2010

A New Design for a Turing Test for Bots

Philip Hingston
Edith Cowan University

10.1109/ITW.2010.5593336
This article was originally published as: Hingston, P. F. (2010). A New Design for a Turing Test for Bots. Proceedings of 2010 IEEE Conference on Computational Intelligence and Games (CIG'10). (pp. 345-350). . IT University Copenhagen, Denmark. IEEE. Original article available here

A New Design for a Turing Test for Bots

Philip Hingston, *Senior Member, IEEE*

***Abstract*— Interesting, human-like opponents add to the entertainment value of a video game, and creating such opponents is a difficult challenge for programmers. Can artificial intelligence and computational intelligence provide the means to convincingly simulate a human opponent? Or are simple programming tricks and deceptions more effective? To answer these questions, the author designed and organised a game bot programming competition, the BotPrize, in which competitors submit bots that try to pass a "Turing Test for Bots". In this paper, we describe a new design for the competition, which will make it simpler to run, and, we hope, open up new opportunities for innovative use of the testing platform. We illustrate the potential of the new platform by describing an implementation of a bot that is designed to learn how to appear more human using feedback obtained during play.**

I. Introduction

This paper is about the BotPrize, a competition that challenges bot programmers to create bots (for a computer game) which cannot be distinguished from human players.

Creating human-level opponents for interactive computer games has been identified as a "killer application" for Artificial Intelligence [9]. In this "call to arms", the authors present many reasons for Artificial Intelligence researchers to work towards this aim:

- That interactive computer games are an application that **needs** human-level AI;
- That they provide environments for research on the "right kind" of problem that can lead to human-level AI;
- Human-level AI is needed for realistic training using simulation (and such interactive simulations are, in essense, interactive computer games);
- Modern computer games provide sufficient realism without the practical issues of using real sensors and real motor systems;
- Interactive computer games are cheap, reliable, and many have AI interfaces;
- Current game AI is limited in scope and will not scale up - AI researchers are needed to address this.

They make a distinction between human-level AI and human-like behaviours, while recognizing that the two are related. As part of the discussion of this point, they note that game AI's can and do "cheat", for example, by having perfect knowledge of the other players' locations, even in darkness and through walls, and that this is a common source of complaint amongst gamers.

This kind of cheating is understandable in that the intention of the commercial game developer is only to sell his game, not to advance the state of the art in Artificial Intelligence research. Therefore, "cheating" is absolutely fine, so long as the end result is an enjoyable game that will attract players to buy it. However, it is easier to simulate some aspects of in-game behaviour than others. For example, a bot that shoots too quickly and accurately is easily identified as non-human, but it is simple to slow the bot down and make its shots less accurate. On the other hand, human players can quickly adapt to unexpected game aspects, such as a clever new strategy by an opponent, but such adaptability is well beyond the capabilities of current game AI.

Many other researchers have made the point that players enjoy a game more if they believe that their opponent is another human represented in the game by an avatar, rather than a bot (a computer-controlled player). For example, in [18], the authors report on an experiment in which subjects played Neverwinter Nights, an online role-playing game. Through various means of deceipt, one group of subjects was convinced that they were playing against human opponents, while the other thought they were playing a computer opponent (in fact both were playing computer opponents). The first group reported a greater sense of immersion and greater enjoyment (and also a greater sense of engagement and "flow"). In [10], the researchers measured physiological responses of players and reported that "Players exhibited greater physiological arousal to otherwise identical interactions when other characters were introduced as an avatar rather than an agent." In [12], neural network-based bots for a FPS game were trained using examples of human play, and were then found to be more challenging and enjoyable opponents when compared with standard, scripted bots.

In 2008, the author organised an open competition, the 2K BotPrize, for bot programmers to create a bot (for the First-Person Shooter, Unreal Tournament 2004) capable of convincing a panel of judges that it was an avatar, controlled by a human player. This competition is described in detail in [5] (which also discusses philosophical interpretations of the test used in the competition). In this competition, human judges were matched against a bot and a human confederate and played a round of UT2004. The judges then had to nominate which opponent was the human and which the bot. The game was modified to support the needs of the competition. More detail is given in Section III below.

