Body Mind and Emotion

An Overview of Agent Implementation in Mainstream Computer Games

Stuart Slater, Kevan Buckley, Kamal Bechkoum

School of Computing and Information technology
Wolverhampton University
s.i.slater@wlv.ac.uk
k.a.buckley@wlv.ac.uk
k.bechkoum@wlv.ac.uk

Abstract

As both the growing demand for more immersive games continues and a widening of the buying audience increases, there is a growth in the quality and quantity of research and development in the field of "believable" computer game characters. This believability is necessary for the continued improvement in realism for computer games [Smith et al 2002] and is a key factor in an ongoing drive by developers who are creating greater immersive experiences for the gamer with each new generation of game. The purpose of this paper is to look at the elements that game developers use to give the agents in their games, human like qualities.

Introduction

Artificial Intelligence (AI) is a broad term, which for this paper applies to all work involved in making agents behave in a more challenging and believable way. This is often achieved through an agent's behaviour/actions appearing more human-like, thus creating the illusion that the gamer is playing against a real human player [Saltzman 2000].

During the early 1980's agent believability was not needed in the arcade style games being developed. Instead developers concentrated on implementing only enough simple behaviour to enhance game play. This behaviour was driven by simplistic algorithms many of which were implemented to allow simple path planning such as the ghost's movement in the *Pac Man* game or simple finite state machines (FSM's) for controlling agent actions. These were traditionally implemented in very small game environments such as single screen or tile based games and required almost no agent planning of goals but instead were more akin to scripted behaviour. Later games utilised more advanced AI techniques such as A* algorithms [Higgins 2002] which are used for agents path planning simply because virtual worlds have become considerably more complex. These techniques were implemented with limited processing resources and without regard for creating agents that can either exhibit human behaviour or look real. By the late 1990's games such as 'first person shooter' (FPS) style games were becoming more popular and as a consequence of consumer expectations in graphics, sound and AI the development became considerably more complex as the requirement for agents that could interact with each other and the player increased. This created a requirement for agents to be able to both look and behave more realistically within the very tight constraints of both developer technical skill and processor limitations. This was especially true when more realistic agents meant implementing new technologies such as realistic sensory systems, planning and more human-like agent physiology.

Agent Appearance

One of the key areas for improved agent believability has evolved from a rapid evolution of graphic processing technologies. This has resulted in higher polygon throughput and incorporation of new technologies such as vertex and pixel shaders, capable of giving surfaces a more "photo realistic look". Using pixel shaders, developers are getting closer to giving agents' skin a human like appearance without the need for complex programming. This reduction in complexity is due to the complementary programming methods and languages including NVIDIA's Cg [Fernando et al 2003] and Microsoft's High Level Shader Language (HLSL) [Fosner 2003]. These offer developers a more simplistic method of shader programming through a "scripting style" language that allows the graphic artists to implement the technologies as well as the programmers. The increase in polygon throughput has itself led to a greater dependence on packages such as *3D Studio Max 8* [Autodesk 2005a] and *MAYA 7* [Autodesk 2005b], which are used to create models and animations for cutting edge games such as *Doom 3* [Id 2004] and *Half Life 2* [Valve 2005]

Agent Physiology

With an evolving realism in the outward appearance of game characters there has been a growing amount of commercial development in game agent physiology such as inverse kinematics [Scarowicz 2004] and "ragdoll" simulation [Karma 2005]. Inverse kinematics include techniques to allow more realistic limb movements in agents, these have been coupled with graphical techniques to allow a smoother transition between actions by agents. "Ragdoll" is a term for the growing area of physics application to agent physiology such as allowing agents to fall and move in a realistic human way.

Top selling titles such as *Half Life 2* combine both the advances in graphics, physiology and physics modeling to infuse game agents with human like movements and appearances. Animals seemingly jump at players and agents fall realistically when shot with extreme weapons. This has provided the gamer with a new level of realism and a higher level of interaction expected with each new generation of game.

Agent Decision Making

With clear visual improvements of agents, the games industry has also experienced an increase in the allocation of resources and research being carried out within project teams. This is supported by evidence from the roundtable moderator's reports for the last six years of the game developer conferences [GDC 1999-2005] that show up to around 10% CPU usage in 1999 rising to between 15 and 50% for a majority of development teams in 2003. This increase parallels the rise of many of the FPS games and strategy games such as Microsoft's Age Of Empires and Westwood Studios'Command and Conquer Games.

