PLAYING SMART – ANOTHER LOOK AT ARTIFICIAL INTELLIGENCE IN COMPUTER GAMES

Eike F Anderson
The National Centre For Computer Animation
Bournemouth University, Talbot Campus
Fern Barrow, Poole, Dorset BH12 5BB, UK
E-mail: eanderson@bournemouth.ac.uk

KEYWORDS
artificial intelligence, computer games, non-player characters, state of the industry.

ABSTRACT

With this document we will present an overview of artificial intelligence in general and artificial intelligence in the context of its use in modern computer games in particular. To this end we will firstly provide an introduction to the terminology of artificial intelligence, followed by a brief history of this field of computer science and finally we will discuss the impact which this science has had on the development of computer games, looking at how artificially intelligent behaviour can been achieved in games.

INTRODUCTION

Modern computer games usually employ 3D animated graphics (and recently also 3D sound effects) to give the impression of reality. This alone however does not necessarily make the experience of playing the game realistic, especially if the behaviour of computer controlled NPCs (non-player characters = virtual entities) in the game does not "feel right". The behaviour displayed by the NPCs is usually generated with the aid of "artificial intelligence" algorithms and techniques. Before one tries to explain the term "artificial intelligence" one first needs to ask the question "what is intelligence". This question has been asked for thousands of years, by science as well as by philosophy. The Greek philosopher Aristotle tried to identify the rules of "right thinking", logical reasoning, by establishing patterns by which a true precondition would always lead to a true goal state (Figure 1). The dictionary definition for intelligence is "the capacity for understanding; ability to perceive and comprehend meaning" [Collins 2000]. This is a valid description of what could be called human intelligence or human-level intelligence, but it does not really provide a usable answer for the original question, as this definition uses a number of terms that are hard to quantify without further definitions. More questions would have to be asked – "what is understanding, perception, comprehension, meaning?" Is this really what we are looking for in our quest for intelligence? In fact, from the computer games perspective in most cases we are far more concerned with behaviour – something visible - rather than with thinking when we look for intelligence in a computer controlled entity, i.e. an agent (*a decision-making entity*). Ethology – the science of behaviour in living entities – is based on observation of behaviour (an entities reaction to a situation/to an external influence) and is not concerned with the inner workings of the mind which may cause the behaviour. Although planning which requires knowledge, understanding, and to some degree also reasoning would lead to a behaviour, it cannot be observed and is an internal process. Early studies in experimental psychology tried to analyse and explain these internal processes but were not successful as the data gained was based on subjective descriptions of feeling and there was no reliable evidence. Opposed to this kind of research were the followers of behaviourism (*Watson and Thorndike*) which until the 1960s (when it was replaced by cognitivism) was the most recognised approach to the understanding of how learning works. Behaviourism concentrates on the analysis of "stimulus-response mechanisms" which had promising results with animals but less success with humans. This approach is well suited for the development of artificial behaviour in computer games. That is because what we would call "intelligent behaviour" could more accurately be described as

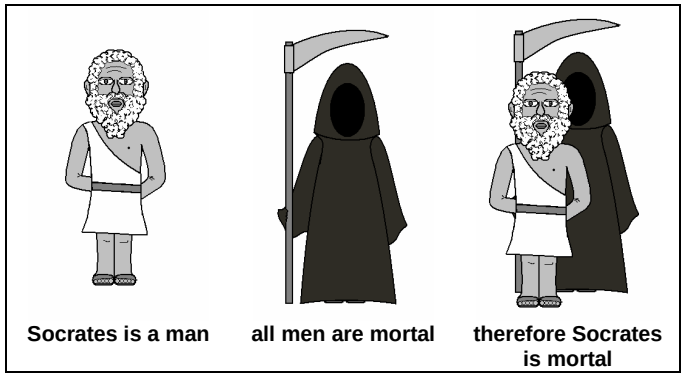

Figure 1 – deductive reasoning (Aristotle)

"behaviour that appears life-like" or more formally as "the display of an action which seems appropriate in the context of the current situation". This describes something which is perceived as intelligent and provides the illusion of intelligence without actually having to be intelligent.

Once the question of "intelligence" has been answered, one can turn to asking for the meaning of "artificial intelligence". This is defined by the dictionary as "the study of the modelling of human mental functions by computer programs". If one looks at the history of this science however, one will easily discover that this description is far less than accurate as AI is not necessarily confined to the simulation of methods that are biologically accurate or biologically possible [McCarthy 2004]. Another definition for artificial intelligence for instance is the ability "to solve problems that would require intelligence if solved by humans" [Johnson and Wiles 2001], or the ability of a system to adapt to its environment through learning. Whatever definition is used, the goal of artificial intelligence is always the same: to understand and to create intelligent entities.

An early measurement for the presence of a kind of human-like intelligence that would comply with these aims is the turing test [Turing 1950]. The turing test, also known as the imitation game, can be explained in simple terms. It requires a set-up of a closed room containing a human test person (the interrogator) at a computer terminal running a chat program, which has two connections. One connection is to a second human operated terminal in a different room and the second connection is to a computer running an intelligent program which pretends to be a human (chatterbot). The interrogator now has to decide which of the two chat partners is human and which one is the chatterbot. If the chatterbot manages to convince the interrogator that it is human, then it has passed the turing test. Every year there is an international competition (Loebner Prize [Loebner 1990]) using the turing test, which aims to find the most convincing and life-like chatterbot.

