

Believable Self-Learning AI for World of Tennis

Maxim Mozgovoy, Marina Purgina
The University of Aizu
Aizu-Wakamatsu, Japan
{mozgovoy, d8172102}@u-aizu.ac.jp

Iskander Umarov
Helium9 Games, s.r.o.
Prague, Czech Republic
isk.umarov@gmail.com

Abstract—We describe a method used to build a practical AI system for a mobile game of tennis. The chosen approach had to support two goals: (1) provide a large number of believable and diverse AI characters, and (2) let the users train AI “ghost” characters able to substitute them. We achieve these goals by learning AI agents from collected behavior data of human-controlled characters. The acquired knowledge is used by a case-based reasoning algorithm to perform human-like decision making. Our experiments show that the resulting agents indeed exhibit a variety of recognizable play styles, resembling the play styles of their human trainers. The resulting AI system demonstrated stable decision making, adequate for use in a real commercial game project.

Keywords—game AI; believability; case-based reasoning.

I. INTRODUCTION

The task of designing a game AI begins with a fundamental question: what is the purpose of AI in the given particular game? What kind of AI will contribute to the overall success of the game? Various aspects of “good game AI” are currently being discussed in the community, and include, in particular, believability, fun, and high skill level [1, 2]. The diversity of these aspects can be explained with the diversity of computer games and game genres. As Kevin Dill summarizes, “*The one thing that is universally true is that games are about creating a particular experience for the player—whatever that experience may be. The purpose of Game AI... is to support that experience*” [3].

In the present work, we will describe the AI system for the upcoming mobile free-to-play game *World of Tennis* [4]. Since free-to-play games are typically designed “*for a (very) long duration of play*” [5] to increase in-app spending (and *World of Tennis* is no exception), one of the principal goals of AI in this case is to provide a diverse and lasting experience, keeping player attention for prolonged time periods. Consequently, we decided to focus on the following game elements:

- *Believable, diverse AI characters.* To ensure long-term player retention, AI characters have to be diverse, fun to play with, and exhibit distinct play styles.
- *Different AI characters for different player profiles.* Like most free-to-play games, *World of Tennis* implements an extensive system of character upgrades, which encourage players to experiment with their play styles. For example, characters with low *running speed* skill values should generally stay close to the central axis of the court to maximize their chances of receiving

opponents’ shots. However, faster characters encourage more experimental and risky play styles. The AI system should provide interesting and challenging opponents for all variations of game characters.

- *The ability to train your “ghost” character.* Human-trainable “ghost” characters provide additional elements of gameplay. We want the AI to be able to serve as a substitute for the user, and complete the game session automatically if the user has no time or wish to do it.

These game elements are consistent with the goals of our previous research projects, dedicated to the AI of boxing and soccer [6, 7]. Thus, we decided to rely on the same method: learning AI play styles from real people, and using case-based decision making during the game. The ultimate goal of such an AI system is to replicate the actions of its human trainer. By accomplishing it, the AI will be able to support the aforementioned game elements, since the diversity of actual *World of Tennis* players will ensure the diversity of AI agents, and the players will be able to train their own “ghosts”.

II. RELATED WORK

While *train your “ghost” character* capability (similar to “creating your own ghost” function in *Tekken 6* [8]) as a user-end feature immediately suggests a method based on human behavior observation, in general the task of building diverse and believable agents can be accomplished in different ways. We should note that the problem of creating a large family of AI-controlled characters is rarely addressed. Among them we can mention the work [9] that proposes to use multiobjective evolutionary algorithms to create the whole populations of NPCs. Due to stochastic nature of evolutionary algorithms, they can produce any required number of distinct characters.

When believability is explicitly stated as a goal, one of the following three general approaches is typically used:

1. *Rely on expert knowledge.* Decision-making logic can be directly hand-coded according to an expert view of the given domain. In certain cases this solution can be adequate, and result in a solid AI system [10].
2. *Mimic human decision making process.* A system can be designed on the basis of contemporary psychological theories of human behavior [11].
3. *Rely on actual logs of human behavior.* In the most straightforward form, this approach can be reduced to replaying human decision-making logs verbatim [12], but usually it is implemented via machine learning:

agents are trained on human data to provide the same actions as human players in similar situations [13, 14].

Our approach falls into the third category. We use logs of actual human-controlled characters to train the AI system. When the AI agent performs decision making, it relies on the acquired knowledge for case-based reasoning. The most relevant predecessor of the present project is our earlier work [7], dedicated to the game of boxing. Tennis and boxing actually share several notable aspects: both games feature one vs. one gameplay in a closed rectangular-shaped space; neither game requires long-term strategical thinking, but allow enough room for exhibiting a variety of play styles.

