Identifying player behavior styles in an arcade fighting game

Toru Ito s1210045 Supervised by Maxim Mozgovoy

Abstract

We analyzed human and AI player behavior styles in Universal Fighting Engine, and built a visualization tool for the convenience of analysis. In the experiment, we examine characters' internal status and actions. We calculated cosine similarity to compare players by examining combo chains and individual actions. Our experiments show that both human players and AI agents exhibit distinguishable behavior patterns.

1. Introduction

The technology of artificial intelligence is developing every year. Building AI that behaves like a human is a goal of many research projects. We are interested in AI that behaves like a human, such technologies have numerous applications in computer games and simulations. In our research, we are aiming to build AI behaves like a human in an arcade fighting game. As a preliminary step, we think it is helpful to analyze human behavior styles in the game. In this paper, we will examine whether a human's playstyle of arcade fighting game can be reliably identified.

2. UFE: experimental testbed

In our experiments, we used Universal Fighting Engine (UFE, Figure 1) [1]. It is a publicly available game engine for Unity development environment [2], and convenient for our purposes. Players can operate game characters by controlling six attack buttons and four-direction keys. Also, the players can make own character perform a special action such as fireball and uppercut when the player presses some keys consecutively. Furthermore, the players can perform additional actions by using unique commands when a special gauge in the bottom of the display is full. In addition, we developed a visualization tool (Figure 2) for experiment's convenience. This tool shows us internal description of each game snapshot, including attributes such as character internal state, location, hit points, special gauge, etc.

Figure 1. Universal Fighting Engine

3. Visualization tool

This tool uses GDI+ [3] for drawing game situations. When we analyze player's behavior, we get dumps of game situations in a log file. We use that log file to visualize game situation. The log file contains character's position (x y z coordinates), current state (constructed by character's state that is clearly recognized by a human observer), current sub state (additional features of character's state that is difficult to recognize), current basic move (non-attack action and passive action), current move (attack action), and more status fields such as hit points and special gauge. When a log file is imported, this tool picks these values on every frame, and visualizes them using a simplified

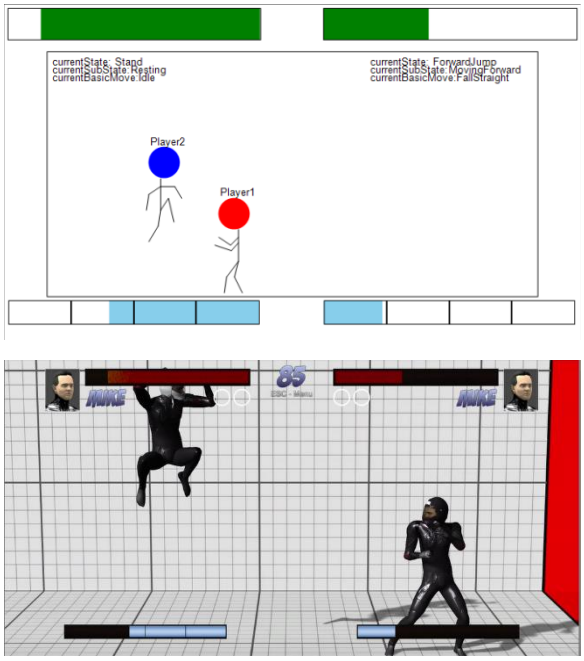

Figure 2. Above: Visualization tool; below: Game window showing the same frame

graphical representation. For example, when this tool reads character's state, it draws a picture that is equivalent to character's state. Some information that is difficult to display as a picture is displayed in textual form in window corners. The main purpose of this visualization tool is to display detailed information about the game states on every frame. It is not available in the real game window, so it is difficult to understand the details of each game situation just by watching game movie clips. The main screens of the visualization tool (Figure 2, above) and the real game window (Figure 2, below) show roughly the same information about characters (such as location and posture). However, a separate "GS Tree View" window of the visualization tool shows numerous additional elements of the game state, not available in the real game window (Figure 3). The posture of a character in the visualization tool main window is determined by the value of `currentState` field in the game state. There are four possible values, and hence four possible postures for now; however, we are planning to support more types of posture by analyzing the value of `currentMove` in the future.

