Turing test for player characters in an arcade fighting game

Tatsuhiro Rikimaru s1210063 Supervised by Maxim Mozgovoy

Abstract

This paper describes a method used to conduct a Turing test for player characters in an arcade fighting game. The idea to implement a human-like AI in a fighting game is attractive, but it makes sense only if people are able to identify distinct play styles in the game, i.e. the game engine provides enough flexibility for human-like behavior. We verify that people are able to identify players' playing styles, and that human players are distinguishable from AI-controlled opponents. This is accomplished with two types of Turing test: (1) Matching game clips, and (2) Grouping game clips. As a result, we show that people are really able to identify players' playing style, and that human players are distinguishable from AI-controlled opponents.

1 Introduction

A computer-controlled player (or a non-player character, NPC) needs to emulate human behavior to entertain people when they play video games together with an NPC. Especially, in a fighting game where a person versus person play is the real fun, it is important to implement diverse AI behaviors for a variety of computer-controlled opponents. Fighting games are regarded as a type of Electronic Sports (e-Sports) and competitions such as "Evolution Championship Series" [6] are held. Then, most people study playing style of others and practice to win in competitions or defeat a particular player. However, it is difficult to really fight against target opponents to practice. Therefore, it will be useful if AI that imitates an individual playing style is developed. However, if people are unable to identify playing styles and distinguish human and AI characters, there is no real possibility for the AI to imitate human players.

In the present work, we verify that people are able to identify players' playing styles, and that human players are distinguishable from AI-controlled opponents by using two types of a Turing test.

2 Game and AI

2.1 Universal Fighting Engine

We use Universal Fighting Engine (UFE) due to its flexibility. In particular, it allows to perform a match between two AI-controlled opponents, an option not typically available in most fighting games. UFE is designed to help to make 2.5D/3D fighting games with Unity 5 [1, 2] (see Figure 1). Therefore, it provides numerous options needed for fighting game developers.

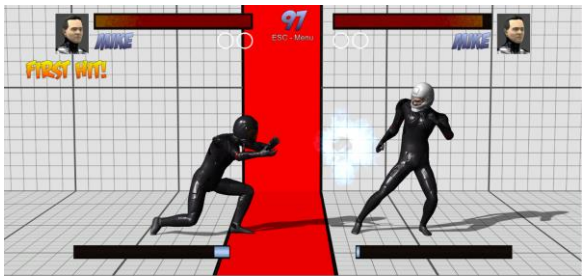

Figure 1. Screenshot of Universal Fighting Engine.

The players in this game can use 10 different keys to perform game actions, such as move right, jump, punch and kick. Furthermore, the player can let the own character perform a special action ("combo"), when the player presses some keys consecutively.

2.2 Fuzzy AI: UFE Addon

This add-on uses fuzzy logic to evaluate the information of the scene and calculate the desirability of each given action, translating the AI decisions directly into user input [3].

In this paper, we use three AI-controlled characters, based on different Fuzzy AI settings:

- Very easy (acts slowly, and is not aggressive, and does not use combos).
- Normal (it acts fast, and is little aggressive, and uses some combos).
- Impossible (it acts rapidly, and is more aggressive, and uses many combos).

In the experiments we used two AI agents in each Turing test (the first test used Very easy and Normal, the second test used Normal and Impossible) to get more data to evaluate.

3 Turing Tests

The original Turing test is an "imitation game" proposed by Turing [4]. In the imitation game, there are three persons: a guesser, a woman and a man (who is an opponent). The guesser knows other two people as A and B, and his task is to determine which of them is the woman. Then, he asks them some questions,

and their answers are exchanged by using notes or chat. Next, the subjects A and B are substituted with a man and a computer, and the game is played again. Finally, the results of a man and a computer are compared, and we can determine whether computer has intelligence.

Computer Game Bot Turing Test is a test for game AI based on imitation game [5].

In order to verify whether human players exhibit identifying play styles and whether human player are distinguishable from AI-controlled characters, we prepared two variations of a Turing test.

3.1 Matching Game Clips Test

We show to human evaluators five game clips of players A-E (Human-1, Human-2, Human-3, Very easy AI and Normal AI) against a random opponent. After that, we show them five more clips of the same A-E players against a random opponent We ask to pay attention to only to the character corresponding to the players A-E.. Next we ask them, not taking into account the outcome of the game, to decide:

- Question 1: Which clip contains the same player in both first 5 clips and second 5 clips.
- Question 2: Which A-E player is human or AI in each piar.
- Which factors are most valuable in making a guess.

A tester gets one point for each correct pair or answer, and we get them to do the above procedure twice. Therefore, the best possible score is 10 for each question.

