Concise paper

Games Programming AI

James Jordaan
Eastern Institute of Technology
Hawks Bay, New Zealand
jjordaan@eit.ac.nz

ABSTRACT

Artificial intelligence (AI) programming has a wide variety of aspects of which only the decision system is usually focused on. This paper puts the decision system into perspective with the rest of the artificial intelligence system and then proceeds to categorize different types of artificial intelligence by function or efficacy rather than method of implementation. The focus of this paper is on games programming artificial intelligence where two types of artificial intelligence are defined followed by placing the entire artificial intelligence system into perspective. Then a taxonomy model is presented upon which artificial intelligence can be categorized. This taxonomy model also forms a foundation upon which artificial intelligence should be implemented in the development of a game and upon which all previous, present and future artificial intelligence can be gauged.

1. INTRODUCTION

Currently most artificial intelligence (AI) programming focuses on just one aspect of the overall AI process and that is the decision system. There exists two main streams of AI: Firstly Successful AI which is the efficacy of the AI system to produce human-like responses which is based on the original philosophy of Alan Turing's (1950) work. Secondly Industrial AI which tries to provide the best solution for a given situation every time. This is usually argued by those that do not like Turing's work (Whitby, 1997, French & Michie 1996).

2. TWO TYPES OF ARTIFICIAL INTELLIGENCES

Alan Turing (1950) introduced what is now being called the Turing Test which, in essence, states that if a person can not determine whether it is artificial intelligence that they are communicating with or another person then the artificial intelligence is successful. The Turing Test does not imply any grade of intelligence nor does it imply any method used in the implementation of that artificial intelligence. Only the effect of the artificial intelligence on the person interacting with it is of importance here.

Many researchers such as Whitby (1996), Watt (1996) and Collins (1997) argue against the Turing Test saying that it is a very narrow minded way of looking at artificial intelligence and that Alan Turing's (1950) philosophy on artificial intelligence has done the field of study considerable damage. As a result of this there are now many acceptable forms of artificial intelligence that have nothing to do with people. A good example here would be Fuzzy Logic (Dewey, 2001) which has been used successfully in industrial embedded applications such as motor speed controllers. These types of AI have been referred to by Copeland (2000) as Applied AI (or advanced information-processing or "smart" systems).

Copeland (2000) also defines two other types of artificial intelligence: Strong AI and cognitive simulation. In Strong AI, he describes an AI that will mimic the entire intelligence processes that a human being is capable of. However he (Copeland, 2000) does admit that in five decades scientists have still not been able to produce anything that is as intelligent as a worm (*Caenorhabditis elegans*) which only has 302 neurons (White *et al*. 1986), and which has had its entire neural network system perfectly mapped out.

In cognitive simulation, Copeland (2000) describes a decision system which does pattern matching in the attempted duplication of the process

