# Learning Human Behavior from Analyzing Activities in Virtual Environments

C. BAUCKHAGE[1], B. GORMAN[2], C. THURAU[3] & M. HUMPHRYS[2]

[1)] *Deutsche Telekom Laboratories, Berlin, Germany*
[2)] *Dublin City University, Dublin, Ireland*
[3)] *Czech Technical University, Prague, Czech Republic*

*Keywords: computer game agents, imitation learning, believability testing*

**Abstract**

Present day multiplayer video games offer an interesting perspective for researching artificial cognitive systems. In this contribution, we focus on the problem of learning believable behavior models for artificial characters. Recordings of the network traffic of modern games allow for applying machine learning techniques to realize artificial agents that act more human-like than conventional current game characters. We detail an imitation learning approach and present the results of an extensive believability study that was carried out on the Internet.

## 1. Introduction

Computer- and video games have turned into an integral part of our popular culture. Given their short history, it occurs that video games must exert a deep fascination otherwise their success would be inexplicable. Indeed, thanks to the technical performance of current computing hardware, well selling genres like action games, adventure games and simulation games do nowadays create a haunting and engaging experience for the player. They are set in dynamic, atmospheric virtual worlds of high complexity and they display an astounding level of physical accuracy and graphical detail. Moreover, data transfer over local networks or the Internet enables sharing the game experience with other human players making it a lot more unpredictable and exciting.

This amazing state of the art in game technology correlates with the fact that game development has become big business. Although modern game development requires considerable intellectual and financial efforts because it involves large teams of programmers, authors, designers and marketing specialists, the game business yields considerable revenues. Recent reports actually see the world market for video games and edutainment software rapidly closing in on $20 billion a year[1].

Apart from entertainment and financial gains, however, present day computer games also provide interesting perspectives for research in disciplines such as sociology, psychology, or computer science. In this contribution, we elaborate the latter claim. More specifically, we discuss benefits computer games might offer for *machine learning* and explore the problem of behavior learning for game characters.

Our interest in the topic arose from our background in artificial cognitive systems designed for dynamic real world settings. It occurred to us that observing human players performing tasks of different complexity in a virtual 3D world might provide new insights into intelligent behavior modeling. Accordingly, we were surprised to find that –even in modern games– rather old-fashioned ideas such as preprogrammed scripts, finite state machines, or tree searches dominate behavior programming. Of course, maturity does not imply ineptitude of a programming technique. But none of these methods is well known for their generalization capabilities. Consequently, common approaches to artificial intelligence (AI) for games may lead to ennui and frustration for experienced players. After some time of playing, the actions of computer controlled characters tend to appear artificial and lack the element of surprise human opponents would provide. If a human player acts in a way not envisaged by the game programmers, game characters simply appear to behave 'dumb' (Cass 2002).

This might be different if game characters (often called *gamebots*) were to learn from experienced human players. In fact, the idea of learning from demonstration to produce more human-like behavior is popular in cognitive systems research (cf. e.g. Schaal 1999). Until now, however, this research focused on autonomous machines intended for deployment in the physical world. This focus led to a situation where research aimed at behavior representation and learning still first and foremost struggles with issues arising from uncontrollable environmental dynamics and noisy sensors. Unlike present day robotics, however, virtual environments and computer games allow for actually concentrating on cognitive aspects of complex behaviors. While in robotics the problem of sensor noise widely prevents investigating *reactive*, *tactical*, and *strategic* decision making, computer games offer a less cumbersome avenue.

As a consequence, we find ourselves amongst a growing number of researchers who are discovering that game worlds provide challenges and opportunities for intelligent systems research. In contrast to most recent contributors, however, we pursue an approach of statistical machine learning rather than of deliberative AI. In the following, we will outline basic concepts from statistical learning and discuss how they may apply to human-like behavior modeling for virtual characters. Then, we shall survey related work in this area and present some of our results which were evaluated in an extensive online survey.

---

[1]See, for instance, a study released in 2004 by the British Dept. of Trade and Industries: http://www.dti.gov.uk/sectors/games/index.html.

# 2. Machine Learning and Video Games

The capability to learn from what we perceive and experience is essential for our everyday live. Just consider the fact that almost everything that constitutes our personality had to be learned at some point in our lives. When born, none of us knew how to walk, how to talk, or how to behave in public. As these examples indicate, learning enables flexibility and adaptation. Once we learned how to walk in our nursery, we were able to transfer this knowledge to other terrains. Thus, whenever we refer to learning in this paper, we are interested in the phenomenon generalizing from something known in order to act appropriately in a novel situation or to better perform in a familiar one.

