2008

Bots Trained to Play Like a Human are More Fun

Bhuman Soni
Edith Cowan University

Philip Hingston
Edith Cowan University

10.1109/IJCNN.2008.4633818

Bots trained to play like a human are more fun

Bhuman Soni, Philip Hingston *Senior Member, IEEE*

***Abstract*— Computational intelligence methods are well-suited for use in computer controlled opponents for video games. In many other applications of these methods, the aim is to simulate near-optimal intelligent behaviour. But in video games, the aim is to provide interesting opponents for human players, not optimal ones. In this study, we trained neural network-based computer controlled opponents to play like a human in a popular first-person shooter. We then had gamers play-test these opponents as well as a hand-coded opponent, and surveyed them to find out which opponents they enjoyed more. Our results show that the neural network-based opponents were clearly preferred.**

I. Introduction

In this paper, we report on a study in which we attempted to use machine learning to create interesting opponents for human players in a video game. Many researchers have observed that video games are a natural and very attractive application area for computational intelligence methods.

Artificial intelligence researchers have long used board games and card games as challenge problems, focussing on achieving human or even super-human levels of play. In video games, while similar techniques can be used, the aim is to create *interesting* opponents (variously called "bots" or "non-player characters" or "NPCs"). While the question of what makes for an interesting opponent may be debated, super-human levels of play are more likely to be frustrating than interesting.

It has been suggested that for maximum enjoyment, the skill level of a computer opponent should roughly match that of the human player. Additionally, it is thought that humans prefer "human-like" opponents. One aspect of being human-like is obviously the level of play. Another related factor is how predictable the play of the computer opponent is. A more predictable opponent is easier to defeat, and predictability is generally thought to be an indicator that one's opponent is following a fixed set of instructions, computer-like.

Programming a computer opponent to convincingly imitate a human is not an easy task - it amounts to a sort of restricted version of the Turing test. At the very least, it would be likely to need a lot of programming time. An alternative approach is to teach a computer opponent to play like a human using machine learning techniques.

The aim of this work is to test out this idea. We designed bots for a commercial video game that use a neural network to select actions. We recorded data from a human player, and used this to train the network. We then tested the resulting bots by having human players compete against them, and asking them about their impressions of their opponents.

In the remainder of this paper, we describe the agent architecure used for these bots, the methods we used to gather the training data, and the results of our analysis of the feedback from the human players.

II. Related work

Other researchers have used AI techniques to improve the performance of bots in computer games, where "performance" is taken to mean strength in playing the game. For example, Spronck et al. [6] used an evolutionary algorithm (EA) to evolve artificial neural networks (ANNs) offline. Their evolved bot successfully outperformed its scripted opponent bot and even discovered flaws in the script. In later work [7], they introduced *Dynamic Scripting*, an online learning technique based on the use of reinforcement learning to adjust a mechanism for selecting between various scripted behaviours. The stochastic selection mechanism and the online learning provided a degree of unpredictability to the play, but again, the primary aim was to make the player stronger. Later still, Ponsen and Spronk [4], further developed this idea by using an evolutionary algorithm offline to derive a better set of scripts, which are then used in dynamic scripting during play.

Another example is [2], where the authors report on their work using reinforcement learning to train an ANN for a bot in a fighting game. Once again, their interest was in showing that the training was an effective way to develop a bot that performs well against bots created by hand. In [10], neural networks were trained using a genetic algorithm to perform several tasks, with mixed success. No human testing was reported, except for a comment that the bot was "far from being competitive or fun to play".

In contrast to the work described above, Yannakakis and co-researchers have carried out studies aimed at using machine learning methods to create interesting bots rather than proficient ones. In [8] they used a constructed figure of merit for "interestingness", taking into account level or play (not too hard or too easy), as well as diversity and activeness of bot behaviour. They then used an evolutionary algorithm to evolve weights for a neural network controlling the bot, using "interestingness" as the fitness value.