3.2 Grouping Game Clips Test

For this test we prepared the clips of the players A-E consisting of three human and two AI (Normal AI and Impossible AI), and a random AI player for the opponent. Each of five players fights against an opponent AI three times, and we record those games. We showed these 15 clips to the testers, and ask the testers to:

- Question 1: Find which clips belong to each player A-E.
- Question 2: Separate the players into the “human group” or “AI group”.
- Note whatever they notice in each player’s behavior.

The tester gets 2 points if three chosen characters of clips are controlled by same player. The tester gets 1 point if two out of three of them are controlled by same player, so the maximum score is 2*5 = 10. When each players' controller of group and controller who chosen by tester are same, tester get 1 point. For example, one tester guesses that players' controller is human on the group 1 (corectly details are human, human and AI), then tester gets 2 points. So, the maximum score is 3*5 = 15.

4 Random Guessing

The obtained results of the Turing tests need to be compared with “baseline performance” obtained from random guesses of answers. We calculate the baseline performance by using the following random guessing algorithm that outputs results similar to our Turing tests.

4.1 Random Pairing Algorithm

This algorithm outputs the same type of result as described in the Matching game clips test, but the five resulting pairs are created by pseudorandom guess. We generate a vector `vec` of five distinct random values that serve as pairs for the values of 0…4. This way, the final random pairing would be (0, `vec[0]`), …, (4, `vec[4]`).

```
std::vector<int> vec{0,1,2,3,4}
std::random_shuffle(vec.begin(),
                    vec.end())
int PairTestScore = 0
FOR i = 0...4
   If(CorrectAns[i] == vec[i])
    PairTestScore++
```

Similarly, to guess whether a certain player is a human or an AI system, the algorithm generates a random number (0 = human or 1 = AI) and checks the guess knowing that the players 0-2 correspond to humans in the system, while the players 3-4 correspond to the AI.

```
int AITestScore = 0
FOR j = 0 … 4
   int checker = random_device() %
2
   If(0 <= vec[j] <= 2)   vec[j] =
0
   If(3 <= vec[j] <= 4)   vec[j] =
1
   If(vec[j] == checker)
    AITestScore++
```

4.2 Random Grouping Algorithm

This algorithm outputs a result as same as Grouping game clips test, but each of 3 players per a group are

decided by pseudorandom number. We generate a vector `vec` of fifteen distinct random values, then we unify numbers that serve as groups for the values of 0...4. The 3 consective elements of `vec` are considered a group (`vec`[0]...`vec`[2] are group 1, `vec`[12]...`vec`[14] are group 5).

```
std::vectoer<int> vec{0,1,…,13,14}
std::random_shuffle(vec.begin(),
vec.end())
```

```
FOR i = vec.begin()...vec.end()
   If(0 <= *i <= 2) *iterator = 0
   If(3 <= *i <= 5) *iterator = 1
   If(6 <= *i <= 8) *iterator = 2
   If(9 <= *i <= 11) *iterator = 3
   If(12 <= *i <= 14)*iterator = 4

int GroupTestScore = 0
For j = 0...4
   If(vec[3*j] == vec[3*j+1]
           == vec[3*j+2])
         GroupTestScore += 2
   If(vec[3*j] == vec[3*j+1] or
     vec[3*j] == vec[3*j+2] or
     vec[3*j+1] == vec[3*i+2])
         GroupTestScore += 1
```

In the task of distinguishing between human and AI, group (elements number 0-2 are human and 3-4 are AI) was chosen randomly whether human or AI like in the Grouping game clips test to use remainder operation. The variable checker is positive integer get randomlly 0 or 1, 0 means human and 1 means AI.

```
AITestScore = 0
FOR j = 0 … 14
   checker = random_device() % 2
   If(0 <= vec[j] <= 2) vec[j] = 0
   If(3 <= vec[j] <= 4) vec[j] = 1
   If(vec[j] == checker)
    AITestScore++
```

5 Results

Let us first discuss a “Matching game clips” Turing test. We carried out this experiment with 10 people and simulated the test by running a random pairing algorithm 200 times. Table 1 is first Turing test targets details, Table 2 and Table 3 are the results of each test. Table 2 shows that the testers can be roughly divided into three groups: three “skillful” testers scored 7-8 points, four “average” testers scored 5-6 points, and three “poor” testers scored only 2-4 points. Interestingly, the results of our two tests (“pairing” and “human/AI”) seem not to be related. For example, tester 1 scored well on the first test, but performed poorly on the second test. Similarly, tester 6 got a high score on the second test, but scored average on the first test. Furthermore, there is no clear relationship between the testers’ experience in fighting games and obtained test scores For example, tester 8 and tester 10 have not played a fighting game, but tester 8 get a high score, tester 10 get a low score.

