2018 International Conference on Advances in Computing and Communication Engineering (ICACCE-2018)
Paris, France 22-23 June 2018

Intelligent PC Games: Comparison of Neural Network Based AI against Pre-Scripted AI

Aakash Sehrawat
Computer Science and Engineering Department
Amity University, Uttar Pradesh, Noida
aakash.sehrawat@yahoo.com

Gaurav Raj
Computer Science and Engineering Department
Amity University Uttar Pradesh, Noida
graj@amity.edu

***Abstract* – Artificial Intelligence in PC games is a known concept and refers to designing the game such that even while playing games alone, there is a degree of intelligence in actions performed by the non-playable characters (NPCs) simulating human like behavior. It is often argued that video game AI is not truly intelligent and simply follows a set of instructions scripted by the developer. Hence, the objective is to study and compare traditional AI against a machine learned AI and make inferences based on the result. A game is developed in unity engine which is complex enough such that there is a balance of complexity in terms of difficulty to learn the game while also being simple enough to generate results as fast as possible.**

Keywords: Artificial Intelligence, Game Development, Machine Learning.

I. INTRODUCTION

AI stand for Artificial Intelligence and can vary from field to field in its application typically found in computer science robotics, graphics, control theory, etc. Video game AI however is often not considered true AI by enthusiasts as it is simply assigning pre-determined commands (called scripts)to a virtual entity which limit the number of responses it can have in any scenario. It does not encompass any sort of learning or intelligent behavior on its own.

A typical example of AI usage in games is Chess where the decision taken by the AI opponent is based on brute force technique using the "minimax algorithm".

Modern games make use of AI in:

1. Combat design,
2. Path finding by NPCs to find the shortest path without collision with terrain.
3. Detecting human players by visiting markers or by in game senses like hearing footsteps or vision.

Hence, there is a wide application of AI concepts that gives an illusion of intelligence to the players.

II. LITERATURE REVIEW

A. Literature Survey

With advancements in storage devices and processing power of machines, more efficient AI are being developed. Games offer the best examples where AI directly interacts with a human on a decision-making level. This helps us getting insight which can be useful in other fields involving humans and machines.

The OEP function in "Playing Multi-Action Adversarial Games: Online Evolutionary Planning versus Tree Search" is a very useful technique when the AI is incapable of going deep enough into the tree branches to find the optimal solution to a problem [1].

"Review of the Use of AI Techniques in Serious Games: Decision Making and Machine Learning" provides a lot of information about AI development techniques in the past decade and categorizing them into various columns for comparison [2].

Hyper heuristics are a great improvement over tradition heuristic techniques as mentioned in - "A Hyper Heuristic Methodology to Generate Adaptive Strategies for Games". It also helps in determining new strategies in games which have pre-established strategies [3].

The "Informed Hybrid Game Tree Search for General Video Game Playing" is an AI that is capable of playing a wide variety of games by familiarizing itself with the game objects. It is hybrid as it makes use of several previously mentioned search techniques at the same time to give the optimum solution [4].

B. Current Trends and Techniques

Elon Musk, a popular enthusiast in the world of Artificial Intelligence and the founder of OpenAI worked on a similar project with his company to develop a bot using neural networks by feeding it data sets of the game DOTA 2.

Within a span of 6 months the bot became so skilled that it was able to beat the top players of the game that had been honing their skill for 7 years.

978-1-5386-4485-0/18/$31.00 ©2018 IEEE

The bot did not use imitation learning or tree search to solve the problems in the game. It learned the games mechanics from scratch and built its own strategies which were unpredictable and challenging to even the most seasoned players.

C. *Advantages Over Current Techniques*

Developing an AI using tradition search methods is very tedious and becomes more and more complicated as the game mechanics and number of scenarios/decisions increase. Developing using neural network allows the AI to program itself by observing the decision-making patterns of humans repeatedly. Hence, it saves time and effort.

AI developed using traditional search techniques follow the same strategies each time for same scenario and become very predictable and repetitive in their decision making. Neural network-based AI is ever evolving and if existing strategy fails then it tries to develop a new strategy for the given scenario.
Unity is a Game Development Engine which is capable of developing games and animations for multiple platforms ranging from PC to mobile devices. It was initially exclusive for OS X but was later made available to 27 other platforms.

It is an all-purpose engine which allows rapid development of games by allowing the user to make use of its pre-rendered physics and ability to create simple 2D and 3D models within the application itself. Unity also allows user to make use of cloud services to be able to access their project from anywhere making it easier to manage their assets. The community also has a vast collection of free to use assets which significantly reduces the hassle to design your own models or purchase them otherwise.