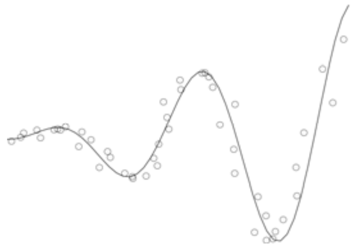
Furthermore, learning is based on examples. Without the analysis of exemplary input or role models there will be no extension of knowledge and capabilities. Given the importance of this mechanism, it is no surprise to find it to be innate and even 'hardwired' into our brains. Especially if it comes to behavior learning, experiments in behavioral science document that already newborn infants endeavor to reproduce activities they observe in their surroundings (Rao & Meltzoff 2003). Recent neurophysiological examinations even indicate that there are particular brain areas specialized in imitation (Kohler, Keysers, Umiltà, Fogassi, Gallese & Rizzolatti 2002).

## 2.1 Machine Learning and Pattern Recognition

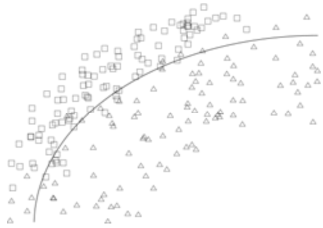
*Machine learning* (ML) is an area of computer science that tries to mimic the flexible learning capabilities of the human brain. It deals with the development of algorithms that learn from examples and apply this knowledge in order to produce reasonable output if confronted with input they never saw before. Note, however, that even though the performance of an algorithm that has learned from examples rather depends on the analysis of data sets than on the intuition of engineers, human intuition cannot entirely be abandoned. The designer of a ML system still must specify the data to learn from as well as the method to analyze it.

Acquiring knowledge requires mechanisms to represent knowledge. Structural methods like rule bases or grammars, for instance, encode relations among pieces of *symbolic* information. *Statistical* machine learning, in contrast, deals with numerical data. The standard approach is to consider vectors $\mathbf{x} = [x_1, x_2, ... x_n]^T$ whose components encode numerical values of features that characterize certain entities. Given a training set of exemplary data, the task of a statistical ML system is to find mathematical functions which provide an abstract description of the examples. This is usually done by adapting the parameters of a given method for function approximation. Common such methods are Gaussian mixture models, neural networks, or support vector machines[2].

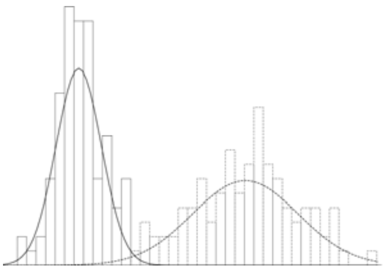
[2] For the sake of completeness, we should note that there also are hybrid machine learning techniques. So called *graphical models* such as Bayesian networks or Hidden Markov Models process numerical data but come along with an underlying graph structure with weighted edges. Suitable values for the weights are learned from examples.



(a) Regression



(b) Classification



(c) Density estimation

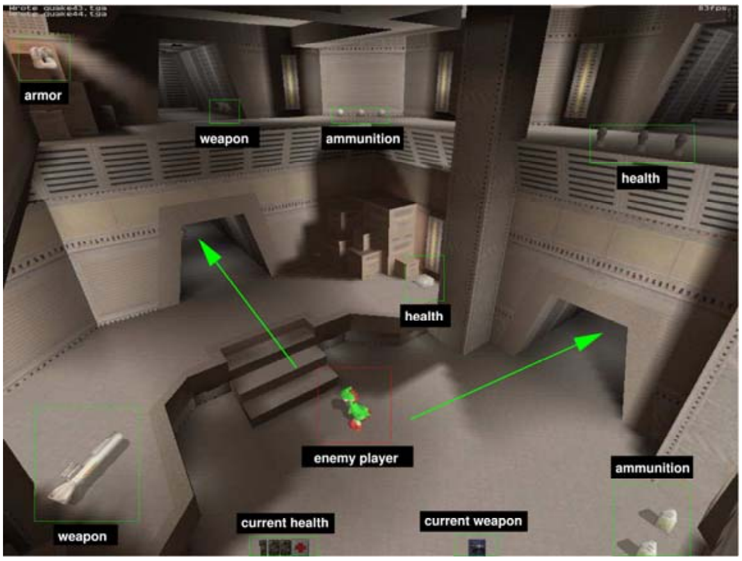
*Figure 1: Three topics of statistical machine learning: 1(a) regression fits a function into a set of data points; 1(b) classification searches for boundaries between classes of data points; 1(c) density estimation determines how data points are distributed.*

Figure 1 exemplifies what statistical ML may accomplish. Given a training set of pairs $\{(\mathbf{x}^{\alpha}, \mathbf{y}^{\alpha})\}$, $\alpha$=1, ..., $N$, the *regression* task tries fitting a function $f$ to the data such that a new input $\mathbf{x}$ will yield the most plausible output $\mathbf{y} = f(\mathbf{x})$. More formally, if we assume the in- and output of the system to be random variables $X$ and $Y$, respectively, the objective is to estimate the expected value $E(Y \mid X = \mathbf{x})$. A typical application for this would be time series forecasting in stock market analysis.

