Proc. Asia Simulation Conference 2009 (JSST 2009), October7-9, 2009, Shiga, Japan.

Implementation of a Human-Like Bot in a First Person Shooter: Second Place Bot at BotPrize 2008

Daichi Hirono[1] and Ruck Thawonmas[1]

[1] Graduate School of Science and Engineering, Ritsumeikan University, Japan

Abstract— A first person shooter game is a game genre where fun in the game resorts to the skill of the opponents. In particular, when the opponents are bots, adjustment of their skill levels as well as implementation of their human-likeness are important research issues that many research groups have been studying recently in order to make bots more interesting to play with. This led to a contest called BotPrize 2008 held in December 2008 where bots and human players competed with each other for human-likeness. In this paper, we analyze play video clips of all bots participated in this contest, including our bot ranked 2nd in terms of human-likeness, and derive five factors related to bot's human-likeness. In addition, we further improve our bot considering these five factors and conduct an experiment comparing our original bot, the one submitted to the contest, and the improved one. Experimental results confirm the effectiveness of our improvements, showing that the human-likeness of the improved one has risen significantly.

Keywords—FPS, Unreal Tournament, Human-likeness, Believability, Bot

1 Introduction

Amusingness of a match with bots in First Person Shooter (FPS) games is related to their level of humanness. Hence, if bots can conduct better humanlike actions, players will enjoy the game more. Because of this, a contest called BotPrize 2008 [1] was held in December 2008 for the purpose of evaluating the human-likeness of bot behaviours. In this contest, one of five human judges entered Unreal Tournament 2004, the FPS game in use, together with two other characters, one bot and one human-player, and scored these two characters the humanness level from 0 to 4 while fighting against them.

Using a platform called Pogamut 2 [2], we developed an FPS bot, named ICE, and submitted it to BotPrize 2008. ICE was ranked second among those bots participated in the final round. In this paper, we describe the outline of our original bot and its weak points. Then we describe how to overcome such weak points and discuss the comparison between our original ICE, the one submitted to BotPrize 2008, and the improved one.

2 Related Work

There exist a number of researches on strategy decision in FPS games in recent years. For example, a team of bots in [3] uses Hierarchical Task Network (HTN) planning to decide their strategy in a team-battle. The Sarsa algorithm is used in [4] for the purpose of reinforcement learning on item-collection and battle-strategy. The work in [5] adopts the neural networks for their bot in order to improve selection of battle strategies by mimicking human strategies.

Researches on human-likeness in terms of movements and actions in the FPS include [6] and [7]. Therein, a bot was proposed that imitates human behaviour in Quake II, an FPS game. In the former paper, the bot imitates a path used by a human player to collect items. Based on human-player play log, it decides the next item to collect by using fuzzy clustering and the path by using reinforcement learning. In the latter paper, the bot imitates human player's behaviour during the battle. It decides which weapon to use, position to aim, and timing to fire by using three interconnected neural networks. However, these researches do not indicate the bot's human-likeness in a real match like BotPrize 2008.

A research in [8] by a participant team of BotPrize 2008 is worth mentioning. Their bot learns weapon selection and battle strategy with unsupervised learning Fusion Architecture for Learning, COgnition, and Navigation (FALCON). However, there is not much learning about human-likeness because a reward is given when the bot hits or kills his opponent.

3 Our Bot in Botprize 2008

In this section, we introduce the development platform (UT2004, Pogamut 2, GameBots) shown in Fig. 1, and the bot's finite state automaton.

3.1 Platform

Unreal Tournament 2004 (UT2004) is an FPS game, which can be used as an environment for embodying bots. This game is used in many researches because it has the following properties:

(a) Researchers can make modifications (MOD) with Unreal Script, like the one used in the contest.

(b) It is one of the most famous FPS games.

(c) It is mainly made for multi-users online battle. So it is a promising environment to examine bots battling with human players.

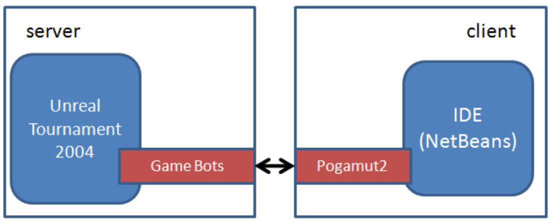

Fig. 1. Development Platform.