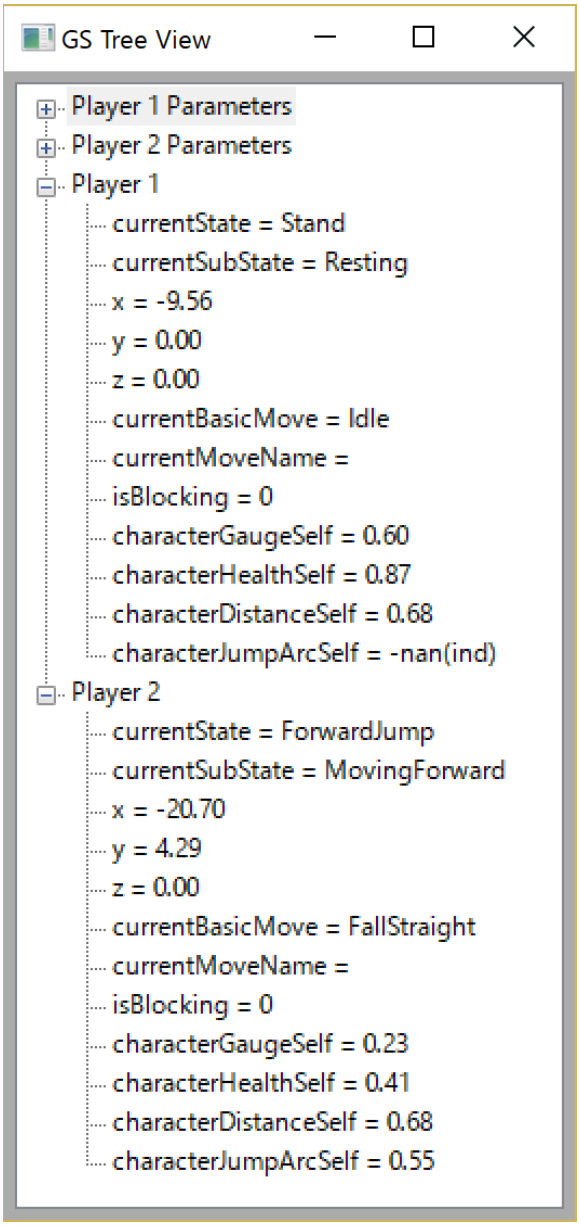

Figure 3. GS Tree View

4. Experimental method [4]

There are some methods to analyze player's

behavior, such as analyzing by watching playing movie by testers (what is called Turing test) [5]. In the following experiment, we tried to distinguish play styles of individual characters by analyzing their actions.

・We relied on five human players A-E and three difficulties of AI players (Ve: very easy, No: normal, Im: impossible). The AI players are provided with UFE.

・These players played three sets of matches using a round-robin scheme. The resulting game logs were recorded.

・To analyze the similarities in play styles, we compared players' "behavior fingerprints" obtained with two different methods. The first method is to prepare a vector of probabilities of individual actions in a certain player's game log. This vector serves as a "behavior fingerprint" and identifies player's play style. The second method is to prepare a matrix of probabilities of two consecutive actions in the game log. To get such combo data, we had to count frequencies for each combination of two possible actions in the game log, and divide all frequency values by the total frame count.

・To compare the fingerprints of individual players, we used cosine similarity measure [6].

$$similarity = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

For example, when we calculate similarity between player A and player B, we begin by calculating behavior fingerprints of A and B using all games where A or B participates, such as player A vs player C, and player B vs player E. Then, we compare the obtained fingerprints by using round-robin scheme, and calculate the average of obtained similarities. In order to compare matrices, we first convert them into vectors by writing down matrix content row after row. In order to calculate similarity between player A and player A, we begin by preparing fingerprints that A participates (A vs B, A vs C, …, A vs Im). Then, we compared the obtained fingerprints by using round-robin scheme, and then, we calculated obtained similarities average. However, the same fingerprint such as A vs B and itself are not compared, because it is self-evident that the similarity of same fingerprint is 1.0.

・After comparing cosine similarity for each possible player pair, we drew conclusion.

5. Results

Table 1. Probabilities of individual actions

punch	uppercut	kick	fireball	…
0	0.0056	0.1124	0.0107	…

Table 2. Probabilities of action combos

	Light kick	Medium kick	Heavy kick	…
Light kick	0	0	0	…
Medium kick	0	0.0667	0.121	…
Heavy kick	0	0.0181	0	…
…	…	…	…	…

Example behavior fingerprints of players obtained with the described above methods are shown in the Table 1 (individual actions) and Table 2 (action

combos). The size of table 1 is 33 rows, because in this game, character has 33 attack actions. In the same way, the size of table 2 is 33×33. In the Table 2 the column name denotes the first action in chain, while the row name is the second action in chain. For instance, the value 0.0181 in the Table 2 corresponds to the sequence of a medium kick and a heavy kick.