An area of AI that has benefited from an increase in development time is agent path-planning, as almost all agent goal-based behaviour relies on agents being able to choose between paths that may lead to the same end goal. This area of AI allows agents to traverse virtual worlds using techniques such as A* and waypoint navigation systems and have been implemented in increasing complexity for many years in some format from *Pac Man* to *Doom 3*. These navigational systems have been implemented with both cognitive modeling and goal-based reasoning, giving the agents the ability to navigate around a virtual world with a purpose, such as the goal of looking for food, as can be demonstrated with the use of the *Renderware AI* tool [Renderware 2005] or the goal of collecting weapons to fight against a human opponent.

Recent AI research has focused in two developing areas, which can be used in conjunction with path planning techniques to create more believable characters. These are sensory input processing and increased agent autonomy through methods such as real-time agent planning [Orkin 2006].

Sensory Input

A growing area of interest in both academia and industry is in the field of agent sensory systems. This has led to methods of providing agents with both the ability to "see" and "hear" items in their environments, but a serious dilemma exists for its implementation, due to the differing goals of industry and academia. Industry chooses to implement only enough technology to provide an element of game play due to processing limitations and is thus only concerned with "emulation and not simulation" [Leonard 2003] academic research on the other hand tends to focus in-depth on areas of interest. Remembering that the vast majority of commercial AI implementations use "smoke and mirrors" techniques there is a definite need to be able to scale down academic research findings so they are applicable within the processing and game play constraints of projects.

Current implementations of agent sight and hearing provide compromises between tweaking virtual worlds so that in some games the scenery broadcasts to the agents as in the *Sims* games [Orkin 2002] and/or the sensory input is driven by either polled or interrupt driven perceptions in order to limit processing load [Kirby 2002]. Therefore if an agent needs to eat, then the agent will actively seek out food using some form of planning, which blends navigation and either pre-scripted behaviour or real-time decision making. Whilst navigating, polled perceptions will provide information about the environments to the agent's sensory system on a continuous basis, if the agent is stationary and a player gets within its sensory range then an interrupt drive perception will feed the information to the agent, this gives a much less processor intensive form of implementation. This research has led to a greater scope in game play such as the ability to sneak behind enemies in games such as *Thief* [Eidos 2002], which added sophisticated auditory and visual senses to agents in the game [Leonard 2003]. *Half Life 2* and *Thief* have agents that can "seemingly" see and hear human players as they wander around their virtual world. This has meant that for the first time agents can be made aware of human players based on similar constraints to those of real human hearing or sight, or at the very least the first steps in simulating these sensory systems. The usage of sensory systems in recent games such as *Far Cry* [UBI 2004] have been used to allow not only elements of stealth, but the ability to distract agents by throwing rocks near them or sneaking past them whilst their backs are turned.

Emerging areas of interest are inter-agent communications as seen in the "walkie talkies" in *Far Cry* or cries for help from agents in *World of Warcraft* [Blizzard 2005] allowing agents to get support from other agents. Similar techniques have been used in *Call of Duty* [Activision 2003], when bullets are fired near enemy agents this causes a change of behaviour that allows the agent to dive for cover allowing human controlled game characters to advance forward to capture areas or attack.

With the implementation of agent sensory systems has come a greater use of data storage for the sensory input for agents. These storage systems are linked to agent decision making through back-end management systems and have created a new area of research in fast data access and storage mechanisms such as spatial data structures [Reynolds 2000]. These new systems and technologies are crucial to real time considerations in modern computer games.

Implementations of senses such as touch and smell though not presently adding much to the game play are beginning to appear as in the use of smell for agents in *Half Life 2* which allows agents in the game to "sense" a player through an artificial olfactory system.

Decision Making

As mentioned previously, decision making has focused on mainly goal based reasoning and path planning. This allows agents, such as those seen in *Unreal Tournament 2004* [Atari 2004] to work co-operatively or adversarial with or against the player. Agents are given goals, such as "kill gamer", supplemented with inter-agent communication frameworks that provides co-operative team play with other agents or the gamer.

Agent decision making traditionally focuses on the use of finite state machines (FSM) which are based on deterministic programming i.e. if-then production rules that are commonly used for controlling agents' behaviour in games [Carlisle 2002].

An example is

"IF player in view AND gun loaded then FIRE"

The limitations of FSM's for computer games is that they rely on conditional *true* or *false* variables resulting in actions that are predictable and could be perceived as being limited. To enhance FSM's the Boolean variables can be replaced with fuzzy variables that have a much larger range of values. This allows for a more complex set of actions i.e.