Pogamut 2 [9] is a platform that facilitates the development of bots in UT2004 and is provided by Charles University (Czech Republic). This freeware has a plug-in for NetBeans. Pogamut 2 communicates to UT2004 through GameBots 2004 (GB2004). It contains functions such as acquisition of sight information, controls to bot's basic actions, and path finding based on the A* algorithm.

GB2004 is a MOD for UT2004, which exports information from UT2004 to the bot and vice versa [10]. It is provided by University of Southern California. As GB2004 only exports and imports text messages, a parser is needed for translation purpose.

3.2 Bot's Finite State Automaton

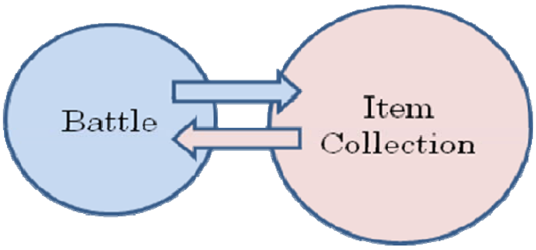

Fig. 2. Bot's finite state automaton.

We developed an FPS bot, ICE, behaving based on Finite State Automaton having two states: the item-collection state (the initial state) and the battle state (Fig. 2). The ICE's state is changed to the battle state when some opponents are found in the ICE's sight. It is changed to the item-collection state when one of the following three conditions is met.

(a) ICE defeats its current opponent

(b) ICE loses sight of its opponent and is in bad condition, i.e., not having bullets for its weapons, except for the initial assault rifle, or having the health level below 30

(c) ICE follows the opponent to the last seen location, but cannot find the opponent

3.2.1 Item-Collection State

Fig. 3. Screenshot of the item-collection state.

In this state, ICE runs to collect items (weapons, bullets, health items, and armours). Figure 3 shows a screen shot of ICE in the item-collection state when it is moving to the nearest armour. This state contains four steps: *reflexive action*, *determination of the target item type*, *determination of the target item and path*, as well as *movement* described in detail as follows:

Reflexive Action

If ICE does not jump when it crashes to a wall, all human players will easily size him up as a bot. The same holds for the situation where it is shot from behind. As a result, ICE is designed to jump in the former case and occasionally look back in the latter situation.

Determination of the Target Item Type

ICE makes a decision on which type of items among weapons and curative items (health and armour items) that it needs. Table1 shows the ICE's rules for determination of the target item type. If the resulting item type is the same as the previous one, the next step will be skipped, and ICE will stick to the present path.

Table 1. Determination of the target item type.

Condition	Target Item Type
ICE does not have bullets for its weapons, except for the initial assault rifle	Nearest weapon
ICE's health level is below 80	Nearest curative item
ICE has less than five weapons that can be used	Nearest weapon
Sum of the health level and the armour level is below 140 OR The health level is below 100	Nearest curative item
Otherwise	Random weapon

Determination of the Target Item and Path

If the target item type is "random weapon", ICE will choose the target weapon at random. In the other cases, it determines the target item, having the type decided at the previous step, to the nearest one. In this regard, those ineffective items for the current situation are ruled out in advance, i.e., ICE does not select the health pack if its life level is already at the upper limit because it cannot further increase the level. Once the target item is decided, the shortest path to it is determined. We determine the path with A*.

Movement

In this step, ICE moves following the decided path. To proceed smoothly, we consider ICE reaches the goal when it has reached to a location within a specific distance from the goal.

3.2.2 Battle State

In this state, ICE encounters or chases the opponent. Figure 4 shows a screen shot of ICE in the battle state. This state consists of four steps: *strategy selection*, *weapon selection*, *aim-point determination*, as well as *shooting and moving*, described in detail as follows.

Fig. 4. Screenshot of the battle state.

Strategy Selection

Our design policy is to keep ICE moving in the battle. If it sometimes stops or moves without any purpose during the battle, its opponent can judge on the instant that it is a bot. On the other hand, if ICE moves with a felicitous strategy, it is hardly possible to be detected as a bot by the opponent. Therefore, ICE must select a suitable strategy. ICE selects one from the following four strategies: *pick the nearest weapon*, *pick the nearest curative item*, *move toward the opponent*, and *move away from the opponent*. Table 2 shows how the strategy is determined.

Table 2. Determination of the strategy.

