Practical Applications of AI and Machine learning in different areas within the Game Development Industry and multiple implementation approaches.

Albert Rando, RESEARCH METHODS IN COMPUTING AND IT - GMIT, G00330058
Professor - Dr. Martin Kenirons

Abstract—This paper draws a rough sketch of what the gaming industry represents in the current world and how AI and Machine Learning became almost core to it. It explains some areas in video games where AI is implemented and the different approaches used in order to maximize "human like" behaviour when required and how AI can be used to improve performance in pathfinding for the game agents.

Index Terms—AI, Machine Learning, Gaming, Game Design, Pathfinding, Agents, RTS, AI Behaviour.

1 Introduction

We have come a long way since first video games were created back in the 1950's. The first of which was 'Tennis for Two', introduced on October 18, 1958, at Brookhaven National Laboratory, An electronic tennis game that would become the father of modern video games[1]. But what is essentially a video game? As defined by [2] video game [noun] "A game played by electronically manipulating images produced by a computer program on a monitor or other display". Born in laboratories the first video games were developed by scientists putting the computational frames on Universities to the test.

On the other hand Artificial Intelligence as a concept was born earlier culturally in the form of robots on films and comic books. Alan Turing a British Mathematician started to tread the path by suggesting that just like human process information to make decisions, machines could equally do it. However, it was not until the 1950's that computational capabilities were powerful enough, not only to perform operations but also store to commands, to do some actual practical research. AI as such was conceived as such in a famous conference presented at the Dartmouth Summer Research Project on Artificial Intelligence (DSRPAI) hosted by John McCarthy and Marvin Minsky in 1956 [3].

As technology progressed very much in accordance to Moore's law, the increase in computational power allowed both fields, AI and Video Games to develop even further from the expected. As video games have grown both in scope and depth, AI has become an entwined part of it. From bots capable of defeating humans in games such as Chess or Go, to better gaming experiences and more realistic behaviours through NPC's (Non-playable Characters), AI provides with more enjoyable challenges an ever growing sector. According to a newzoo article[4] about the Gaming Industry's revenue in 2017, it is expected that 2.2 billion gamers worldwide will generate $108.9 billion in this year's exercise, which represents a significant increase from previous years.

In this paper, we pretend to introduce some of the most common uses for AI that predominate in video games, and provide a definition as well as to explain some approaches to implement them. We will also provide an insight of how machine learning has improved both capabilities and performance of some of the proposed models and algorithms and how can they be tuned to increase outputs.

2 Pathfinding

Pathfinding is a technique or a process carried by a software running in a computer from which the shortest path or route between two points is extrapolated. Pathfinding is, not only important in video games, but also in other areas such as delivery, transport services and intelligent storage among others. Pathfinding algorithms are usually, although not only, heavily based on the Dijkstra's algorithm[5] which implies what follows: Let the node starting be X. Let the distance of node Y be the distance from X to Y. Dijkstra's algorithm will assign some initial distance values and will try iteratively to improve them step by step.

However there are other complementary approaches using different techniques that seek to improve such algorithms.

2.1 Techniques for Pathfinding

Pathfinding is usually a two-step process: first, a graph generation algorithm followed by a Pathfinding algorithm to calculate the best path. Zeyad Adb Algfoor et al[0] published a brief but thorough compendium of the most common techniques developed during the last 10 years.

2.1.1 Grids

In Pathfinding grids are represented as graphs, a list of vertices or points connected by edges to represent the map, the navigation performance of which is based on its attributes. The most popular grid approaches are: regular grids and irregular grids.

Regular grids use triangles, squares and hexagons to describe tessellations of a given terrain. The most prominent of which are the following:

- 2D hexagonal Grid
- 2D Square Grid
- 2D Triangular Grid
- 3D Cubic Grid

Whereas irregular grids use different techniques to define terrain topology such as:

- Visibility Graphs
- Mesh Navigation
- Waypoints

Further information on the actual implementation of these pathfinding techniques can be found in Zeyad et al[9]'s paper.