III. WORLD OF TENNIS: ENGINE AND GAMEPLAY

Our previous experience shows that even minor changes in gameplay may require notable modifications of the AI learning and decision making procedures, so let us briefly discuss the game mechanics of World of Tennis, as it affects AI design.

World of Tennis is a mobile version of a conventional one vs. one tennis sports game. The player sees the whole court shown with a fixed camera: there is no need of scrolling or camera adjustments. The player always controls the bottom character, while the top character is controlled by the AI system (see Fig. 1).

There are only two possible action types available to the player. By tapping the screen area of the own side of the court, the player directs the character to the desired target point (*SetMovePoint* action). By tapping the screen area of the opponent side of the court, the player sets the target point for the next shot (*SetHitPoint* action). The game engine calculates the optimal shot parameters according to the current player skill values. For example, a player with a higher *shot power* value is able to hit the ball with higher speed. By tapping the opponent side of the court twice, the player forces the game engine to perform a high lob shot, typically used to loft the ball over the opponent.

In Fig. 1, the bottom player's active move point is shown with a cube (located near the center of the player side of the court), while the active hit point is represented with a sphere (located near the opponent).

When the player performs a *SetHitPoint* action, the game engine displays a *shot circle* (shown as a white circle near the center of the opponent side of the court in the Fig. 1). This circle represents the accuracy of the next player shot, and starts shrinking over time as soon as the opponent returns the shot. The actual target of the shot will be randomly selected within the circle limits. This way, the game engine encourages the players to choose their hit points as early as possible to ensure higher accuracy. However, the shot circle shrinks much quicker for the players with high *accuracy* skill values, so they have more flexibility in tactics.

The actual shot is performed as soon as the ball comes close to the player, and the next hit point is set. There are no actions required for making the shot: setting a hit point is sufficient. Furthermore, once the ball is shot towards the player, the system automatically starts steering the player to the

optimal ball receive location (it also depends on character skill values). Hence, the game process is strictly divided into distinct phases. When the ball is moving towards the player, the player can only set the next hit point, since its movements are controlled automatically. When the ball is moving towards the opponent, the player can only move to the specified court location. While it is also possible to set the next hit point in this phase, the game engine will not process it until the opponent returns the shot.

The general motivation of these design decisions is to facilitate tactical rather than pure arcade gameplay. The player is encouraged to select winning hit point and move point locations, while the game engine takes care of the rest.

If the player has no time or desire to finish an ongoing game, it is possible to pass the control to the "ghost" character at any moment. Ideally, the "ghost" should be able to replicate the player behavior style as closely as possible.

IV. AI AGENTS: IMPLEMENTATION DETAILS

A. Learning Human Behavior

The core element of our AI system is its knowledge representation mechanism. We rely on the somewhat extended notation of a finite-state machine (FSM), later referred as *acting graph* [15]. Acting graph consists of N independent automatically generated FSMs representing an individual agent's knowledge with different levels of granularity (see Table I). Each state of the i -th machine incorporates A_i attributes representing a certain game situation. The states are connected with action-labeled edges. Unlike states, actions do not have levels of granularity, and are always represented with a complete set of attributes:

- Action type (*SetHitPoint* or *SetMovePoint*).
- Shot type (*None* for *SetMovePoint* actions, *Auto* or *Lob* for *SetHitPoint* actions).
- Target point position (x and y coordinates on the court).

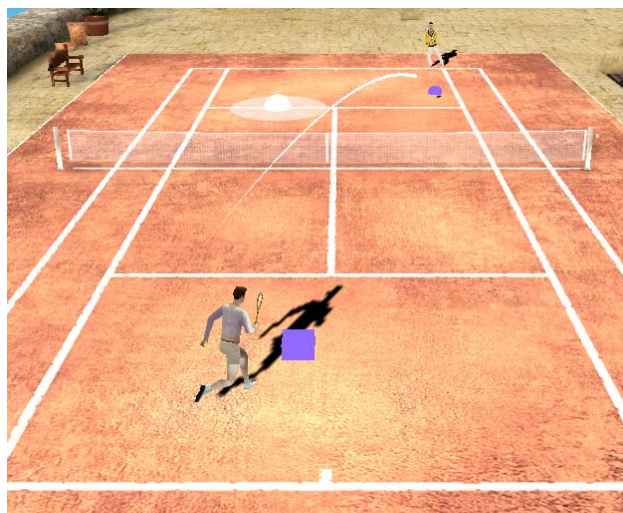


Fig. 1. Actual screenshot of *World of Tennis*.

