International Journal of Engineering & Technology, 7 (2.28) (2018) 204-205

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Development of Non-Character Player Using Self-Learning Algorithm for Artificial Intelligent Games

M. A. Noryushan[1], N. Zamin[2], H. A. Rahim[3], M. A. Sahari[4], N. I. Hassan[5], Z.M. Fauzee[2]

[1]Student, University Malaysia of Computer Science and Engineering, Putrajaya, MALAYSIA.
[2-3]Assistant Professor, University Malaysia of Computer Science and Engineering, Putrajaya, MALAYSIA.
[4-6]Lecturer, University Malaysia of Computer Science and Engineering, Putrajaya, MALAYSIA.
**Corresponding author E-mail: azimnoryushan@gmail.com*

Abstract

In most of video games, the Non-Playing Character (NPC) behavior and movement are usually scripted. Players who have exploited the NPCs weaknesses will be able to beat them easily and there will be no freshness in player experiences. However, if the character can adapt and learn from the environment, it will be more interactive since players need to find new weaknesses to exploit. In this project, an agent that can learn by itself in the game which is introduced. This ongoing project investigates and compares the available self-learning algorithms used in game development and will be implemented as the intelligent agent. The Fourth Industrial Revolution (IR 4.0) has the potential to raise global income levels and improve the quality of life through Artificial Intelligence (AI) programs. AI has made possible new products and services that increase the efficiency and pleasure of our personal lives such as dynamic games that can learn from its environment.

Keywords: *Artificial Intelligence, Self-learning Algorithms, Intelligent Agent, Non-Player Character.*

1. Introduction

Games development needs a specialize tools to capture the movement for the move set of character which cost around USD1500 per day refers to [1]. On top of that, for a big game like Role-Playing Game (RPG) needs lot of more characters to input the move set for each of them. Artificial Intelligence (AI) in game development provides intelligent behavior to the NPC and it has been a great substitute for players to interact without another human player. Conventional games are fixed and learn from a set of rules and states. They do not learn from environment. Additionally, most AI game movement still have some deficiency with it for example the movement is not smooth and natural. In game AI, characters should appear dynamic and adaptive to the changing environment to create extra challenges and fun for human players. This ongoing research investigates the possibility of using unsupervised learning method known as Neuro-Evolution of Augmenting Topologies (NEAT). It is a method for evolving artificial neural networks with a genetic algorithm. NEAT was introduced by Ken Stanley from Austin University [2] implements the idea that it is most effective to start evolution with small, simple networks and allow them to become increasingly complex over generations.

2. Self-Learning Algorithm

NEAT is a complex reinforcement learning technique for evolving neural networks using Evolutionary Algorithm (EA). NEAT is the combination of the usual search for appropriate network weights with complexification of the network structure. This allows the neural network behaviour to become more sophisticated over generations. Figure 1 shows a genotype represented as a phenotype. The connection between nodes 2 and 4 is disabled so that it is not expressed in the phenotype.

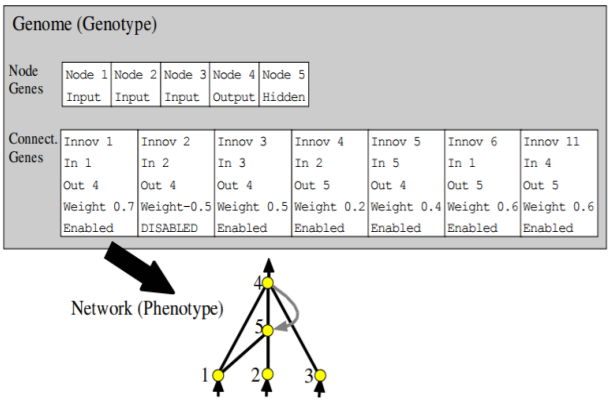


Fig. 1. NEAT genotype to phenotype mapping example [2]

Each genome in NEAT has a list of connection genes which shows the two nodes genes is connected. Each connection has its own in-nodes, out-nodes, the weight of the connection, whether the connection gene is expressed, and an innovation number, which is used for finding corresponding genes during crossover [2]. NEAT has been used in training a human like muscular-skeleton arm model. The uniqueness of its smooth and natural movement as well as low energy cost has attracted many researchers' interest. The muscular-skeleton arm model has three link driven by nine muscles and propose a control model with two

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

neurocontroller trained by the NEAT algorithm. To evolve the network, sets of training is generated by using forward kinematics, geometry relationship and muscle mechanic equation. By using two neurocontrollers, Position-Angle and Angle-Activation, the control model can deal with the redundancy and non-linear problems in separate neurocontroller at the same time [3]. The review of several types of NEAT algorithm is presented in **Table 1**.

Table 1. Summary of NEAT series.