III. TARGET GAMEPLAY

A. *Setup*

The game environment is an arena with desert terrain and many obstacles inside it. The arena is covered by a fence and the characters are only allowed to move within its perimeter.

It consists of 2 characters where one is controllable by the player and the other is the AI we are designing.

The game has a top-down point of view and the camera is aligned to always have the main character in its center.

B. *Mechanics*

1. Movement: The movement scripts are bound to W, A, S D on our keyboard. Pressing the keys will move the character in forward, left, backward, and right direction respectively in the arena.

2. Dash: Pressing the same movement key twice in rapid succession results in the character 'dashing' in that direction. This ability can be used twice before going into cooldown for 10 seconds.

3. Primary Attack: The primary projectile is bound to the 'left mouse button.' Upon pressing, the player character shoots in the direction of the mouse cursor. This ability can be used 10 times after which the player has to 'reload' for 3 seconds. The player has 30% reduced movement speed while 'reload' is in process.

4. Stun Attack: This attack throws a 'stun' projectile in the direction of the mouse and is triggered by pressing the 'right mouse button.' This attack stuns the enemy opponent for 3 seconds. It has a 10 second cooldown.

5. Heavy Attack: This attack hurls a 'heavy' bullet in the direction of the mouse and is triggered by holding the left mouse button for 2 seconds. This bullet does damage equal to 15 primary projectiles. It has a 20 second cooldown.

IV. DEVELOPMENT AND MODELING

We start by developing the game environment and the main character. Unity offers the convenience of creating terrain and other basic models instantly so we can start testing out the mechanics right away.

First, we create a basic 3D object over a terrain.

Next, we try to implement the local scripts for the various mechanics in the game one at a time.

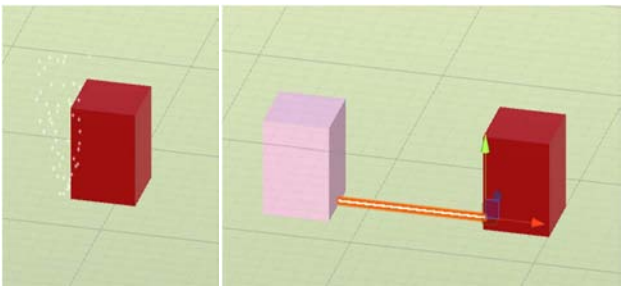

Fig 1: Dash and Movement

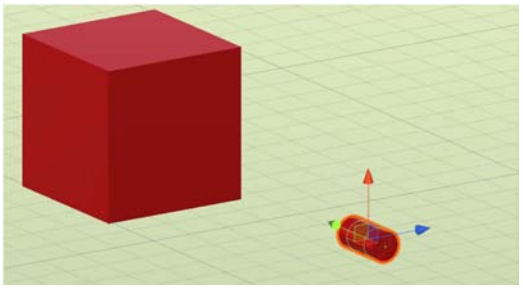

Fig2: Projectile shooting

3DS Max is an application that is necessary for importing and rendering 3D models that will later replace our simple shapes like cubes and planes.

978-1-5386-4485-0/18/$31.00 ©2018 IEEE

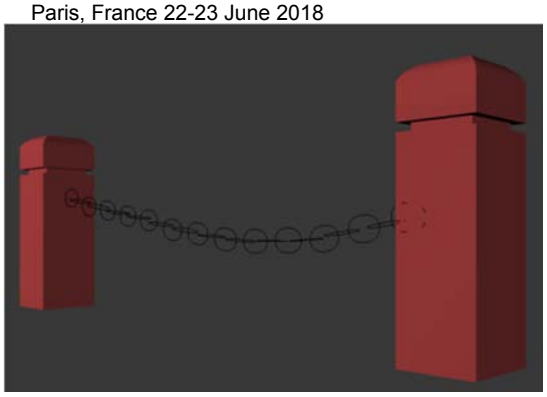

Fig 3: Replaced Shape with 3D models

V. NEURAL NETWORK INTRODCUTION

Artificial Neural network (ANN) is an extremely superior computing algorithm which has its simple concept taken from the structure of BNNs determined in an anatomy. ANNs can also be known as "synthetic neural structures," or parallel distribution processing systems, or connection gadget. ANN wishes a huge group of units which might be connected to every different in some styles and it to let's sign verbal exchange among the unit's flows. those gadgets or units, also are known as nodes or neurons, which perform parallel to each different. each neuron is connected to different neuron thru a connection of numerous links.