Condition	Strategy
ICE does not have bullets for its weapons, except for the initial assault rifle	Pick the nearest weapon
ICE's health is below 80	Pick the nearest curative item
The distance to the opponent is below 100	Move toward the opponent
Otherwise	Move away from the opponent

Weapon Selection

In this step, we calculate the ineffectiveness of each weapon ICE is holding and select the one with the smallest value. Such a value is calculated by the absolute value of the difference between the distance to the opponent and the best range of a weapon of interest. However, the initial weapon, the assault rifle, is so much weaker than the other weapons that it is eliminated from the candidates.

Aim-Point Determination

Basically, ICE aims at the current position of the opponent. However, we add certain randomness to this in order to prevent ICE from becoming too strong and thus not behaving like a human player. In particular, we add to the opponent's x, y, and z axis positions random numbers generated proportional to the distance between ICE and the opponent.

Shooting and Moving

Finally, ICE shoots and moves in accordance with the decisions made earlier in the previous steps. When ICE loses sight of its opponent, if it has bullets for its weapons, besides for the initial assault rifle, and the health level above 30, it will follow the opponent to the last seen location; otherwise, its state will be changed to the item-collection state.

4. Improvements of ICE

4.1 BotPrize 2008 Analysis

In the contest, all bots competed with each other to see which one is the most humanlike. Each of the five judges played the death-match mode with one of the five human players and one of the five finalist bots for ten minutes. After that, the judge scored these two characters the human-likeness from 0 to 4, 4 when the judge thought it a human and 0 for a bot without a doubt.

Table 3 shows the contest result. ICE came the second among the finalist bots, but there is a significant difference in the human-likeness between the human players and the bots. From the Table 3, one can see that all human players have scores higher than even the best bot.

Table 3. Human-likeness of each character at the contest.

Character Name	Human-likeness (0~4)
Byron（human）	4
Andrew（human）	3.8
Roderick（human）	3.8
Keith（human）	3
Seb（human）	2.6
AMIS（bot）	2.4
ICE（bot）	**2.2**
ISC（bot）	2
UTexas（bot）	0.8
Underdog（bot）	0.4

To seek bots' weak points on human-likeness, we asked five students in the laboratory, including the first author, to examine the contest movie files following the think-aloud protocol. In consequence, we drew five factors for being improved which were pointed out by two or more of the aforementioned students, i.e., *items acquisition*, *attacking method*, *weapon change*, *combat against multiple opponents*, *combat-movement locus*.

4.2 Five Improvement Factors

4.2.1 Items Acquisition

Here we needed to tackle three problems. First, once the target item was decided, ICE tried to capture it although this target item had been already acquired by an opponent. Second, it moved pass and did not pick an item which was not the target item. Third, if ICE lost sight of its target opponent, it always stopped collection of the target item, and soon started pursuing its opponent. In the improved version of ICE, all of these problems are solved accordingly. For example, in a situation where the bot's health level is low and its opponent is lost from the bot's sight, Fig. 5 shows a screenshot of the original ICE that stopped looking for a curative item and started to follow its opponent while Fig. 6 shows a screenshot of the improved ICE that continued collection of a curative item.

Fig. 5. Screenshot of the original ICE when its health level is low and its opponent is lost of sight.

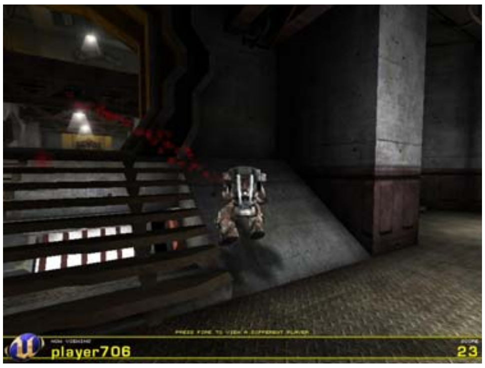

Fig. 6. Screenshot of the improved ICE when its health level is low and its opponent is lost of sight.

4.2.2 Attacking Method

ICE did not select weapons according to their attributes and the current combat situation. For example, it sometimes selected an unsuitable weapon and an improper type of attack, say, charging the opponent with a weapon suited for long-distance combat. To solve this, in the improved version of ICE, we adopt an alternative policy for attack and weapon selection based on the current strategy.

Moreover, weapons in the game have two different kinds of attack mode called "special attack" and "normal attack". However, ICE never used the special, but only used the normal one. Human players could easily judge it as a bot on this point. Therefore, we also use the special attack mode in the improved ICE.