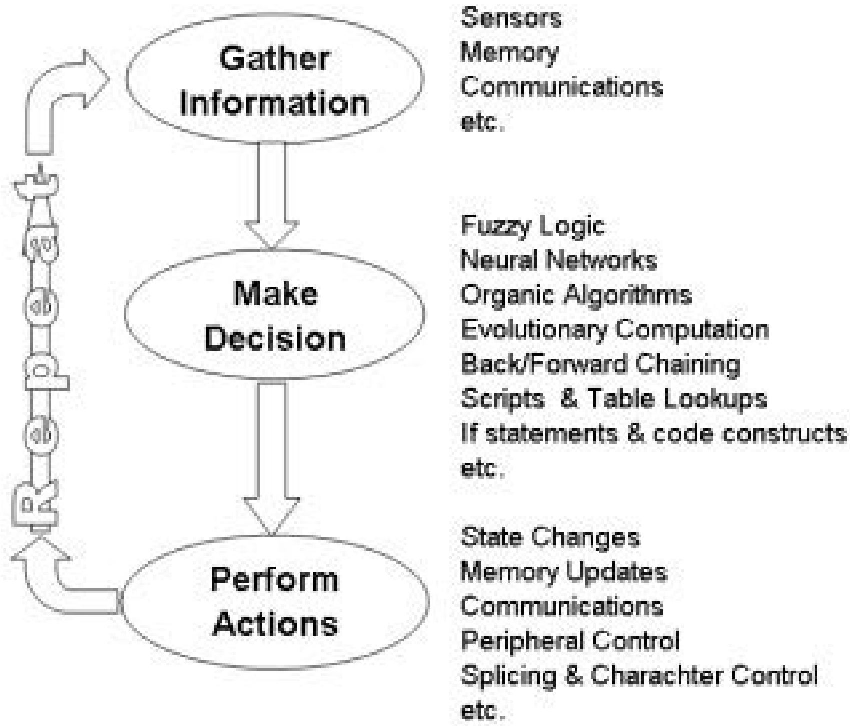

Figure 1: AI cycle

that biological neural networks use. However this is not a type of artificial intelligence in itself but merely an implementation concept used for the decision system that forms only a part of an artificial intelligence system as a whole.

3. SUCCESSFUL AND INDUSTRIAL AI

Turing (1950) introduced the Turing Test and although researches such as Whitby (1997) argue against its validity as artificial intelligence, it has become clear that Turing was describing a type of artificial intelligence which serves only to mimic the responses of a person. This is then called 'uccessful AI. Whereas other AI concepts which are being used to fly planes, control conveyor belts and authenticate users in human computer interface devices all have an industrial, "smart systems" (Copeland, 2000) application to their purpose. This is why they can rather be referred to as 'Industrial AI'.

Successful AI in its purpose to mimic people, makes mistakes, has individual characteristics, learns new tricks, is unreliable and amongst other things, is unpredictable. Industrial AI on the other hand has the purpose of making the best, optimal choice possible within its environment, every time, with no exception.

Artificial intelligence can be seen as the decision made by an artificial agent based on the agent's interpretation of its environment. As such the method used to implement the agent does not serve to classify the agent as an artificial intelligence or not. Therefore artificial intelligence can be as little as a single if statement in the right place, to advance neural network implementations (Negnevitsky, 2002, Kurzweil, 2002 and Wok, 2001).

4. COMPOSITION OF AN AI SYSTEM

An AI system contains more than just the decision subsystem. Figure 1 shows that there is an iterative process of gather information, decide what to do and act on the decision. The action however is not a physical action that is being referred to but an internal restructuring of the system such as memory, state flags or charge weightings, which depend on the type of AI implementation being used. The decision system may be the most difficult part of the system to implement and in some case forms the largest part of the system but the decision system has a resulting action based on the environment. This action and

	Level	Example	Key Definition
Low Level AI	Level 1	Explosion Laser Bolt	Manipulation of an object as a result of internal properties
	Level 2	Guided Missile Terrain Following	Manipulation of an object as a result of external environment
	Level 3	Laser Turret Cluster Bomb	This object will spawn new objects in the virtual world.
	Level 4	Tank Aircraft	Switches through AI states and can be cascaded: sub states, sub-sub ?
	Level 5	Tower, Airfield Platoon Sargent	Has the ability to switch states in other objects. (not properties)
High L. AI	Level 6	Head Quarters	General: Action/Reaction for all existing objects under its command
	Level 7	Capitol City Emperor	Govenor: Planner/Strategist all construction, diplomacy, technology

Figure 2: Taxonomy of AI

environment are two important subsystems that are usually ignored in discussions about artificial intelligence programming.

In games programming (specifically strategy games like Warcraft and Starcraft) most artificial intelligence is done at the lower level and serves a secondary role as the "eyes, ears, arms and legs" of the high level AI (High level & Low Level explained later in the paper). In this way the low level AI operations function as the sensor subsystem that feeds the memory of the high level AI. Once the high level AI decision subsystem has performed the required tasks, the memory and states are set in the high level AI (action) which become directives that are then fed into the low level AI operations.

Using this approach one can say that the AI collective exists as a result of interactions of all the objects that exist in the virtual world of a computer game. Low level AI objects are used to feed an AI system with information where the high level AI objects then make decision so that low level AI objects can act upon those decisions. Another observation taken from games programming is that there are different levels of AI objects which in turn can execute different levels of AI functions (or methods).