Algorithm	Encoding	Evolutionary Algorithm	Aspect Evolved	Strength	Application	Highlight
Neuro-genetic Evolution [2]	Direct	Genetic Algorithm	Network weights	Have multiple offspring, they can explore the solution space in multiple directions at once [4]	Prediction of software development effort.	Nil
NeuroEvolution of Augmenting Topologies(NEAT) [3]	Direct	Genetic Algorithm.	Structure and parameter	Track genes with history markers to allow crossover among topologies. Apply evolution of species to preserve innovations.Develop topologies incrementally from simple initial structures	Neuroevolution of augmenting topologies based musculor-skeletal arm neurocontroller	Track genes using historical marks to allow crossover between different topologies, protects innovation via speciation.
Real-time NeuroEvolution of Augmenting Topologies (rtNEAT) [3]	Direct	Genetic Algorithm	Structure and parameter	Able to train the intelligent agent in real-time. Enhancement of NEAT.	Neuro-Evolving Robotic Operative game (NERO)	Upgraded from NEAT with the capability of player interact with the evolving agent real-time.
Hypercube-based NeuroEvolution of Augmenting Topologies (hyperNEAT) [3]	Indirect	Genetic Algorithm	Parameter, and structure fixed	Enhancement of NEAT. Able to exploit the geometry of the problem	Multi agent learning Controlling leg robot Evolving Objects that can be 3D-printed Autonomous robots that learn how to drive	Used compositional pattern-producing networks. Able to evolve large neural network.
Online Distributed NeuroEvolution of Augmenting Topologies (odNEAT) [4]	Indirect	Genetic Algorithm	Weight and network topologies	Able to train the intelligent agent in real-time. Enhancement of NEAT	E-puck robot.The open racing car simulation.	Distributed evolutionary algorithm for online learning in groups of agents

This research adopts HyperNEAT algorithm because it has some unique features which can be used to train the character or intelligent agent as described in the next section.

3. Game Development in 3D Format

HyperNEAT algorithm will be implemented in a modelling software. The 3D model will then be applied in Unity3D to implement the move sets and also implements the algorithm. Unity3D allows the 3D model to get an input which is the move sets and produce the output which is the simulation of the 3D model as shown in **Figure 1**. It will be developed using C#.

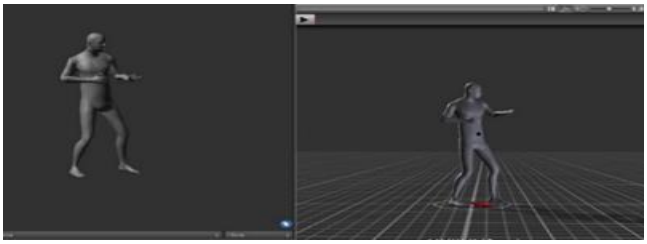


Fig. 1. Example of 3D model used in Unity3D.

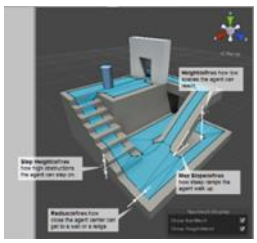


Fig. 2. An example of training course.

At this stage, the training of the intelligent agent is in progress. The intelligent agent will be trained from a generation to a generation and stops when it has produced the best and consistent results. The intelligent agent will have to pass several courses as shown in **Figure 2** with obstacles such as walking on a steep path and uneven terrain. This will test the stability of the intelligent agent and making them learn on how to stabilize themselves. In each iteration of generation, the data is recorded to be analyzed later. The algorithm of NEAT begins with calculating the statistic for each specie (mean fitness, target size, number of offspring to produce). The algorithm includes trimming the species to produce only the elite genomes which will be evaluated.

4. Conclusion

Game development requires the character to be animated and flexible so that human players will have fresh experience when playing the game. On top of that, animating the game characters usually requires high cost. Many existing technologies are using static codes on the NPC thus making the game predictable. The research proposed on the evolution of the NPC using HyperNEAT algorithm so that they can adapt and bring fresh experience to players. The interest in gaming had shifted from learning how best to beat human players to enhancing human players' gaming experience with AI. The most common role for AI in video games is the NPCs. Many research have found that video games can distribute intelligence between the player and artificially intelligent virtual characters such as the NPCs. Smart and intelligent machines are one of the key components in IR 4.0. Thus, intelligent game development may have significant primary market research at present and in future.

References

[1] Motion Capture NYC. “How Much Does Motion Capture Cost?” motioncapturenyc.com. http://www.motioncapturenyc.com/ (accessed October 11th, 2017).

[2] Stanley, Kenneth O., and Risto Miikkulainen. "Evolving Neural Networks Through Augmenting Topologies." Evolutionary computation 10, no. 2 (2002): 99-127.

[3] Wen, Ruoshi, Zixi Guo, Tong Zhao, Xiang Ma, Qiang Wang, and Zhaojun Wu. "Neuroevolution of augmenting topologies based musculor-skeletal arm neurocontroller." In Instrumentation and Measurement Technology Conference (I2MTC), 2017 IEEE International, pp. 1 - 6. IEEE, 2017.

[4] Marczyk, Adam. "Genetic Algorithms and Evolutionary Computation." The Talk Origins Archive: http://www. talkorigins/faqs/genalg/genalg.html (2004).