4.2.3 Weapon Change

Before improvement, ICE frequently changed its weapon more than necessary because we determined the best weapon as the one with the minimum value in the difference between "the distance to the nearest opponent" and "the best distance of each weapon". Thereby, the frequency of changing the weapon in use was too high. In order to solve this problem, in the improved version of ICE, we subtract a small value, the distance a bot can move in one cycle, from the above value of the current weapon; this is to give a higher priority to the current weapon.

4.2.4 Combat against Multiple Opponents

In the contest video clips, we found that ICE could be easily noticed as a bot when it combated with two or more opponents. To put it concretely, we noticed that ICE showed its back to the judge while chasing for the opponent character of the human player who was lost from its sight. To solve this problem, we add two rules in the improved version of ICE. One is that ICE changes the target opponent when it loses sight of the previous target opponent. The other rule is that it does not select the move-toward-the-opponent (the nearest one) strategy when there are two or more opponents in sight.

4.2.5 Combat-Movement Locus

In the contest, ICE selected the move-toward-the-opponent (the nearest one) strategy more frequently than necessary. So it sometimes went straight to the opponent in spite of its low level health value and thus suddenly died. Our improvement here is that the bot does not select this strategy when the health level is below 100 (the initial value), and the new strategy, move-around-the-opponent as shown in Fig. 7, is added, mimicking a human player in the contest. This should be effective in reduction of straight paths.

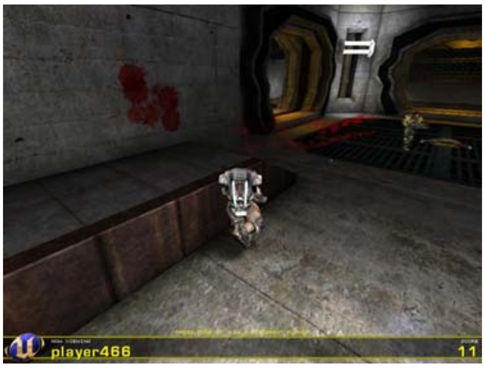

Fig. 7. Screenshot of the improved ICE in silver color conducting the move-around-the-opponent strategy.

5 Experiment

Having improved ICE along the ideas described above, we conducted an experiment to verify their effectiveness. We assembled sixteen judges, university students who had finished first or second place in the Death Match mode with three "Skilled bots", indicating that they would not easily get killed while judging.

We conducted the experiment in almost the same way as Bot Prize 2008. But we changed three points. First we asked each judge to think aloud about his feeling on a character of interest, e.g., "I think Player 1 is a bot because its path is too straight" and recorded their voices and movies of the game screen. This was for checking the contribution of each improvement. Second, each of these judges played against two bots: the original ICE and the improved one. Finally, we extended the judge time from ten minutes to fifteen minutes, because we thought we should increase the chance to combat against the two bots. Each judge was asked to participate in two games.

5.1 Experimental Result

We summarized the human-likeness score by all judges of each bot in Table 4 showing the average score of the two bots. This table indicates that the improved version is better than the original version. Moreover, under T-test, p falls below 0.01. So the difference in the score of the original ICE and that of the improved one is statically significant.

Table 4. Human-likeness of each bot.

Original ICE	Improved ICE
1.87	3.03

5.2 Discussions

To research which improvements are effective, we classified judge comments into each improvement factor and counted them. Table 5 shows the number of utterances about each improvement. Below we discuss the effectiveness of each improvement factor and what are necessary for further improvements.

Table 5. Classification of Utterances.

	Original ICE		Improved ICE	
	Positive Comment	Negative Comment	Positive Comment	Negative Comment
Items Acquisition	1	7	5	1
Attacking Method	1	9	1	1
Weapon Change	0	2	1	0
Combat against Multiple Opponents	0	3	0	2
Combat-Movement Locus	3	5	4	5

5.2.1 Items Acquisition

After improvement, the number of comments in which judges negatively rated the bot in human-likeness decreases while that they positively rated the bot in human-likeness increases. So, it can be said that this improvement is successful. Our movie analysis also reveals that many judges felt human-likeness in the improved bot when it got an item that was aimed by them.