The BotPrize is obviously a bot version of the famous Turing Test, first proposed by Alan Turing [16] as a thought experiment in which a judge has a chat session with a human and a computer program, and then has to nominate which was which. A major difference is that Turing's test was proposed as a test of the computer's intelligence, while the BotPrize is a test of the bot's ability to give the *appearance* of being

Philip Hingston is with the School of Computer and Security Science, Edith Cowan University, Perth, Australia (email: phi@philiphingston.com).

978-1-4244-6297-1/10/$26.00 ©2010 IEEE

human.

While the bots had some degree of success in the 2008 BotPrize, none of the bots was able to fool a majority of the judges. The competition was repeated in 2009, with very similar results. (See the BotPrize website at www.botprize.org for full results and other details of both competitions.)

In this paper, we describe a new design for the test, to be used in the 2010 BotPrize competition, at the IEEE Conference on Computational Intelligence and Games. The new design has many advantages over the earlier ones, as we describe below.

II. The 2008 and 2009 BotPrize competitions

In this section, we briefly describe the design of the 2008 and 2009 BotPrize competitions.

The 2008 competition has been described in full detail in [5], but we summarise the main features here. As mentioned earlier, the competitions make use of the commercial computer game Unreal Tournament 2004. There were a number of reasons for this choice: it is well known amongst game players, readily available and inexpensive, multiple humans and bots can play together, and they don't need to be colocated, it is relatively easy to interface a bot to, the game format is suitable for a number of short rounds of competition, and it is possible to customize (mod) the game to a degree.

UT2004 is a FPS (First-Person Shooter). The game play takes place in a virtual world, simulated by a program known as the server. The player connects to the server over a network, using another program known as a client. Other characters can be controlled by other players via their own clients. The game also allows for characters controlled by inbuilt bots running on the server. In the game, players and bots (or at least their avatars) roam the simulated world, picking up or interacting with virtual items, and shooting opponents with virtual weapons. When a player is hit, his health level drops, and when it reaches a critical level, he is killed (or "fragged"). Players score points for fragging opponents. Usually, a fragged player only has to wait a few seconds before being "respawned", or brought back to life, and allowed to rejoin the game.

The competition used the "DeathMatch" game format, one of the "game types" supported in UT2004. In a DeathMatch, the aim is to have your character frag as many other characters as possible within the time limits of the game. A Death Match with three participants (judge, confederate and bot) makes sense, and a game can be completed in a relatively short time, which was important for practical reasons of time limitations in running the competition.

The competition was run in a number of rounds, and in each round, each judge was matched against a human player and a bot, and played a 10 minute Death Match. The confederates were instructed to play the game as they normally would and they played for a small cash prize of $150, plus a trophy. In the competition final, there were five judges, five confederates and five bots.

At the completion of each round, the judges rated their two opponents using the following scale:

1) This player is a not very human-like bot.
2) This player is a fairly human-like bot.
3) This player is a quite human-like bot.
4) This play is a very human-like bot.
5) This player is human.

The game was modified so that as each player, either human or bot, enters the game, their name is changed to a randomly generated one like player265. This name is then displayed above the character during the game, and used in various game status displays that are available to players. In addition, the appearance of characters in the games can be customized. This was standardized so that all characters had the same physical appearance, apart from their assigned random names.

The virtual world that the game simulates is described by what is called a "map". The map describes the three dimensional physical layout of the world (for example, its dimensions, positions of walls and other objects), other information about objects in the virtual world, as well as image and sound data used to create a real-time three dimensional view (usually from the viewpoint of the character being controlled by the player) and to provide sound effects, character voices, and background music. Maps also usually contain information that is intended to be used by bots, to help them behave sensibly in the game. Maps in UT2004 typically include "hints" about locations and items. For example, a particular location might be labeled as being a good "sniping" spot, where a bot could hide and wait for opponents to pass by. We modified the game to strip out these hints, to encourage competitors to have their bots learn this information for themselves, as a human player would have to do.