In [1], the authors used self-organizing maps and multi-layer perceptrons to learn various aspects of player behaviour. Some of the same authors [3] later used imitative Bayesian learning to train bots to imitate human players' movement patterns, recorded sample 20 second movie clips of the bots, along with clips of human-controlled characters, and hand-coded bots. They then showed these clips to subjects and surveyed them to measure their judgement of how "human-like" the play was. This differs from our work chiefly in that our subjects actually played against different bots, which arguably is a truer test of how human-like the bots appear in

978-1-4244-1821-3/08/$25.00

actual play. There is the additional benefit that our subjects could be asked questions about other aspects, such as their enjoyment of the game, which is a judgement that could not be made on the basis of viewing clips. In addition, our subjects' judgements are made on the basis of approximately 15 minutes of play, rather than a 20 second clip.

There is also a recent trend of increased interest in modelling, measuring and attempting to improve player satisfaction in games. Bot behaviour is just one aspect of this more general concern. For example, in [9], the authors used physiological measurements (heart rate) in a physical play game for children to infer levels of interest. The idea is that heart rate can then be used to adjust aspects of gameplay. In [5], the authors propose a formalism called *Declarative Optimization-based Drama Management*, which might be used to estimate a player's preferences in terms of gameplay and plot, so that these can be customised to improve player enjoyment.

III. Method

In order to investigate gamers' preferences in computer-controlled opponents, we created a number of modified versions ("mods") of a popular commercial video game. We then tested the mods by having gamers play each one, and answer a questionnaire about their experiences and impressions.

A. The game

The game we chose is a first-person shooter (FPS). In an FPS, the player takes the role of a character in a simulated scenario, whose actions the player can control. Typically the player can move about the simulated world (walk, run, crouch, jump etc), can pick up objects (health packs, weapons, ammunition etc), and use objects in various ways (shoot weapons, open doors etc). The aim is usually to achieve some strategic objective such as capturing a base. In the course of play, characters inflict damage on each other using various weapons. When a player suffers too much damage, he/she/it dies. But in the game world, death is a temporary setback: the character usually "respawns", re-entering the game in full health, after a short period in limbo.

Specifically, the game we used was "Unreal Tournament 2004" (UT2004). This is a popular game and there is an active "modding" community and good tools and support for modding. We chose the "Deathmatch" style of game, in which two players, or a player and a bot, play against each other in a given scenario, and where the winner is the first player to achieve a preset number of "kills" or "frags".

UT2004 comes equipped with a standard bot, which is controlled by hand-coded scripts based on a fuzzy finite state machine. These scripts take into account the current game situation to select from a number of possible predefined "states". Each of these states represents some high-level behaviour or goal, and the bot behaves differently depending on its current state. The states that we used in this study are:

- **Roaming:** the bot is not engaged with the enemy. It moves about the virtual world, picking up useful objects.
- **Hunting:** the bot follows after the enemy, which has gone out of sight.
- **Ranged Attack:** the bot fires the current weapon at the enemy from a long distance.
- **Charging:** the bot charges at the enemy while firing the current weapon.
- **Shield Self:** the bot takes a defensive stance and fires its current weapon at the enemy.
- **Tactical Shoot:** the bot moves in random directions while shooting at the enemy in order to dodge hostile fire and confuse the enemy.

B. Mods

We created two mods, each featuring a different neural network-based bot. These bots are like the standard bot, except that they use a neural network to select an action from one of the predefined states. The difference between the two was a different action-selection mechanism.

Each bot was trained using a dataset created using the recorded actions of a human player in a number of games with different scenarios. Another mod was created for the purpose of recording this data.

We now describe each of these mods:

1) Recorder mod: For this mod, we added functionality to the game so that approximately every 100 ms, the current game situation and the player's current action were stored in a text file, to be used later in the training of a neural network.

To represent the game situation, we used a vector of features that are readily available to the human player:

- **Enemy distance:** the distance between the character and the enemy.
- **Health:** the character's current health level.
- **Shield:** the current strength of the character's shield.
- **Weapon:** a nominal attribute - the current weapon being used by the character (in this game, one of Shield gun, Assault rifle, Bio-rifle, Minigun, Shock rifle, Link gun, Flak cannon, Rocket launcher or Lightning gun).
- **Enemy weapon:** likewise for the enemy.
- **Ammo:** the amount of ammunition left in the current weapon.
- **Enemy firing:** a boolean, indicating whether the enemy is currently shooting his/her/its weapon.