Table 3. Cosine similarity (combo chains)

A	**0.813**							
B	0.456	**0.848**						
C	0.314	0.360	**0.850**					
D	0.342	0.557	0.662	**0.688**				
E	0.478	0.659	0.604	0.617	**0.733**			
Ve	0.440	0.715	0.313	0.526	0.559	**0.907**		
No	0.445	0.436	0.439	0.504	0.500	0.507	**0.798**	
Im	0.446	0.690	0.364	0.537	0.584	0.730	0.598	**0.798**
	A	**B**	**C**	**D**	**E**	**Ve**	**No**	**Im**

Table 4. Cosine similarity (individual actions)

A	**0.894**							
B	0.465	**0.780**						
C	0.262	0.287	**0.964**					
D	0.258	0.385	**0.780**	0.733				
E	0.491	0.601	0.708	0.628	**0.745**			
Ve	0.464	0.551	0.419	0.507	0.531	**0.732**		
No	0.457	0.279	0.543	0.547	0.453	0.609	**0.919**	
Im	0.485	0.518	0.450	0.559	0.509	0.730	0.815	**0.856**
	A	**B**	**C**	**D**	**E**	**Ve**	**No**	**Im**

Table 3 shows player cosine similarity values calculate on the basis of Table 2 (combo chains). For example, the similarity of players in a pair A-A is 0.813. It is the highest value in all match player A relates. The Table 4 shows player similarities calculated using a cosine similarity value for vectors of probabilities of individual actions. In general, we can see a similar tendency in these values and the previous table values, such as high similarity between the fingerprints of the same player, and much lower similarity between the fingerprints of distinct players. The only exception is the pair C-D, that is higher than D-D.

6. Discussion and Conclusion

Considering combo chains, high cosine similarity appears in the analysis of the same player's fingerprints. There are some lower cosine similarity values such as for the pair D-D, however, it is still higher than any other pair that includes D. Therefore, we can say that an individual human's playstyle can be identified by analyzing character actions forming a combo chain.

However, analyzing individual actions is less reliable, as shown with the high similarity value between the player C and D in the Table 4. It means that the player C is more like the player D than the player D oneself, which is unexpected. We think this situation shows that individual actions are not enough to build a precise behavior fingerprint. For example, suppose there are two combos punch-punch-kick and punch-kick-punch. When we analyze their actions

independently, these two combos consist of the same actions (two punches and one kick). However, if we consider combinations of two consecutive actions, we will get distinct “combos” (punch-punch, punch-kick and punch-kick, kick-punch). Therefore, cosine similarity of combo chains takes into account more details. Furthermore, we calculated average similarity of same players and different players’ using both method of similarity assessment, and compared their difference. We found that difference of average in combo chain is larger than for individual actions. Therefore, we can say that using cosine similarity of combo chains is superior to individual actions. Finally, we conclude that identifying human playstyle is possible, but we recommend relying on combo chains rather than individual actions for building behavior fingerprints.

We think this research will helpful to analyze human playstyle, and to build AI that behaves a human eventually. We hope that our research will contribute to develop computer fighting game AI and the AI that behaves like a human.

References

[1] Universal Fighting Engine (UFE). http://www.ufe3d.com

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

[3] GDI+ Graphic. https://msdn.microsoft.com/ja-jp/library/aa984108(v=vs.71).aspx

[4] Toru Ito, Tatsuhiro Rikimaru, “Tracing Human Behavior Styles in a Computer Fighting Game”, Symposium on Big Data Analytics in Science and Engineering, 31 Oct. – 2 Nov. 2016.

[5] P. Hingston, “A Turing Test for Computer Game Bots,” IEEE Transactions on Computational Intelligence and AI in Games, vol. 1, no. 3, September 2009, pp. 169-186.

[6] Nguyen, Hieu V., and Li Bai. "Cosine similarity metric learning for face verification." *Asian Conference on Computer Vision*. Springer Berlin Heidelberg, 2010.