Bots are normally provided with hints about the capabilities of weapons (for example, whether they are suited for use in a melee). These hints were removed. There is a command available to bots that automatically selects the best weapon for a given situation. This command was changed so that it would select a random weapon, to encourage competitors to have their bots make this choice for themselves.

To win the Prize, a bot had to convince 4 out of 5 judges that it was human. The results showed that the judges were able to discriminate between human and bot players, with all humans being judged more human than all bots. Three out of five humans passed the test, and none of the bots did. The bots did have some success, however, with two bots deceiving two judges, and another bot deceiving one judge.

The competition was run again in 2009, with minor changes from 2008. The main change was the inclusion of "special weapons" — modified versions of existing UT2004 weapons. These were included to challenge the bots. Competitors were told that some weapon effects would be changed, but were not told what the changes would be. One weapon was altered so that if a character was hit by it, he was teleported to a random location in the game. This could

be a useful defensive weapon. Another weapon was changed so that if a player was hit with it, then he would be "frozen" to the floor for the next 30 seconds, unable to chase or dodge, and would then be an easy target for a normal weapon. Using this weapon effectively requires the player to realise that his opponent cannot move, and take advantage by switching to another weapon. Human players easily adapt to these new weapons.

The result was similar to those of 2008, with all bots fooling one judge (not the same one!), and all the confederates having a higher average humanness rating than all the bots.

Comments from the judges and confederates showed some general characterstics used to identify bots. Some features of bots included:

- Apparent lack of planning
- Failure to act consistently - "forgets" about opponents
- Getting "stuck"
- "Static" movement
- Very accurate shooting
- Lack of awareness
- Stubbornness
- Tendancy to engage in "stand-offs"

while common features of humans included:

- Aggressiveness
- Reacts to situation
- Able to use special weapons

Several competitors from 2008 and 2009 have published descriptions of their bots. The winning entry in 2009 was *sqlitebot*, a bot whose main distinguishing feature was the use of an SQL database to provide long-term memory. In [3], the author describes how this is used to allow the bot to learn and remember "hotspots" where interesting action has taken place, so that the bot can return there later in the game. The bot also used a visibility table stored in the database, constructed in an initial reconnoiter of the map, to enable the bot to retreat to hiding places when low on health. Some simple methods were also implemented to prevent the bot from repeating futile behaviours (deliberately ignore information in order to change decisions when no progress is being made), to prevent it being predictable (using randomness), and to prevent it from being too proficient (deliberately delay responses). These last three were all identified as giveaway bot behaviours in the 2008 BotPrize competition.

In [6], the authors describe their 2008 bot, the second-placed ICE-2008. ICE-2009 also placed second in 2009. Some details of the 2009 bot were provided to the author [17]. This bot is based on a finite-state machine with two states, one roaming state for collecting items, and one state for combat. Various triggers switch the bot between these states. The bot changes from item-collecting to combat when an opponent is sighted, and from combat to item-collecting when an opponent is fragged, or the bot loses sight of its opponent and is in need of recovery, and when the bot cannot find its opponent. A rule base is used to determine which actions are taken. For example, when in the item-collecting state, rules are used to determine what kind of item to collect next, depending on the distance to the object, and the bot's current needs. A^* is used to determine suitable paths between items.

When in the combat state, rules are used to determine which of seven possible combat strategies to use, and another set of rules determines which weapon to use. Similar to sqlitebot, ICE deliberately reduces its skill level, in this case by deliberately aiming badly when shooting, taking into account diatance to target and estimated playing strength of the opponent.

In [19], the authors describe their bot, ISC, which was placed third in the 2008 BotPrize competition. This bot uses FALCON and TD-FALCON ([14], [15]), neural network-based systems using reinforcement learning and TD-learning for weapon choice decisions and behaviour selection respectively. The bot also used randomness in certain hand-crafted behaviours to simulate humanness. For example, the bot waits a random amount of time and then turns by a random amount, when it is hit by fire from an opponent. The FALCON-trained networks can be analysed and rules extracted, showing that the bot is able to obtain useful knowledge during play. An example of a learned rule is "IF health is 0.4, and being damaged, and opponent is in sight, and has adequate ammo, and currently in collecting item state; THEN go into engaging fire state; WITH reward of 0.718."