In order to create a training set, we had to somehow select one of the predefined states to match with the human player's current behaviour. To do this, we considered the distance between the player and the enemy, the player's direction of motion relative to the enemy, and whether the enemy was in sight, and used the matching criteria given in Table I.

2) Neural bot mods: Both the neural bot mods used sockets to communicate with an external program that implements a standard multi-layer perceptron. While the enemy is not in view, the bots remain in the Roaming state. When the enemy is in view, the current game situation is passed to the external program, which feeds these into the neural network. The external program then uses the outputs of the neural network to select one of the non-Roaming states, and passes

TABLE I

CRITERIA USED TO DETERMINE THE PLAYER'S STATE

distance	moving	enemy in sight?	state
far	toward	yes	ranged attack
far	backward	yes	shield self
far	sideways	yes	tactical shoot
medium	toward	yes	charging
medium	backward	yes	tactical move
medium	sideways	yes	tactical move
close	toward	yes	charging
close	backward	yes	tactical move
close	sideways	yes	tactical move
any	toward	no	hunting
any	not toward	no	roaming

this back to the bot, which then changes to this state (if not already in the selected state).

Both bots use a neural network with an input layer, one hidden layer of 10 neurons, and an output layer with 5 neurons, one for each non-Roaming state. We called these two bots the *feedforward bot* and the *recurrent bot*.

For the feedforward bot, the input layer contains 23 neurons as shown in Fig 1. The state selected is the one whose output neuron has the highest activation level. Thus, the feedforward bot is deterministic.

For the recurrent bot, the input layer has 6 additional neurons, one for each state. These are used to provide the network with the bot's current state. The next state is selected using "roulette wheel" selection based on the activation levels of the output neurons. Thus, the recurrent bot makes stochastic choices.

The hope was that this would make the recurrent bot more unpredictable, and therefore more challenging and more human-like. The current state was provided as additional input to the neural network so that the bot would not "thrash" between states, which would be both unnatural and ineffective. This mechanism arguably reflects the way a human would play - tending to stick with their last choice unless a change of circumstance dictates a different choice, at least most of the time.

Both bots were trained using back-propogation on the example data obtained using the recorder mod, after inferring player selections as in Table I. Since some states occurred less often than others, training examples for less frequent states were replicated so as to balance the number of examples in each category. The replicated dataset contained approximately 10,000 training examples. Both networks were trained over 100 epochs. This resulted in cross-validated accuracy of 72% for the feedforward bot, and 94% for the recurrent bot.

C. Procedure

To test whether the neural bots made interesting opponents, we recruited subjects from amongst the student population and had them play the standard game, the game with feedforward bots, and the game with recurrent bots (in randomised order), and then had them complete a questionnaire.

The subjects were 21 males and one female, of different nationalities, between the age of 18 and 50. All had prior experience in playing first-person shooters.

In each session, the subject first went through a short in-game tutorial to familiarise with the controls and rules of the game. The subject was then given 15 minutes to play each version of the game. The versions were presented in a randomised order, and subjects were not told which version of the game they were playing. After each 15 minute session, subjects answered a series of questions about the bot they had just played. After playing all three bots, they were asked additional questions in which they were asked to compare the three versions. The questions used are listed below.

D. Questionnaire

After playing one of the bots for 15 minutes, the subjects were given a questionnaire consisting of statements about the game, to which they were asked to respond on a Likert scale. The questions could be divided into four categories according to what construct they were intended to measure, as listed in the following subsections. For the first three categories of question, -2 means strong disagreement, -1 disagreement, 0 neutral, 1 agreement, and 2 strong agreement:

1) Perception of human-ness: The following statements relate to the subject's perception as to how human-like the bot's behaviour was:

1) the bot's combat skills made it appear human-like
2) the bot's dodging skills were human-like
3) the bot's movement was human-like
4) the bot's behaviour was human-like
5) the bot appeared as if a human player was controlling it

2) Predictability: The following statements relate to the subject's judgement as to how predictable the bot was:

1) the bot surprised me with its unpredictable combat strategies
2) the bot's unpredictable combat strategies made me rethink my strategies
3) the bot's movement was often unpredictable

3) Entertainment value:

1) the bot's combat skills made it an interesting opponent
2) you enjoyed the game because of the bot's combat skills
3) the bot was fun to play against
4) the overall gaming experience was enjoyable

Subjects were also asked how challenging they found the bot as an opponent. For the second question in this group, the choices were 2 for very difficult, 1 for difficult, 0 for average, -1 for easy, and -2 for very easy. For the third question, they were 2 for very strong, 1 for strong, 0 for average, -1 for weak, and -2 for very weak:

4) Challenge:

1) the bot was effective at all aspects of combat
2) how difficult an opponent was the bot?
3) how do you rate the bot's combat skills?

