OPPONENT-BASED TACTIC SELECTION FOR A FIRST PERSON SHOOTER GAME

David Thomson, Antonija Mitrovic
Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand
david.thomson@pg.canterbury.ac.nz, tanja.mitrovic@canterbury.ac.nz

Keywords: Opponent modeling, Adaptive AI, Machine Learning, Student Modeling, Video Games, Quake 3

Abstract: Video games are quickly becoming a significant part of society with a growing industry that employs a wide range of talent, from programmers to graphic artists. Video games are also becoming an interesting and useful testbed for Artificial Intelligence research. Complex, realistic environmental constraints, as well as performance considerations demand highly efficient AI techniques. At the same time, the AI component of a video game may define the ongoing commercial success, or failure, of a particular game or game engine. This research details an approach to opponent modeling in a first person shooter game, and evaluates proficiency gains facilitated by such a technique. Information about the user is recorded and used by the existing Artificial Intelligence component to select tactics for any given opponent. The evaluation results show that when computer characters use such modeling they are more effective than when they do not model their opponent.

1 INTRODUCTION

For better or for worse, video games are becoming a significant part of our society. While once little more than a curiosity, the video game industry has grown rapidly, to the point where it generated about USD$9.5 billion in the US in 2007, and USD$11.7 billion in 2008 (Entertainment Software Association, 2010). Video games are an interesting environment for developing new techniques for writing software, with complicated settings, efficiency demands, and often very realistic physics engines. In particular, games are useful testbeds for evaluating new Artificial Intelligence techniques.

This research uses the video game Quake 3 to evaluate an implementation of user modeling. User modeling is a general term used for collecting and processing data about each user (Kobsa, 2001).

This research takes methods from student modeling (modeling students to help teach them), and adapts them to a First Person Shooter (FPS) game. Student modeling, used in Intelligent Tutoring Systems (ITSs), has been shown to be effective at increasing both cognitive (Hartley & Mitrovic, 2002) and meta-cognitive (Mitrovic & Martin, 2007) ability. When used in ITSs, student modeling allows adaptive problem and feedback selection, with the aim of keeping the student at an appropriate difficulty level; not too hard, but not too easy. This level is called the Zone of Proximal Development.

2 BACKGROUND

Opponent modeling has been incorporated into a number of games. Opponent modeling refers the technique of using user modeling for the opponents in a video game. Some such games that have been used to evaluate opponent modeling include Poker (Billings et al., 1998) Racing games (Togelius et al 2006) and Real-Time Strategy Games (Schadd et al., 2007). Machine Learning techniques have also been incorporated into Quake 3 (Zanetti & Rhalibi, 2004), with mixed results. A detailed description of all these systems has not been included due to space constraints.

In this paper, we use Quake 3 as the context of our research. Quake 3 (full name Quake 3 Arena) is a death-match style FPS game. Quake 3 was released on December 2nd 1999 by id software. It focuses on multi-player action, and has no story line. In single player mode, the player competes in a series of matches against computer-controlled characters. The most common form of play is the death-match. In a death-match, all players compete against each other, on a single map. The winner is the player who scores the most points (points are achieved by killing other players) in the time limit, or the player who reaches the score limit first.

When a character shoots at and hits an opponent, the opponent loses some of its health. How much health it loses depends on the weapon that was used. If this causes the characters health to reduce below zero, the character dies. After a brief pause, the character re-spawns (comes back to life) at a different location and with only basic weapons. This ensures players continuous game-play, even if they get killed a lot. There are a number of different weapons in the game that can be found scattered around a map. A player can pick up such weapons and use them against their enemies.

3 OPPONENT MODELING IN QUAKE 3

Although the computer-controlled characters in Quake 3, or bots, as they are known, are already capable of playing the game in an effective manner, we wanted to examine whether modeling opponents would make a measurable difference to the bots skill. We believe an aspect of Artificial Intelligence (AI) has been overlooked in many computer games, namely, adaptation. Being able to adapt is a vital skill humans have, and must be a key goal if we wish to reach the pinnacle of artificially intelligent computers.

We believe adaptive AI will be increasingly useful for developers to differentiate their games. While some work is surely occurring in industry, we believe this is an exciting field for research. Adaption in computer games is also crucial to the effectiveness of educational computer games, a field that is rapidly growing in both size and importance.