2.2 Using Potential Fields

In 1985, Ossama Khatib discussed a new concept in regards a real-time obstacle avoidance for manipulators and mobile Robots. He named it Artificial Potential Fields[6]. In 2008, Johan Hagelback et al.[10] used it to create a Multi-agent Potential Field-based Bot for Real-Time Strategy Games(RTS) in which objects in the map where assigned charges (like magnetic chargers positive or negative) and such charges were used as forces to attract the agents to specific destination or to repel such agents to avoid collisions or to delimit the terrain.

They proposed two-different scenarios using Open real-time strategy (ORTS) engine to deploy the bot. In the first scenario the task of the bot was to recollect as many resource as possible in a limited amount of time. To do so Johan Hagelback et al. had 20 agents (workers) using a finite state machine (FSM) to define their behaviour and assigned the charges to the agents, to the resource gathering point, the base were resources had to be delivered and ”sheep” moving around the map to provide moving obstacles. Even though the results shown that the bot was prone to disconnect from the game in some occasions, it beat bots developed in previous years improving the average of resources gathered.

The second scenario, using ORTS, was a tank game where two players had a number of tanks and bases and the goal of the game was to destroy both enemy bases and tanks. After outlining the rules for the game agents, they assigned the charges accordingly and programmed the bot to avoid situating tanks in the middle of enemy clusters. The results of such scenario, tested against other bots, were of a 98% of victory rate over their opponents.

2.3 Aesthetics in Pathfinding

While most of the algorithms focuses lies on obtaining the best suitable path, such path does not always meet the expectations by using an unrealistic behaviour that differs greatly of what a humans would perform. In order to find more ”human-like” behaviour in Pathfinding some authors such as Coleman, R.[7] introduce the idea of Aesthetics in it.

The experiment's approach was to use Mandelbrot's Fractal Dimension Analysis[8] to determine the aesthetics level of certain paths, an A* Pathfinding algorithm was used to create the path and then tweaked to reward paths that were obstacle-prone to simulate a behaviour of ”stealthyness”. By doing so the levels of aesthetics were higher than in those were paths were determined as best suited by the algorithm.

3 MACHINE LEARNING

Even though Machine Learning has been a field of stud in computer science for the last 30 years, it was not until very recently that game developers started to implement it in their games. The lack of enthusiasm for adaptable behaviour in games typically responds to the fact that it is not really that important for a game to ”learn”. However, more and more developers release their games as services rather than products. Replayablity has become a huge factor, if using the same strategy over and over achieves always victory it gets tedious, more a chore than a challenge. A game that can learn adapt to its players has a higher chance to provide its users with more enjoyable situations.

3.1 Defining Behaviour with AI

Most games nowadays, either single or multi-player, have NPCs. Whether they are enemies, allies, unnamed or story characters it is important to define a behaviour pattern for therm to perform. However, players tend to reject erratic or inconsistent behaviour from agents, breaking game immersion. To create realistic behaviours that act in a more ”human-like” manner is the aim of many AI researchers in the video game industry. In some cases, the goal of the agent is not to perform an action as perfect as a computer would, being the most efficient, but rather try to simulate imperfection. As well, in order for NPC to act as other players is necessary to emulate the lack of information a real player might have such as not being able to detect an enemy that is out of sight but act alert to the possibility of him coming. With that in mind Marvin T. Chan et al [12] designed an AI that emulated driving like a human would do through a race track providing a more challenging experience for the player. As stated in the paper ”Although the players goal within the game is to win the race against the game-controlled car, the AI techniques adopted in the game are primarily designed to give the player an enjoyable time racing his or her car. In other words, the objective of winning by either side is not given the highest priority.”