In addition, states and actions have auxiliary attributes, containing their usage frequency, timestamps, and other optional data fields.

The number of independent FSMs (i.e., levels of granularity) as well as the choice of their attributes is currently done manually in accordance with expert knowledge. The present design of attribute sets is a trade-off between AI quality, pure business goals, and mobile platform restrictions. While the AI, in principle, should be tuned for the best accuracy, we also had to set reasonable limits for the expected knowledgebase size, CPU load and required learning time. One of the business decisions was to ensure that any given “ghost” is able to reproduce basic player style elements after 7-10 minutes of realtime training.

TABLE I. ACTING GRAPH CONFIGURATION

FSM	Next level status ^d	Game Situation Attributes
FSM ₀		Player phase (serve / hit / receive / etc.)
		Player coordinates ^a
	M	Player <i>move point</i> coordinates ^{ac}
	M	Player actual <i>hit point</i> coordinates ^{ac}
	M	Opponent coordinates ^a
	R	Opponent <i>move point</i> coordinates ^{ac}
	R	Opponent actual <i>hit point</i> coordinates ^{ac}
	R	Player shot type (serve / top spin / lob / etc.) ^f
	R	Opponent shot type ^c
FSM ₁	M	Player phase (serve / hit / receive / etc.)
		Player coordinates ^a
		Player <i>move point</i> coordinates ^{bc}
		Player actual <i>hit point</i> coordinates ^{bc}
		Opponent coordinates ^b
FSM ₂		Player phase (serve / hit / receive / etc.)
		Player <i>move point</i> coordinates ^{bc}
		Player coordinates ^b
		Player actual <i>hit point</i> coordinates ^{bc}
		Opponent coordinates ^b

^a. An integer value inside a 10×10 grid.

^b. An integer value inside a 5×5 grid.

^c. If exists.

^d. M: attribute will be modified on the next level;

R: attribute will be removed on the next level.

During the game, the AI system observes the actions of the human-controlled character and inserts into the FSMs the (*state, action, state'*) triplets, actually occurred in the tennis match [15]. As a result, we obtain a system of static FSMs that encode agent knowledge (this approach is similar to learning behavior trees from observation, discussed in [16]). The resulting structure can be also treated as a set of three Markov decision processes, representing human behavior.

B. Decision Making Mechanism

In the simplest case, an AI agent can choose the next action by finding a match for the current game situation in the acting graph. If the agent learned a triple (*state, action, state'*), it can decide that the best move in the game situation *state* is to perform *action*, and the anticipated outcome will be *state'*. In practice, however, it is improbable that the agent will be always able to retrieve actions from the FSM₀ graph, most accurately representing user intentions. The procedure of case-based reasoning requires approximate matching, based in World of Tennis on the following three instruments.

1. If searching FSM₀ yields no results, we can repeat the same operation with reduced attribute sets for FSM₁ and FSM₂. The resulting matches will be less precise, as any state retrieved from FSM₁ or FSM₂ represents the actual game situation with fewer details. Therefore, the retrieved actions might not accurately reflect user intentions anymore.
2. We can optionally turn off the requirement of action adjacency in decision making mode. For example, if the agent performed the action A_1 in the state 1, it should then continue acting from the state 2 (see Fig. 2). However, if the perfect match for the current game situation is a state other than 2, we can ignore the current sequence of actions forming player strategy, and continue from the best matching state. This relaxation can greatly increase the number of relevant actions on higher levels of abstraction. Say, if the state (2, 4) of FSM₁ corresponds to the states 2 and 4 of FSM₀, and the previous agent action was A_1 , the only match in FSM₁ that preserves action adjacency would be A_3 . By relaxing the adjacency requirement, we can retrieve both A_3 and A_4 .
3. Numerical attributes (such as player coordinates) can optionally be matched approximately with a specified range of error. In the current implementation we only allow errors within a range $[-1, 1]$, i.e., for a given value V the values $V-1$, V , and $V+1$ will be treated as acceptable matches.

The complete decision making procedure performs a sequence of queries until at least one matching action is found (see Table II). Each subsequent query implements further relaxations and thus can potentially retrieve more actions, albeit at the cost of reduced accuracy.