If a program manages to pass the turing test, i.e. manages to convince a human that it is human (and therefore intelligent) itself, that program can be considered somewhat intelligent. However, a number of people claim that the turing test alone would not be enough to allow judgement of the artificial intelligence of a program. John Searle's "Chinese Room argument" [Searle 1980] states that just by following a set of rules regarding a language one might be able to pass the turing test in a language one does not even understand (Chinese in the case of his argument) which would mean that the turing test itself could not be used to measure intelligence or understanding. In addition to that, during the turing test the interrogator knows that he is participating in a game, generating some form of bias in which the interrogator's imagination makes him perceive intelligence where there is none. There are a number of philosophers who question if AI can ever reach a level of intelligence that could be compared to that of a human. However, not everyone thinks of human-level intelligence as a goal for the development of AI. Each different interpretation of the term "artificial intelligence" is associated with different approaches to achieving AI. In turn, each of those approaches is more or less suitable for the different areas of AI research.

A BRIEF HISTORY OF ARTIFICIAL INTELLIGENCE

Artificial intelligence is almost as old as computer science itself, although it took some time for the field to be recognized as such. Research in artificial intelligence even existed a very long time before the term "artificial intelligence" was first used. The roots of AI can be found as far back as ancient Greece when philosophers (Socrates, Plato, Aristotle) discussed the way in which the human mind functions and how intelligent decisions are made. The study of what we now call AI is very much rooted in the study of philosophy and the quest for the understanding of mind and body:

"How can the scientific understanding of how the body and the brain works be combined with the thinking mind?"

Rene Descartes (1596-1650) suggested that an independent entity, the soul, would interact with the brain – a view which is nowadays hardly considered acceptable. This line of thinking is called the theory of "interactionist dualism". Opposed to this is the doctrine of materialism, going back to Wilhelm Leibniz (1646-1716) among others, which suggests that even a computer could have a mind of its own, provided that it would be given an appropriate program. Only the materialist position in philosophy allows for the existence of artificial intelligence, while "interactionist dualism" denies it. In fact, the study of logic, which is one of the foundations of AI, had been considered a purely philosophical problem until George Boole developed his mathematical concepts of symbolic logic in the mid-nineteenth century (published 1847). Only after the development of boolean algebra, logic was considered as a field of science, which ultimately made the development of computers and modern computing possible. Before the term "artificial intelligence" itself was used, scientist tried to develop intelligent machines, including mechanical machines for reasoning or for playing

year	Major developments/events in AI
1931	Gödel shows that some mathematical theorems that are known to be true cannot be proven by mathematical and logical means
1937	Church-Turing Thesis states that all problems that a human can solve can be broken down into an algorithm
1943	McCulloch & Pitts develop a model of artificial neurons
1948	Wiener publishes the book "Cybernetics" on information theory
1949	Hebb presents learning process for neural networks
1950	• Shannon develops early chess-playing program • Turing states the idea for the turing test • Asimov states the three laws of robotics
1951	Minsky & Edmonds build SNARC – first neural network computer
1956	Dartmouth Conference – term "Artificial Intelligence" used for 1st time by McCarthy
1958	• McCarthy develops LISP – first dedicated AI programming language • Simon makes a number of predictions of the future of AI: computer will prove mathematical theorem (*happened 1996*), computer will be chess champion (*happened 1997*)
1963	ANALOGY by Evans solves problems from human IQ tests
1965	Weizenbaum programs ELIZA (*early chatterbot*)
1967	DENDRAL rule-based system for analysing molecular structures created
1972	PROLOG AI language developed
1974	MYCIN developed by Shortliffe – first expert system (*medical diagnostics*)
1975	learning program Meta-Dendral makes first scientific discoveries (*chemistry*) by a machine
1983	Laird and Rosenbloom work on SOAR – an AI architecture which is now also applied to computer games (*SOAR Quakebot*)
1990	Koza develops genetic programming (*GP*) – programs that evolve
1996	• Kasparov beats Deep Blue (*world's most powerful chess computer*) • computer proves Robbins problem (*1st creative proof of a mathematical theorem by a machine*)
1997	• Deep Blue finally wins against Kasparov • first official Robo-Cup soccer match

Table 1 – a brief timeline displaying some of the major developments in AI

chess end-games. This however was independent from the development of computers and computing and did not result in notable successes.

artificially intelligence and computers

Alan Turing was probably the first researcher to recognise that electronic computers were better suited to the development of AI than dedicated machines. As the timeline (table 1) shows, the term "artificial intelligence" for this field of research was coined in 1956 when a number of researchers interested in the study of intelligence and neural networks took part in a workshop (Dartmouth Conference) organised by John McCarthy [McCarthy 1955]. In the beginning researchers were incredibly enthusiastic about artificial intelligence. Computers were something new and revolutionary. At first it was thought that all that computers could do was arithmetic calculations – all other uses were considered to be nothing more than science-fiction. As a result, people were amazed as soon as a computer program managed to do something that at least seemed to be an intelligent action. Once it became apparent that computers could not only handle numerical data, but also symbols for the representation of concepts, the use of computers for AI became a real possibility and by the mid-1950s most AI research was conducted using computers. Most of the first attempts in creating AI were mainly focussed on replicating the way that the human mind works – a difficult undertaking since it is still unknown how the human mind works. Since then there have been a number of different trends in AI research. Since many areas of AI research overlap it is hard to find clear distinctions between the different approaches that were used, but generally one can say that quite early on AI research split into two different camps [McCarthy 2004][Russell and Norvig 1995]:

One group of researchers used a biological approach, trying to imitate human physiology and psychology to create intelligent systems that think and act like humans. Their motivation was to determine which methods and techniques would explain real intelligence. This line of research to which many cognitive scientists and psychologists have contributed is concerned with the scientific goal of AI.