In humans, the ability to adapt comes from our ability to remember. When faced with a decision, we recall, either sub-consciously or consciously, previous situations where we had to make similar decisions. We examine the decision we made on that occasion, and whether the result was desirable, or not. If our previous decision led to a desirable outcome, we are likely to make a similar decision. If, on the other hand, the previous decision led to undesirable outcomes, we are likely to make a different decision. In essence, we are remembering what worked well in the past, and what did not. An important aspect of this skill is in identifying previous situations that are relevant to the current decision.

Our research in student modeling has taught us that there is no point modeling what you cannot use, and you cannot use what you cannot model. Once we know what is desirable, we can identify decisions in situations that lead to desirable outcomes. If a decision does not at all influence the outcome, there is little point modeling it. If we cannot change a decision that influences the outcome, even with perfect information, then there is also no point modeling that decision.

This research has added opponent modeling to Quake 3. Bots model opponents in real time, as they compete in a match.

During the match, every bot is constantly evaluating and deciding what they want to do. One aspect of this is what weapon to hold. Other aspects are concerned with goals such as seek weapon, seek health etc. Opponent models are now used in three specific aspects of decision making: choosing a weapon to hold, choosing which weapons to seek, and deciding whether to chase or flee an opponent. These decisions were chosen because they are related to opponents and weapons. The information in the model is limited to the success of different weapons, so is therefore only helpful when making decisions regarding opponents or weapons.

```
models lucy
{
  opponent ranger
  {
    GAUNTLET 0 0 0 *0.73*
    MACHINEGUN 0 *0.9*
    SHOTGUN 1 *1.10*
    GRENADELAUNCHER 0 *0.9*
    ROCKETLAUNCHER 1 1 0 1 *1.2*
    LIGHTNING 0 0 1 *0.89*
    RAILGUN 1 *1.1*
    PLASMAGUN 1 0 0 0 *0.8*
    BFG10K 1 *1.1*
  }
}
```

Figure 3: An opponent model for a Quake 3 bot.

Figure 3 shows a section of the opponent models for the bot Lucy, showing the model for the opponent Ranger. Each bot will have a similar model for every opponent they have ever fought.

For each opponent, a history of interactions between the opponent and the bot is recorded. An interaction is defined as a kill or a death. During a match other interactions may occur, such as a hit that does not kill the target, or a shot that misses. It is however hard to define the meaning to the result of such interactions, so they are not recorded. For each weapon, a history of kills and deaths is recorded.

Following the history of interactions is the success value for the current weapon. The success value is calculated by multiplying the current value as the history is parsed, by 1.10 for each “1”, and 0.9 for “0” observed, starting with a value of one. A second method of calculating the success value has been investigated, but is not detailed here due to space constraints.

The purpose of the success value is to improve the performance. As bots are constantly evaluating what they wish to do, the opponent models will be read very often. Maintaining a success value for each weapon means the whole history for each weapon does not need to be read every time the bot makes a decision. The history only needs to be read when re-calculating the success value, which is only necessary after a kill or a death.

With our opponent modeling implementation, bots select weapons based on a combination of the success value for every weapon against the current opponent, and the bots own preference for each weapon. This means the bots still act differently (they each have different preferences), but combine that with knowledge about the opponent

The opponent model is also used when the bot must decide if they want to chase a fleeing opponent, and if they themselves want to flee an opponent, or keep fighting. The bots are now more likely to flee from a string opponent, and more likely to chase a weak opponent, than when opponent modeling was not used.

4 EVALUATION

The purpose of this research is to make the bots in Quake 3 Arena smarter. This is achieved by effectively giving the bots memory. The bots remember what worked well in the past, and use this information to influence their strategy. The result is that a bot who uses opponent modeling should score more points, by killing more opponents, than if they were not using opponent modeling. The bots scores can then be used to measure the benefit realised from opponent modeling.

Although opponent modeling is intended to be used against human players, the bots in Quake 3 have been specifically designed to play like a human player. Evaluation can thus be performed between computer characters, without the need for human participants. The information modeled by the computer characters is also low-level enough that the same information can be extracted from modeling computer characters, as would be from a human player. Computer players also must choose a weapon from the selection of available weapons, and kills and deaths occur in the same manner. Computer players competing in matches against one another are therefore used to evaluate the effectiveness of opponent modeling in this project.

Ten matches were played between two bots, *Mynx* and *Orbb*, with a score limit of 100; the first bot to score 100 kills is the winner. Neither bot was using opponent modeling in the first ten matches. In these matches Mynx is the superior player, winning every game. Orbbs score fluctuates around 40.

Orbb was then given opponent modeling, but no existing model. Orbb would have to build the model while competing in the next ten matches. The same model was used and built upon in all ten matches for the second set. Orbbs performance on both sets of matches is summarised in Table 1 and Figure 4.