If the $n$ dimensional vector space $V^n$ of input data is partitioned into $K$ different classes, one might want to know to which class an input vector $\mathbf{x}$ belongs. Instances of this problem are automatic speech recognition or object recognition in computer vision. For training, the *classification* task requires a set of pairs $\{(\mathbf{x}^{\alpha}, y^{\alpha})\}$ where $y^{\alpha} \in \{1,...,K\}$ denotes the class index of the pattern vector $\mathbf{x}^{\alpha}$. The goal is to learn a function $f: V^n \to \{1,...,K\}$ that partitions the input space and maximizes the probability $P(Y \mid X = \mathbf{x})$.

Finally, given a data set $\{\mathbf{x}^{\alpha}\}$, statistical ML can produce a functional description of the distribution $p(\mathbf{x})$ of the data. Applications of this task of *density estimation* can, for instance, be found in several data compression technologies.

Machine learning algorithms are often categorized with respect to the training data that is provided. *Supervised learning* characterizes algorithms that generate a function which maps inputs to desired outputs. The tasks of regression and classification would thus typically be dealt with by supervised learning techniques. Algorithms for *unsupervised learning* generate a model for a set of inputs; density estimation would hence be an example for unsupervised learning. In *reinforcement learning*, the algorithm itself creates pairs of input/output vectors and has to apply a trial-and-error strategy to determine whether they lead to a desired goal.




(a) 3D game environment

(b) Training data generation at a LAN party.

*Figure 2: Complex 3D computer game worlds are popular among players.*

## 2.2 How Does It Relate to Video Games?

Our tendency to imitate successful behavior certainly also apply to the way we learn to play a video game. Since imitating tricks and routines of experienced players leads to more success, learning from demonstration may also provide an avenue to programming engaging behaviors for computer game characters.

For the reminder of this contribution, we will consider behavior learning for the game QUAKE II® in which the player moves through a virtual 3D world (also called a *map*) which (s)he perceives from the first person perspective (see Fig.2(a)). The map is loosely based on the physics of the real world. Players can move freely only constrained by the game physics. Though tactical variations exist, the player's task is to gain as many points as possible by battling other characters. In doing so, the player will loose health, armor, and ammunition but can compensate it by collecting corresponding items distributed all over a map. Items will reappear at the same position shortly after having been picked up. This induces strategies into game play. Winning will be facilitated by *item control*, which means moving through the map such that you secure the best items for yourself and leave the weaker ones for your adversaries.

Obviously, the state of a game character can be characterized by its current position and view on the map and its current armament and health conditions as well its distance to possible foes. If these features are thought of as components of a *state vector*, the current state of the character corresponds to a point in a high dimensional state space. The history of states a character assumes during a game will form a path in this state space. Neglecting a possible dependency on former actions and assuming the state of player *p* at time *t* to be given by a vector $\mathbf{x}_t^p$, a simple first order approximation of the player's state at the next time step *t*+1 could hence be modeled as $\mathbf{x}_{t+1}^p = F(\mathbf{x}_t^p, \mathbf{y}_t^p(\mathbf{x}_t^p), \mathbf{e}_t)$ where *F* is some unknown function, $\mathbf{e}_t$ denotes environmental influences at time *t* and $\mathbf{y}_t^p(\mathbf{x}_t^p)$ represents the action player accomplishes according to his current state. Restating this expression as $\mathbf{y}_t^p = f(\mathbf{x}_{t+1}^p, \mathbf{x}_t^p, \mathbf{e}_t)$ reveals that this model corresponds to what Arkin (1998) calls *reactive behaviors*. The actions of a player only depend on his or her state and on the current environmental influence. We also recognize that our model resembles the regression task in machine learning. Thus, given suitable training data, prototypical actions $\mathbf{y}^{p,t}$ or situated behaviors might be *learnable*. Since they simply correspond to sequences of actions $\{\mathbf{y}_{t_i}^p, \mathbf{y}_{t_{i+1}}^p, ..., \mathbf{y}_{t_{i+n}}^p\}$, techniques like neural networks, support vector machines, or Bayesian learning may apply.

Since a demo contains recordings of the network traffic of a multiplayer game, it encodes the series of states $\mathbf{x}_0^p, \mathbf{x}_1^p, \mathbf{x}_2^p, ..., \mathbf{x}_T^p$ the recording player *p* underwent during a game. It also includes information about nearby items and other players as well as temporary entities. There is no need for a visual analysis of a game scene, since all necessary information is already available on a cognitive higher level. The same applies to the player actions; they are included as simple velocity and position vectors.