Defining the behaviour of the AI requires the expertise to be able to analyse and describe accurately such behaviour and script it accordingly. Chek Tien Tan and Holun Cheng[11] proposed an agent personality representation model to provide an adaptable agent that can perform across a variety of games of different genres exhibiting a plausible adapted behaviour. The Tactical Agent Personality (TAP) is a framework model of progressive learning, to test its capabilities and evaluate the adaptability of the model three scenarios were created. In each scenario, TAP had a series of predefined behaviour with randomly assigned weights to start with and after every iteration a process starter to assimilate new weights (greedy search) and to assign a random value to seek new paths.

In the first scenario the model was tested against a First Person Shooter (FPS) environment. Three sets of experiments were performed. In the first set the AI was determined at random. In the second set the AI adapted its behaviour based on player performance. In the third set the AI adapted its behaviour base on the NPCs performance. The output of 500 tests per set demonstrated that the highest level of adaptability occurred when AI adaptation was based on the NPCs behaviour rather than player performance.

In the second scenario the model was tested against a Real-Time Strategy (RTS) environment. In addition to TAP, another layer of decision making was added, the Strategic Agent Personality (SAP). While TAP defined NPC behaviour at an individual level (short-term decisions), SAP determined their actions as a group (long-term decisions).

In the third scenario the model was tested against a Role-Playing Game (RPG) environment. In this test, the model was modified to switch the weight from the nodes or actions to the edges representing the temporary transitions between actions, Temporary Tactical Agent Personality (TTAP). The game consisted in two groups of characters 1 player and 5 NPCs. The first group implemented TAP while the second implemented TTAP. After running the experiment 500 times, the TTAP demonstrated a quicker adaptation but at the 500 the difference between the two models grow shorter and draws became more frequent.

TAP presents some interesting points such as high versatility and adaptation that reduces the need of specific scripting for actions escaping from more traditional approaches like Finite State Machines. TAP and SAP present a good performance and the scalability has low impact, however TTAP does not represent a better model over time and presents scalability issues as calculating the weight on the edges increments the overhead by a significant amount.

3.2 Perfect versus Imperfect Knowledge

As previously mentioned, in modern player-bot engagement-based video games, bot responses and behaviour are designed to be as realistic as possible. Is because of that, bots are designed to detect players just within its visibility area and not further than that. That is called Imperfect Knowledge. In games such as checkers or chess, both player and computer have complete visibility of the area (board) and all information about what is in it, thus we call this Perfect Knowledge. To provide the player with a sense of fairness it is necessary to program the bot to behave as it has imperfect knowledge. To do so several approaches make use of the principle of ”neighbourhood”

In their article, Peter K. K. Loh and Edmond C. Prakash, evaluate the performance of the existing Moving Target Search Algorithms through simulation:

- Basic Moving Target Search (BMTS)
- Weighted Moving Target Search (WMTS)
- Commitment and Deliberation Moving Target Search (CDMTS)

After simulation the results are compared against the Abstraction Moving Target Search Algorithm (AMTS) design, proposed in their paper. When compared, AMTS outperforms the three MST algorithms, with higher exploration moves, but the lowest MTS move. However further investigation is required as scalability might become an issue. Even though it has downsides, the algorithm presents a interesting approach in the matter.

Another approach includes Machine Learning to learn and generate strategies to increase performance while maintaining imperfect knowledge. Beaulac and Larribe [13] proposed a model based on Hidden Markov Models to narrow down the possible locations of a hidden mobile agent and pair it with machine learning to create a heat map where ”hotter” areas are more likely to be traversed by such agent and design strategies around that. To test the model, the experiments presents a simple game (pirate-themed), where the player has to reach two different points in the map. There are ”parrots” that indicate to the AI if the mobile agent is in its vicinity, providing more information but still maintaining the imperfect knowledge premise. The model demonstrated its effectiveness but the model requires further adjustments.

