Creating Adjustable Human-like AI Behavior in a 3D Tennis Game with Monte-Carlo Tree Search

Kaito Kimura, Yuan Tu, Riku Tanji, Maxim Mozgovoy
The University of Aizu
Tsuruga, Ikki-machi, Aizu-Wakamatsu, Fukushima, 965-8580 Japan
Email: {s1250131, m5252118, s1270139, mozgovoy}@u-aizu.ac.jp

***Abstract*—Interaction with opponents is a core element in video sports games. Thus, user experience in single-player matches heavily depends on the quality of AI opponents, who are expected to vary in their skill level and play styles. One way to achieve this goal is to learn game-playing behavior from real human players and to improve it if necessary with an automated optimization method. Monte-Carlo tree search (MCTS) has been successfully used for this purpose in several card and board games, such as chess and poker. We explore the possibility to apply MCTS in an action sports game of 3D tennis, and show how a dataset of pre-recorded tennis games can be used to train an MCTS-based AI system, exhibiting believable and reasonably skillful behavior.**

***Index Terms*—tennis, game AI, MCTS, believability**

1. Introduction

The genre of sports games provides an interesting challenge for game AI research and development. The environment of a typical sports game is relatively simple, limited with a bordered playing field and governed with a set of predefined rules, mirroring those taken from real-life sports event. Thus, the enjoyability of the game process is to a large extent determined with player-player interaction. If teammates and/or opponents are controlled by an AI system, its quality can make a significant impact on user experience.

A large number of research works explore the nature of enjoyability in computer games, and discuss how AI can contribute to user enjoyment [1]–[5]. Within the context of simulation-like sports games the most relevant factors appear to be *skill diversity* [3], *behavior diversity* [3], and *believability* [5]. The AI system is expected to provide sufficient challenge for players of different skill level; it is expected to behave in a realistic, human-like manner, and to exhibit a variety of play styles, avoiding repetitive acting patterns.

Many authors discuss how to achieve these goals in practice. Most approaches seem to fall into three categories: 1) handcrafting ”human-like” behavior; 2) engineering ”cognitive architectures” that simulate human decision making process to a certain degree; 3) learning and reproducing actual human behavior patterns. Some systems implement elements of two or all three categories. From a practical game development perspective, a certain capability of the given method may be decisive for a particular game project. For example, the importance of the ability to make long-term strategic decisions, to learn from scant data, or to provide an easy way to adjust AI skill level is highly game-dependent.

Monte-Carlo tree search (MCTS) [6] is a decision making method, growing in popularity due to its high flexibility, predictable computational demands, and the capacity to create strongly performing game AI agents. While this method is mostly used in board games, it was also successfully applied to more action-oriented genres, such as arcade and fighting games [7], [8]. Some authors suggest that MCTS can be used to obtain believable AI behavior [8].

The target environment for our AI system is a popular free-to-play mobile game *World of Tennis: Roaring ’20s* (see Fig. 2). It aims to represent a relatively accurate experience of a tennis game, comprising realistic player movements, ball physics, and typical acting patterns. Since *World of Tennis*, like most other free-to-play games, is designed *”for a (very) long duration of play”* [9], its AI system has to provide a diverse and lasting experience for the players. Thus, the game implements a believable machine learning-based AI system, aimed to mimic a large variety of human play styles [10]. Currently, there is no online multiplayer capabilities in the game: all matches are between human players and AI-controlled bots. The present AI solution is able to learn from human players and reuse their behavior patterns in subsequent matches. As a result, people play against diverse ”virtual clones” of their peers.

This approach ensures believability and diversity of behavior, but has no capability to control the skill level of the given AI agent. In *World of Tennis* we have enough ”virtual clones” to choose from, so finding an opponent of a desired level is not hard (a leaderboard rating of its human ”trainer” can be used to estimate agent’s skills). However, in general case the ability of an agent to optimize own behavior can be useful in a variety of scenarios. For example, one may want to have ”easy” and ”difficult” versions of the same character. An optimized ”virtual clone” of a player can be used as an intelligent assistant, suggesting the next possible moves. Finally, self-optimizing agents can find potential defects in the game engine, leading to unexpectedly degenerate winning strategies.

MCTS routine can be supplied with an existing tree, reflecting a certain play style of an agent. MCTS is also able to expand this tree and adjust action probabilities, optimizing the initial behavior. These capabilities motivated us to try introducing MCTS-based AI agents into *World of Tennis*. The present work reports our preliminary results, showing that MCTS is able to generate believable AI characters, playing on par with human participants. The simplicity of our current approach also leaves room for further improvements.