C. Decision Making Points

One of the surprisingly difficult tasks for the AI subsystem design is to decide *when* to act. The frequency of AI decisions should be carefully balanced. Short time intervals between decision making points help the AI to react to ongoing changes on the court. However, it may cause jolted movements of the AI agents, since subsequent calls to the AI system override previous move and hit points. Furthermore, they increase CPU load. The present solution is based on the following observations:

1. The player moves to occupy a certain advantageous court location. It is unlikely that a reasonable game

strategy requires complex movements between the player shot and the subsequent opponent shot. The next move point can be set right after the player shot.

2. The player can set a hit point right after opponent shot. It is unlikely that a reasonable game strategy will require more than one revision of a hit point (since it will reduce the accuracy of the shot).

Hence, the interval between two consecutive player shots contains three decision making points: 1) right after the player shot (*SetMovePoint* expected), 2) right after the opponent shot (*SetHitPoint* expected), and 3) right after the ball covers a half of the distance between the players (*SetHitPoint* expected). One may notice that this algorithm actually shortens AI reaction time comparing to its human trainer.

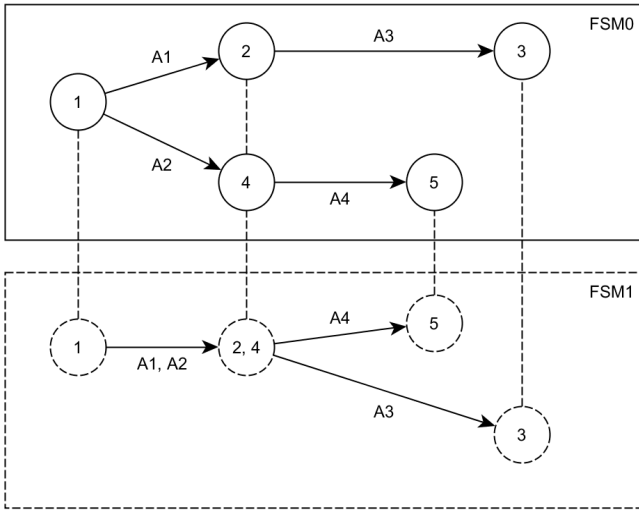


Fig. 2. Optional preservation of player strategy.

D. Action Ranking

When the decision making procedure has to rely on relaxed search conditions, it often returns a list containing a relatively large number (dozens) of actions, only approximately applicable in the current game situation. To improve agent behavior, we implemented the action ranking algorithm, based on the following simple rules.

1. Initially, each action is assigned a numerical weight equals to $freq \times age$, where $freq$ is the number of occurrences of the given action in learning sessions, and age is the cardinal number of the current learning session. This way, both frequent and more recent actions are prioritized. The primary goal of age is to make the AI adjust to new patterns of player behavior that emerge due to character upgrades.
2. If the resulting list of actions contains an action similar to the one currently being executed by the agent, the system aborts decision making, and lets the agent to continue. The rationale is to keep the ongoing player activity if the AI system can prove that the current action is still relevant (i.e., present in the action list). Two actions are considered similar if the Euclidean

distance between their target points is less than *MinDist* value, currently set to one meter. This rule effectively suppresses unnecessary *SetHitPoint* actions set at the third decision making point.

After ranking, a weighted random choice is used to select the action to be returned to the game engine.

TABLE II. CASE-BASED REASONING WITH QUERIES

Query	FSM level	Require action adjacency	Range search on attributes
1	0	Yes	—
2	0	No	—
3	1	Yes	—
4	1	No	—
5	1	Yes	Player coordinates
6	1	No	Player coordinates
7	2	Yes	—
8	2	No	—
9	2	Yes	Player coordinates Player actual <i>hit point</i> coordinates Opponent coordinates
10	2	No	Player coordinates Player actual <i>hit point</i> coordinates Opponent coordinates

V. PLAY STYLE COMPARISON METHOD

Since an AI-controlled character in the game is typically intended to serve as a “ghost” of a certain human player, it has to exhibit a similar play style and demonstrate a comparable skill level as its human trainer. Evaluating skill level of an AI agent is a rather straightforward task: we can play a series of matches between two agents and check the final scores. In contrast, comparing play styles of two agents is a far more complex problem.

The task of play styles comparison is often discussed in connection with a more general task of evaluating character’s believability, i.e. its ability to provide the illusion of being controlled by a real human player. Since believable characters are not necessarily obtained by learning from human behavior, “gold standard” human behavior patterns might not be available. Consequently, believability assessment is often implemented in a form of a Turing test, where game observers have to evaluate believability of characters they see [17].