"IF player in sensory range AND gun has enough bullets THEN fire weapon ELSE look for ammunition"

which would mean that an agent might not fire the gun if they do not have enough bullets to kill the player, thus they might go instead to retrieve ammunition. This allows developers to expand agent goal options by using linguistic rules to define behaviour in conjunction with tiered goal systems involving primary and sub goals to allow a breaking down of complex tasks [Waverren 2001]. This tiered approach to tasks enhances the behaviour options of agents to offer them choices depending on the primary task set. A difficulty with giving agents sub tasks is that if the agent's option path is highly varied, then a situation may arise where conflicting goals will need to be carefully managed to avoid a gridlocked agent response.

Making more adversarial players is not the only way that AI has been improved. Games such as *Creatures* [CyberLife 1996] blend techniques such as neural networks, and aspects of biochemistry to create agents with unique behaviours that can interact and mutate into new agents with unique behaviours [Stern 1999].

Many developers have serious concerns about agents that could try and move beyond the constraints of the game architecture if they exhibited unpredictable behaviour. Therefore developers rely on a more scripted-behaviour approach to avoid any kind of adverse emergent behaviour. Another concern of games developers is that there are serious concerns with AI adversely affecting game play due to both the processor time required and the speed of the response. This may be due to the software waiting for the next agent action or simply overly complex AI that interrupts the player's immersion. In some cases this has led to developer's reducing agent capabilities such as in *Ultima Online* [Stern 1999]. Techniques to decrease system load and responses times for complex AI include AI handlers running as separate threads and/or using level of detail AI architectures dependant on situation and processing availability [Woodcock 2003].

Agent Emotions

Commercial implementations of intelligent agents continue to provide a reasonable challenge to the gamer, but most games still lack any implementation of agent emotion's and thus agents can appear devoid of emotion, which could appear to the gamer that the agent is lifeless and shallow. Agents in commercial games currently cannot get annoyed by failed goals, show satisfaction for a kill against a tough enemy, or run away in fear.

Some developer's script facial animations to appear on agents' faces at intervals, to give the gamer the illusion of agent emotion i.e. when a player is killed by an agent the facial emotions might show a smile. The game *Halo* [Bungie 2003] featured simple finite state machines for the emotions surprise, anger and awe and upon activation of a particular state the agent would flee in terror, go berserk and attack, or retreat into a defensive position, this was complemented with suitable facial animations.

Conclusions

Recent games featuring large-scale environments such as *World of Warcraft* have virtual worlds filled with hundreds of agents and millions of gamers across multiple servers. The agents in these worlds need to appear to behave as realistically as possible to provide a satisfactory degree of immersion to the gamer and therefore are programmed with some awareness of their environments and a level of autonomy. These agents seemingly make decisions on goals given to them by the developer and thus appear to act in a similar way to a real player. Some games even implement "planning models" such as in F.E.A.R [Orkin 2006] to allow agents to choose options in real-time, based on their goals.

Currently agents can be equipped with sophisticated sensory systems such as sight and sound (hearing) and are able to traverse virtual worlds. They can plan, choose between options to complete goals and thus create the illusion of autonomy. The impact on the developer for implementing these new technologies is the increased processing and resource implications for this new generation of games. This has meant that developers have been forced to look at ways of optimising games in such a way as to maintain a consistent game experience for the user whilst creating ever evolving virtual agents for the gamer to interact with. In part this has been achieved through a greater use of more intelligent environments, some of which can broadcast information about useful items in "view" to the agents, such as those found in the *Sims* games.

This evolving of both environmental design and AI architectures is necessary for the continued development of more sophisticated behavioural models for agents, in order that these agents are able to affect and interact with the player and their environments [Todd, P et al 1997]. These optimisation techniques along with a growth in visual improvements of agents is currently leading developers to look at other areas of improving agent believability. One area of interest is in the use of human emotion modeling.

If agents can be developed with more sophisticated cognitive and goal architectures then it is feasible that they could exhibit a level of emotions such as the universal emotions [Damasio 1999], fear, happiness, sadness, anger, surprise and disgust. These emotions could be linked to the agents' goal structure [Johnson-Laird 1989]:

- Happiness that a goal has been achieved or progress made.
- Sadness if a goal not completed or loss of a goal.
- Anger if goal challenged or failed because of an external entity
- Disgust if a goal is violated.
- Anxiety if a goal or the goal of self-preservation is threatened by a future event.

Plus

- Fear if an immediate goal is in danger or the goal of self- preservation is immediately threatened.
- Surprise – unexpected successful completion of a goal.

This may initially be limited to both a blend of facial [Ekman 2004], physical i.e. posture, voice and emotional state storage, but could subsequently be expanded to actually affect primary goals and the choosing of sub goals that might best satisfy any emotional needs. This could be the agent experiencing rage at a player killing his comrade and then choosing to punish the player by killing his teammate rather than the player. This of course would open up the possibility that an agent may become unpredictable, but would certainly offer the gamer a much more human like opponent.