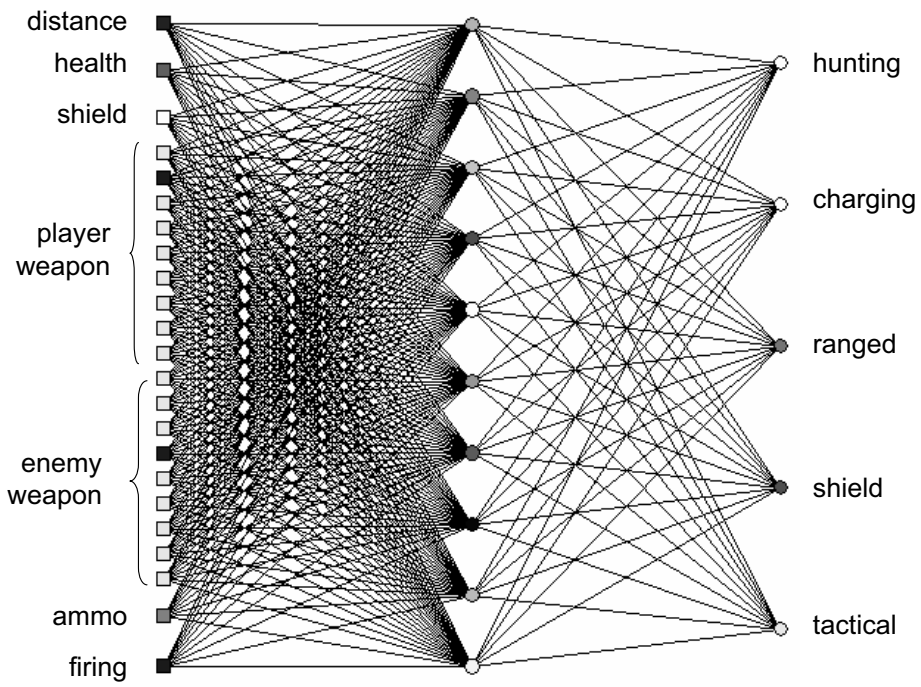

Fig. 1. Architecture of the neural network for the feedforward bot

Finally, after playing all three bots, subjects were asked about their overall experience. The bots were given meaningless code names for this purpose.

The questions were:

5) Comparison:

1) Which bot did you enjoy playing against the most?
2) Which bot did you enjoy playing against the least?
3) Which bot displayed the most human-like behaviour?
4) Which bot would you most want to play against again?

Subjects were also asked the following question: *Did you find that 15 minutes was sufficient to distinguish between the three bots?*

IV. Results and discussion

In this section, we present the questionnaire results and a simple statistical analysis. Note that 17 of the 23 subjects felt that they had sufficient time in their session to differentiate between the bots. Four felt that they did not and two were undecided.

A. Perception of human-ness

First, we present the results pertaining to our main focus: whether the subjects perceived the neural bots, trained using examples of human play, to be more human-like than the default, hand-coded bots.

Fig 2 shows the results in graphical form. Visually, the plot suggests that subjects found both neural bots to be human-like, and the hand-coded bot to be less so, and perhaps even not human-like. The results are less clear for the first and last questions, perhaps because they are phrased differently, asking whether the bots *appear* human-like, rather than whether they *are* human-like. A 2-way analysis of variance, bot type(3) by question(5), confirms that there is a difference between the bots, $F(2, 44) = 3.35, p = 0.044$. The effect of the question is not significant, $F(4, 88) = 0.365, p = 0.833$, but there is a significant interaction between bot type and question, $F(8, 176) = 3.906, p = 0.005$. An analysis of variance including only the neural bots shows no significant effect, $F(1, 22) = 0.642, p = 0.431$ and no significant interaction $F(4, 88) = 1.089, p = 0.367$.