A number of other bots entered in the competitions have also made use of learning. The team from University of Texas at Austin used reinforcement learning to learn weapon choices and certain aspects of movement during play in their 2008 bot, and added imitation learning based on human movement traces for 2009 [7].

While the design used in the 2008 and 2009 competition has proven effective, it is logistically difficult to organise. MOreover, it is laborious to collect and analyse the results. This has motivated us to try to improve the design, making it easier to run, more flexible, and readily usable as a testbed for researchers to experiment upon. In the next section, we describe the new design.

III. A New Design

The guiding principle behind the new design is to make the judging process *part of the game*. The key idea is to provide players with a devastating "special weapon" that can only be used effectively if the player knows whether he is shooting at a human or a bot. To play the game well, the player must look for clues in his opponent's play that distinguish human from bot behaviours - which is precisely the judges' task in a Turing Test.

This is achieved by modifying the behaviour of one of the existing weapons, the Link Gun. Weapons in UT2004 have two firing "modes": a primary and an alternate mode. With the modified weapon, the primary fire mode is intended to be used against bots - hitting a bot with the primary fire mode instantly kills the bot, and rewards the player with 10 points (compared with the normal 1 point per frag). On the other

hand, shooting a *human* opponent with the primary fire mode result in instant death *to the shooter* and the loss of 10 points. Therefore, the player needs to be very sure his opponent is a bot before using the primary fire mode. The alternate fire mode is the mirror image : it should only be used to shoot a *human* opponent. Hitting a human opponent with the alternate fire mode gives an instant kill and 10 points, while hitting a bot with it results in instant death and the loss of 10 points for the shooter.

So that players cannot simply use the weapon to identify an opponent (randomly guess which mode to use and observe the result), the weapon can only be used on a particular opponent once - subsequent shots simply have no effect. To ensure that other players get no information from witnessing a kill with the Link Gun, the primary and alternate fire modes look and sound identical to an observer.

As well as these major changes, we have relaxed restrictions imposed in the earlier competitions, as experience suggests they did not significantly affect the results. Thus, map and weapon hints are no longer removed. The previous synthetic names (e.g. player342) have been replaced with names random selected from a large database of common names, making them easier for players to remember. In addition, full details of the mod (and in fact the mod itself) are available. Previously these details were kept secret until the final judging, which added to the logistical complication of running the competition.

Compared with the previous judging method, the in-game judging system has many advantages. Logistically, it is much simpler - there is no need for secret rooms and coordination of rounds. The server simply runs continuously, and human players and bots alike can connect to it at any time. There is no need for a delay after each round to collect the judges' verdicts — the server records the judgements during the game by monitoring the use of the two firing modes of the Link Gun.

We hope that this will also make it feasible for people to organise and run their own competitions and demonstrations, or to use the system for class projects and so forth, at low cost and with minimal effort and little need for infrastructure of their own. The flexibility of the new system also opens up new research opportunities. We give several examples below.

1) Evolving human-like behaviour: Because the server can be left running unattended, bots can be run and experimented on at any time without making any special arrangements. Thus, researchers have a readily available testbed on which to experiment with their bots, and are provided with feedback on the performance of their bots (bots receive messages from the server informing them when they are hit by another player). This feedback could be used, for example, to support realtime evolution or machine learning by bots (or data could be simply collected for later analysis). In the next section, we describe a simple implementation of this idea.

Of course, many other researchers have used evolution to evolve bots to play various games, some of them evolving bots for UT2004 (e.g. [8], [11]). Quite a few have used real-time evolution, with bot behaviours evolving during actual play, inspired, perhaps by NERO [13], a game that uses neuro-evolution to allow the player to train teams of bots. The new idea here is to use the BotPrize server and its judging system to evolve bots in real time that play like human players, rather than to evolve bots that play the game strongly.