In our case, it is possible to evaluate human-likeness of an AI-controlled character via direct comparison of AI-generated and human-generated samples of game play. In its turn, there is no universal scheme of game style comparison, as relevant game style features highly depend on a particular game. For example, promising results for a first-person shooter game were obtained with player trajectories comparison [18], while in Super Mario Bros. a scalar *performance score* function incorporating various player achievements was shown to serve as a reasonable criterion of behavior similarity [13].

The discussion of features that constitute a play style for tennis deserves a separate study. For the current purposes, we decided to develop the simplest possible scheme that can separate human players reliably. Presumably, humans exhibit distinct play styles, while the same person keeps a consistent

play style across games. Therefore, the desired similarity function in most cases should produce higher similarity values for the same player in different games, and lower values for different human players.

One of the approaches that turned out to be unreliable was based on plain heat map comparison. We represented the character side of the court with a two-dimensional table (10×10 cells), then used actual game logs to calculate the probability of character presence in each court cell. Such probability tables (heat maps) were transformed into one-dimensional arrays (vectors), and compared using a dot product, yielding a similarity value within a range $[0, 1]$. However, the experiments showed that even pairs of completely unrelated characters often generate similar heat maps.

Our current comparison algorithm uses a modified heat map approach, based on a presumption that a play style is defined with explicit player actions rather than character presence in certain court locations. The algorithm first builds an independent heat map for each character's *SetMovePoint* and *SetHitPoint* actions found in a game log:

```

For each SetMovePoint action a:
    MoveF[a.x, a.y]++
    MoveCount++

For each SetHitPoint action a:
    HitF[a.x, a.y]++
    HitCount++

For x = 0...9 and y = 0...9:
    MoveP[x, y] = MoveF[x, y] / MoveCount
    HitP[x, y] = HitF[x, y] / HitCount

```

Action target coordinates *a.x* and *a.y* are here represented with scaled integer values lying in the range $[0, 9]$. Two resulting heat maps of each character are then converted into vectors and concatenated. The obtained vector serves as a character's "behavior fingerprint", and can be compared with other vectors with a dot product.

VI. EXPERIMENTAL RESULTS

To evaluate the quality of the resulting AI system we performed a series of experiments, answering the following research questions:

*Q*₁. Do human players exhibit distinguishable play styles?

*Q*₂. Do AI "ghost" characters exhibit distinguishable play styles, similar to the styles of their human trainers?

*Q*₃. Do AI "ghost" characters exhibit tennis skills, comparable to the skills of their human trainers?

A. Preparation of Experimental Dataset

In our experiments we rely on behavior data of eight human players H_1, \dots, H_8 , and one "AI coach". Human players highly vary in their skill level, but all of them have some experience of World of Tennis gameplay. The AI coach is a "ghost", deliberately trained to be a reasonable-skilled opponent for entry-level players. All the characters in the game have comparable skill values (running speed, shot power, etc.), kept with no significant adjustments during the experiments.

A single *game session* in the experiments consists of a short match that lasts until a) one of the players scores 7 points; and b) the difference between scored points of the opponents is at least two. On average, the actual duration of an individual game session is around 2 minutes. Any number of game sessions can be used to train a character's AI "ghost" or to construct a behavior fingerprint.

The actual sessions were designed to replicate the real everyday World of Tennis activities. All the human players were put into the same "league" and assigned opponents according to a round robin scheme. Thus, each player has to play seven game sessions as a host against another player's AI "ghost", and seven game sessions as a guest, serving as an AI "ghost" opponent for another player. Each league day begins with a training, when a player has a chance to play 1-2 game sessions against the AI coach or, starting from day two, against the AI "ghost" of the player currently leading in the tournament. After the training, the player has to play the next scheduled game.

In total, it gives us (per player) 7 league games played as a host, 7 league games played as a guest, and 7.88 training games on average. Our present experiments do not take into account that the "ghost" are trained on the fly, which means that their skill level and ability to capture human play styles should presumably be higher for the latter game sessions.

B. Primary Experiments

Experiment 1. We used the obtained game data to build a behavior fingerprint of each human player H_i , and compared the fingerprints of every human character pair (H_i, H_j) . To check whether a character's play style is consistent across different games, we also compared two fingerprints of the same human characters, obtained on three randomly selected sessions. The results of fingerprint comparison are shown in Table III.