B. Predictability

Next we examine the questions relating to perception of predictability. We expect to see that the bots percieved as human-like will also be considered unpredictable. Fig 3 confirms that the neural bots are judged similarly unpredictable, and the hand-coded bot much more predictable, and that this is consistent across all the questions. Analysis of variance shows that the bot types are different with respect to perception of predictability, $F(2, 44) = 8.968, p = 0.001$, the effect

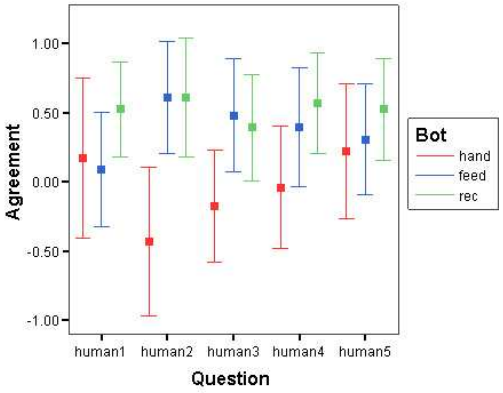

Fig. 2. Mean agreement with questions regarding human-ness. Error bars show 95% confidence intervals around the mean.

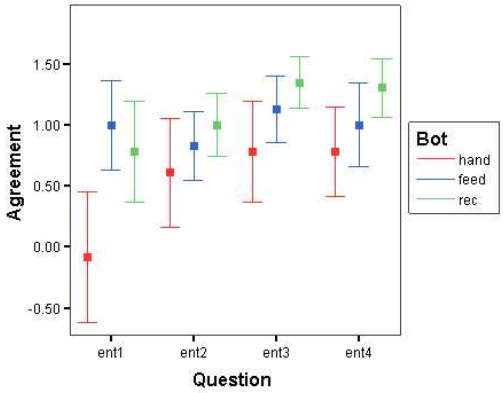

Fig. 4. Mean agreement with questions regarding entertainment value. Error bars show 95% confidence intervals around the mean.

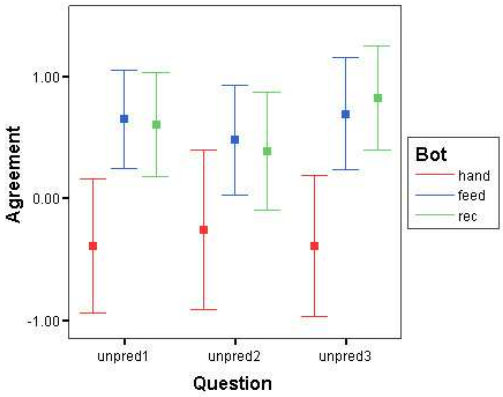

Fig. 3. Mean agreement with questions regarding unpredictability. Error bars show 95% confidence intervals around the mean.

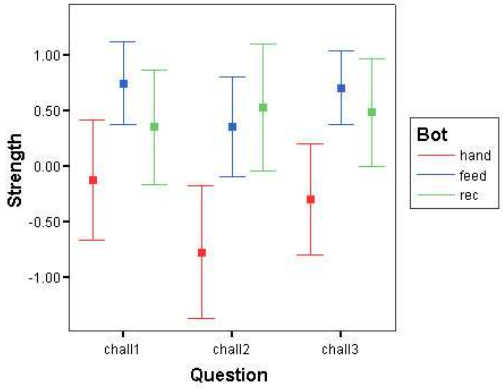

Fig. 5. Mean agreement with questions regarding level of challenge. Error bars show 95% confidence intervals around the mean.

of question is not significant, $F(2, 44) = 1.095, p = 0.343$ and there is no significant interaction $F(4, 88) = 1.222, p = 0.307$.

C. Entertainment value

Of course, the ultimate aim is to create entertaining opponents. Fig 4 shows the results for this group of questions. Once again, the neural bots are reported as more entertaining.

Analysis of variance shows significant effects for bot type, $F(2, 44) = 6.531, p = 0.003$, for question, $F(3, 66) = 12.026, p < 0.001$ and a significant interaction $F(6, 132) = 3.868, p = 0.001$. Visually, the first question appears to have different responses, possibly because subjects did not equate *interesting* with *entertaining*. If we omit this question and rerun the analysis, the interaction is no longer significant, $F(4, 88) = 0.316, p = 0.866$.