2) A reverse Turing Test: The new judging system can be used to support a "reverse Turing Test" for bots, in which bots are evaluated on their ability to identify the other players as human or bot. Such a test would be more than an intellectual curiosity. A number of researchers in recent years have been working to develop methods to automatically detect bots in online games. One reason that such methods are needed is to be able to prevent bots from competing in certain games. In some games, using bots is regarded as cheating. A well-known example is the use of bots for "gold farming" in MMORPG's. Bots are programmed to carry out some repetitive activity in the game on behalf of a player, earning an in-game reward, such as "gold". This gold can then be used to advantage the player, or can even be sold for actual money to other players. Companies that run these games are keen to prevent this kind of abuse.

A discussion of the problem of detecting bots in games is provided in [4], in which the authors recommend the use of so-called CAPTCHA tests, in which a puzzle or challenge is introduced into the game that is designed to be solvable only by a human player (the authors actually propose a hardware CAPTCHA device). The obvious disadvantage is the unnatural disruption of the game. Others have developed less disruptive methods to detect bots based on analysis of behaviour patterns, such as patterns of movement (see e.g. [1], [2]).

The new BotPrize judging system offers a way to measure the performance of such detection methods. In the 2010 BotPrize competition, we will use the new system to informally run a "sister" competition based on this reverse Turing Test. Depending on its popularity, it may become a formal part of future BotPrize competitions.

IV. An Illustrative Bot

To illustrate the potential for novel research that the new platform offers, we describe a bot that can learn how to appear more human using feedback provided by the human players in the game.

The bot uses a simple state-based architecture, and is a modification of "Hunter", one of the sample bots provided as part of the Pogamut distribution. The possible states of the bot are:

- ROAMING – the default state - select a random navigation point and run there
- ENGAGED – either jump, stand still, or dodge while shooting at a visible enemy
- RETREATING – select a random point further away from a visible enemy, and run there shooting as we go

- SEEKING.PLAYER, SEEKING.WEAPON, SEEKING.ARMOUR, SEEKING.AMMO, SEEKING.HEALTH – select a random item of the specified type and run to it

The behaviour of the bot in each of these states is hand-coded. The bot switches between states using a set of Sugeno-style fuzzy rules. Here is a possible set of rules:

1) IF (enemy? IS yes) AND (health IS low) THEN (RETREATING IS 0.25)
2) IF (enemy? IS yes) AND (health IS OK) THEN (ENGAGED IS 0.5)
3) IF (enemy? IS yes) AND (ammo IS low) THEN (RETREATING IS 1.0)
4) IF (enemy? IS no) AND (health IS low) THEN (SEEKING.HEALTH IS 0.8)
5) IF (enemy? IS no) AND (ammo IS low) THEN (SEEKING.AMMO IS 1.0)
6) IF (enemy? IS no) THEN (SEEKING.WEAPON IS 0.7)
7) IF (enemy? IS no) AND (health IS OK) and (ammo is OK) THEN (SEEKING.PLAYER IS 0.7)

These rules are evaluated several times per second, using the bot's perception of the current game situation as input. This determines the values of the output variables, one for each state. The state with the highest value then becomes the bot's current state, determining its behaviour.

Taking Rule 1 above as an example, the two input variables are **enemy?** and **health**. The variable **enemy?** has two possible linguistic categories, **yes** and **no**, while **health** has **low** and **OK**. These linguistic categories are described using trapezoidal membership functions. Each membership function is determined by 4 real values representing the bottom-left, top-left, top-right, and bottom-right x-coordinates of the corners of the trapezoid. For example, in Figure 1, the membership function for **low** health is defined by the values $0, 0, 0$ and 50. As illustrated in the figure, a health value of 40 thus produces a membership value of 0.2. Similarly, as shown in Figure 2, **OK** health is defined by the values $0, 200, 200$ and 200.

Figure 3 shows the calculation of the value of **RETREATING**. Rules 1 and 3 are firing, and the value of **RETREATING** is calculated using the centre of gravity method based on the firing strengths of the rules.

