COMP3330 - HiPhi Report - AI In Games

- Beau Gibson - C3146845
- Tyler Haigh - C3182929
- Simon Hartcher - C3185790
- Rob Logan - C3165020

Introduction

Video Games form a rapidly growing industry moving towards an artistic form that both programmers and players can appreciate (Wexler, 2002). The term "Game AI" encompasses programming and design practices, including path-finding algorithms, neural-networks, models of interaction, state machines, and decision trees, all with the common goal of providing some intelligent behaviour (Mateas, 2003). Game AI aims to establish the belief in a living entity within the computer that can act independently of the player.

The Founding Fathers

Game AI shares its foundations with the rise of digital processing. Claude Shannon and Alan Turing started developing chess programs as early as 1950. Board games were the popular choice for experimenting with Artificial Intelligence due to their nature of perfect information - the entire state of the game can be captured by the computer. They are typically non-trivial search problems with no well-defined path to a given solution as the involvement of an opponent introduces a degree of uncertainty (Middleton, 2002).

The Early Years

During the early years of video games, AI was not a feature due to the simplicity of the games produced, and human centred design.

In 1949, Claude Shannon published the paper *Programming a Computer for Playing Chess* in which he describes how a machine can decide how to move such that it can minimise the possible loss in the worst case scenario (Shannon, 1949). His reasons for using chess as the basis for his paper include:

- The legal move-set and the end goal is well defined
- It is neither a trivial problem, nor too difficult to calculate a viable solution
- There is a element of skilful thinking involved
- The discrete structure of chess fits well into the design of computers

A computerised version of Nim was published in 1952, and was able to consistently win games against players 95% of the time (The New Yorker, 1952)

The Golden Age

The Golden Age of Video Games revolutionised how we develop and play games today.

Game of Life

Game of Life is a deterministic, zero-player based, cellular automaton game developed by British Mathematician John Conway. Conway's rules for genetic change are designed to make behaviour of the population unpredictable (Gardner, 1970)

- Every counter with two or three neighbouring counters survives for the next generation
- Every counter with four or more neighbours dies from overpopulation
- Every counter with one neighbour or none dies from isolation
- An empty cell adjacent to exactly three neighbours is a birth cell
- All births and deaths occur simultaneously constituting a single generation

Discrete Logic

Atari released the infamous Pong in 1972, a table tennis simulator that can be played between two people, but more importantly with a computer controlled opponent. The core intelligence of these games was established using discrete logic (i.e. hard-coded logic gates). (Monfort, Bogost, 2009).

PacMan, released in 1980, introduced AI patterns in maze environments, with individual personalities (or strategies) assigned to each enemy ghost. Rather than having a knowledge or awareness of the maze, or even a rudimentary pathfinding algorithm, the ghosts were given specific "goal tiles" depending on their personality and their current "mode" (scatter/chase/frightened) which could be exploited by players who knew the underlying discrete logic.

One example of this is the pink ghost's goal tile being set 4 tiles in front of PacMan's position. This would generally lead to the ghost following the player's movements (leading to the ghost's nickname, "shadow"), however could result in the ghost losing a game of "chicken" against the player. When head-to-head, the ghost could recalculate it's goal tile and find that it lies behind itself, thus leading to the ghost radically changing direction away from the player. (GameInternals, 2010)

This discrete logic only gives the illusion of intelligence. This illusion provides little or no actual game logic or decision making and was mainly used due to memory constraints at the time.

Randomness

Random number generators are intelgral to Game AI. It makes the games feel more immersive and "real". Random numbers drive many decisions and events for in-game objects such as graphical effects (rain or transitions), frequency and choice of sound effects, and enemy behaviour.

(Doom, 1997) is a primary example of using randomness to establish interesting features in a game. It uses an array of 256 integers that are sorted randomly, and stored in a static lookup table. The game references these "seeming random" values throughout the game to make the game more dynamic.