2. 3D Tennis Game Environment

The core game process in *World of Tennis* can be divided into the following phases:

- **Serve.** The player can walk to a desired serve location and hit the ball to a desired area.
- **Receive Serve.** The player can walk to a desired location to prepare for the opponent's serve.
- **Recovery Movement.** When the ball is moving towards the opponent, the player has a chance to run to an advantageous position at the court.
- **Incoming Ball.** When the ball is moving towards the player, the game engine directs the player character to the receiving location. The only task of the player is to perform the next shot, i.e., specify the target location of the shot and its type with a swipe gesture (three shot types are supported).

This way, *World of Tennis* incorporates the elements of both action and strategy, and the main challenge is to decide quickly where the next recovery and shot points should be set. A typical game episode (from a serve to a point scored) consists of interleaving "shot" and "move" actions.

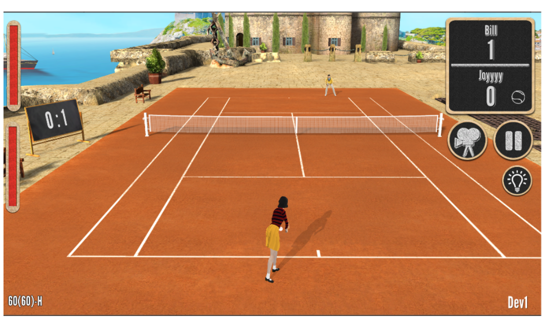

Figure 1. World of Tennis: Roaring '20s

The game also implements a system of character upgrades: the players can improve their characters' abilities, such as shot accuracy or speed of movements as well as equipment, such as rackets or shoes. These abilities are designed to compensate each other. For example, improving shot power decreases shot accuracy, so after upgrading the power ability one has to upgrade accuracy as well to keep shots both strong and accurate. Thus, as the players progress, the game becomes more fast-paced, but winning playing strategies remain essentially intact, as shown in our earlier work [11]. The existence of "accuracy" factor also shows that the outcome of each shot is probabilistic. A successful strategy has to balance risks: for instance, a shot sent to a corner of the court is hard to return, but it is also most likely to go out.

Matches are played in a tiebreaker mode: a player has to score at least seven points and be at least two points ahead of the opponent. Thus, while a typical match ends with a score 7:5 or 7:4, occasional results like 10:12 are also possible. An individual game session lasts around 1.5–3 minutes.

As noted previously, the game's existing AI system strives to provide diverse and believable opponents by "cloning" behaviors of actual human players. Behavior profiles are represented with Markov decision processes [10] and reflect around 30 last game sessions (older observations are removed from the knowledge model). We estimate that playing 10–15 sessions is enough to obtain a reliable clone, able to compete with people.

3. Adapting MCTS for Tennis

The use of MCTS-family methods require the game flow to be represented as a tree, where individual nodes correspond to certain game situations. As described in the previous section, basic game episodes (from serve to point scored) in *World of Tennis* consist of move/shot action sequences, effectively corresponding to a turn-based gameplay process. Thus, our game tree is similar to that of a board game (see Fig. 2).

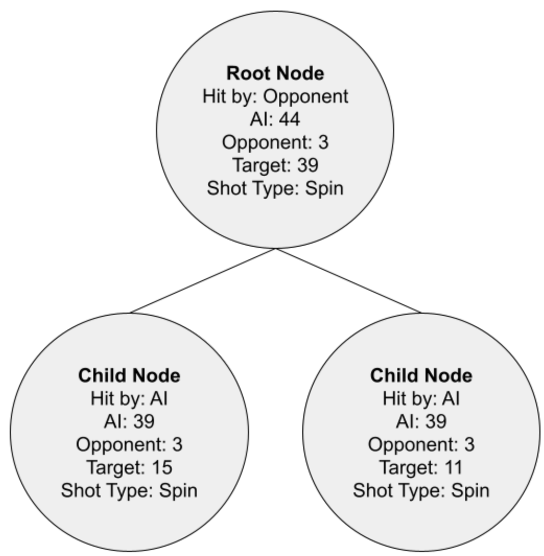

Figure 2. Game tree fragment

Each node contains the following elements:

- **Shot side**: AI side or Opponent (human player) side.
- **AI position**: on-court location of an AI-controlled character.
- **Opponent position**: on-court location of an opponent-controlled character.
- **Target position**: on-court target location of the current shot.