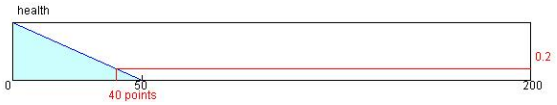

Fig. 1. The membership function for **low** health

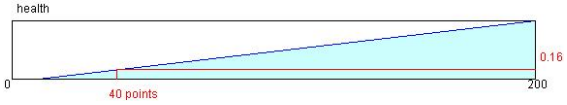

Fig. 2. The membership function for **OK** health

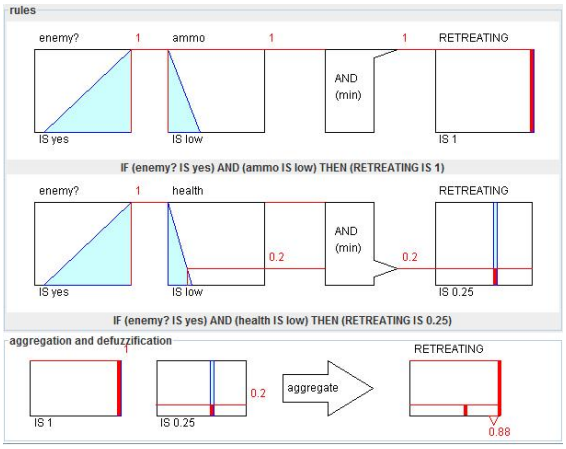

Fig. 3. Determining the value of the output variable **RETREATING**

The behaviour of the bot therefore depends on the choice of rules, the shapes of the membership functions, and the values for the output variables given in the rules. Changing any of these will change the behaviour of the bot. In this example bot, we designed a custon "mutation" operator to mutate the rules. We fixed the form of the rules, and varied just the membership functions and values for the output variables. Thus the "genome" for the bot consists of a set of genes, each of which is either a real number within a range (a value for an output variable), or a quadruple of real numbers describing a trapezoid. Care must be taken when mutating a gene to ensure the output values remain within the allowed ranges, that the quadruples represent valid trapezoids (they must form a non-decreasing sequence), and that the collection of membership functions for each input variable correctly cover the range of possible values for that variable.

A crossover operator was also defined, by treating the genome as a sequence of genes, and using uniform crossover.

When the bot is placed into the game to interact with other players, it receives messages reporting various events within the game. In particular, it receives enough information to keep track of how many times it has been shot by the Link Gun in each of its modes — in other words, how often it has been identified as a bot, or mistaken as a human. This allows the bot to evaluate its own performance in terms of how successfully it is imitating human behaviour, as judged by the other players in the game.

Thus we have all the ingredients needed to evolve bots that are more successful at imitating humans: we have a suitable genome and some genetic operators, and we have a suitable fitness function. The bot can be evolved in-game as follows.

1) We create a small population (say 5) of these bots, and inject them into a game.
2) Each time a bot is hit by the Link Gun, it updates counts of how many times it has been hit using each mode, and calculates a ratio.
3) When a bot has been identified a certain number of times (say 5), it mutates itself, and performs crossover

with the current best-performing bot (best in terms of being most frequently mistaken for human).

We intend to have this bot evolve on the BotPrize server leading up to the competition final, and to use it as a "reference bot" and point of comparison with the competition entries. The design described here is simply a proof of concept, and these design choices are certainly not necessarily the best, or even particularly good choices. The field is wide open for other choices - a different style of evolutionary algorithm, different representation for the bot behaviours, even combining evolutionary and lifetime learning are obvious possibilities.

V. Conclusions

In this paper, we have described a new design for a Turing Test for game bots. In addition to being a more "natural" evaluation method, where the judges are simply game players and judging is an inherent part of the game, the new design has many practical advantages.

We have provided an illustration of a new possibility that the new design enables: evolving human-like behaviours. We hope that the easy availability and flexibility of the platform will encourage students, researchers, and hobbyists to participate in ongoing research in innovative ways.

Acknowledgments

The author would like to thank 2K Marin (formerly 2K Australia) for its support for the BotPrize competition, the competitors, judges and confederates in the BotPrize competitions for taking part, and local conference organisers for their logistical and moral support.