Table 1: Average score and Standard Deviation for Orbb.

	Average(n=10)	Std. Dev.
No modeling	37.7	6.75
Modeling	73.3	13.43

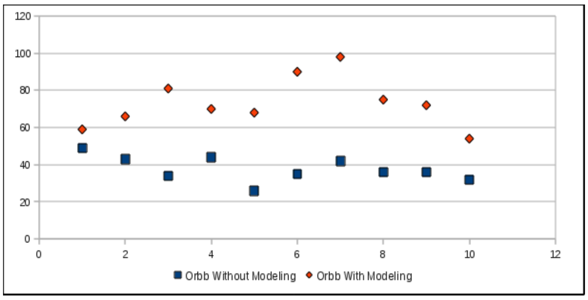

Figure 4: Orbbs scores with modeling (red diamonds) and without modelling (blue squares).

The scores for Orbb in the second set of matches are significantly higher (t=-7.5, p=0.000002) than in the first set. The only difference between the two sets of matches in that opponent modeling was enabled for Orbb in the second set. These results show that this opponent modeling technique does improve the bots performance. Although Mynx still wins every match, Orbb is much more competitive, and very nearly wins in match seven. It appears that after two matches the model is sufficiently accurate, although even in the first match Orbbs score is much better than in any of the matches from the first set.

There is a general upwards trend, with the highest score being in match seven. Importantly, in every match in the second set, Orbb gets a better score than any match from the first set.

Table 1 shows the average score and standard deviation from the two sets of matches. Mynx's score are not included as she scored 100 in every match. The average score when Orbb uses opponent modeling is twice that of when no modeling is used. The standard deviation is also much higher, showing that Orbb's scores are more variable when opponent modeling is used.

Further tests were performed with more than two competitors, the results were excluded due to space constraints. When more players compete in a match the effect of opponent modeling was less clear, but a small increase in performance was regularly observed. There are many potential reasons for this, such as more game time needed to build models (more models need to be built), and less time available to collect weapons and implement strategies.

In addition to recording scores, we have performed analysis on the generated models. This analysis showed that a bot will fairly reliably produce the same model for the same opponent, if many games are played, each game starting with an empty model. Analysis also showed that a bot will produce different models for different opponents.

CONCLUSION & FUTURE WORK

These results show that opponent modeling, based on techniques from student modeling, can provide useful models of opponents in Quake 3. Now that we have shown that useful information can be obtained from such a model, more research can be performed on how best to use such information. In particular, the success value calculation could be further refined. Future research will show us how to best utilise model information, with the potential to develop similar systems for other games.

This research has two aspects of significance. One is the scale of the video games industry; this is definitely big business. Additionally, perhaps the next generation of educational games could utilise similar opponent modeling techniques, to make the game component of an educational game more stimulating and challenging.

REFERENCES

Entertainment Software Association. 2010. http://www.theesa.com/facts/salesandgenre.asp

Darse Billings, Denis Papp, Jonathan Schaeffer, Duane Szafron, 1998. Opponent modeling in poker. Proceedings of the fifteenth national/tenth conference on Artificial Intelligence/Innovative applications of artificial intelligence, p. 493-499.

Hartley, D., Mitrovic, A. 2002. Supporting learning by opening the student model. In: S. Cerri, G. Gouarderes and F. Paraguacu (eds.) Proc. 6th Int. Conf on Intelligent Tutoring Systems ITS 2002, Biarritz, France, LCNS 2363, 453-462.

Kobsa, A. 2001. Generic User Modeling Systems, User Modeling and User-Adapted Interaction, v.11 n.1-2, p.49-63.

Mitrovic, A., Martin, B. 2007. Evaluating the Effect of Open Student Models on Self-Assessment. International Journal of Artificial Intelligence in Education, Special issue on Open Learner Modeling. 17(2), p.121-144.

Schadd, F., Bakkes, S., & Spronck, P. 2007. Opponent modeling in real-time strategy games. In 8th International Conference on Intelligent Games and Simulation (GAME-ON 2007), M. Roccetti, Ed., p. 61-68.

Togelius, J., De Nardi, R., & Lucas, S.M. 2006. Making racing fun through player modeling and track evolution. Proceedings of the SAB Workshop on Adaptive Approaches to Optimizing Player Satisfaction.

Zanetti, S, Rhalibi, A. El. 2004. “Machine Learning Techniques for First Person Shooter in Quake3”, International Conference on Advances in Computer Entertainment Technology.