D. Challenge

On the questions related to level of challenge, Fig. 5 shows that the neural bots are consistently judged as stronger players. Analysis of variance shows that bot and question are both significant effects, $F(2, 44) = 10.033, p < 0.001$ and $F(2, 44) = 4.354, p = 0.019$, while there is no significant interaction, $F(4, 88) = 1.969, p = 0.106$.

E. Comparison

The results here are summarised in Table II. The figures in each column give the number of subjects who choose each bot type as their answer for the corresponding question. We carried out a χ^2-test for each question, and the significance levels are given in the last row.

While the figures might be suggestive, the difference from equal counts is not significant for any of these questions, due

TABLE II

DIRECT COMPARISON OF BOT TYPES

	most enjoyable	least enjoyable	most human	most replayable
hard coded	5	11	7	3
feed forward	7	6	7	8
recurrent	11	6	9	12
χ^2	2.435	2.174	0.348	5.304
p	0.296	0.337	0.840	0.070

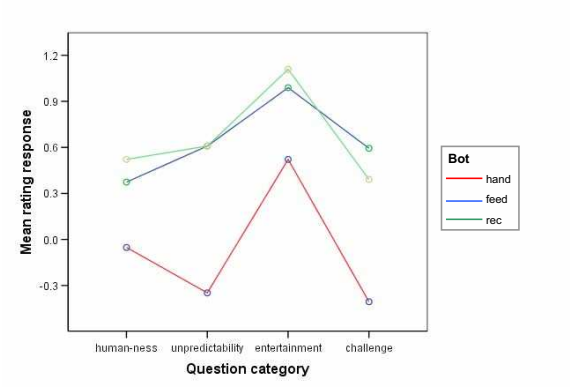

Fig. 6. Summated rating scales showing mean responses for each bot type, for each category.

to the low power of the tests. Therefore, any conclusions we draw from these data are at best tentative. The first, second and third questions, taken together, might suggest that the neural bots have more entertainment value, supporting the results shown in Fig 4. It is also interesting that, although when asked to rate each bot type separately (as in Fig 2), subjects gave a higher human-ness rating to the neural bots, when asked to compare the three, they appear less certain. Perhaps this is because the neural bot "vote" is split between the feedforward and recurrent versions. If so, it may have been better to ask subjects to order the bots rather than to select one.

F. Summary

In order to provide a summary of the overall perceptions of the subjects for the three bots in terms of human-ness, unpredictability, entertainment and challenge, we created a number of summated rating scales, by averaging the responses of the subjects for all questions in each question group, for each bot type. The results are then plotted in Fig 6, where it is obvious that the neural bots received higher ratings for all four categories.

Finally, we also constructed correlation matrices to determine the correlations between mean responses for different question catagories for each bot type, in Tables III, IV and V. In the case of the hand-coded bot, the four categories are all strongly correlated. For the feedforward bot, the correlations are weaker and less significant, especially in the relationship

TABLE III

CORRELATION MATRIX FOR THE HAND-CODED BOT

Category	1	2	3	4
1. human-ness	-			
2. unpredictability	.81**	-		
3. entertainment	.76**	.81**	-	
4. challenge	.67**	.90**	.78**	-
** correlation is significant at 0.01 level				
* correlation is significant at 0.05 level				

TABLE IV

CORRELATION MATRIX FOR THE FEEDFORWARD BOT

Category	1	2	3	4
1. human-ness	-			
2. unpredictability	.35	-		
3. entertainment	.49*	.66**	-	
4. challenge	.09	.30	.42*	-
** correlation is significant at 0.01 level				
* correlation is significant at 0.05 level				

between human-ness and challenge. The correlations are also weaker for the recurrent bot, although they are close to being significant in all cases. Overall, there is good support for the notion that the four categories are related in the context of this study.

V. CONCLUSIONS

We set out to test the idea of using machine learning to create interesting computer-controlled opponents for video games. Ultimately, the aim is to provide opponents that are enjoyable to play against, and will keep players coming back to play again and again. While this is not a new idea, and many researchers have stated similar aims, we know of no other reported testing with actual gameplay.

