Incremental Learning Algorithm for Online Action Game System

Junichi Hoshino and Hiroshi Mori

University of Tsukuba, Graduate School of Systems and Information Engineering,
1-1-1, Tennodai, Tsukuba-shi, Ibaraki, Japan
{jhoshino,hmori}@esys.tsukuba.ac.jp

Abstract. One of the limitations of computer opponents in action games is that the character AI is constructed in advance, and players may become bored quickly. We have built an online action game system in which a non-player character (NPC) can incrementally learn sequences of action and combinations. NPCs can adopt different fighting strategies for fighting with different players.

Keywords: action game, imitation learning, non-player character.

1 Introduction

One of the limitations of computer opponents in action games is that the character AI is constructed in advance, and players may become bored quickly. A human player can play the same game repeatedly, learn the behavior of the computer-controlled game character, and win easily. This is one major reason why human players soon grow tired of "fighting with a computer." Thus, many players prefer playing with another human player to playing with a computer.

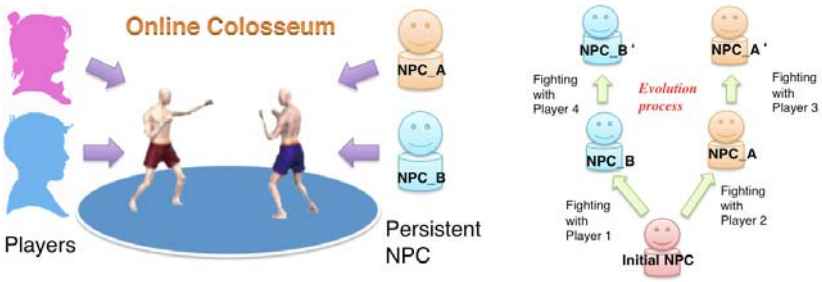
In this paper, we propose an online action game system in which a non-player character (NPC) can incrementally learn sequences of action and combinations in real time. NPCs can adopt different fighting strategies for fighting with different players. Individual fighting styles can be generated from the unique fighting history. We have developed a new engine of action learning that analyzes a human player's action pattern automatically and extracts the effective fighting sequences. Action control trees are generated automatically and incrementally added to a player's action profile.

2 Learning Game Character

Fig. 1 shows the concept of an online colosseum game system in which non-player characters (NPCs) can incrementally learn sequences of action and combinations by fighting with human players. NPCs can adopt different fighting strategies for fighting with different players. Fig. 2 shows the growing process of NPCs. Individual fighting styles can be generated from the unique fighting history.

S. Natkin and J. Dupire (Eds.): ICEC 5709, pp. 319–322, 2009.

A player profile is created from the play data of the imitated player. The information recorded in a player profile includes the player's tactics, which we discussed in the preceding sections, the tactic sequences that represent the player's strategies, and the frequency with which each of these sequences appears. When the system applies a profile to the actual playing of the game, it reads the tactics and strategies and creates a chart of the tactic graph based on the tactic sequences (strategies) and stores it in the pool of graphs. While playing a game, the computer chooses a tactic graph that matches the selection criteria closest to the situation at a specific time, chooses the tactics in the tactic graph that corresponds to the situation that such a tactic graph refers to, and applies the actions to the game character by referring to these tactics. In the following sections we will discuss how the tactics and strategies are chosen.



(a) (b)

Fig. 1. Concept of an online action game system. (a)Many players and persistent NPCs can fight each other on the virtual colosseum. (b)NPCs can obtain different skills by fighting with different players.

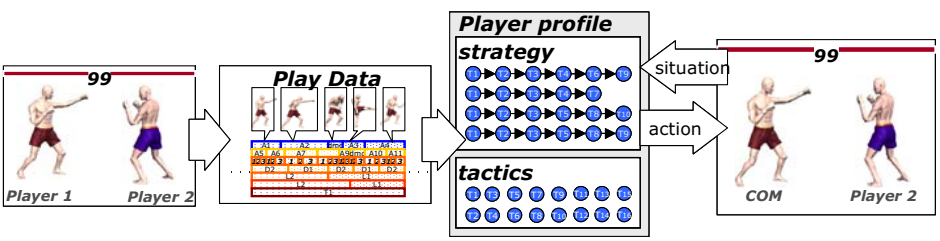


Fig. 2. Overview of the player profile sytem

3 Learning Strategy from a Human Player

1) Common Sequences of Tactics

Searching for a usable action pattern from accumulated action patterns is a technique often used for creating fighting strategies. In this paper, this is called method 1. The

