward an understanding of flow in video games. *Computers in Entertainment (CIE)* 6, 2 (2008), 20.
- [8] Pierre Dangauthier, Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. Trueskill through time: Revisiting the history of chess. In *Advances in Neural Information Processing Systems*. 337–344.
- [9] John Dekkers and Stephen Donatti. 1981. The integration of research studies on the use of simulation as an instructional strategy. *The Journal of Educational Research* 74, 6 (1981), 424–432.
- [10] Simon Demediuk, William L Raffae, and Xiaodong Li. 2016. An Adaptive Training Framework for Increasing Player Proficiency in Games and Simulations. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*. ACM, 125–131.
- [11] Simon Demediuk, Marco Tamassia, William L. Raffae, Zambetta Fabio, Xiaodong Li, and Florian “Floyd” Mueller. 2017. Monte Carlo Tree Search Based Algorithms for Dynamic Difficulty Adjustment. In *Computational Intelligence and Games (CIG), 2017 IEEE Conference on*. IEEE.
- [12] Daniel Druckman. 1995. The educational effectiveness of interactive games. *Simulation and gaming across disciplines and cultures: ISAGA at a watershed* (1995), 178–187.
- [13] AE Elo. 1961. New USCF rating system. *Chess Life* 16 (1961), 160–161.
- [14] Daniel Gopher, Maya Well, and Tal Bareket. 1994. Transfer of skill from a computer game trainer to flight. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 36, 3 (1994), 387–405.
- [15] Shenan Hahn. 2010. Transfer of training from simulations in civilian and military workforces: Perspectives from the current body of literature. *Unpublished manuscript* (2010).
- [16] Ya’nan Hao, Suoju He, Junping Wang, Xiao Liu, Wan Huang, et al. 2010. Dynamic difficulty adjustment of game AI by MCTS for the game Pac-Man. In *Natural Computation (ICNC), 2010 Sixth International Conference on*, Vol. 8. IEEE, 3918–3922.
- [17] Mance E Harmon, Leemon C Baird, and A Harry Klopff. 1995. Reinforcement learning applied to a differential game. *Adaptive behavior* 4, 1 (1995), 3–28.
- [18] Robert T Hays, John W Jacobs, Carolyn Prince, and Eduardo Salas. 1992. Flight simulator training effectiveness: A meta-analysis. *Military Psychology* 4, 2 (1992), 63.
- [19] Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. Trueskill™: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems*. 569–576.
- [20] Robin Hunicke and Vernell Chapman. 2004. AI for dynamic difficulty adjustment in games. In *Challenges in Game Artificial Intelligence AAAI Workshop*. 91–96.

- [21] Ritsumeikan University Intelligent Computer Entertainment Lab. 2017. Fighting Game AI Competition. <http://www.ice.ci.ritsumei.ac.jp/fgaic/> (2017).
- [22] Jesse Orlansky and Joseph String. 1977. *Cost-Effectiveness of Flight Simulators for Military Training. Volume 1. Use and Effectiveness of Flight Simulators*. Technical Report. DTIC Document.
- [23] Jesse Orlansky and Joseph String. 1979. *Cost-Effectiveness of Computer-Based Instruction in Military Training*. (1979).
- [24] Da Pierty. 1977. *A Comparative Simulation Game Research*. *Simulation and Games* 8 (1977), 255–268.
- [25] Paul R Pintrich, Barbara A Schrauben, et al. 1992. Students' motivational beliefs and their cognitive engagement in classroom academic tasks. *Student perceptions in the classroom* 7 (1992), 149–183.
- [26] Josephine M Randel, Barbara A Morris, C Douglas Wetzel, and Betty V Whitehill. 1992. The effectiveness of games for educational purposes: A review of recent research. *Simulation & Gaming* 23, 3 (1992), 261–276.
- [27] Brent D Ruben. 1999. Simulations, games, and experience-based learning: The quest for a new paradigm for teaching and learning. *Simulation & Gaming* 30, 4 (1999), 498–505.
- [28] Sandra Sampayo-Vargas, Chris J Cope, Zhen He, and Graeme J Byrne. 2013. The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game. *Computers & Education* 69 (2013), 452–462.
- [29] Roger Smith. 2010. The long history of gaming in military training. *Simulation & Gaming* 41, 1 (2010), 6–19.
- [30] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. Vol. 1. MIT press Cambridge.
- [31] Penelope Sweetser and Peta Wyeth. 2005. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)* 3, 3 (2005), 3–3.
- [32] Takashi Nishiyama. 1987. *Street Fighter*. Game [Arcade]. (30 August 1987). CAPMCOM, Osaka, Japan.
- [33] Julian Togelius, Renzo De Nardi, and Simon M Lucas. 2006. Making racing fun through player modeling and track evolution. (2006).
- [34] K Williams, Joel Goldberg, and M Gilder. 1982. *Validation of the Effectiveness of Simulator Experience vs Real-World Operational Experience in the Port of Valdez*. Technical Report. DTIC Document.
- [35] Georgios N Yannakakis and John Hallam. 2006. Towards capturing and enhancing entertainment in computer games. In *Hellenic Conference on Artificial Intelligence*. Springer, 432–442.
- [36] Georgios N Yannakakis and John Hallam. 2009. Real-time game adaptation for optimizing player satisfaction. *Computational Intelligence and AI in Games, IEEE Transactions on* 1, 2 (2009), 121–133.
- [37] Georgios N Yannakakis and Julian Togelius. 2011. Experience-driven procedural content generation. *Affective Computing, IEEE Transactions on* 2, 3 (2011), 147–161.