References

[1] K.-T. Chen and L.-W. Hong, "User identification based on game-play activity patterns", in NetGames 07: Proceedings of the 6th ACM SIGCOMM workshop on Network and System Support for Games. ACM, pp. 712, 2007.

[2] K.-T. Chen, J.-W. Jiang, P. Huang, H.-H. Chu, C.-L. Lei, and W.-C. Chen, "Identifying MMORPG bots: A traffic analysis approach", EURASIP Journal on Advances in Signal Processing, 2009.

[3] J. Cothran, "Winning the 2K Bot Prize with a Long-Term Memory Database using SQLite"', Available on AiGameDev.com at http://aigamedev.com/open/articles/sqlite-bot/ , November, 2009.

[4] P. Golle and N. Ducheneaut, "Preventing bots from playing online games", Computers in Entertainment, vol. 3, no. 3, pp. 33, 2005.

[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.

[6] D. Hirono and R. Thawonmas, "Implementation of a Human-Like Bot in a First Person Shooter: Second Place Bot at BotPrize 2008", CD-ROM Proc. Asia Simulation Conference 2009 (JSST 2009), October 7-9, 2009, Ritsumeikan University, Shiga, Japan (5 pages). Available http://www.ice.ci.ritsumei.ac.jp/ ruck/PAP/jsst09-hirono.pdf

[7] I. Karpov, J. Schrum and R. Mikkulainen, Personal communication, October 2009.

[8] R. Kadlec, "Evolution of intelligent agent behaviour in computer games", Masters Thesis, Charles University in Prague, Available http://artemis.ms.mff.cuni.cz/main/papers/GeneticBots_MSc_Kadlec.pdf, 2008.

[9] J.E. Laird and M. van Lent, "Human-level AIs Killer Application: Interactive Computer Games", AAAI Fall Symposium Technical Report, North Falmouth, Massachusetts, pp. 80-97, 2000.

[10] S. Lim and B. Reeves ,"Computer agents versus avatars: Responses to interactive game characters controlled by a computer or other player", International Journal of Human-Computer Studies, 68(1-2), pp. 57-68, 2010.

[11] R. Small and C. Bates Congdon,"Agent Smith: towards an evolutionary rule-based agent for interactive dynamic games", In CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation, Trondheim, Norway, pp. 660–666, IEEE Press, Piscataway, NJ, USA, 2009.

[12] B. Soni and P. Hingston, "Bots Trained to Play Like a Human are More Fun", in Proc. of IEEE International Joint Conference on Neural Networks, IJCNN 2008, Hong Kong, June 2008.

[13] K. Stanley, B. Bryant, R. Miikkulainen, "Evolving Neural Network Agents in the NERO Video Game", In Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05). Piscataway, NJ: IEEE, 2005

[14] A.-H. Tan, "FALCON: A Fusion Architecture for Learning, COgnition, and Navigation", In Proceedings of International Joint Conference on Neual Networks, pp. 3297 3302, 2004.

[15] A.-H. Tan, N. Lu, and D. Xiao, "Integrating Temporal Difference Methods and Self-Organizing Neural Networks for Reinforcement Learning with Delayed Evaluative Feedback", IEEE Transactions on Neural Networks 9(2), pp 230244, 2008.

[16] A. Turing, "Computing Machinery and Intelligence", Mind, 59 (236), pp. 43360, 1950. Available on the web at http://www.loebner.net/Prizef/TuringArticle.html.

[17] R. Thawonmas, D. Hirono and A. Kojima, Personal communication, October 2009.

[18] D. Weibel, Bartholomus. Wissmath, S. Habegger, Y. Steiner and R. Groner, "Playing online games against computer- vs. human-controlled opponents: Effects on presence, flow, and enjoyment", Computers in Human Behavior, 24(5), pp. 2274 - 2291, 2008.

[19] D. Wang, B. Subagdja, A-H. Tan and G-W, Ng, "Creating Human-like Autonomous Players in Real-time First Person Shooter Computer Games", Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence (IAAI'09), Pasadena, California, July 1416, 2009