# 3. Related Work

During the past two years, we could witness an increased academic interest in the problem of believable computer game characters. One of the driving factors behind this interest was already mentioned in the introduction and has also been noted by authors such as Cass (2002) or Nareyek (2004): on the one hand, commercial games still mainly rely on well seasoned, deliberative AI techniques like finite state machines or game trees. On the other hand, statistical machine learning as a tool to produce believably acting game agents has been largely neglected by the scientific community. This, however, seems to be changing.

Recent work by Spronck, Sprinkhuizen-Kuyper & (2003) introduced reinforcement learning to the task of rule selection for agent behavior in a commercially available role playing game. Earlier, the same authors reported on a hybrid coupling of genetic algorithms and neural networks for offline learning in a simple strategy game (Spronck, Sprinkhuizen-Kuyper & 2002).

The idea of using human generated data to train game agents was first reported by Sklar, Blair, Funes & (1999) who collected the key-strokes of people playing *Tron* in order to train neural networks. Just recently, Le Hy, Arrigioni, Bessière & (2004) described advanced probabilistic action selection for a commercial game using Bayesian networks which are trained by means of human generated input.

Next, we will summarize some of our own results in using machine learning techniques for producing human-like bot behavior for modern video games.

# 4. Imitation Learning

In this section, we outline our current approach to imitating human movement and strategic behavior in QUAKE II®. The model discussed by Gorman, Thurau, Bauckhage & (2006) focuses on two core aspects of human behavior; *strategic planning* and *motion modeling*. Several investigations (Laird 2001, Livingstone 2006) have found that the ability of an agent to exhibit long-term strategic planning faculties is a crucial factor in determining how human-like its behavior appears. The importance of *motion modeling* is equally evident because human players frequently exhibit actions other than simply moving along the environment surface. In many cases, the player can only attain certain goals by performing one or more such actions at the appropriate time; they therefore have an important *functional* element. From the perspective of creating a believable agent, it is also vital to reproduce the *aesthetic* qualities of movements and activities.

## 4.1 Learning Goal-Oriented Strategic Behaviors

In QUAKE II®, experienced players roam the environment methodically, controlling important areas of the map and picking up *items* to strengthen their character. Thus, a player's long-term goal can be seen in systematically collecting items found at certain points of a map. By learning the mappings between the player's status and his subsequent item pickups, the agent can adopt observed strategies when appropriate, and *adapt* to situations which the player did not face.

We first read the set of all player locations $\mathbf{l}=[x,y,z]$ from the recording, and cluster them to produce a *goal-oriented* discrimination of the level's topology. We also construct an $n \times n$ matrix of edges $E$, where $n$ is the number of clusters, and $E_{i,j} = 1$ if the player was observed to move from node $i$ to node $j$ and 0 otherwise. The player's *inventory* –the list of what quantities of which items he currently possesses– is also read from the demo and unique state vectors are obtained; these *inventory prototypes* represent the varying situations faced by the player during a game. We can now construct a set of *paths* which the player followed while in each such situation.

Having obtained the different paths pursued by the player in each inventory state, we turn to reinforcement learning to learn his or her behavior. The topological map of the game environment may now be viewed as a *Markov Decision Process* (MDP), with the clusters corresponding to states and the edges to transitions. In this scenario, the MDP's actions are considered to be the choice to move to a given node from the current position. Thus, the transition probabilities are $P(c' = j \mid c = i, a = j) = E_{ij}$ where $c$ is the current node, $c'$ is the next node, a is the executed action, and $E$ is the edge matrix. We assign an increasing reward to consecutive nodes in every path taken under each prototype, such that the agent will be guided along similar paths to the human when facing similar situations.

To model player's intuitive *weighing* of strategic objectives, and his understanding of *object transience*, we introduce a weighted *fuzzy clustering* approach and an *item activation* variable $m_p(\mathbf{s})$. Its membership distribution implicitly specifies the agent's current goals, which will later facilitate integration with the Bayesian motion-modeling system. The final utilities thus result from:

$$U(c) = g^{e(c)} \sum V_p(c) m_p(\mathbf{s}), \quad c_{t+1} = \max_y U(y), \quad y \in \{x \mid E_{c,x} = 1\} \tag{1}$$

where $U(c)$ is the final utility of node $c$, $\gamma$ is the discount, $e(c)$ is the number of times the player has entered cluster $c$, $V_p(c)$ is the original value of node $c$ in state prototype $p$, and $E$ is the edge matrix.