The other group consisting mainly of engineers and computer scientists aims for the engineering goal of AI and is concerned with creating intelligence by defining common-sense rules by which real-world problems can be solved through AI. Their research concentrates on the creation of systems that think and act rationally and which can act as tools that augment human thinking.

symbolic AI

The latter of these two approaches led to the rise of symbolic AI during the 1960s. Symbolic AI originates from research concerned with chess playing and the proof of mathematical theorems. It requires the programmer of the system to know what algorithms will be needed for solving problems, so that the programs can be built to contain the necessary

"knowledge". This is then used to perform some kind of planning or use various search strategies to find intelligent ways for finding an appropriate solution. Results of that research were a number of reasoning systems which were capable of solving logical problems using sets of rules (rule-based systems). Other developments in the 1960s were in the area of subsymbolic AI which uses systems that are modelled after neurons, i.e. neural networks. Here knowledge and planning for finding intelligent solutions is emergent, rather than built-in. This was mainly used for research on machine learning, natural language processing (recognition and understanding) and speech processing (understanding and reproduction).

knowledge based systems

The 1970s were the decade of knowledge-based systems. Before, researchers had tried to create mostly generic AI systems with general-purpose reasoning methods. This kind of approach was now called "weak AI". Scientists realised that the only solution was to increase the system's knowledge about the problems that they had to solve (domain knowledge) by combining rule-based systems with probabilistic reasoning techniques and huge domain-specific databases (knowledge-bases). Because these systems are optimised for finding solutions to their specific problems, their methods are called "strong AI". The result of research in this area were the first expert systems for solving problems in chemistry (structural analysis of molecules) and medicine (diagnostic tools). AI systems were now able to solve some real-world problems, opposed to the microworld problems (problems originating within a very small and confined domain) that had been the subject of earlier research. However, most of the predictions about AI that had been made about 20 years earlier did not happen and the early enthusiasm started to disappear. Towards the end of the 1970s it looked like there was no future for AI research. The great expectations that had been held from the late 1950s on were destroyed as research funds were cut because of a lack of useful results and many researchers started looking for other areas of computer science.

a second chance for AI

Fortunately though this decline of AI came to an end when a number of business ventures found ways to exploit AI. Parallel to the proliferation of personal computing, the discovery of commercial uses for some AI techniques – mainly expert systems - led to a second wave of enthusiasm in AI which in turn gave a boost to AI research. New advances in information technology and computing made it possible to research in previously untouched fields of AI like computer vision. This new wave also involved a revival of neural networks which had virtually disappeared in the early 1970s and a move towards evolutionary computing. Researchers had made some experiments in machine evolution, which is now called genetic algorithms, as far back as the late 1950s, but the AI renaissance of the 1980s as well as the rapid development of cheap and fast computer systems allowed more and more scientists to research this area of AI which eventually led to the establishment of genetic programming as a field of research. These techniques can find solutions in complex search spaces while requiring only little knowledge. Natural computation (evolutionary techniques, neural nets, complex and chaotic systems) was the new trend which became AI "state of the art" during the late1980s and early 1990s. And finally, scientists started researching AI for computer games.

ARTIFICIAL INTELLIGENCE IN COMPUTER GAMES

The AI found in most computer games is no AI (in the academic sense), but rather a mixture of techniques which are - although related to AI - mainly concerned with creating a believable illusion of intelligence. Some might argue, that in the case of AI in games the term AS (artificial stupidity) or "artificial instincts" might be a better description for the level of intelligence that is found there. As a rule of thumb one can say that creating a simple AI for games is easy as very little is required to fool the human brain. For example, the ghosts that oppose the human player in the classic arcade game PacMan make the player believe that his enemies are hunting him like intelligent pack-hunters. This perception of group-intelligence however is only an illusion. Each of the ghosts is given a slightly different variation of the same algorithm which is a very simple alternative selection of the direction with a weighted random component. As this weighting is different for each of the player's opponents it seemingly provides them with an individual personality. The result of this simple method is a personification of each of the ghosts by the player *(through subconscious projection*) as he will perceive the ghost's behaviour as that of an intelligent character. A complex AI on the other hand is actually quite invisible and will hardly be recognised. The concept of "less is more" can therefore be applied to AI in computer games.

The main requirement for creating the illusion of intelligence is perception management, i.e. the

organisation and evaluation of incoming data from the AI entity's environment. This is mostly acting upon sensor information but also includes communication between AI entities in environments with multiple NPCs. The decision cycle of those NPCs constantly executes three steps [van Lent et al 1999]:

1. perceive (accept information about the environment – sensor information)
2. think (evaluate perceived information & plan according actions)
3. act (execute the planned actions)

One could argue that this approach is far too simple and therefore might be unsuitable for creating an enjoyable gaming experience, however that is not so. In fact, video games do not need realistic NPCs that are as sophisticated and capable as the humans who play the game. Games are meant to be fun and the AI should never be too good and make a game impossible to win. Instead it should allow the player to win the game in interesting and challenging ways.