- **Shot type**: one of three possible shot types (basic shot, lob, power shot).

Locations are defined with integer cell indices inside a 6×8 court grid (see Fig. 3).

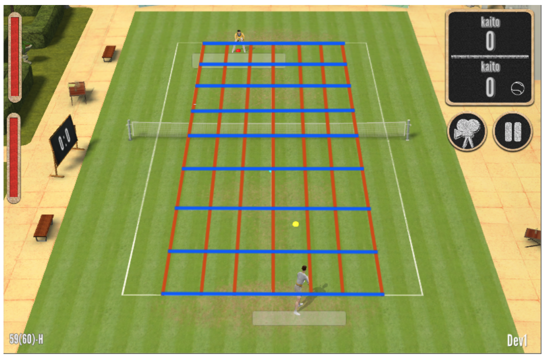

Figure 3. Tennis court grid

The situation shown in Fig. 2 can be understood as follows. The opponent is in the upper court half (cells 0–23), and the AI-controlled character is in the lower court half (cells 24–47). In the current state the ball is moving towards the AI character, and will land in the cell 3. As a response, the AI can return the ball to two different target positions (11 and 15) using a "spin" shot. A complete set of responses would include 24 possible return positions and three possible shot types.

In the present study, we use a popular MCTS variation known as UCT with UCB1 tree policy [6]. The core MCTS routine attempts to build and grow a partial game tree by traversing and expanding a tree obtained during previous algorithm runs. One of the key contributing factors to the performance of a particular MCTS variation is the choice of *tree policy*, governing the selection and creation of leaf nodes. The resulting algorithm assigns node scores according to the formula

$$UCT = X_i + 2C_p\sqrt{\frac{2\ln N_i}{n_i}}, where$$

$$X_i = p_i + (1 - p_i)\frac{w_i}{n_i}$$

The reward X_i is calculated on the basis of the immediate scoring potential of the next action p_i. The value of w_i denotes the number of wins for the node considered after the *i*-th action, and n_i stands for the number of simulations for the node considered after the *i*-th action (N_i is the number of simulations for its parent node). Thus, the value of w_i/n_i represents a possibility of winning of the current action, and is used as the reward value. The scoring potential p_i is calculated from actual shot patterns in the process of game simulation. The constant C_p governs the degree of bias towards unexplored nodes. It is set to $1.5/\sqrt{2}$ in the current version of our system.

4. Reusing Human Behavior Patterns

The algorithm described above represents a basic MCTS variation, aimed to identify winning strategies for the game. Other traits of the obtained behavior, such as human likeness, are not addressed. However, several works discuss how to adjust MCTS operation to improve believability of AI agents. These works also reflect different understanding of what exactly constitute "human-like behavior". For instance, Ishihara et al. [8] rely on the proposal of Demediuk et al. [12], suggesting that that believability is ensured by an improved MCTS action selection policy. It is proposed to treat all actions of low estimated reward (lying within a specified range around zero) as equal, and use random choice to select one. This should encourage more experimental behavior and avoid repetition of the same actions. A somewhat similar idea is implemented by Khalifa et al. [13]. The authors of this work deal with arcade games, and one of their concerns is human tendency to keep the same gamepad button pressed for more than one game frame. It effectively means repeating the same action, which is not a natural property of MCTS-driven AI agents. Thus, they suggest to modify MCTS action selection policy to adjust the probability distribution of same-action chains to match the patterns observed in human game recordings. Biasing MCTS selection towards a particular probability distribution of individual actions is proposed by Delvin et al. [14]. The target distribution is obtained from a collection of games played by people.

Thus, "human-like" behavior is sometimes considered simply as "possessing certain human-like traits", such as avoidance of repeating the same acting patterns or proclivity to proactive and exploratory behavior. It can also be seen as characterized by a particular generalized "human-like' action distribution, obtained from an aggregated collection of human-supplied game sessions.

While in *World of Tennis* we strive to represent a variety of play styles, corresponding to individual human players, we opted for generalized "human-likeness" in the present work. The probabilistic outcome of tennis actions as well as large branching factor of the game tree makes it hard for MCTS to reach a reasonable playing strategy without any *a priori* knowledge. Thus, we rely on data, extracted from 10000 actual human-vs-AI games, containing approximately 1400 distinct winning actions. We define an action to be "winning", if it resulted in player's point in more than 50% cases across 10000 actual games. Each action contains an action type (move or shot), its target location, and a shot type for shots. Unlike game states, actions use fine-grained coordinates, not bound to court grid cells. This gives us the ability to estimate action outcomes more accurately at the expense of losing player-specific acting patterns.