5. THE NEED FOR SUCCESSFUL AI

Although most types of AI implementations such as discrete constructs, genetic/organic algorithms, neural nets and memory driven methods have been tried in games at some point or another, developers have realized that at least the high level AI needs to be unpredictable in order to be well received by the players. This implies that regardless of the method of implementation the type of AI that run at the high level needs to be successful AI. Coincidently the more successful the AI, the more successful the game.

A taxonomy of AI has been compiled (figure 2) as a result of categorizing a wide variety of games for the functions/methods that are generally used in games programming to produce the overall AI in those games. All game genres are included in this taxonomy such as Tactical, Strategy, First Person Shooter (FPS), Arcade, Role Playing, Multi User Domain (MUD), Massive Multiplayer Online (MMO) and many other variations as well. This taxonomy fits more than 500 games that I have compared and also fits Industrial AI applications as well.

6. THE AI TAXONOMY EXPLAINED

Figure 2 has a concise summary of the AI taxonomy which shows firstly that there are 7 levels of AI defined where level 1 is the lowest and level 7 is the highest. The first five levels together serve as the lower level AI's and the last two levels serve as the high level AI's. The level of the AI is categorized according to the function (method) that the object performs. In the case of objects that operate on levels 4 and above, the level of the object is classified according to the highest function that it can perform. Also virtual world objects can be classified according to the taxonomy as well as support procedures (functions/methods) which those objects may make use of. (A virtual world object refers to the units in a game; such as tanks, aircraft, projectiles, clouds, explosions etc., and although it shows a remarkable similarity in behavior to programming language objects they are not always the same thing.)

6.1 Level 1

The manipulation of an object as a result of its own internal properties.

Example 1: Explosion animation.

The explosion procedure/method only changes the displayed shape of the object until it fades away.

Example 2: Laser Bolt.

The laser position X/Y(/Z) is updated every time until it has reached its max range.

6.2 Level 2

The manipulation of an object as a result of internal properties and external environments.

Example 1: Guided Missile.

The missile moves through (X/Y/Z) and will eventually expire, but it also aims itself towards a target, which is an external environment.

Example 2: Path finding

Path finding is a two stage system that finds a route (stage 1) and then follows that route. This system operates on level 2.

6.3 Level 3

The manipulation of an object as a result of internal properties and external environments. Objects can also spawn (create) other objects in the virtual world.

Example: Laser Turret.

The turret has to detect an enemy presence (level 2), aim (elevation, bearing - level 2), and when the gun is ready to fire it creates (level 3) the laser projectile which is then a level 1 AI object.

6.4 Level 4

The manipulation of an object as a result of internal properties and external environments. The object can spawn (create) other objects in the virtual world. Object can switch between multiple states within itself. Each of these states could be operating on levels 1-4 so that a level 4 object can be cascaded: sub states, sub-sub states etc.

Example 1: Tank.

The tank may have several states such as go to X/Y location, patrol route, attack target, hide or return to head quarters. The go to X/Y would be a level 2 only but the attack target is an example of a cascaded sub-state because attack can be broken down into approach target and aim/fire.

Example 2: Aircraft.

The aircraft state may include takeoff, land, go to way-point etc. The land state can be broken down further into holding pattern (level 1), final approach (level 2), touchdown, taxi etc.

6.5 Level 5

The manipulation of an object as a result of internal properties and external environment. Objects can also spawn other objects in the virtual world. Object can switch through multiple states (cascaded) but can also switch states (not properties) of other AI objects.

Example 1: Control Tower (Airfield).

The control tower holds separate states of each of its runways/hangars. A runway is open or closed. It "grants permission" for other objects (aircraft), by changing their state to allow a final approach on runway X.

Example 2: Platoon Sergeant.

An AI object, platoon sergeant, may order all those under his command to fall in formation, all the soldiers (level 4 AI's) will then be given go to X/Y - a level 2 action, or they could be given a command to attack a target which will switch the all the platoon's states to attack target.

6.6 Level 6

This is a high level AI, often called the General or Action-Reaction. The main responsibility of the general is that it commands all active units (objects) under its command. The term units, refers to mobile objects that currently exit. The General also divides units up as defense, offence and mobile. It must attempt to cover the goals and priorities as set by the level 7 AI and does a threat analysis based on data available to it. The general should be an object in the virtual world and can be destroyed like any other object. There can be more than one general per faction or side. It does not directly spawn any objects and is not directly cascade-able. There can be different types of generals. This AI object is a good candidate for the use of neural net, fuzzy logic and organic/genetic algorithms and should be of a successful AI type.