Jonathan Dowland examined the behaviour of Doom when all values in the "random value" lookup table were set to all 0 (0x00 in Hexadecimal) or 255 (0xFF) (Downland, 2015). The following are some of his findings:

- Monster make the same sound effect
- Hitscan weapons (shotgun, chaingun) have no spread
- Lights will strobe, or never flicker
- Damage values don't vary
- Monsters will never make an idle sound with 0x00, but with 0xFF, they constantly play, drowning everything out

The 90's Explosion

As the arcade and video game industry began to wane in the late 1980's, gamers began to crave a better quality of gameplay. The goal of the industry then was to create a more challenging and lifelike opponent, which would react to the player like a human.

Dune II - Real Time Strategy

Early strategy games implemented simple "turn-based" gameplay. Dune II was the first Real Time Strategy (RTS) that would redefine strategy based games. Concurrent management of multiple units and build strategies within a 3D perspective allowed Dune II to set a new standard fo strategy-based gameplay and AI which would continue into games of today. The AI which was employed set individual goals for it's agents early in the game according to an overall strategy being employed (e.g. build vs combat). This overall strategy could then be

switched dynamically by the AI accoring to timing or in response to the player. Despite it being an early precursor to the familar RTS styles of today, the game's AI was less than impressive when measured against today's high standards. For example, it would often pursue a player's single unit with a single minded obsession while the rest of it's forces and base were being attacked by the player.

In interesting element of RTS games is that of AI "cheating". This relates to the amount of knowledge that an enemy AI could or should have about a player's movements. Obviously, a computer player has access to every input by the player at any time, and a sophisticated enough AI would be able to quickly counter any strategy a human player could devise, making the game virtually impossible to beat. Because of this fact, although a certain level of cheating by enemy AI is considered integral to the RTS game environment, it is important for this not to become too sophisticated (Buro, 2004).

Colin McRae's Rally 2 - Use of Neural Networks

Leveraging off his PhD work, Jeff Hannan utilised trained neural networks to control the opposing racers in Colin McRae's Rally 2. For the training process, he used a Resilient Back-Propagation (RPROP) training algorithm with random weights and a single hidden layer, providing sensor outputs his own driving as the input training data (Hannan, 2001).

There were two components used: Racing Lines and the Driving Model. If the track was wide enough, the racing line needed to be calculated in advance and used as the target for the agent. This racing line represents condensed knowledge about the track such as corners the agent is able to cut, obstacles, or off-road sections, and could be configured for different levels of skill. For narrow roads, the racing line could be considered as the centre of the track.

In order to get the agents to follow the racing line, a driving model is used. This model takes information about the car's state and the racing line, and calculates the controls for the car. After an unsuccessful attempt at manually defining a set of rules to drive the agents, Hannan turned to neural networks for training.

Current Research Topics

The primary goal of Game AI is to entertain. As such, developers tend to avoid using true "academic" AI, opting for short-cuts and efficiency, whilst maintaining some level of intelligence in the system.

Instead, researchers have redefined the role of artificial intelligence in video games by replacing all human interaction with intelligent agents. This concept of General Game Playing requires agents to be able to play numerous types of games without reliace on dedicated algorithms, designed specifically for a particular game. In this case, IBM's Deep Blue, the first intelligent computer to defeat the world chess champion, would not be sufficient as it was designed to play chess, and hence would not perform as well when playing checkers. (Genesereth, Love, Pell, 2005)

Google AI - DeepMind Technologies

Founded in 2011 by Dennis Hassabis, and acquired by Google in 2014, Google DeepMind aims to combine "the best techniques from machine learning and systems neuroscience to build powerful general-purpose learning algorithms" (DeepMind, 2015).

In 2013, DeepMind published a report detailing the success in using Reinforcement Learning environments, and sensory data, to train a neural network to play Atari 2600 games (NIPS, 2013). Using a modified Q-Learning (a model free) reinforcement learning algorithm, and stochastic gradient descent to update the network weights, their paper proposed a new deep learning model for reinforcement learning and demonstrated its effectiveness at mastering seven Atari 2600 games using only the raw pixels and game score as input.

In an interview with Dennis Hassabis, their network starts off with no prioir knowledge of the game (The Verge, 2015). Early training interactions involve the network randomly pressing mapped buttons in order to determine the basic controls and inputs. As expected with

Reinforcement Learning, the network adapts to the new game by playing. An interesting observation made was that the network was unable to plan long term actions and hence, struggled with learning maze-based games, such as PacMan.