Table 2. Turing test terget detail

	Age	Gender	Fighting game experience
Tester 1	22	Man	10 - 50 hours
Tester 2	22	Man	Over 100 hours
Tester 3	22	Man	1 - 10 hours
Tester 4	22	Man	1 - 10 hours
Tester 5	22	Man	10 - 50 hours
Tester 6	22	Man	10 – 50 hours
Tester 7	21	Man	Over 100 hours
Tester 8	21	Man	Nothing
Tester 9	21	Man	Over 100 hours
Tester 10	22	Man	Nothing

Table 2. Matching Scores

	Pairs Score	Human/AI Score
Tester 1	8	4
Tester 2	6	8
Tester 3	5	5
Tester 4	8	6
Tester 5	5	6
Tester 6	5	8
Tester 7	4	4
Tester 8	7	0

Tester 9	3	3
Tester 10	2	6
Average Score	5.3	5
Standard deviation	2	2.4

In Table 3, a random routine got scores that making correct pairs become unipolar with low score: average is low score too.

Table 3. Random Pairing Algorithm Scores

	Pairs pattern	Human/AI pattern
Average Score	1.9	4.8
Standard deviation	1.4	1.6

Next, we carried out the “Grouping game clips” Turing test we with the help of 9 people and simulated the test by running the random grouping algorithm 200 times. The Table 4 show second Turing test targets details, and the Table 5 show the results of it. In Grouping scores has become polarized with high score and low score, so standard deviation is higher than others value. However, “human/AI” scores become unipolar with high score and average score is high. Same as “Matching game clips” Turing test, the results of “grouping” and “human/AI” seem not to be related. For example, tester 4 scored well on the first and second test, but tester 6 got the highest score on second test in spite of low score on the first test. Also, It is not necessarily tied to high score that tester has played fighting games for a long time: in tester 6 and tester 8, the experience of playing fighting games is over 100 hours, but they did not score well on the first test.

Table 4. Turing test terget detail

	Age	Gender	Fighting game experience
Tester 1	22	Man	10 - 50 hours
Tester 2	22	Man	1 - 10 hours
Tester 3	22	Man	1 - 10 hours
Tester 4	22	Man	10 - 50 hours
Tester 5	22	Man	10 – 50 hours
Tester 6	21	Man	Over 100 hours
Tester 7	21	Man	Nothing
Tester 8	21	Man	Over 100 hours
Tester 9	22	Man	Nothing

Table 6 show that random program find whether correct group or not. These random answers become unipolar with l ıe as Table 2.

Table 5
Grouping Scores

	Grouping Score	Human/AI Score
Tester 1	3	9
Tester 2	6	8
Tester 3	7	7
Tester 4	8	9
Tester 5	5	8
Tester 6	2	10
Tester 7	7	9
Tester 8	1	8
Tester 9	3	9
Average Score	4.7	8.6
Standard deviation	2.5	0.88

Table 6. Random Grouping Algorithm

	Selected pattern	Human/AI get correctly
Average Score	2.1	7.5
Standerd deviation	1.2	1.9

6 Discussion

In question 1 on the first Turing test, we compare person's performance with a random performance. The human average is 5.3 points, and the random average is 1.9 points. The human's point is considerably higher,

so this test show that people really can identify playing styles, and there are some people who have great identifying skills.

In question 2 on the first Turing test, the human average is 5 points, and the random average is 4.8. These values are comparable, so while humans point is higher that the score of a random guesser, we consider this result as inconclusive, so we cannot be sure that human players are distinguishable from AI-controlled opponents in this game.

In question 1 on the second Turing test, the human average is 4.7 points, and the random average is 2.1 point. The human score is considerably higher, , so this test again demonstrate that people are able to identify distinguishable playing styles.

In question 2 on the second Turing test, the human average is 8.6 points, while the random average is 7.5 points. Here the difference between the human graders and the random procedure is higher than in the previous test, so we tend to believe that people are able to distinguish human players from AI characters.

As a result of two conducted Turing tests we conclude that people can identify player's style, and human players are generally distinguishable from AI-controlled opponents.

7 Conclusion

In this paper, we show that people can identify player's playing style, and human players are generally distinguishable from AI-controlled opponents. We also show that Turing tests participants do not need experience of playing fighting games or to play it for a long time. However, we did not identified a profile of a person with high guessing skills. Therefore, we believe that it is necessary to increase the number of participants.

We hope that this work will help to build high-quality AI systems for fighting games that are able to mimic individual human behavior.

Reference

[1] UniversalFightingEngine (UFE). http://www.ufe3d.com/doku.php

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

[3] UniversalFightingEngine (UFE) A.I. Editor. http://www.ufe3d.com/doku.php/ai:start

[4] Turing, Alan M. "Computing machinery and intelligence." Mind, vol. 59, no. 236, pp. 433-460, 1950.

[5] Hingston, Philip. "A Turing test for computer game bots." IEEE Transactions on Computational Intelligence and AI in Games, vol. 1, no. 3, pp. 169-186, 2009.

[6] Evo Championship Series. http://evo.shoryuken.com