Towards Self-Learning AI for the Videogame of Tennis

Akane Yamada
The University of Aizu
Tsuruga, Ikki-machi,
Aizuwakamatsu, Fukushima, Japan
+81-242-37-2664
m5191102@u-aizu.ac.jp

Maxim Mozgovoy
The University of Aizu
Tsuruga, Ikki-machi,
Aizuwakamatsu, Fukushima, Japan
+81-242-37-2664
mozgovoy@u-aizu.ac.jp

ABSTRACT
While behavior patterns of AI-controlled videogame characters are typically designed manually, self-learning AI systems possess unique attractive features. They can be used to create distinct character personalities of different skill levels, and "train your own character" can be a major user-end feature of a game. In this paper we present our work-in-progress attempt to develop a self-learning AI for the game of tennis. We show how an AI system can learn behavior patterns from user behavior, and act accordingly in similar game situations.

Categories and Subject Descriptors
I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems – *games.*

General Terms
Algorithms, Experimentation

Keywords
Case-based reasoning, learning by observation, behavior capture.

1. INTRODUCTION
Artificial intelligence systems that control game characters are essential to most computer games, ranging from simple arcade games like Pacman to modern virtual game worlds, inhabited with computer-controlled non-player characters (NPCs). Typically, books on game AI suggest a variety a methods that use handcrafted rules, decision trees, goal hierarchies, etc. [1] These well-established solutions, however, do not help game designers to create diverse and realistic behavior of NPCs, as each behavior pattern has to be designed by hand.

One possible way to remedy the situation is to use learning by observation techniques. We propose the following analogy: in mid-90s, the technology of *motion capture* revolutionized game animation, since the designers could use motion patterns obtained from real people instead of inferior hand-drawn animation. Similarly, handcrafted behavior cannot replicate the whole complexity of realistic NPC reasoning, so it can be reasonable to take the same approach: instead of designing behavior patterns, one can build a "behavioral knowledgebase" by observing human-controlled game characters.

Unfortunately, behavior acquisition is not a straightforward process, and it cannot be applied in a universal way to any computer game. However, for certain types of games this approach is feasible and not exceedingly complex. In this paper, we will describe our work-in-progress effort to create such a self-learning character for a video game of tennis. We will show that the character is able to learn a variety of complex strategies, and we will discuss our preliminary results on character acting.

*IWAIT'*15, Oct. 8–10, 2015, Aizu-Wakamatsu, Japan.

2. TENNIS GAME ENGINE
As a vehicle for the experiments, we use a custom-designed tennis game engine, developed with Unity3D [2] (see Figure 1).

Figure 1. Tennis game engine.

Each player in the game can perform one of the two actions: run to the specified location and shot the ball into the specified point on the court. The game physics is accurate, so, for example, if a player tries to send a ball coming at a high speed to a nearby point on the opponent's side of a court, a backspin shot will be performed, so the ball will fly high over the net.

3. AI LEARNING MODE
To learn practical tennis behavior patterns, we played a series of five mini-games, approximately 1-1.5 min each. During a single

game, each player typically performs 100-150 actions. Every game session was recorded to a file for subsequent processing.

Before learning behavior patterns, we stripped recording fragments containing undesirable actions (made due to human errors). In learning mode, our AI system processes actions of a specified player (so it learns from one player in the given game). When an action occurs, the system extracts observable features of the current game situation, and stores the resulting <situation, action> pair in a game graph (see Figure 2). In our current experiments we actually build a family of game graphs for the different sets of observable features. The most accurate graph uses 27 features, while the most abstract game representation relies on 15 features only. These principles are explained in more detail in our earlier work [3].

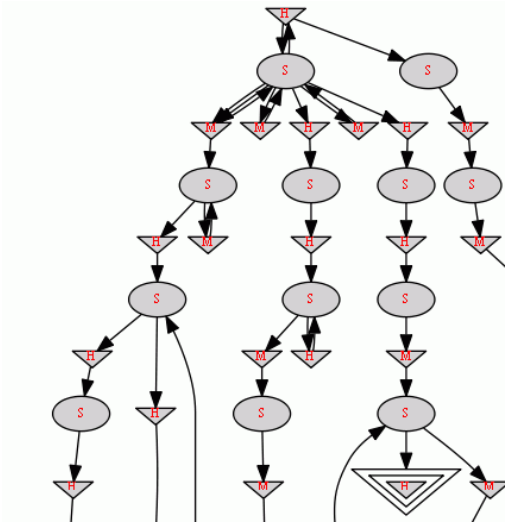

Figure 2. A fragment of a game graph (visualized with AT&T GraphViz).

This representation preserves the specific behavioral patterns of an observed player, since all the actions are stored in the graph.

Even at this stage, our system can be useful for the purpose of game analysis. Game graph provides a "bird's-eye view" on a game, and can reveal various game peculiarities, such as repeatability (whether two different game sessions share a significant fraction of identical situations), the differences in acting styles of distinct players, average duration of a game episode between subsequent ball servings, etc. We believe this type of instrument can be also applied in real conditions, where digitalized recordings of actual tennis games are used instead of video game data.

4. AI ACTING MODE

In AI acting mode the learned game graph is used to support decision making process. When a player needs to act, our AI subsystem polls the graph to find the matching game situation, and uses a weighted random choice to select one of the attached actions. The procedure also takes into account the previously performed action, and first checks whether the next game situation in the action chain matches the current game situation, so the next action in the chain can be applied.

If the search yields no result, the procedure is repeated for a less detailed graph (learned with fewer observable attributes). The algorithm fails only if no search returns a matching action.

Our preliminary experiment show that the current agent is able to reproduce typical game patterns such as serving and responding to an opponent's serve. However, the agent often fails to reproduce longer strategies, involving over 7-10 consecutive actions. We are actively working on these issues; previously we successfully addressed similar problems in a game of boxing [4].

One of primary challenges of tennis is caused by the necessity to determine reasonable decision making points. Human player can act at any time, but as stated above, a typical observed acting rate is around 100 actions per minute. Therefore, our AI system should also act with certain non-uniform frequency, reflecting typical tennis gameplay strategy. In the current implementation the AI system acts every 10 frames, which causes frequent change of behavior patterns. It should be noted, though, that this problem is relevant to any tennis AI system regardless of the underlying approach.

5. CONCLUSION

Game AI is a very specialized topic, since computer games set unusual requirements for AI, such as human-likeness, unpredictability, skill level adjustments, and *fun factor*, a core of any entertainment. Self-learning AI systems have a potential to address these challenges, since they can directly learn from human opponents, and thus exhibit distinct human-like behaviors of different skill levels.

In this paper, we briefly discussed our work-in-progress AI systems for the game of tennis. While our AI did not reach a production-quality stage yet, it demonstrated the ability to learn and repeat simple behavioral patterns of human players. We plan to achieve the desired quality of acting within this year.

6. REFERENCES

[1] Millington, I., Funge, J. Artificial Intelligence for Games, 2nd Ed. CRC Press, 2012, 896 p.

[2] Unity 3D Game Engine. Project website: http://unity3d.com [Accessed: 20 July 2015].

[3] Mozgovoy, M., Umarov, I. Behavior Capture with Acting Graph: a Knowledgebase for a Game AI System. *Lecture Notes in Computer Science*, 2011, vol. 7108, pp. 68-77.

[4] Mozgovoy, M., Umarov, I. Building a Believable and Effective Agent for a 3D Boxing Simulation Game. *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology*, vol. 3, Chengdu, China, 2010, pp. 14-18.