Each set is related in connection with a corresponding weight that continues the information approximately corresponding input sign. that is the maximum helpful statistics for each neuron has the capacity to know and clear up a specific problem because the weight generally corresponds the signal this is being transmitted or dispatched throughout. every neuron is present with a state internally, that is referred to as its activation signal. Output alerts, which can be transmitted after the aggregate of the input signals and activation rule are then in addition units are introduced.

VI. BIOLOGICAL NEURON

In living organisms, nerve cells are responsible for carrying sensitive information all across the body. There are up to 10^{11} neurons in the human body having as many as 10^{15} connections joining them. This is a good approximation to understand how complex a human's decision making can be.

Working of a Biological Neuron

There are 4 parts of a Biological Neuron. Explaining the working of each will help us better understand its function

Dendrites − These are thread-like structures, which are responsible for receiving information from neighbouring neurons. They act like 'ears' of the neuron.

Soma –It is the body of the neuron. The information received by Dendrites is processed here.

Axon –It is the long central section of the neuron which carries the processed information form soma to the synapses.

Synapses –Its function is to send the information to dendrites of other neurons.

Following table shows the terminology which have the same meaning for BNN and ANN respectively.

Table 1: BNN vs ANN terminology comparison.

Biological Neural Network (BNN)	Artificial Neural Network (ANN)
Dendrites	Input
Soma	Node
Axon	Output
Synapse	Weights

The following table is a comparison between ANN and BNN for the various tasks that they perform:

Table 2: BNN vs ANN function comparison.

Criteria	BNN	ANN
Processing	Highly Parallel, slow but more efficient than ANN	Highly parallel, fast but lower quality than BNN
Size	10^{11} neurons and 10^{15}interconnections	10^2 to 10^4 nodes {it mostly depends on the kind of application and network designer)
Learning	They can infer vast data.	To the point, structured and formatted data is being needed to understand vague data.
Fault tolerance	Performance can go lower with even a minute damage.	It is designed for outstanding performance, therefore can tolerate any kind of fault.
Storage capacity	It saves the data in the form of synapse	It saves the data in continuous memory locations

VII. DESIGNING AND LEARNING ARCHITECTURE

As already mentioned, games lack intelligence as they do not possess any sort of learning techniques. To deal with that, machine learning is used where a mechanism is provided for the AI to gather data and turn it into knowledge. The mechanism varies from system to system and needs to be

978-1-5386-4485-0/18/$31.00 ©2018 IEEE

carefully crafted so that the AI collects right information and make predictions from it. In the case of this game, the AI is guided by a reward and punish ladder which tells it what decision is best in a given scenario. Reinforcement learning is a suitable technique for such a learning model.

A. Reinforcement Learning

In this technique, the AI is not given a set of instructions as to what is to be done. Instead it is simply told what is right and wrong and based on them the AI learns the game. The AI will then perform random actions in an attempt to learn from its environment where every right thing earns it a reward and every wrong thing a punishment. Doing this repeatedly given enough time will lead the AI to learn what actions to perform which are more likely to lead to a positive reward.

Unity consists of **Machine Learning Agents** (or **ML-Agents** which are simply a Python API that allows the AI to make use of deep reinforcement learning. It is an advanced form of machine learning where the AI learn about their environment from their actions and make changes to their behaviour based on that. This give the game a level of dynamic difficulty where more the agent plays the game, more efficient it becomes.

Follow is a flow chart that represents State and Reward function in the AI.

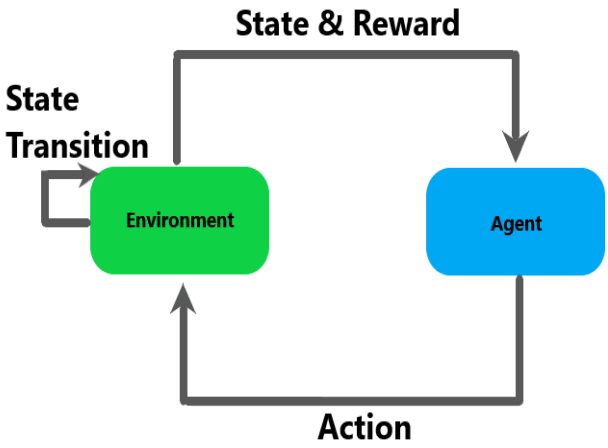

Fig4: The reinforcement learning training cycle.