On the other hand, there is one negative comment after improvement. A judge commented, under the read-aloud protocol, that "It is too smooth. I think a human player can’t smoothly walk backward during the battle like this". In this particular situation, the improved bot walked backward very precisely without any stops in order to acquire the target item. We agree with the judge that this does not look like human. To improve this, it is absolutely essential that ICE should relinquish to acquire the target item if it is located too far to go backward.

5.2.2 Attacking Method

From Table 5, this improvement does not contribute to human-likeness (positive comments), but it is an asset to decrease bot-likeness (negative comments). We think that in order to increase the degree of human-likeness, it is necessary to learn attack patterns from human players. The reason simply comes from the fact that attacking styles differ greatly among individuals. For example, some players frequently use the special attack mode while others do not use this mode. One negative comment after improvement comes from a judge who never used the special attack mode.

5.2.3 Weapon Change

After improvement, the number of negative comments slightly decreases while that of the positive ones slightly increases. However, the number of utterances is too low to confirm the effectiveness of this improvement.

5.2.4 Combat against Multiple Opponents

We consider that this improvement is not successful. In particular, we need to come up with a way to deal with an opponent not in sight of ICE.

5.2.5 Combat-Movement Locus

This improvement is not effective for the following reasons. First of all, the original ICE also sporadically behaved like the move-around-the-opponent strategy by chance. Secondly, ICE could not get behind its opponent because the opponent moved ceaselessly at high speed. Finally, a couple of judges rated a straight path as good. These problems remain to be solved.

6 Conclusions and Future Work

In this paper, we described our FPS bot, ICE, participating in BotPrize 2008 and finished second in the contest. In addition, we derived the five important factors on human-likeness in the FPS game and improved them. As a result, we achieved an advancement of bot’s human-like behaviours for two factors and also found new problems.

From this work, we realized the necessity of adjustment of each parameter and the provision of the individual difference according to player skill. Therefore, we adjusted the hit probability and the skill for searching the opponent before the pre-final of BotPrize 2009 [11], where our newest version of ICE was ranked third and qualified for the final contest.

Moreover, we plan to add two improvements on the individual difference, namely, adjustment of the firing accuracy and selection of strategies according to the opponent’s battle-skill. We plan to use Dynamic Scripting [12] for the latter one.

Acknowledgements

The authors would like to express our deepest gratitude to Yuna Dei for his contributions to bot development. We wish to thank also all judges, human players, and the organizer of BotPrize 2008 for their hard work.

References

[1] BotPrize 2008, http://www.botprize.org/2008.html

[2] Pogamut 2, https://artemis.ms.mff.cuni.cz/pogamut/tiki-index.php

[3] Hoang, H., Lee-Urban S., and Munoz-Avila, H.: Hierarchical Plan Representations for Encoding Strategic Game AI, Proc. of Artificial Intelligence and Interactive Digital Entertainment 2005 (2005)

[4] McPartland, M. and Gallagher, M.: Learning to be a Bot: Reinforcement Learning in Shooter Games, Proc. of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, pp. 78-83 (2008)

[5] Soni, B. and Hingston, P.: Bots Trained to Play like a Human Are More Fun, Proc. of IJCNN 2008, pp. 363-369 (2008)

[6] Gorman, B. and Humphrys, M.: Towards Integrated Imitation of Strategic Planning and Motion Modeling in Interactive Computer Games, Proc. 3rd ACM Annual International Conference in Computer Game Design and Technology (GDTW 05), pp. 92-99 (2006)

[7] Gorman, B. and Humphrys,M.: Imitating Learning of Combat Behaviors in First Person Computer Games, Proc. 10th International Conference on Computer Games: AI, Mobile, Educational & Serious Games (2007)

[8] Wang, D., Subagdja, B., Tan, A. and Ng. G: Creating Human-like Autonomous Players in Real-time First Person Shooter Computer Games, Proc. IAAAI -09 (2009)

[9] Kadlec, R. et al.: POGAMUT 2 - A Platform for Fast Development of Virtual Agents’ Behaviour, Proc. of CGAMES 07 (2007)

[10] Kaminka, G.A. et al.: GameBots: a Flexible Test Bed for Multiagent Team Research, Communications of the ACM, Vol.45, No.1, pp. 43–45 (2002)

[11] BotPrize 2009, http://www.botprize.org/

[12] Spronck, P. : Dynamic Scripting., Game AI Programming Wisdom 3, Ed. Steve Rabin, pp. 661-675, Charles River Media, Hingham, MA (2006)