The ability of MCTS to adapt to ongoing changes in the game process can also be considered important for a human-like AI system. As players familiarize with the game and upgrade their abilities, they also learn to play better and discover new winning strategies. Thus, exploratory capabilities of MCTS decision making should also contribute to the overall user enjoyment.

5. AI Decision Making Process

In the current version of the system, the AI follows the standard Selection / Expansion / Simulation / Backpropagation scheme of MCTS. The UCT/UCB1 approach is used to balance between exploration and exploitation of child nodes. The UCT formula relies on scoring potentials of individual actions and the number of simulations made to calculate it. Thus, MCTS decision making process incorporates both the frequency (or popularity) of a particular action, and its scoring potential.

We calculated a scoring potential and a frequency of every action found in human-provided game recordings. Our dataset contains approximately 9300 unique actions, described with (shot side, AI position, opponent position, target position, shot type) tuples. Actions are distributed very unevenly: the average number of occurrences of an action is 3.35, while the most frequent actions appear more than 200 times in the dataset. This data shows some objective properties of tennis, such as higher scoring potential of shots into court corners or higher frequency of specific patterns typical, e.g., for serve actions. However, it also shows human players' preference for certain types of actions.

In our algorithm, MCTS starts with action frequencies and scoring potentials taken from the dataset. Thus, its choice of actions is biased by human-supplied behavior data. However, the outcomes of subsequent games affect node values, steering the MCTS-controlled agent towards more efficient (in terms of higher chances to win) strategies.

6. Evaluation: Skill Level

The implementation of player progress capability in *World of Tennis* makes it hard to evaluate the objective skill level of the MCTS AI system. On one hand, there is a great diversity of existing AI-supported players that can be used to benchmark the MCTS-based agent. On the other hand, we do not have any objective measurements of their skills: higher-ranking players might simply possess higher abilities such as shot accuracy or movement speed, which helps them to win, but does not improve their decision making. We have access to performance indicators such as the percentage of matches won, but the matchmaking system always supplies to each player a certain fixed percent of lower-ranking opponents to keep the game challenging and not frustratingly difficult.

For the first test, we took a random sample of twelve existing AI-supported players of different "player levels" (reflecting the total number of skill points allocated to the player) and played ten matches between the MCTS-based AI and each of the chosen players. The abilities of the MCTS AI were upgraded to match those of its opponent in each case. In this experiment we do not save the MCTS tree between the runs to evaluate the baseline performance, obtained from human-supplied behavior data. The results are summarized in Table 1, showing the total number of points scored by each opponent in the course of ten matches.

Data shows that the MCTS-driven character is stronger than one half of the opponents. In some cases the difference in score is quite small, so the skills of both opponents can be considered nearly equal. While this result represents the baseline performance, we still see at least three options for possible improvements. First, the training set of 10000 matches is not very large, and thus award scores for many leaf nodes cannot be considered reliable. Second, the grid we use is relatively coarse, so decision making might benefit from finer court partitioning. Third, the treatment of "move" and "shot" actions as independent does not reflect atomic units of tennis strategy. It would be more accurate to consider "shot and move" as a single action, leading to a certain outcome.

Player ID	**Level (1-9)**	Experience (matches)	Final score (player vs MCTS)
1	4	488	45:61
2	3	282	**70:41**
3	7	965	**70:18**
4	5	1203	**65:48**
5	9	1261	46:62
6	5	549	**69:54**
7	5	277	45:72
8	4	1684	62:67
9	1	345	26:27
10	2	305	**58:53**
11	6	2508	**54:48**
12	8	604	53:59

TABLE 1. FINAL SCORES (BUILT-IN AI VS MCTS AI)

The skill level of an MCTS-based AI is also expected to increase as a result of learning. To verify this effect we set up a quick test. Four players (ID 2, 3, 4 and 6) were selected from the twelve players used in the previous run to play a series of 20 matches against the MCTS AI. These players were found to be much stronger than the basic MCTS agent in the previous test, so there is a considerable room for possible improvement. This time MCTS tree is preserved between matches, so the agent is able to retain previously learned information. Despite small number of training sessions, MCTS agent was able to improve significantly, eventually becoming stronger than any of its opponents (see Fig. 4). The chart shows AI progress in four independent test sessions. Each test session consists of 20 games against a particular opponent. Chart bars show how the ratio of MCTS AI victories increases with every 5-game sub-session, and how the average game score shifts in favor of the MCTS AI. This gradual improvement provides a method to the game designers to fine-tune MCTS AI in terms of its skills, since the process of behavior optimization can be stopped at any time, freezing AI at the current skill level.