## 4.2 Bayesian Motion Modeling

It is not sufficient to simply identify the player's goals and the paths along which (s)he moved to reach them; it is also necessary to capture the actions executed by the player in pursuit of these goals. Here, we apply a Bayesian inverse-model for action selection in infants and robots due to Rao & Meltzoff (2003). The choice of action at each time step is expressed as a probability function of the subject's current position $c_t$, next position $c_{t+1}$ and goal $c_g$:

$$P(a_t|c_t, c_{t+1}, c_g) = \frac{P(c_{t+1}|c_t, a_t) P(a_t|c_t, c_g)}{\sum_u P(c_{t+1}|c_t, a_u) P(a_u|c_t, c_g)} \tag{2}$$

This model fits into the strategic navigation system almost perfectly; the clusters $c_t$ and $c_{t+1}$ are chosen by examining the utility values, while the current goal state is implicitly defined by the membership distribution. In order to derive the probabilities, we read the sequence of actions taken by the player as a set of vectors $\mathbf{v}$. We then cluster these action vectors to obtain a set of *action primitives*, each of which amalgamates a number of similar actions performed at different times into a single unit of behavior.

Several important adaptations must be made in order to use this model in the game environment. Firstly, Rao's model assumes that transitions between states are instantaneous, whereas multiple actions may be performed in QUAKE II® while moving between successive clusters; we therefore express $P(c_{t+1} \mid c_t, a_t)$ as a soft-distribution of all observed actions on edge $E_{ct,ct+1}$ in the topological map. Secondly, Rao assumes a single unambiguous goal, whereas we deal with multiple weighted goals in parallel. We thus perform a similar weighting of the probabilities across all active goal clusters. Finally, Rao's model assumes that each action is independent of the previous action. In QUAKE II®, however, each action is constrained by the action performed on the preceding time step; we therefore introduce an additional dependency in our calculations. The final probabilities are computed as follows:

$$\sum_g m_g P(a_t|c_t, c_{t+1}, c_g) \frac{P(a_t|a_{t-1})}{\sum_u P(a_u|a_{t-1})} \quad (3)$$

## 5. Believability Testing

Still, there exists no standard method of gauging the 'believability' of game bots. Given that one of the central aims of our work lies in improving this quality of such agents, we need to address this shortcoming. The most obvious means of determining the degree to which agents are perceived as human is to conduct a survey. This, of course, raises questions of subjectivity, experimenter influence, and so on. In order to produce a credible assessment of agent believability, any proposed system must be designed with these concerns in mind. Our aims, then, are as follows: i) to construct a framework which facilitates rigorous, objective testing of the degree to which game agents are perceived as human; ii) to formulate a *believability index* expressing this 'humanness', and allowing comparisons between different agents.

To counteract any potential observer bias, our test takes the form of an anonymous Internet-based survey (see Fig. 3 for a screenshot of one of the forms). Respondents are presented with detailed instructions covering all aspects of the test. They are not asked for personal data such as age or gender, but are required to estimate their gaming experience at one of five different levels:

1. Never played, rarely or never seen
2. Some passing familiarity (played / seen infrequently)
3. Played occasionally (monthly / every few months)
4. Played regularly (weekly)
5. Played frequently (daily)

Upon proceeding to the test itself, respondents are presented a series of pages, each of which contains a group of video clips. Each group shows similar sequences of game play from the perspective of the in-game character. Within each group, the clips may depict any combination of human and artificial players; the respondent is required to examine the behavior of the character in each clip, and indicate whether (s)he believes it is a human or artificial player. The clips are marked on a gradient, as follows: 1: Human, 2: Probably Human, 3: Don't Know, 4: Probably Artificial, 5: Artificial

This rating is the central conceit of the survey and will later be used to compute the believability index. Additionally, the respondent may provide an optional comment explaining his/her choice. In cases where (s)he indicates that (s)he believes the agent to be artificial, (s)he will be further asked to rate how "human-like" (s)he perceives its behavior to be, on a scale of 1 to 10. This more subjective rating is not involved in the computation of the believability index, but may be used to provide additional insight into users' opinions of different agents.