Then there is "Game AI" which is used in game theory, which is not at all the same thing as the AI used in video games. This kind of AI is mainly concerned with various approaches to tree search (used in chess playing and for other board games). The uses of this kind of AI for video games is very limited, mainly because the prohibitively huge amount of computation that it requires, but it has been used in the game Worms which is a turn-based game. For real-time games, which usually deal with large numbers of NPCs that need to act simultaneously, this is not possible.

historical stages of computer game AI

To understand, what the requirements of a typical AI in modern real-time computer games are, it is useful to look at the various stages in the history of computer game AI:

From the first games with computer controlled players and NPCs on, AI was used for creating believable adversaries/enemies to compete/fight against the human player. Depending on whether this was a tactical opponent in classical board-games or a monster in a role-playing or arcade game, the methods used for creating the AI were different, but their purpose was ultimately the same – to create a life-like opponent to provide the player with a challenging and fun experience.

As computers became more powerful and games grew bigger, incidentals (background character & creatures) were added to enrich the virtual game world without actively contributing to the plot of the game (an example would be neutral NPCs in role-playing games like Fallout or the pedestrians in the GTA series of games). People going about their own business in the background of the game action or secondary animation like flocks of birds in the virtual sky above, generate a sense of reality which aids in the player's immersion within the game world.

The development of the internet and networking technology for local area networks (LANs) soon led to the creation of games in which multiple players could engage over a network connection. In the first of these multi-player games, the players were opponents, competing or fighting against each other, but soon other ways of playing emerged, in which players co-operated and formed teams that would play against each other. Because of the overwhelming success of these team-based multi-player games, developers started looking for ways to generate the same sensation to single-player games. As a result, the latest addition to NPCs are artificial team-mates for the player (collaborative NPCs).

Whereas the quality of graphics and the number of polygons that could be displayed simultaneously on screen used to be the selling point for many video games, the realisation that graphical realism alone does not make a good computer game has led to the replacement of this development trend with a drive to improve the complexity and believability of the artificial characters that populate the virtual game worlds. If the behaviour of NPCs appears natural, they seem to be more life-like and real which is a crucial factor for the success and popular acceptance of a computer game. This has now become more important than ever.

problems of computer game AI

One of the greatest problems that faces games AI programmers is the requirement for the NPCs to work in real-time. This automatically excludes a number of AI techniques from being used in games, as it would be unacceptable for an NPC to spend minutes of game-time with decision making. The AI has to be made to work so that to the player it looks like the decisions are made as the NPC plays along. Another problem that is closely related to the real-time requirement for games AI is the fact that the AI has to share the computer's processing resources with the rest of the game which will include graphics, input processing sound processing and possibly even networking. In early games AI was given very small importance and was therefore allocated only little processor time. Only after the development of graphics accelerators in the mid-1990s when more and more elements of the graphics pipeline were redirected onto dedicated graphics hardware, AI got higher priority and with it additional resources. At first, CPU budgets for AI exploded and a number of games spent up to 30% of

their processor time doing AI calculations, but this has now levelled off at about 10% of CPU time.
While the exact range of problems that an artificial character within a computer game will need to solve depends on the virtual environment in which the character exists, the most common problems found in modern computer games to which the intelligent actions of NPCs are restricted to are:

- path finding / path planning
- decision making
- steering / motion control

While each of these problems can usually be solved with relatively simple methods, it is the combination and balancing between those methods that create the illusion of intelligence in games.

GAMES AI – STATE OF THE INDUSTRY

Just like games have come a long way over the past two and a half decades, so have the AI techniques that are employed within those games. The greatest changes in the use of AI in games however have involved the choice of AI to solve different problems rather than the choice of AI techniques. Some of the more proven and successful techniques have changed little over time and those techniques are almost always the first choice of the developers when they need to implement AI in their games. However over the past decade more and more novel ideas and methods for games AI have filtered into the game development process [Sweetser 2003].

rule based techniques

Rule-based techniques are the most commonly found AI methods used in computer games. They can be easily implemented and they provide a robust and reliable solution to a wide range of problems but are often used for decision making.

Finite State Machines

Finite state machines (FSMs) are the most commonly used type of AI used in games [Fu and Houlette 2004]. In an FSM the behaviour of the NPC is arranged in logical states – one state per possible behaviour – of which only one state is active at a time. A state is a boolean value which is either active or inactive – on or off. When the current behaviour needs to be changed to a different behaviour, for example a transition from a guarding stance to an attack on the closest opponent, the FSM will switch between the states. It is relatively simple to program a very stable FSM that may not be very sophisticated but that "will get the job done". The main drawback of FSMs is that they can become very complex and hard to maintain, while on the other hand the behaviour resulting from a too simple FSM can easily become predictable. To overcome this problem sometimes hierarchical FSMs are used. These are FSMs where each state can itself be an FSM. A recent example for FSMs in games are the game-bots in the Quake series of first-person shooters in which each NPC has a number of states which define the character's current behaviour.