The learning environment to be configured for the ML-agents consists of 3 primary objects:

- *Agent*: All agents have a unique set of states, observations and actions, each having rewards with event associated with them.
- *Brain:*Brain in this case (much like a human brain decides what actions are to be taken for particular scenarios.
- *Academy:* This object contains all the brains within the environment. However, this is not needed in case of a single AI.

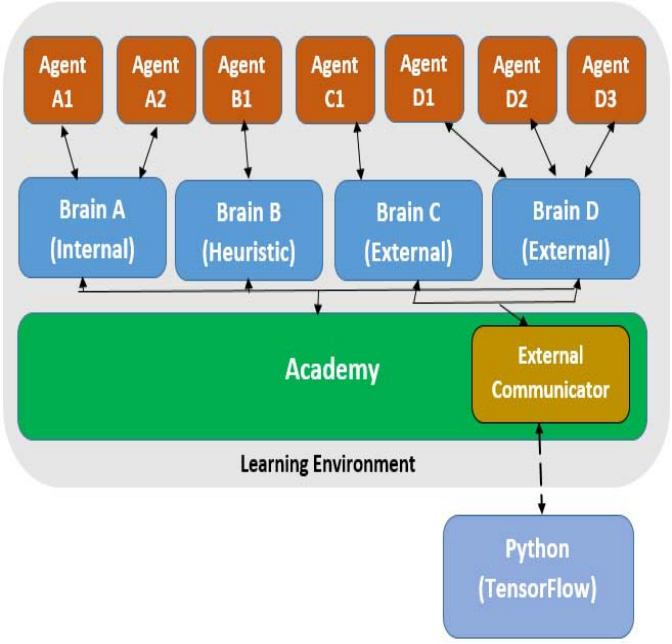

Fig 5: Diagram representing the learning architecture**.**

Here, reinforcement learning is used to build a neural. As we know, Neural Networks are created using neurons which collectively represent layers. Theinputlayer interacts with the environment and gathers input information. On the other hand, the Output layer is responsible for storing the outputfor a given set ofinputs. Lastly, the hidden layerslie in the middle of these two and is simply an additional layer of input sandwiched to create more outcomes. This allows few parameters to create a complex network of outcomes which is as close to intelligent as it can get. Neurons of each layer are connected such that output of any given layer becomes the input of the succeeding layer. They often have weights which determine the degree of link between two neurons of two layers.

The learning process happens by having the agents perform actions in the game and try to beat the human player. The results of those actions are fed back to the agent along with the respective rewards and punishments associated with those actions.

The learning environment contains an “academy”that contains information about the parameters that are to be used during training (shots hit, shots received…) as a script. Those parameters are then transferred to the brain – an entity guiding an agent with a training model. Nextan Agentis connected to the Brain, which tells it what actions to take and the information for those action is stored again to complete the learning cycle. Training is performed by, usingadditional

978-1-5386-4485-0/18/$31.00 ©2018 IEEE

functionality called TensorFlow python script which connects the brain to an external library for learning. Once one iteration of training is complete, the TensorFlow makes a new model which is actually a new brain for the agent to take commands from. This acts as a version control as the AI progresses.

B. The reward function

The reward function is fairly simple, hit the target with bullets until it dies before time runs out. On the other hand, a punish function exist to punish the AI for receiving shots from the human player and ultimately dying.

Fig 6: Punish and Reward ladder

After many attempts, the AI started improving by making greater use of the in game mechanics like dashing to dodge the human bullets and shooting the human with superior bullets. It also improved in accuracy and was better able to predict the pathing take by a human. The reward function works so well that it helped ultimately helped in identifying elements of the game that were unnecessary. For example, if the AI is at a point that it is able to beat a human opponent easily without using a particular feature in the game, that feature can be downright removed as the algorithm does not believe that using it will help increase the chances of winning.

VIII. STATISTICS AND ANALYSIS

After a total of 150 games played against traditional and neural network based AI, the following statistics were recorded:

Fig 7: Statistics for shots successfully hit by both AI.

The above graph shows the shots that an AI successfully hit a human player with. The It shows a generally increasing trend for neural network and decreasing in case of standard AI. This is due to the fact that while neural network AI uses previous games output as input and improves upon it, standard AI repeats the same strategy again and again and grows predictable over time.