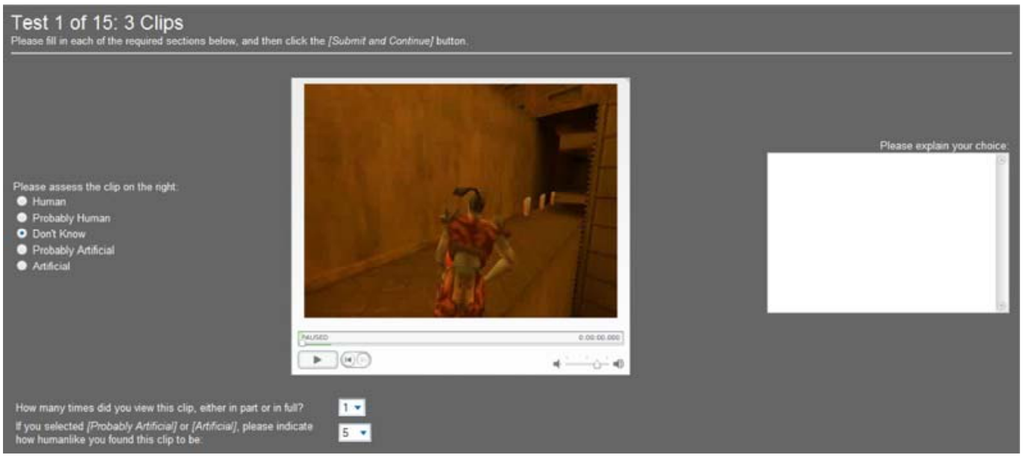



*Figure 3: Extract from the believability test questionnaire.*

## 5.1 Subjectivity, Bias and Other Concerns

Aside from the observer effect, there are several areas in which the potential for subjectivity and the introduction of bias exist. Since our aim is to provide an objective measure of believability, these must be eliminated or minimized.

The first obvious pitfall lies in the selection of video clips. The selector may deliberately choose certain clips in an effort to influence the respondents. To guard against this, we first ensure that the number of samples is sufficient to embody a wide variety of behaviors, and secondly, we cede control of the selection of the specific behaviors to an unbiased arbiter. In our case, we wished to compare the believability of our imitation agents against both human players and traditional rule-based bots; thus, we first ran numerous simulations with the traditional agent –over whose behavior we had no control– to generate a corpus of game play samples, and then proceeded to use human clips embodying similar movements and activities both in the believability test and to train our imitation agents.

Similarly, the order in which the videos are presented could conceivably be used to guide the respondents' answers. To prevent this, we randomize the order in which the groups of clips are displayed to each user, as well as the sequence of clips within each page; the test designer thus has no control over the order of the samples seen by the user.

Another issue concerns the possibility that users will choose the 'Probably' options in a deliberate effort to artificially minimize their error and 'beat' the test, or that they will attempt to average out their answers over the course of the survey – that is, they may rate a clip as 'human' for little reason other than that they rated several previous clips as 'artificial', or vice-versa. To discourage this, we include notes on the introduction page to the effect that the test does not adhere to any averages, that the user's ratings should be based exclusively upon their perception of the character's behavior in each clip, and that the user should be as definitive as possible in their answers. A related problem is that of user fatigue; as the test progresses, the user may begin to lose interest, and will consequently invest less effort in each successive clip. We address this by including a feature enabling users to save their progress at any point, allowing them to complete the survey at their convenience.

It is also imperative to ensure that the test is focused upon the variable under investigation – namely, the believability of the agent's movement and behavior. As such, the survey must be structured so as not to present 'clues' which might influence the respondents. For instance, the tester should ensure that all clips conform to a standard presentation format, so that the respondent cannot discern between different agents based on extraneous visual cues. To this end, we run a script over the demo files to remove all such indicators. In the resulting clips, all agents are rendered using the same model, they are given the same name, and the display perspective is homogenized to a common point of view. In the specific case of our imitation agents, this requirement that all extraneous indicators be removed raises a conflict between two of our goals in conducting the survey. If the players in two of the three clips we use on each page begin from the same location and exhibit near-identical behavior, the respondent may conclude through pure logical deduction that (s)he is probably viewing a human and imitation agent, and consequently that the remaining clip is more likely to be a traditional artificial agent. Note that this might not necessarily be true, but even an incorrect answer based on factors other than believability will adversely affect the accuracy of the results. We circumvent this problem by training imitation agents with different (but similar) samples of human game play to those actually used in the test. The resulting clips are therefore comparable, but do not 'leak' any additional information; respondents must judge whether or not they are human based solely on their appearance. At the same time, however, we obviously wish to test how accurately our agents can capture the aesthetic appearances of their human exemplars. To satisfy both requirements, a small minority of imitation agents are trained using the same human data as presented in the survey; in the experiments described below, 2 of the imitation agents were direct clones, while the remainder were trained on different data.

## 5.2 Evaluation of Results

Before evaluating the results of the survey, one should ensure that there have been a substantial number of responses with a decent distribution across all experience levels; a good 'stopping criterion' is to run the test until the average experience level is at least 3 (i.e. a typical, casual games player). Standard analyzes (precision, recall, etc) can be carried out on the results; however, as mentioned earlier, we also wish to formulate a believability index which is specifically designed to express the agent's believability as a function of user experience and the certainty with which the clips were identified.