Looking ahead there are several hurdles to overcome in implementing agent emotions amongst which are:

1. Constraining agents to virtual world architectures thus their cognitive model and domain knowledge would need to be carefully developed which might require extensive resources and time.
2. Limitations due to the current state of the art in agent decision making that may not be advanced enough for the realization of complex emotional architectures.
3. A possible side effect of adding 'human like' emotions to agents is that this unpredictability could be hard to replicate for testing and potentially memory and processor intensive.
4. The defining of a suitable emotion architecture that can adequately model human emotion within the constraints of developer resources and current state of the art in AI games technologies.

Web Reference

[Activision 2003] Call of Duty, 2003. Activision. www.callofduty.com

[Atari 2004] Unreal Tournament 2004. Atari. www.unrealtournament.com

[Autodesk 2005a] 3D Studio Max 8. www.discreet.com

[Autodesk 2005b] Maya 7. Autodesk formerly Alias. www.alias.com

[Blizzard 2005] World of Warcraft, 2005. Blizzard. www.worldofwarcraft.com

[Bungie 2003] Halo . Bungie Studios. http://www.microsoft.com/games/halo/default.asp

[CyberLife 1996] Creatures, CyberLife Technologies,1996.

[Eidos 2002] Thief. Eidos Interactive 2002.http://www.eidos.co.uk/games/info.html?gmid=34

[GDC 2001-2005] archive section of the Games Developers conferences 2001-2005. www.gdconf.com

[Id 2004] Doom 3, Id Software, 2004. www.doom3.com

[Karma 2005] Karma Physics Engine http://udn.epicgames.com/Two/IntroToKarma

[Renderware 2005] Renderware A.I game engine http://www.renderware.com/

[UBI 2004] Far Cry 2004. UBI soft. www.ubisoft.com

[Valve 2005] Half Life 2, Valve Software, 2005. www.valvesoftware.com

Reference

Carlisle, P. 2002.*Designing a GUI Tool to Aid in the Development of Finite State Machines* (pg 71-77), AI Game Programming Wisdom: Charles River Media.

Damasio, A. 1999. *The Feeling of What Happens.*. Published by Vintage.

Ekman,P. 2004. *Emotions Revealed.* Published in the UK by Phoenix

Fernando, R; Kilgard, M.J. 2003. *The Cg Tutorial*. Addisson Wesley.

Fosner,R. 2003. *Real-Time Shader Programming*. Morgan Kaufmann Publishers.

Funge, J.D. 1999. *AI for Games & Animation – A Cognitive Modelling Approach*. AK PETERS LTD. Natick, Massachusetts.

Higgins,D. 2002.*How to achieve Lightning Fast A**. AI Game programming Wisdom. Charles River Media.
Johnson-Laird, P.N. & Oatley,K. (1989). The language of emotions: An analysis of a semantic field. Cognition & Emotion, 3, 81-123.
Kirby, N. 2002 *Moderator's report – AI Roundtables*. Game Developers Conference 2002. San Jose, California.
Leonard, T .2003. *Building AI Sensory Systems – Lessons from Thief and Half-Life*. Game Developers Conference 2003. San Jose, California.
Orkin, J. 2002. *Tips from the Trenches*, AI Game programming Wisdom. Charles River Media.
Orkin, J. 2006. *Three States and a Plan: The A.I. of Fear.* Game Developer's Conference 2006. San Jose, California.
Reynolds, C. 2000. *Interaction with Groups of Autonomous Characters*, in the proceedings of the Game Developer Conference 2000, San Francisco, California.
Saltzman, M.1999. Secrets of the Sages. Chapter 9. Macmilan Digital Publishing USA.
Scarowicz,A. 2004. *Artificial Intelligence for Animated Autonomous Agents*. PhD Thesis Kingston University.
Smith, P; Hjelstrom, G. 2002. *Polygon soup for the programmer's soul: 3D Pathfinding.* Game Developers Conference 2002. San Jose. California.
Stern, A .1999. *AI Beyond Computer Games.* (www.lggwg.com/wolff/aicg99/stern.html
Todd, P.;Geoffrey F. 1997. How Cognition shapes cognitive evolution. IEEE EXPERT: Intelligent Systems and their applications, 12(4), 7-9.
Waverren, JMP.2001. *The Quake 3 Arena Bot.* University of Technology, Delft Facults, ITS
Woodcock, S. 2003. *AI Roundtables Moderators' Report.* Game Developers Conference 2003. San Jose, California.

Biography

Stuart Slater is a senior lecturer in IT & Computing at Wolverhampton University. His current research involves the development of an "agent emotion architecture" for use with commercial AI solutions.