4 Conclusion

Our aim in this research is to discuss some of the approaches to AI implementation in video games and what are the higher impact areas in that regard. We also described in what manned the industry requires AI to be developed to match players expectations. It is still a field in expansion, and even though more than 60 have elapsed since the creation of the first video game and almost as much since the first game implementing a basic AI system, video games evolved always using cutting edge technologies to deliver better experiences. Machine learning provided video games with a new edge. However due to this very same reason, there are little to no standards or frameworks that unify such technologies. The future of AI in gaming, as approached by Safadi el al.[15] by conceptualizing video games, ”it becomes possible to create solutions for common conceptual problems and use them across multiple video games. Developing solutions for conceptual problems rather than specific video games means that AI design is no longer confined to the scope of individual game projects and can be more efficiently refined over time”.

A unified framework would serve as a core, around which game developers would be able to build add-ons, decreasing the amount of resources for creating pathfinding models and behaviours from scratch and providing a stronger foundation for AI in gaming.

REFERENCES

[1] Brookheaven National Laboratory, "The First Video Game?", Brookheaven National Laboratory, 1958. [Online] Available: https://www.bnl.gov/about/history/firstvideo.php

[2] Oxford Dictionaries, " Video Game", 2017. [Online] Available: "https://en.oxforddictionaries.com/definition/video_game"

[3] John McCarthy, Marvin L. Minsky, Nathaniel Rochester,and Claude E. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence", August 31, 1955.

[4] Emma McDonald, "The Global Games Market Will Reach $108.9 Billion in 2017 With Mobile Taking 42", Aprl 20, 2017. [Online] Available: https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/

[5] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". Numerische Mathematik. 1: 269271. doi:10.1007/BF01386390.

[6] O. Kathib. "Real-time obstacle avoidance for manipulators and mobile robots". The international Journal of Robotics Research, Vol 5, no.1,pp. 90-98, 1986.

[7] Ron Coleman, Fractal Analysis of Stealthy Pathfinding Aesthetics, International Journal of Computer Games Technology, vol. 2009, Article ID 670459, 7 pages, 2009.

[8] B. Mandelbrot, The Fractal Geometry of Nature, W. H. Freeman, San Francisco, Calif, USA, 1982

[9] Zeyad Abd Algfoor, Mohd Shahrizal Sunar, and Hoshang Kolivand, "A Comprehensive Study on Pathfinding Techniques for Robotics and Video Games" International Journal of Computer Games Technology, vol. 2015, Article ID 736138, 11 pages, 2015.

[10] Johan Hagelbck and Stefan J. Johansson, A Multiagent Potential Field-Based Bot for Real-Time Strategy Games, International Journal of Computer Games Technology, vol. 2009, Article ID 910819, 10 pages, 2009.

[11] Chek Tien Tan and Ho-lun Cheng, Tactical Agent Personality, International Journal of Computer Games Technology, vol. 2011, Article ID 107160, 16 pages, 2011.

[12] Marvin T. Chan, Christine W. Chan, and Craig Gelowitz, Development of a Car Racing Simulator Game Using Artificial Intelligence Techniques, International Journal of Computer Games Technology, vol. 2015, Article ID 839721, 6 pages, 2015.

[13] Peter K. K. Loh and Edmond C. Prakash, Performance Simulations of Moving Target Search Algorithms, International Journal of Computer Games Technology, vol. 2009, Article ID 745219, 6 pages, 2009.

[14] Cdric Beaulac and Fabrice Larribe, Narrow Artificial Intelligence with Machine Learning for Real-Time Estimation of a Mobile Agents Location Using Hidden Markov Models, International Journal of Computer Games Technology, vol. 2017, Article ID 4939261, 10 pages, 2017.

[15] Firas Safadi, Raphael Fonteneau, and Damien Ernst, Artificial Intelligence in Video Games: Towards a Unified Framework, International Journal of Computer Games Technology, vol. 2015, Article ID 271296, 30 pages, 2015.

Albert Rando Student in GMIT, currently attending fourth year in BSC in Software Development.