Fuzzy State Machines

Fuzzy state machines (FuSMs) are a permutation of FSMs which uses fuzzy logic instead of boolean logic. As a result states in FuSMs are not limited to being on or off but they can hold an intermediate value. This means that at any one time more than one state may be active and to some degree be on and off. While this makes the construction of FuSMs slightly more complicated than the creation of an FSM the existence of simultaneously active states greatly reduces the predictability of the resulting behaviour. It also dramatically reduces the complexity of the state machine, as a wider range of different behaviours can be encoded with fewer states. FuSMs are a relatively new games AI technique that can be used in almost all of the areas in which FSMs are usually found. Recent games that have made use of FuSMs are "The Sims" and "Civilisation: Call to Power".

machine learning and machine intelligence

In recent years the use AI techniques that involve machine learning in games to achieve emergent behaviour has become more frequent and surprisingly effective [Graepel et al 2004]. The implementation of systems that "learn to play good" can be done without too much effort, but the downside is their unpredictability which makes them unsuitable for may games. The danger with learning algorithms is always that instead of making the AI seem smarter by behaving clever, it could in fact behave more stupid by learning to act in a less desirable manner. To prevent this from happening the learning is often done before the game itself is published and the commercial product often only uses the previously learned behaviour.

neural networks

Neural networks are a primitive simulations of animal brains in which the neurons are modelled using nodes that are interconnected which allows the network to learn and improve itself. Using a

neural network can enable games to adapt to the way that the player plays by updating itself during gameplay. Neural networks are used in strategy games but they have also been successfully implemented in adventure games or action games like "Heavy Gear" in which the robots controlled by the player use neural networks to improve its skills in line with the player's performance.

decision trees

Decision trees that grow as they learn new information are another machine learning method that is used in computer games. They are one of the most reliable and robust learning methods available and usually the preferred choice if a game AI requires to predict future outcomes or classify situations. When it is generated the decision tree will store situations and their outcomes within its nodes, allowing it to "remember" the best cause of action in case a similar situation is encountered in the future. A prominent example for the use of decision trees is "Black & White" which uses reinforcement-learning for the generation of the decision trees that are used to control the player's creature.

evolutionary techniques

Evolutionary techniques are the least often used machine intelligence methods used in computer games. In these techniques a basic initial set of problem solving strategies is usually evolved over time using a range of selection methods as well as random mutations, which are then evaluated until an optimal solution is found. While these solutions are usually very robust and reliable it can take a long time to reach that level of competence which makes evolutionary techniques unsuitable for real-time games. Nevertheless a number of games have made use of evolutionary techniques like genetic algorithms (GA) which played a major role in the game Creatures which employed a number of machine learning methods. In addition to GA, genetic programming (GP) has also been used for evolving agents for a number of games (Figure 2), including arcade games [Anderson 2002].

Other machine intelligence methods that have been used in computer games include artificial life techniques like flocking [Reynolds 1987] which is sometimes used for crowd simulations or to achieve squad-movements in strategy games.

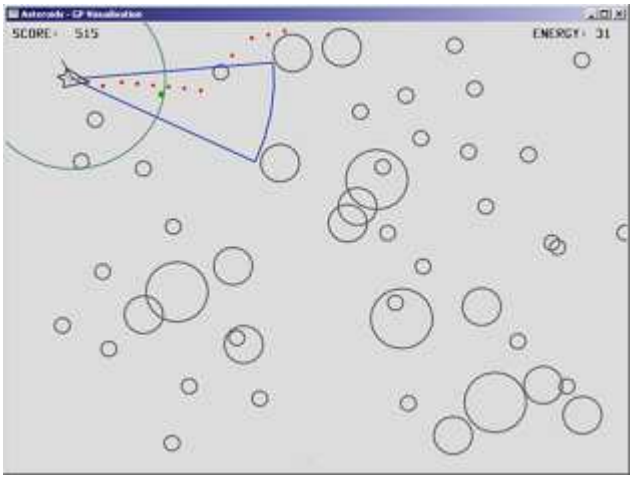

Figure 2 – Asteroids player evolved using Genetic Programming

extensible NPC intelligence

A recent trend in computer games is to make them extensible by allowing users to modify them to their needs. Some games even offer software interfaces, allowing parts of the games to be reprogrammed. One of the main areas in which games can be modified in this way is the game AI.

parameter tweaking

The most simple way for modifying AI behaviour is by modifying the parameters or rules that are used internally by the game AI. There are a number of games that allow to do this – some games even have graphical user interfaces to make this as simple as possible. Other games employ very simple scripts (see scripting systems below) to achieve this effect. During program runtime these scripts are usually only executed once at program startup while the application is initialising. In most cases this kind of script is used only to set internal program parameters to the values in the script which is why initialisation scripts are often nothing more but lists of values, sometimes using additional syntactic elements to make scripts easier to read and edit [Tapper 2003].

plug-in interfaces

Some games like Quake contain software interfaces that allow plug-ins to be written that can change the AI of NPCs in the game [Laird 2001]. Some games even have complex SDKs (software development kits) to simplify the modification of the game behaviour.

scripting systems

Many new games contain complex scripting systems that allow the game AI to be defined