The "Mario Lives!" Project

The Association for the Advancement of Artificial Intelligence (AAAI) holds an annual video competition to showcase the advances in artificial intelligence in a fun and exciting means.

Computer scientists from the Cognitive Modelling Group at the University of Tubingen, Germany, submitted the *Mario Lives* project as part of this competition in 2015. Their aim was to develop an artificial agent that can become alive in its own world.

In the demonstration, Mario is shown to learn from his own experiences. After jumping on Goomba, he is asked "What do you know about Goomba?". The speech recognition and context grammar that underpins the agent allows Mario to respond with "If I jump on Goomba, then it maybe dies". Mario is als shown to learn from what it is told. Mario is told "Goomba dies, when you jump on Goomba". He is able to store this information and refer to it as he responds "If I jump on Goomba, then it certainly dies". (Ehrenfeld, et. al., 2015)

"Mario develops a schema based knowledge base, and maintains internal emotive states". These emotive states are used to drive behavioural actions. When he is hungry, Mario will collect coins.

References

1. James Wexler, *Artificial Intelligence in Games*, http://www.cs.rochester.edu/~brown/242/assts/termprojs/games.pdf, University of Rochester, 2002
2. Grant, Lardner, *IT*, http://www.newyorker.com/magazine/1952/08/02/it, The New Yorker, 1952
3. Michael Mateas, *Expressive AI: Games and Artifical Intelligence*, http://lmc.gatech.edu/~mateas/publications/MateasDIGRA2003.pdf, The Georgia Institute of Technology, 2003
4. Zak Middleton, *Case History: The Evolution of Artificial Intelligence in Computer Games* http://web.stanford.edu/group/htgg/sts145papers/zmiddleton_2002_1.pdf, Stanford, 2002
5. Nick Motfort, *Racing the Beam: The Arati Video Computer System (PLatform Studies)*, http://books.google.co.uk/books?id=DqePfdz_x6gC&lpg=PP1&dq=racing+the+beam&pg=PA40&hl=en#v=onepage&f=false, ISBN: 9780262261524
6. Botea, Herbrich, Graepel, *Video Games and Artificial Intelligence*, http://research.microsoft.com/en-us/projects/ijcaiigames/, Microsoft Research, 2015
7. *Interview with Jeff Hannan*, http://www.ai-junkie.com/misc/hannan/hannan.html, AI-Junkie, 2001
8. Siyuan Xu, *History of AI Design in video games and its development in RTS games*, https://sites.google.com/site/myangelcafe/articles/history_ai, Xu, n.d.
9. Claude Shannon, *Programming a Computer for Playing Chess*, http://www.pi.infn.it/~carosi/chess/shannon.txt Shannon, Published in Pilosophical Magazine, 1950
10. *DeepMind Technologies*, http://deepmind.com/ DeepMind Technologies, 2015
11. Mnih, et. al. *Playing Atari with Deep Reinforcement Learning*, http://arxiv.org/pdf/1312.5602v1.pdf, DeepMind Technologies, Published by NIPS, 2013
12. The Verge, *Google's AI can learn to play video games*, http://www.theverge.com/2015/2/25/8108399/google-ai-deepmind-video-games, 2015
13. Genesereth, Love, Pell, *General Game Playing: verview of the AAAI Competition*, http://www.aaai.org/ojs/index.php/aimagazine/article/download/1813/1711, AI Magazine, 2005
14. Ehrenfeld, et. al., 2015, *Mario Lives! An Adaptive Learning AI Approach for Generating a Living and Conversing Mario Agent* https://www.youtube.com/watch?v=AplG6KnOr2Q, AAAI, 2015
15. Martin Garnder, *Mathematical Games. The fascinating combinations of John Conway's new solitare game "life"*, http://web.archive.org/web/20090603015231/http://ddi.cs.uni-

potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm, Scientific American, 1970
16. Jonathan Dowland, Deterministic Doom, http://jmtd.net/log/deterministic_doom/, JMTD, 2015
17. Chad Birch, *Understanding PacMan Ghost Behaviour*, http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior, GameInternals, 2010
18. Michael Buro, *Call for AI Research in RTS Games*, https://skatgame.net/mburo/ps/RTS-AAAI04.pdf, American Association For Artificial Intelligence, 2004