TABLE III. HUMAN PLAY STYLE SIMILARITY

H₁	0.96							
H₂	0.58	0.94						
H₃	0.54	0.76	0.93					
H₄	0.75	0.58	0.55	0.93				
H₅	0.88	0.49	0.56	0.57	0.97			
H₆	0.77	0.59	0.67	0.83	0.64	0.96		
H₇	0.81	0.69	0.60	0.74	0.69	0.68	0.96	
H₈	0.74	0.78	0.62	0.81	0.57	0.72	0.89	0.93
	H₁	H₂	H₃	H₄	H₅	H₆	H₇	H₈

$$\begin{aligned} \text{mean}(\{(H_i, H_j), i=j\}) &= 0.95, \sigma = 0.02 \\ \text{mean}(\{(H_i, H_j), i \neq j\}) &= 0.68, \sigma = 0.11 \end{aligned}$$

The similarity values show that different fingerprints of the same human player are consistently closer than the fingerprints of two distinct players. Since fingerprints are essentially heat maps, it is also easy to visualize them to show the differences in human play styles (see Fig. 3).

One may notice that the present fingerprinting scheme was *explicitly designed* to separate human players, and therefore cannot reliably prove that humans indeed exhibit play styles that are perceived as distinct by external observers. However, since the fingerprinting algorithm is a very straightforward

simplification of human behavior data, we believe that major contradictions between the automated scheme and human evaluations are unlikely. Obviously, the current algorithm is unable to detect subtle differences in individual play styles, thus the average similarity between different human players is high (0.74).

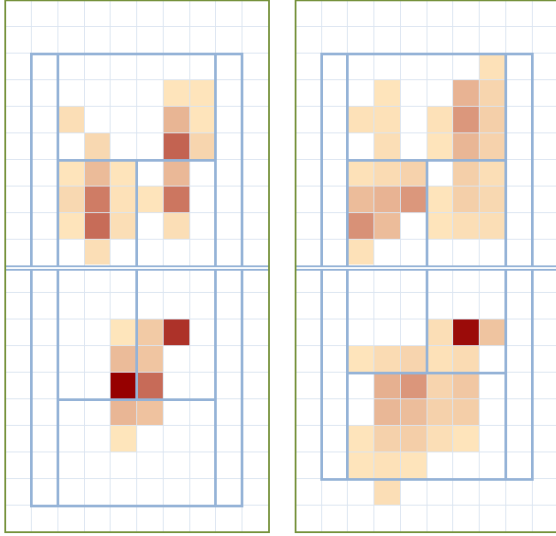


Fig. 3. Heat maps of H_1 , and H_2 (players are always in the bottom half of the court; the top half is used to show *SetHitPoint* targets).

Experiment 2. We repeated the steps of *Experiment 1* for the games played by AI “ghost” characters G_1, \dots, G_8 , and compared behavior fingerprints of human and “ghost” characters to obtain human/ghost and ghost/ghost similarity values (see Table IV and Table V).

Table IV shows that the “ghost” players indeed exhibit diversity of behavior, comparable to their human trainers. “Ghosts” tend to be consistent in their play style, and each “ghost” is clearly identifiable. Despite several exceptions (such as relatively low G_7 - G_7 similarity), on average behavior data of “ghosts” is comparable to the results shown by human players.

Table V demonstrates that our current algorithm is indeed able to produce AI “ghosts” exhibiting play style closer resembling their human trainers rather than other human players or other “ghosts”.

TABLE IV. AI “GHOST” PLAY STYLE SIMILARITY

G_1	0.91								
G_2	0.67	0.95							
G_3	0.74	0.77	0.92						
G_4	0.82	0.47	0.69	0.94					
G_5	0.75	0.44	0.64	0.72	0.96				
G_6	0.88	0.59	0.79	0.90	0.74	0.85			
G_7	0.84	0.49	0.69	0.86	0.81	0.87	0.80		
G_8	0.84	0.53	0.69	0.78	0.89	0.82	0.92	0.91	
	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	

$$\begin{aligned} \text{mean}(\{(G_i, G_j), i = j\}) &= 0.91, \sigma = 0.05 \\ \text{mean}(\{(G_i, G_j), i \neq j\}) &= 0.74, \sigma = 0.13 \end{aligned}$$

Experiment 3. We calculated the total number of points scored by the human players and the AI “ghosts” in league games, and visualized them (see Fig. 4). This experiment shows that the “ghosts” can play on par with human opponents. Their slightly better performance can be explained primarily with higher “discipline”: AI-controlled characters never miss their actions, while human players tend to play casually, and sometimes may skip the next hit or move point.

During the experiments, we collected additional data that may be useful to evaluate the resulting AI system. These numbers are summarized in Table VI.