(Figure 3) or extended. Scripting removes a large part of the – previously hard-coded – internal game logic from the game engine and transforms it into a game asset. Modification without the need for the game code to be recompiled becomes possible, a task that can be accomplished by a game designer alone. This enambles the introduction of "parallel development", which means that the programmers' time is freed up as they no longer need to concern themselves with design elements which designers can now manipulate themselves with scripts [Huebner 1997]. A number of games have built-in dedicated scripting languages, like Quake which includes a scripting language called QuakeC or Unreal which has a scripting system called UnrealScript. Other games use existing scripting systems that have been modified according to the game's requirements. An example for this is the scripting language Lua [Ierusalimschy et al 1996] which has been used in a number of games, including the game MDK2 from Bioware who also used scripting in their role-playing games "Baldur's Gate" and "Neverwinter Nights". A type of scripting languages which is domain specific to the creation of NPC intelligence is the behaviour definition language [Anderson 2004]. As their name suggests, behaviour definition languages are used to define the behaviour of virtual characters – often in the form of programs running on a virtual machine which interfaces with the character controls within the game engine.

knowledge based techniques

Knowledge based techniques are rarely used on their own when it comes to games AI, but they are often used as sub-systems of games AI within strategy games. This would include terrain analysis techniques such as influence mapping [Tozour 2001] which allow a strategic AI in a wargame to assess the current situation, to identify choke points for ambushes [Higgins 2002] or to position its troops on the virtual battlefield. Also search strategies are frequently used for path finding for NPCs in a wide range of games like individual units in strategy games.

other techniques

agents and animats

Although the term agent is used frequently, there is no single definition for it, but generally speaking an intelligent agent is "anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors" [Russell and Norvig 1995]. Using this

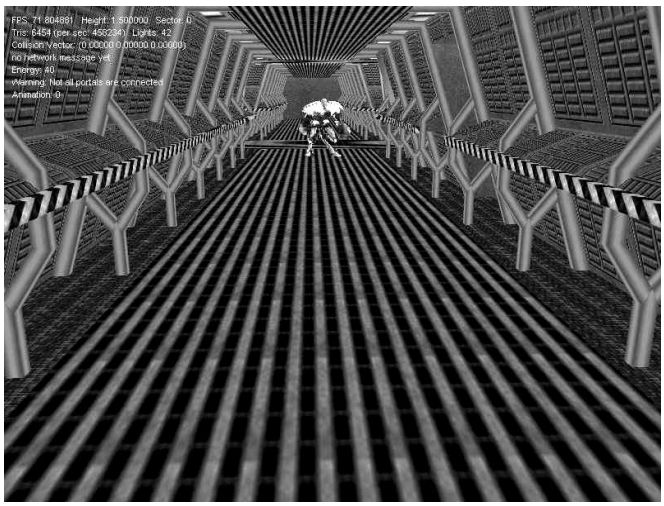

Figure 3 – script controlled NPC in "Pandora's Legacy" (using ZBL/0 [Zerbst et al 2003])

definition almost all NPCs in modern computer games can qualify as agents and it does not come as a surprise that agent-based techniques are used in computer games. Intelligent agents are decision-making entities that are usually constructed from a range of other AI methods. For example, an agent could integrate machine learning techniques with FSMs to be able to analyse the player's behaviour so it can anticipate the player's next move and make appropriate decisions and plans. The computer opponent AI in real-time strategy games is frequently an agent program.

Autonomous agents are agents that are self contained, i.e. agents that base their actions upon the information that they are able to gather themselves and their own knowledge. They do not have inputs that allow for external control but they are perceiving, "thinking" and acting by themselves. Autonomous agents with embodied systems, i.e. virtual beings that interact with their environments using their virtual bodies which take the place of abstract sensors or effectors are called animats [Champandard 2004].

annotated environments

A number of games now use annotated environments (Smart Terrain & Objects) to simplify the simulation of intelligent behaviour. If the environment of the NPC holds all the information necessary for the NPC to interact with it, the NPC can be less complex which not only benefits the development process but also makes the NPC's AI extensible. Objects that make up the virtual game world are "annotated" [Doyle 1999], i.e. the objects contain all of the information that an NPC will need to be able to use them, effectively making the environment smart. In the

game “The Sims” smart objects [Peters et al 2003] were used for behaviour selection. This means that most of the AI is not actually programmed into the Sims characters but into their environment. An object will broadcast information about itself to the Sims around it, including what animations to play for interaction between the Sims and the object [Forbus and Wright 2001].

Since many game AI techniques are repeatedly used in various games, there have been a number of attempts to create games AI SDKs to create generic solutions [Fairclough et al 2001]. However this kind of middleware has so far more followed than led the development of games AI. Innovations have appeared in mainstream games long before they found their way into middleware and as a result these SDKs have found little acceptance in the games industry [Skibak and Stahl 2002]. Although there is a growing market within the game development community, AI middleware solutions are still looked at with suspicion. Recent attempts to formalise the use of games AI, driven by the IGDA AI Standards Committee [Nareyek et al. 2004] however seem to be more successful. Once middleware based on these innovations is created, it will hopefully find acceptance from the industry.