Despite this lack of empirical testing, it is accepted wisdom that for a bot to be enjoyable and to have replay value, it should play "like a human" and in particular should be neither too capable nor too predictable. Therefore, we have created several different bots for a popular video game, by training neural networks to imitate examples of human play, and tested these bots with gameplay sessions followed by surveys.

Our results show that in the context of our study, human players consistently find these bots to be more human-like, less predictable, more replayable, and more challenging than the provided, hand-coded bot. Furthermore, the correlation analysis shows that by and large, a player's perceptions of these bot characteristics are highly correlated.

TABLE V

CORRELATION MATRIX FOR THE RECURRENT BOT

Category	1	2	3	4
1. human-ness	-			
2. unpredictability	.44*	-		
3. entertainment	.38	.40	-	
4. challenge	.50	.67**	.64**	-
** correlation is significant at 0.01 level				
* correlation is significant at 0.05 level				

We attempted to examine the role of predictability in more detail by making one of the bots deterministic and the other stochastic. However, our subjects did not perceive the stochastic bot to be more unpredictable. It may be that longer and repeated gameplay sessions would give players sufficient time to notice the differences, but testing of this hypothesis will have to await future studies. A larger study might also make some marginal effects statistically significant. Further studies with longer and repeated sessions would also allow for investigating the effect of on-line learning, especially with regard to predictability. Whilst the bot architecture we used should support it, we felt our sessions, at only 15 minutes in length, were too short to see any effects of on-line learning.

REFERENCES

[1] C. Bauckhage, C. Thurau, and G. Sagerer, "Learning Human-like Opponent Behavior for Interactive Computer Games", *Pattern Recognition*, Lecture Notes in Computer Science 2781, pages 148-155, Springer-Verlag, 2003.

[2] B.H. Cho, S.H. Jung, Y.R. Seong and H.R. Oh, "Exploiting Intelligence in Fighting Action Games Using Neural Networks", *IEICE - Trans. Inf. Syst.*, vol. E89-D, no. 3, pp. 1249–1256, Oxford University Press, 2006.

[3] B. Gorman, C. Thurau, C. Bauckhage, and M. Humphrys, "Believability Testing and Bayesian Imitation in Interactive Computer Games", In *Proc. 9th Int. Conf. on the Simulation of Adaptive Behavior (SAB'06)*, LNAI, Springer, 2006.

[4] M. Ponsen and P. Spronck, "Improving Adaptive Game AI with Evolutionary Learning" in *Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004)*, pp. 389-396. University of Wolverhampton. 2004.

[5] D. L. Roberts, C. R. Strong, and C. L. Isbell, "Estimating Player Satisfaction through the Author's Eyes", in *Proceedings of the AIIDE'07 Workshop on Optimizing Player Satisfaction*, AAAI Press Technical Report WS-01-01, pp. 31-36, Stanford, USA, June, 2007.

[6] P. Spronck, I. Sprinkhuizen-Kuyper and E. Postma, "Improving Opponent Intelligence through Machine Learning" in *Proceedings of the 14th Belgium-Netherlands Conference on Artificial Intelligence*, pp. 299-306, 2002.

[7] P. Spronck, I. Sprinkhuizen-Kuyper and E. Postma, "Online Adaptation of Computer Game Opponent AI" in *Proceedings of the 15th Belgium-Netherlands Conference on Artificial Intelligence*, pp. 291-298. University of Nijmegen, 2003.

[8] G.N. Yannakakis and J. Hallam, "Evolving Opponents for Interesting Interactive Computer Games" in *Proceedings of the 8th International Conference on the Simulation of Adaptive Behavior (SAB'04);From Animals to Animats 8*, pp. 499-508, Los Angeles, CA, USA, July 13-17, 2004.

[9] G.N. Yannakakis, J. Hallam and H.H. Lund, "Capturing Entertainment through Heart-rate Dynamics in the Playware Playground", in *Proceedings of the 5th International Conference on Entertainment Computing*, Lecture Notes in Computer Science, vol. 4161, pp. 314-317, Cambridge, UK, September 20-22, 2006.

[10] S. Zanetti and A. El Rhalibi, "Machine learning techniques for FPS in Q3", in *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pp. 239–244, Singapore, 2004.