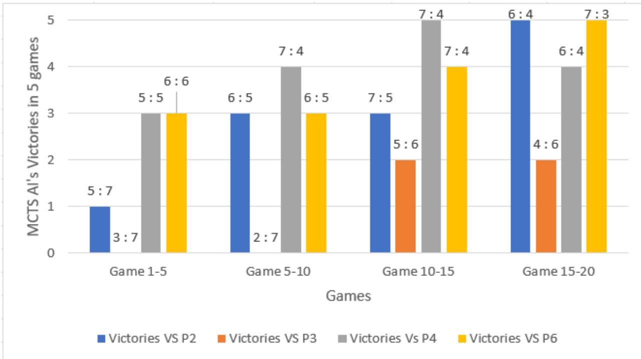

Figure 4. Matches won by the MCTS agent and average scores

7. Evaluation: Believability

Evaluating believability of AI is not an easy task. The most straightforward approach is to implement a variation of a Turing test, where observers are asked to label game characters as humans or bots in a series of video clips [15]. A number of research projects indeed rely on Turing test for assessing believability (one notable example is the 2K BotPrize Contest [16]). However, this method is hard to setup in a way to obtain reliable results and difficult to use for the task of play style identification, i.e. to label game recordings belonging to the same player. In addition, it is not scalable, and hardly suitable for quick assessment of gradual improvements of a system being under development.

Thus, some works propose relying on a certain kind of an automated assessment procedure. While it is arguably hard to assess "human-likeness" with an algorithm, it is possible to compare game logs of AI-controlled characters and of humans to evaluate similarity. We have proposed such a technique for *World of Tennis* in [10]. An automated procedure analyzes game logs to build a "behavior profile" of an individual player, based on a heatmap of its movements and shots. These heatmaps are converted into vectors of probabilities and compared using a cosine similarity formula.

We used this approach to compare the behavior profile of our MCTS player with the profiles of five AI-supported players P1–P5 and with a "Coach AI" — a deliberately weaker AI player used in a tutorial game section to teach the user the basics of the gameplay. The results are summarized in Table 2. They are consistent with our earlier findings [10] that players indeed possess distinctive play styles and generally preserve them across matches. The MCTS agent is seemingly not yet as human-like as our existing agents (however, they should be considered imperfect proxies for real human behavior rather than gold-standard human behavior profiles). Still, some player pairs, such as P6 and P9, also exhibit comparably low behavior similarity.

We also performed a quick Turing test-like survey involving 15 testers, who were asked to answer which of the two onscreen players behaved in a more human-like manner. Seven testers do not play sports video games, seven other people play occasionally (once a week), and one person plays daily. Three testers believe they are proficient sports game players. The recordings included three videos of MCTS AI vs Coach AI matches and three videos of a moderately experienced real human player vs MCTS AI matches. As a result, the MCTS-controlled character was labeled as "more human-like" in 79% of cases in MCTS vs Coach matches and in 64% of cases in MCTS vs human matches. It is surprising to have such a high score in the latter case, but it might be explained with a specific idiosyncratic play style of this particular player. We could observe similar mislabeling in video replays of fighting game sessions [17]. According to tester feedback, a somewhat aggressive and reasonably smart behavior is perceived as human-like. For instance, several testers mentioned aiming the corner or edge of the court as human-like, being risky and aggressive. The testers also noted active court positioning strategies, such as returning to the middle zone of the court after shot and taking into account the current opponent location.

MCTS	98.3						
Coach	5.0	97.6					
P1	33.7	39.7	94.8				
P6	28.4	35.6	60.0	88.1			
P8	22.6	35	70.7	40.1	89.4		
P9	15.7	14.2	50.4	35.1	39.5	96.4	
P10	27.7	16.3	64.6	65.2	57.3	50.2	93.6
	MCTS	Coach	P1	P6	P8	P9	P10

TABLE 2. BEHAVIOR PROFILES COMPARISON (P1–P5, MCTS, COACH AI)

8. Conclusion

In this work, we have created an MCTS-based AI system for a 3D tennis game. By modeling game states adequately, the AI is able to represent the tennis game process as a tree and perform search for winning actions in real time. The algorithm relies on a pre-calculated statistics of action outcomes, obtained from a collection of game recordings, played between actual human players and a built-in AI system based on Markov decision processes.