THE FUTURE OF GAMES AI

With video games gaining acceptance and cultural significance as a form of art and popular culture, games are now more visible than ever. The life-like behaviour of the NPCs that populate the virtual game worlds will become increasingly important. Fortunately the fact that more and more classical AI methods are “spilling over” into the AI techniques used in computer games suggest that in the future the ability of NPCs to project the illusion of life-like behaviour will increase a lot. Here, special note should be given to the so-called animats [Champandard 2004]. These truly autonomous NPCs are very close to what might be regarded as the ideal NPC, as they present a believable virtual creature. It is therefore very likely that they will feature significantly in future computer games. This may lead to AI will becoming the new deciding factor for the success of games, very much like graphics used to be. The introduction of programmable GPUs (Graphical Processing Units) and therefore the advent of programmable shaders for real-time graphical applications in recent years [Lindholm et al 2001] has shown that with relatively little effort, great advances in the graphical quality of computer games can be achieved. The introduction of higher level programming languages for the creation of these shaders [Fernando and Kilgard 2003] has demonstrated that even better graphical quality for games is attainable by providing more powerful tools to the developers. It is our firm belief that to achieve further improvements in the quality of computer games, a similar approach will have to be taken for the creation of the artificially intelligent characters that populate the virtual worlds of computer games, i.e. the creation of a high-level programmable system. Some games AI researchers are convinced that at some point in the future dedicated hardware for games AI, AI accelerator cards – co-processors similar to the GPUs that are used for 3D graphics – will become available [Funge 1999]. Considering that the market for this kind of highly specialised, and therefore expensive hardware is comparatively small, as its main – and possibly only – use would be games, this future scenario might seem unlikely. The target audience for this kind of equipment would be hard-core game players, who make up only a fraction of the total number of computer game players, so the benefits that could be gained by developing a product for such a small market are few and far. However that does not mean that there won’t be any hardware solution for computer games AI. Using GPGPU (general purpose GPU) computation techniques, some AI calculations are already carried out outside the main processor and on GPUs instead [De Chiara 2004]. Furthermore, only recently – at this year’s Game Developer’s Conference, a prototype co-processor for physics and dynamics simulation [Hegde 2005] for use with games was introduced, proving that there are companies that invest time and resources in the research and development of dedicated hardware for games. While the development of different co-processors for different tasks is one possible solution, a different solution which could be introduced within a few years from now could be generic programmable computer chip which would allow for a dynamic modification of the way that chip would function. In that case this kind of co-processor would be adaptable to a number of different problems, including AI, physics and graphics. A program with extensive AI requirements could reprogram the chip on-the fly when the program is initialised, enabling it to then use it as if it was dedicated hardware. Technologically this is possible, but so far it is uncertain if there exists a market for this kind of hardware-software combination – only time will tell.

FINAL THOUGHTS

A large proportion of innovation and new developments in AI come from academic research and

estimates suggest that currently about 30-50% of all computer science research is conducted in artificial intelligence related topics. Modern video games are possibly the most visible application of AI techniques, generating a lot of public interest, which makes them an ideal subject for research. Therefore it does not come as a surprise that in recent years the AI research community has become aware of the possibility to use modern computer games as a platform for AI research. More and more researchers now use virtual game worlds as relatively complex environments in which they can field-test their theses in an inexpensive way. This trend is driven by the high extensibility of many game engines that allow the total modification of the behaviour of NPCs in the virtual game world of the game engines, for example the Quake engine which is used by a number of researchers [Laird 2001]. Other research has concentrated on AI in real-time strategy games, integrating general-purpose AI engines in different games of the genre [Atkin and Westbrook 2001]. The games industry can only benefit from these developments as there are few more ways left for improving games [Hawes 2002]. Graphics have now arrived at a stage where there is little room for additional developments, so now the games industry will have to find other methods to further their cause. The improvement of games AI is the answer. Unfortunately the relationship between academic research and the commercial game development community is still very much one-sided. The high competitiveness between game developers due to their commercial interests make them extremely protective of their intellectual property which results in a lack of co-operation with academic researchers (an important issue which has been discussed at various conferences). Although game developers welcome the development of techniques that they can profit from, they prefer not to share their own results with the research community. Academia on the other hand has a real problem with finding funding for video games related research, as there is a lack of evidence that their findings would actually be used by the game developers. While these issues will hopefully be resolved in the foreseeable future – the work of the IGDA AI Standards Committee [Nareyek et al. 2004] which attempts to develop standardised games AI interfaces is a step in the right direction – they still provide a huge barrier for academic acceptance of computer games as a valid and valuable platform for research and probably prevents major advances in the area of computer games AI.

ACKNOWLEDGEMENTS

An earlier version of this paper was presented at the zfxCON03 Conference on Game Development. In that respect I should mention Ian Millington whose talk on games AI at the NCCA in January 2003 prompted me to prepare this paper.
I would like to thank everyone at the NCCA for their support which made it possible for me to prepare this document, especially Professor Peter Comninos.

REFERENCES

[Anderson 2002] Anderson, E.F. (2002), Off-Line Evolution of Behaviour for Autonomous Agents in Real-Time Computer Games, Parallel Problem Solving from Nature - PPSN VII LNCS VOL. 2439, 689-699.

[Anderson 2004] Anderson, E.F. (2004), A NPC behaviour definition system for use by programmers and designers, Proceedings CGAIDE'2004, 203-207.

[Atkin and Westbrook 2001] Atkin, M.S. and Westbrook, D.L.(2001), Panel Discussion: Collaboration Between Academia and Industry: A Case Study, Working Notes of AAAI Spring Symposium, 5-6.