Recall that each clip is rated on a scale of 1 (definitely human) to 5 (definitely artificial). Obviously, the true value of each clip is always either 1 or 5. Thus, we can express the degree to which a clip persuaded an individual that the visualized character was human as the normalized difference between that person's rating and the value corresponding to 'artificial':

$$h_p(c_i) = \frac{|r_p(c_i) - A|}{\max(h)} \tag{4}$$

where $h_p(c_i)$ is the degree to which person p regarded the clip as depicting a human, $r_p(c_i)$ is person *p's* rating of clip *i*, A is the value on the rating scale which corresponds to 'artificial', and *max*(*h*) is the maximum possible difference between a clip's rating and the value of 'artificial'. In other words, $h_p(c_i)$ will be 0 if the individual identified a clip as artificial, 1 if he identified it as human, and somewhere in between if he chose one of the 'Probably' or 'Don't Know' options. We now weight this according to the individual's experience level:

$$w_p(c_i) = \frac{e_p h_p(c_i)}{\max(e)}, \tag{5}$$

where $e_p$ is the experience level of person *p* and max(*e*) is the maximum experience level. Thus, the believability index is conditioned upon a sufficient level of expertise among respondents; if their average experience level is 1, for instance, then their responses will be weighted into insignificance and the believability index will be correspondingly low. Finally, we sum the weighted accuracies across all clips and respondents, and take the average:

$$b = \frac{\sum_p^n \sum_i^m w_p(c_i)}{nm}, \tag{6}$$

where *b* is the believability index, *n* is the number of individual respondents, and *m* is the number of clips. The believability index is, in essence, a weighted representation of the degree to which a given type of clip was regarded as human. In the context of the survey, then, a 'good' result for an AI agent would involve a high value of *b* for both the agent and human clips.

## 6. Experiments

The main purpose of the experiment described in this section was to examine how believable our imitation agents were in comparison with human players and traditional rule-based artificial agents. It consisted of 15 groups of video clips, with 3 clips in each; these clips were, on average, approximately 20 seconds in length. We first ran numerous simulations involving the rule-based artificial agent to derive a set of game play samples, and then used similar samples of human players both in the test itself and to train our imitation agents. The rule-based agent used was the QUAKE II® Gladiator bot, which was chosen due to its reputation as one of the best bots available.

With the video clips in place, the URL of the survey site was distributed to the mailing lists of several colleges in Ireland and Germany. After a one-week test period, we had amassed a considerable number of responses. After discarding incomplete responses, we were left with 20 completed surveys, totaling 900 individual clip ratings.