TABLE V. HUMAN / AI “GHOST” PLAY STYLE SIMILARITY

	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8
G_1	0.89	0.82	0.66	0.77	0.73	0.75	0.82	0.84
G_2	0.38	0.89	0.69	0.47	0.35	0.50	0.47	0.62
G_3	0.57	0.80	0.95	0.63	0.50	0.75	0.63	0.68
G_4	0.85	0.61	0.61	0.94	0.70	0.92	0.80	0.81
G_5	0.85	0.57	0.69	0.60	0.97	0.69	0.70	0.60
G_6	0.83	0.74	0.73	0.86	0.71	0.88	0.82	0.84
G_7	0.86	0.69	0.70	0.77	0.79	0.79	0.88	0.87
G_8	0.86	0.73	0.73	0.70	0.88	0.73	0.84	0.83

$$\begin{aligned} \text{mean}(\{(H_i, G_j), i = j\}) &= 0.90, \sigma = 0.05 \\ \text{mean}(\{(H_i, G_j), i \neq j\}) &= 0.71, \sigma = 0.13 \end{aligned}$$

TABLE VI. ADDITIONAL EXPERIMENTAL DATA

Actions matched in FSM ₀	18%
Actions matched in FSM ₁	56%
Actions matched in FSM ₂	18%
No matching actions found	8%
Adjacent (strategy-keeping) actions	14%
Average AI “ghost” graph size (FSM ₀)	525 nodes

VII. DISCUSSION

Before discussing obtained results, let us once again express the view that the purpose of game AI is to support a particular game experience. This way, our goal was to create an AI system for a specific type of a mobile tennis game, and certain design decisions were influenced by game designers’ views of typical game/player interaction scenarios. Since the game is still in the development stage, we believe that both business requirements and the game engine will be updated, so we will have to amend certain AI elements accordingly.

Our first goal was to create a variety of believable and diverse characters. As a preliminary step, we had to make sure that the game engine itself allows the players to exhibit distinguishable play styles, since one may argue that the difference between two human players can be merely explained with their different reaction time and differently upgraded characters. However, Experiment 1 shows that each person tends to adhere to the same distinctly recognizable characteristic behavior pattern (at least over a short span of 10-20 game sessions), thus answer positively to the research question Q_1 . Therefore, believable AI characters should also be able to exhibit a variety of recognizable game play styles.

Experiment 2 confirms that our method is indeed able to produce believable and diverse AI “ghosts” that resemble their

human trainers (research question Q_2). The diversity of “ghosts” is comparable to the diversity of human players. A similarity score of two distinct “ghosts” in the experiment is 0.74 on average, while each “ghost” exhibits consistent play style across the game sessions (yielding a self-similarity score of 0.91 on average). These results are comparable to the similarity values obtained from human players in the Experiment 1. The similarity scores between the “ghosts” and their respective human trainers vary. In most cases, the closest match to a “ghost” is indeed its human trainer (with the average similarity score 0.90).

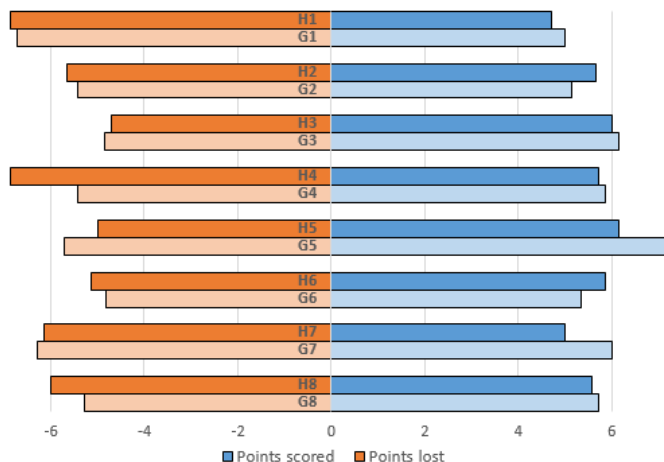


Fig. 1. Number of points lost and scored (on average per game) by the human players and their respective AI “ghosts”.

Experiment 3 demonstrated that in the AI “ghosts” are able to compete with people successfully, and in the present version of the AI system perform even slightly better than their human trainers (probably, due to their perfect discipline). From the business perspective, we are satisfied with this moderate handicap the “ghosts” have. Human players always play against the AI, so the players generally have no expectations about the skills of their opponents. At the same time, each player expects own “ghost” to perform equally well, so in this case stronger AI should be more appealing to the users.