[Champandard 2004] Champandard, A.J. (2004). AI Game Development, New Riders

[Collins 2000] Collins English Dictionary: 21st Century Edition, Harper Collins Publishers

[De Chiara et al 2004] De Chiara, R. et al (2004), Massive simulation using GPU of a distributed behavioral model of a flock with obstacle avoidance, Proceedings of 9th International Fall Workshop Vision, Modelling and Visualization 2004

[Doyle 1999] Doyle, P. (1999), Virtual Intelligence from Artificial Reality: Building Stupid Agents in Smart Environments, AAAI '99 Spring Symposium on Artificial Intelligence and Computer Games.

[Fairclough et al 2001] Fairclough, C. et al (2001), Research Directions for AI in Computer Games, Technical Report TCD-CS-2001-29, Trinity College Dublin

[Fernando and Kilgard 2003] Fernando, R. and Kilgard, M.J. (2003), The Cg Tutorial, Addison Wesley

[Forbus and Wright 2001] Forbus, K.D. and Wright, W. (2001), Some notes on programming objects in The Sims™, Class Notes from Northwestern's Computer Game Design Course,
retrieved from http://qrg.northwestern.edu/papers/papers.htm

[Fu and Houlette 2004] Fu, D. and Houlette, R. (2004), The Ultimate Guide to FSMs in Games, AI Game Programming Wisdom 2, Charles River Media, pages 283-302

[Funge 1999] Funge, J.D. (1999), AI for Games and Animation: a Cognitive Modeling Approach, A K Peters

[Graepel et al 2004] Graepe, T. et al (2004), Learning to fight, Proceedings CGAIDE'2004, 193-200.

[Hawes 2002] Hawes, N. (2002), AI for Computer Games, retrieved from http://www.cs.bham.ac.uk/~jxb/AITA/

[Hegde 2005] Hegde, M. (2005), Physics, Gameplay and the Physics Processing Unit, White Paper, retrieved from http://www.ageia.com/

[Higgins 2002] Higgins, D. (2001), Terrain Analysis in an RTS – The Hidden Giant, In D. Treglia (Ed.), Game Programming Gems 3, Charles River Media

[Huebner 1997] Huebner, R. (1997). "Adding Languages to Game Engines". Game Developer, Vol. 4(1997): nr 9

[Ierusalimschy et al 1996] Ierusalimschy, R. et al (1996), Lua-an extensible Extension Language, In Software: Practice & Experience, Vol. 26(1996): nr 6, John Wiley & Sons

[Johnson and Wiles 2001] Johnson, D. and Wiles, J. (2001), Computer Games with Intelligence, Australian Journal of Intelligent Information Processing Systems, 7, 61-68.

[Laird 2001] Laird, J. E. (2001), It Knows What You Are Going To Do: Adding Anticipation to a Quakebot, Proceedings of the Agents-2001 International Conference on Autonomous Agents

[Lindholm et al 2001] Lindholm, e. et al (2001), A User-Programmable Vertex Engine, Proceedings of ACM SIGGRAPH 2001

[Loebner 1990] Loebner, H., Loebner Prize Home Page, retrieved from http://www.loebner.net/Prizef/loebner-prize.html

[McCarthy 1955] McCarthy, J. (1955), A Proposal for the Summer Research Project on Artificial Intelligence, retrieved from http://www-formal.stanford.edu/jmc/history/

[McCarthy 2004] McCarthy, J. (2004), What is Artificial Intelligence, retrieved from http://www-formal.stanford.edu/jmc/whatisai/whatisai.html

[Nareyek et al. 2004] Nareyek, A. et al (2004), The 2004 Report of the IGDA's Artificial Intelligence Interface Standards Committee, IGDA AIISC – http://www.igda.org

[Peters et al 2003] Peters, C. et al (2003), Smart Objects for Attentive Agents, Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision

[Reynolds 1987] Reynolds, C. (1987), Flocks, Herds and Schools: A Distributed Behavioural Model, Computer Graphics 21(4), 25-34

[Russell and Norvig 1995] Russell, S. and Norvig, P. (1995), Artificial Intelligence: A Modern Approach, Prentice-Hall

[Searle 1980] Searle, J. (1980), Minds, Brains, and Programs, The Behavioral and Brain Sciences 3, 417-457.

[Skibak and Stahl 2002] Skibak, S. and Stahl, M. (2002), KI – State of the Art,
retrieved from http://www.uni-ulm.de/~s_hdamme/

[Stern 1999] Stern, A. (1999), AI Beyond Computer Games, 1999 AAAI Symposium on Computer Games and Artificial Intelligence.

[Sweetser 2003] Sweetser, P. (2003), Current AI in Games: A Review, unpublished - under review for Australian Journal of Intelligent Information Processing Systems

[Tapper 2003] Tapper, P. (2003). Personality Parameters: Flexibly and Extensibly Providing a Variety of AI Opponents' Behaviors. Gamasutra - http://www.gamasutra.com

[Tozour 2001] Tozour, P. (2001), Influence Mapping, In M. Deloura (Ed.), Game Programming Gems 2, Charles River Media

[Turing 1950] Turing, A.M. (1950). Computing machinery and intelligence, Mind, 59, 433-460.

[van Lent et al 1999] van Lent, M. et al (1999), Intelligent Agents in Computer Games, Proceedings of the National Conference on Artificial Intelligence, 929-930.