The resulting system learns very quickly, and is able to win most matches against a build-in AI system after 20 sessions of play. This capability makes MCTS a strong candidate for creating a fine-tunable game AI, able to provide diverse characters of varying skills. Our tests were performed against a small set of pre-selected opponents, so additional experiments are necessary to ensure the robustness of this approach.

The current believability assessment produced mixed results. On one hand, if we presume that our existing agents are believable, then MCTS agent cannot yet reach the same level of believability as the present system. On the other hand, our quick survey shows that external observers could not distinguish an MCTS-controlled character from a real human player. Here we have to note that the original premise of "human-likeness of MCTS" hinges on the presumption that action statistics obtained from human behavior data can sufficiently approximate human-like behavior. However, it is possible that more complex observations of human behavior are needed (such as move-shot sequences or partial decision trees). They can also be supplied as an initial input to MCTS.

Therefore, we can conclude that MCTS is a good candidate for creating AI-controlled opponents. Creating believability behavior is more challenging, but should be possible with more advanced ways of reusing human-supplied behavior patterns.

References

[1] H. Wang, C. Shen, and U. Ritterfeld, "Enjoyment of digital games," *In: U. Ritterfeld, M. Cody, & P. Vorderer (Eds.), Serious Games: Mechanisms and Effects*, pp. 25–47, 2009.

[2] J. M. Quick, R. K. Atkinson, and L. Lin, "The Gameplay Enjoyment Model," *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)*, vol. 4, no. 4, pp. 64–80, 2012.

[3] G. N. Yannakakis and J. Hallam, "Capturing player enjoyment in computer games," in *Advanced Intelligent Paradigms in Computer Games*. Springer, 2007, pp. 175–201.

[4] P. Sweetser and P. Wyeth, "Gameflow: A model for evaluating player enjoyment in games," *Computers in Entertainment*, vol. 3, no. 3, pp. 1–24, 2005.

[5] N. A. Taatgen, M. van Oploo, J. Braaksma, and J. Niemantsverdriet, "How to construct a believable opponent using cognitive modeling in the game of set," in *Proceedings of the fifth international conference on cognitive modeling*, 2003, pp. 201–206.

[6] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo Tree Search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[7] T. Pepels, M. H. Winands, and M. Lanctot, "Real-time Monte Carlo Tree Search in Ms Pac-Man," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 6, no. 3, pp. 245–257, 2014.

[8] M. Ishihara, S. Ito, R. Ishii, T. Harada, and R. Thawonmas, "Monte-carlo tree search for implementation of dynamic difficulty adjustment fighting game ais having believable behaviors," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games (CIG'18)*. Piscataway, NJ: IEEE, 2018, pp. 1–8.

[9] P. Luban. (2011) The design of free-to-play games. [Online]. Available: https://www.gamasutra.com/view/feature/6552/the_design_of_freetoplay_games_.php

[10] M. Mozgovoy, M. Purgina, and I. Umarov, "Believable self-learning AI for World of Tennis," in *2016 IEEE Conference on Computational Intelligence and Games*, 2016, pp. 1–7.

[11] M. Mozgovoy, "Analyzing user behavior data in a mobile tennis game," in *2018 IEEE Games, Entertainment, Media Conference (GEM)*, 2018, pp. 1–9.

[12] S. Demediuk, M. Tamassia, W. L. Raffe, F. Zambetta, X. Li, and F. Mueller, "Monte carlo tree search based algorithms for dynamic difficulty adjustment," in *2017 IEEE conference on computational intelligence and games (CIG)*. IEEE, 2017, pp. 53–59.

[13] A. Khalifa, A. Isaksen, J. Togelius, and A. Nealen, "Modifying mcts for human-like general video game playing," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, p. 25142520.

[14] S. Devlin, A. Anspoka, N. Sephton, P. I. Cowling, and J. Rollason, "Combining gameplay data with monte carlo tree search to emulate human play," in *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[15] J. Togelius, G. N. Yannakakis, S. Karakovskiy, and N. Shaker, "Assessing believability," in *Believable bots*. Springer, 2013, pp. 215–230.

[16] P. Hingston, "A turing test for computer game bots," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, no. 3, pp. 169–186, 2009.

[17] K. Yuda, S. Kamei, R. Tanji, R. Ito, I. Wakana, and M. Mozgovoy, "Identification of play styles in Universal Fighting Engine," in *Game-On Conference*, 2021. [Online]. Available: https://arxiv.org/abs/2108.03599