VIII. CONCLUSION

We demonstrated a practical learning by observation-based method used to create an AI system for a mobile tennis game. The chosen approach provides reliable decision making, and is able to produce a variety of diverse human-like opponents, preserving play style of their human trainers. Believability and diversity of AI-controlled characters were demonstrated with an objective method based on behavior fingerprint comparison. Furthermore, the AI agents are able to achieve the same skill level as human players.

The proposed approach is based on the earlier AI project for the game of boxing, so we believe that this method can be adapted to a variety of game genres. At the same time, every game is different, and even small changes in a game engine or gameplay may require considerable redesign of AI system. In case of tennis, the core elements of decision making system were kept intact, however, the design of granularity levels, the

choice of attributes and queries had to be reconsidered. The resulting AI system is resource-efficient, and can work on an average (iPhone 4-class) mobile device.

Currently several core decision making subsystems (such as configuration of FSMs and the list of queries) are designed manually, on the basis of expert knowledge. While automating these tasks is not a priority for us now, it can be a fruitful topic for future research.

REFERENCES

- [1] N. Taatgen, M. van Oploo, J. Braaksma, and J. Niemantsverdriet, “How to Construct a Believable Opponent using Cognitive Modeling in the Game of Set,” in *5th International Conference on Cognitive Modeling*, 2003, pp. 201–206.
- [2] S. Johnson, “Playing to Lose: AI and “CIVILIZATION” (Lecture),” Game Developers Conference, 2008.
- [3] K. Dill, “What is Game AI?,” in *Game AI pro: Collected wisdom of game AI professionals*, S. Rabin, Ed, pp. 3–10.
- [4] *World of Tennis*. Available: www.worldoftennis.com.
- [5] P. Luban, *The Design of Free-To-Play Games*. Available: http://www.gamasutra.com/view/feature/6552/the_design_of_frektoplay_games_.php (2016, Mar. 31).
- [6] M. Mozgovoy and I. Umarov, “Believable team behavior: Towards behavior capture AI for the game of soccer,” in *8th International Conference on Complex Systems*, 2011, pp. 1554–1564.
- [7] M. Mozgovoy and I. Umarov, “Building a believable agent for a 3D boxing simulation game,” in *Second International Conference on Computer Research and Development*, 2010, pp. 46–50.
- [8] Bandai Namco Games, *Tekken 6 Instruction Manual (PSP)*, 2009.
- [9] A. Agapitos, J. Togelius, S. M. Lucas, J. Schmidhuber, and A. Konstantinidis, “Generating diverse opponents with multiobjective evolution,” in *IEEE Symposium on Computational Intelligence and Games*, 2008, pp. 135–142.
- [10] A. M. Mora, F. Aisa, P. García-Sánchez, P. Á. Castillo, and J. J. Merelo, “Modelling a Human-Like Bot in a First Person Shooter Game,” *International Journal of Creative Interfaces and Computer Graphics (IJCICG)*, vol. 6, no. 1, pp. 21–37, 2015.
- [11] Asensio, Joan Marc Llargues *et al*, “Artificial Intelligence approaches for the generation and assessment of believable human-like behaviour in virtual characters,” *Expert Systems with Applications*, vol. 41, no. 16, pp. 7281–7290, 2014.
- [12] I. V. Karpov, J. Schrum, and R. Miikkulainen, “Believable Bot Navigation via Playback of Human Traces,” in *Believable Bots: Can Computers Play Like People?*, P. Hingston, Ed, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 151–170.
- [13] J. Ortega, N. Shaker, J. Togelius, and G. N. Yannakakis, “Imitating Human Playing Styles in Super Mario Bros,” *Entertainment Computing*, vol. 4, no. 2, pp. 93–104, 2013.
- [14] G. Lee, M. Luo, F. Zambetta, and X. Li, “Learning a Super Mario controller from examples of human play,” in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 1–8.
- [15] M. Mozgovoy and I. Umarov, “Behavior Capture with Acting Graph: A Knowledgebase for a Game AI System,” in *Databases in Networked Information Systems (DNIS): 7th International Workshop*, S. Kikuchi, A. Madaan, S. Sachdeva, and S. Bhalla, Eds, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 68–77.
- [16] G. Robertson and I. Watson, “Building behavior trees from observations in real-time strategy games,” in *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2015, pp. 1–7.
- [17] J. Togelius, G. N. Yannakakis, S. Karakovskiy, and N. Shaker, “Assessing believability,” in *Believable bots*: Springer, 2013, pp. 215–230.
- [18] H.-K. Pao, K.-T. Chen, and H.-C. Chang, “Game Bot Detection via Avatar Trajectory Analysis,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 3, pp. 162–175, 2010.