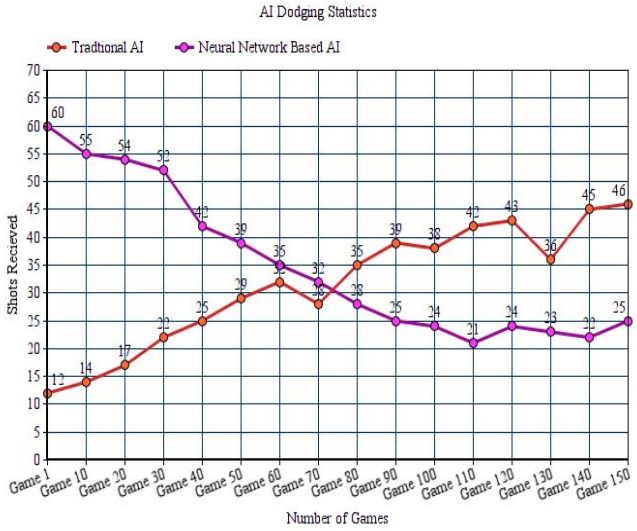

Fig 8: Statistics for shots received by both AI

This graph depicts the ability to dodge shots fired by a human player. A general increase in shots received by standard AI and decrease in case of neural network AI.

The same explanation holds good in this case as well. While neural network grows more and more capable of predicting human shots. Standard AI does not improve upon its pathing and follows its coded path becoming predictable and easier to hit.

Following are some of the limitations faced by neural network system:

1. Requires huge data set: Neural networks are able to find solution to complicated problems the same way humans do, however, they require a very large data sets of how humans are currently solving those problems to build its own network.

2. Unreliable solutions when unexpected event occurs: Neural network-based AI are usually effective. Even so, they sometimes generate unreliable results. This happens in special cases where the problem is completely new to the AI and it does not have a record for it in its data set.

3. Perfect data: In order to make a reliable neural network-based AI, one needs to know the system very well for which it is being made. If the system has variables who values have no correlation whatsoever, then the neural network too will generate solutions which are incapable of working in the long

978-1-5386-4485-0/18/$31.00 ©2018 IEEE

run. The network needs access of all the variables and the factors affecting those variables in order to be successful.

IX. Conclusion And Future Scope

The following study shows that Neural Network based AI are capable of generating solutions to complicated problems for systems involving humans. The standard AI became more and more predictable while the machine learned AI only improved with each game to the point that it became too overwhelming for a human to beat. Deep Learning and neural network are evolving fields and they will only get more and more advanced from this point henceforth. It is a major step in the development of true Artificial Intelligence that can be almost as intelligent as a real human.

This technique can also be used in many other platforms like system which can predict stock market, help in medical diagnostics based on symptoms and past records.

There is still great room for growthin this field and it can potentially be a trigger for the next generation of computers in the coming future

References

[1]. Niels Justesen, Tobias Mahlmann, Sebastian Risi, Julian Togelius, "Playing Multi-Action Adversarial Games: Online Evolutionary Planning versus Tree Search", IEEE Transactions on Computational Intelligence and AI in Games, 11 August 2017.

[2]. Maite Frutos-Pascual, Begoñya García Zapirain, "Review of the Use of AI Techniques in Serious Games: Decision Making and Machine Learning", IEEE Transactions on Computational Intelligence and AI in Games, 25 December 2015.

[3]. Jiawei Li & Graham Kendall, "A Hyper heuristic Methodology to Generate Adaptive Strategies for Games" IEEE Transactions on Computational Intelligence and AI in Games, 21 January 2015.

[4]. Tobias Joppen, Miriam Moneke & Nils Schroder, "Informed Hybrid Game Tree Search for General Video Game Playing", IEEE Transactions on Computational Intelligence and AI in Games, 06 July 2017.

[5]. Pell, "Strategy Generation and Evaluation for Meta-Game Playing", Ph.D. dissertation, Computer Science Research Centre, Trinity College, Cambridge University, Cambridge, United Kingdom, 1993

[6]. Silver et al., "Mastering the game of go with deep neural networks and tree search," Nature, vol. 529, pp. 484–503, 2016.

[7]. S. Risi and J. Togelius, "Neuro evolution in games: State of the art and open challenges," IEEE Transactions on Computational Intelligence and AI in Games, 2015.

[8]. Niels Justesen, Sebastian Risi, "Learning macro management in starcraft from replays using deep learning", Computational Intelligence and Games (CIG) 2017 IEEE Conference on, pp. 162-169, 2017, ISSN 2325-4289.

978-1-5386-4485-0/18/$31.00 ©2018 IEEE