sequences of tactics in the player profile, created from the game data, are evaluated, and patterns that the opponent used repeatedly are given a high value. However, if a tactic is used too much, it might not be imitated. A certain value is chosen that represents the ratio of that action to the total length of the sequence of tactics. The standard for what sequence of tactics is suitable to imitate changes depending on this value. Therefore, setting of this at a different in various characters affects how the character evolves.

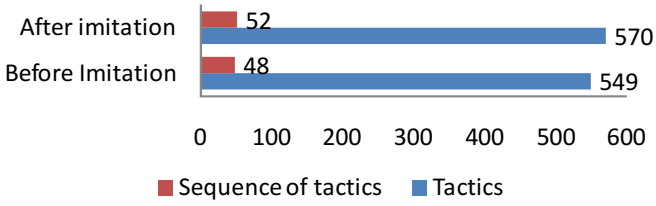
2) The Sequence of Tactics Contains Actions That the NPC Executes
This method of imitating adversary tactics includes using actions that we execute effectively. In this paper, this is called estimation method 2. In this method, new actions that the NPC does not execute are not added, because only actions that the NPC has already used are evaluated. This evaluation adds new conditions for executing actions that have already been used, and new sequences of tactics composed of existing actions.

3) Sequences Containing Actions that the NPC Does Not Execute
Imitating the tactics of an adversary that do not include actions that we execute effectively is a method used when a person imitates how to fight others. In this paper, we call this estimation method 3. In this method, the new actions—ones that the NPC does not execute, are added by evaluating sequences consisting only of these actions. Therefore, adding these sequences enables the NPC to execute new actions. The evaluation value is determined from the success rate, the occurrence count, and the sequence length, as in the previous evaluation method.

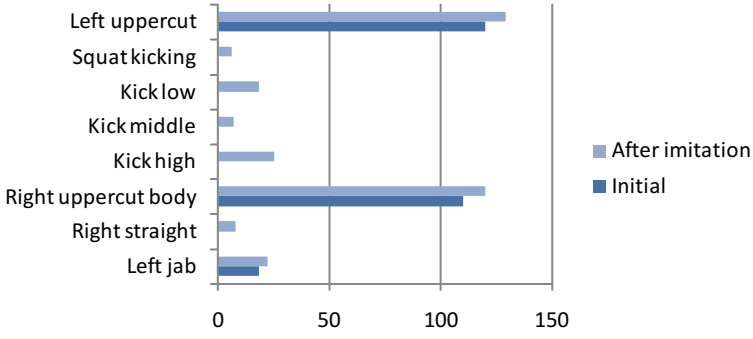
4 Experiments

Fig. 3 shows a screen shot of our fighting game. The initial player profile was made from the play data of 10 games. This initial player profile contains 549 tactics and 48 sequences of tactics. A player played ten games against the NPC based on this player profile, and we checked the added tactics and sequences of tactics as a result of the ten games. The probability of imitating the adversary's sequence was set high because it was an evaluation for only 10 games. As shown in Fig 4(a), four sequences and 21 tactics were added after 10 games. Fig 7(b) shows the number of the major actions.

Fig. 3. Screen shots of the fighting game



(a)The number of tactics and sequence of tactics



(b) The number of the major actions.

Fig. 4. Result of imitation learning

5 Conclusion

In this paper, we proposed a system that enables a computer character to imitate a human player. To do so, the system first acquires tactics and tactic sequences from play data of a player. Then, from the tactic sequences collected it creates tactic graphs that represent the strategic actions of the player. From these graphs, the system selects tactics that suit different situations. We also demonstrated the effectiveness of the system in an evaluation experiment. Furthermore, we created many different behavioral patterns for the computer by changing player profiles, which are the collections of tactic sequences and tactic graphs of the particular players.