Example: HQ or Head Quarters.

This object will consider the plight of all units based under it, and weigh the local threat against it. If the ratio is favorable it could check will other head quarters that may be requesting troops and hence will send some of its troops over to the other head quarter. It may also detect a target of opportunity and send out a group of units to attack or capture it. If its territory is under attack it may send out a request for additional reinforcements from other head quarters or to the Governor AI.

6.7 Level 7

This is a high level AI (object). The governor AI also called the Planner/Strategist, is the AI that handles the prioritizing the acquisition of resources and allocation of resources. It is responsible for construction; roads, bridges, factories and units. It creates trade routes, handles technology, diplomacy spies etc. This AI object is a good candidate for the use of neural net, fuzzy logic and organic/genetic algorithms and should be of a successful AI type.

Example: The "commander" in Total Annihilations or the "palace" of civilizations, However in most cases this AI does not exist in the virtual world.

7. IMPLEMENTATION

This taxonomy has been used in games programming classes as the fundamental guide to implementing AI. Students are required to develop one AI level each week starting from level 1, until they have reached the required level for that paper. The functions of the lower levels have a tendency to support the functions required by higher levels for example: A laser turret (level 3) needs to aim and shoot this requires both a guidance (aim) and a spawn object (shoot) procedure/method. Once this is done the level 4 is easier to make since both guidance and spawn are already available, and tested. The guidance alone is needed in the Goto X/Y, Way Point, Attack, Land, Take-off (and many more) sub states. By following this taxonomy one level at a time students were able to make better sense of how AI worked as a whole, and were able to produced higher level AI objects than before the taxonomy was used.

8. CONCLUSION

In conclusion, a distinction between successful AI and industrial AI is provided as well as a simplified and intuitive taxonomy model which all past, current and future AI developments can be gauged, specifically in games programming. The taxonomy provides a controlled, easily assessed, best practice, way of implementing AI in games programming.

REFERENCES

Collins, H.M, 1997, "The Editing Test for the Deep Problem of AI.", Psycology: Turing Test, http://www.cogsci.ecs.soton.ac.uk/cgi/psyc/newpsy?8.1

Copeland, B.J, 2000, "What is Artificial Intelligence?", May 2000, http://www.alanturing.net/turing_archive/pages/Reference%20Articles/what_is_AI/What%20is%20AI02.html

Dewey, D.S, 2001, "Fuzzy Logic", http://www.omega.com/techref/fuzzylogic.html

French, R and Michie, D, 1996, "Essays in Honour of Alan Turing", O.U.P in Clark, A. and Millican, P. (eds)

Kurzweil, R, 2002, "A Wager on the Turing Test: Why I Think I Will Win", Published April 9, 2002 on KurzweilAI.net http://www.kurzweilai.net/meme/frame.html?main=/articles/art0374.html

Negnevitsky, M, 2002, "Artificial Intelligence: A Guide to Intelligent Systems", Addison Wesley, Harlow, England, 2002. Chapter 8

Turing, A.M, 1950, "Computing machinery and intelligence." Mind, 59, 433-460. http://www.loebner.net/Prizef/TuringArticle.html

Watt, S. 1996, "Naive Psychology and the Inverted Turing Test.", Psycology: Turing Test, http://www.cogsci.ecs.soton.ac.uk/cgi/psyc/newpsy?7.14

Whitby, B. 1997. "Why The Turing Test is AI's Biggest Blind Alley", http://www.cogs.susx.ac.uk/users/blayw/tt.html

White, J.G, Southgate, E., Thompson, J.N, and Brenner, S, 1986, "The structure of the nervous system of C. elegans", Phil. Trans. R. Soc. London 314:1-340, http://citeseer.ist.psu.edu/context/481575/0

Wok, 2001, "AI Scripting Guide", Star Wars Battlegrounds, http://swgb.heavengames.com/article.php?id=27#defining