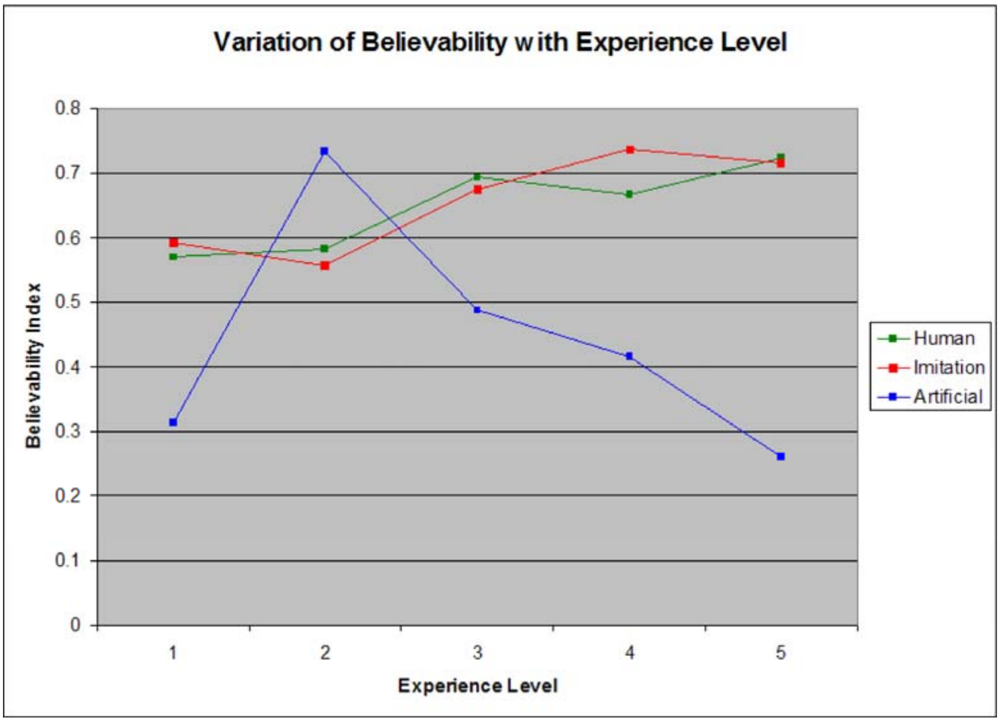



*Figure 4: Variation of perceived believability with experience level.*

The survey produced a very favorable impression of our imitation agents compared to the artificial agent. The believability indices for human, imitation and traditional artificial clips were 0.69, 0.69 and 0.35, respectively. In other words, the imitation agents were misidentified as human 69% of the time, while the rule-based agents were mistaken as human in only 35% of cases (weighted according to experience). Clips which actually *did* depict human players were also identified 69% the time. Essentially, it seems that respondents were generally unable to discern between the human players and our imitation agents. These results are corroborated by the recall values, which indicate that both the human and imitation clips were classified as human in approximately 68% of cases, while the rule-based agent was classified as human only 36.69% of the time. Since the human sources used to train the imitation agents were different than those human clips presented as part of the test, this implies that the results are based on the general abilities of the imitation mechanism, rather than any factors unique to the clips in question.

Further indication of the imitation agents' effectiveness is evident in the graph of believability against experience level shown in Fig. 4. While an in-depth psychological explanation of the curves displayed there is beyond the scope of our work, it is noticeable that, as the experience level rises, respondents correctly identify human clips as human more frequently, and misidentify the traditional agent as human less frequently. The identification of imitation agents as human, by contrast, closely parallels that of genuine human clips. These trends may be explained by the fact that more experienced players have a greater knowledge of characteristically human behaviors – smooth strafing, unnecessary jumping, pausing to examine the environment, and similar idiosyncrasies – which the traditional agent would not exhibit, but which would be captured and reproduced by the imitation bots.

In summary: the results of the believability study suggest that our imitation agents exhibit far greater 'humanness' than even a well-regarded rule-based agent, and indeed are comparable to genuine human players. We consider this to be strong evidence in support of our original premise; namely, that imitation learning has the potential to produce more believable game agents than traditional AI techniques.

## 7. Conclusion

In this paper, we considered virtual computer game worlds as a testbed that allows for studying the problem of modeling human behavior. We reviewed concepts in statistical machine learning and described one of our own approaches to behavior learning from human generated data. Also, we proposed a formal method of quantifying the degree to which different agents are perceived as ’human-like’, in the form of a web-based survey and an objective metric based on both the respondents’ level of experience and the accuracy with which the players/agents were identified. Through our experiments, we showed that our imitation-learning approach produces game bots which are capable of conveying a significantly more human-like impression than traditional rule-based agents, and are often almost indistinguishable from genuine human players.

## References

Arkin, R. C. (1998). *Behavior-Based Robotics*. Cambridge, MA: MIT Press.

Cass, S. (2002). Mind Games. *IEEE Spectrum, 39,* (12), 40–44.

Gorman, B.; Thurau, C.; Bauckhage, C. & Humphrys, M. (2006). Believability Testing and Bayesian Imitation in Interactive Computer Games. In: *Proc. Int. Conf. on Simulation of Adaptive Behavior*. Berlin, Germany: Springer Verlag.

Kohler, E.; Keysers, C.; Umiltà, M.; Fogassi, L.; Gallese, V. & Rizzolatti, G. (2002). Hearing Sounds, Understanding Actions: Action Representation in Mirror Neurons. *Science*, 297, 846–848.

Laird, J. E. (2001). Using a Computer Game to develop advanced AI. *IEEE Computer, 34,* (7), 70–75.

Le Hy, R. ; Arrigioni, A. ; Bessière, P. & Lebeltel, O. (2004). Teaching Bayesian behaviours to video game characters. *Robot. Auton. Syst.*, 47 (2–3), 177–185.

Livingstone, D. (2006). Turing’s test and believable AI in games. *Comput. Entertain., 4,* (1), 6.

Nareyek, A. (2004). Artificial Intelligence in Computer Games – State of the Art and Future Directions. *ACM Queue*, 1 (10), 58–65.

Rao, R. & Meltzoff, A. (2003). Imitation learning in infants and robots: Towards probabilistic computational models. In: *Proc. AISB 2003 Convention: Cognition in Machines and Animals, Aberystwyth, UK.* Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

Schaal, S. (1999). Is Imitation Learning the Route to Humanoid Robots? *Trends Cogn. Sci.*, 3 (6), 233–242.

Sklar, E.; Blair, A.; Funes, P. & Pollack, J. (1999). Training intelligent agents using human internet data. In: *Proc. 1st Asia-Pacific Conference on Intelligent Agent Technology*. Hong Kong: Hong Kong Baptists University.

Spronck, P.; Sprinkhuizen-Kuyper, I. & Postma, E. (2002). Improving Opponent Intelligence through Machine Learning, In: *Proc. GAME-ON*. London, UK: EOROSIS.

Spronck, P.; Sprinkhuizen-Kuyper, I. & Postma, E. (2003). Online Adaptation of Game Opponent AI in Simulation and in Practice. In: *Proc. GAME-ON*. London, UK: EUROSIS.