Project Proposal:

Design of intelligent non-player character in games

Like Chen
October 25, 2018

Motivation and Background

Nowadays, video game becomes an important part of people's life. And more and more computer scientist and software engineer, decide to be a game developer, they may want to develop their own game, or they may want to become in one of the 3A game developer group.

RPG (Role-Playing Game) is part of the biggest kinds of video games. In a RPG game, player control a character in the game world, they can explore the world, they can fight against others, the whole point of RPG game design is to build an immersing and interactive virtual world. NPC (non-player character) is one of the significant parts of RPG, NPC can interact with player such as talking to player, fight against player, they can be player's assistant etc. Moreover, NPC may interact with environment, they may sit down under the tree rest, they may hold weapon and be alert, they may just walk around the street. NPC might interact with each other, they may clash with each other in begin with a quarrel, then start to fight with weapons. May be two NPC has different camp, and fights once they meet each other. It is all about building an immersion game world, to let the player enjoy the game play.

When I was play NPC game such as The Witcher 3, Legend of Zelda: Breath of the Wild, The Last of Us etc. I found that most of the enemy NPC in these games have a lot in common. They patrol some area, they have an 'alert system' (when they first sight player, they do not direct chasing player, they turn into an alert mode, often have a question mark on their head, when alert value full, then they turn in to chase mode, begin chasing player, and when player get out of their sight, they will investigate area that the player finally exists), they interact with game world (pick up weapon when they are chasing player, sitting by the fire, chopping tree with axe etc.), they interact with each other (soldier attack monster, monsters dancing with each other by the fire etc.). In each game developing, developer using much time on design NPC behaviors and NPC decisions. And there is a complicate part is the difficulty of the game, if NPC is too 'smart', the game will be too hard to play, it may frustrate player's confidence, but if NPC is too 'stupid', the game will be too simple to gain sense of achievement. NPC's intelligent level is hard to control by the developer.

Figure 1: NPC alert value UI

Figure 2: NPC changed mode

Hence, I decide to developer a NPC brain that can let NPC have those behaviors in common and let NPC intelligent level controllable. This brain needs to be adaptable to many RPG games, so that a game developer can save much time on the NPC developing, they can just focus on other part of NPC developing such as moving, animation.

I already have many game develop experience in Unity editor, and Unity editor is one of the most popular game engines in the world, so this project I decide to use Unity as developing tools and C# as the programming language. Moreover, Unity has a new open source project call Unity Machine Learning Agent (ML-agent), which can connect TensorFlow to the Unity game environment which can use machine learning to develop a model of the game controller. This is exactly what we need for developing NPC brain. We can train NPC with TensorFlow then get the model, and it can easily control the intelligent level of the model.

Aims and Objectives

The aim of this project is to develop a tool that can reduce game developer's work when they need to develop NPC. This tool should be adaptable and can be used in many RPG game.

The key objectives of this project are:

1. Investigate the utility of machine learning in game environments.
2. Compare the difference between determined algorithm and machine learning model.
3. Develop some demo of RPG game to be a test environment
4. Use the machine learning result to develop the "NPC brain" tool
5. Make this tool adaptable

Project Plan

This project needs a demo to show that the different of determined algorithm and machine learning model. Hence, I must build a game demo then I can do research on it. Since I have many games developing experience, and I do not have enough machine learning knowledge and skill now, I decide to build one or two demos at the early time while I study the machine learning course, then I can start do research on how to link TensorFlow and Unity game with ML-agent. The best result is that I can have two or more demos that can compare determined algorithm and my "NPC brain".

Date		Plan
2018/10/16	2018/10/23	First demo develop should be finish in the first month, the demo should have: a character that can be control by player, a aim that player should acomplish, some NPC against player, a common RPG game scene
2018/10/23	2018/10/30	
2018/10/30	2018/11/6	
2018/11/6	2018/11/13	
2018/11/13	2018/11/20	Learning TensorFlow knowing what machine learning work
2018/11/20	2018/11/27	Figure out how to use ML-Agent to train NPC
2018/11/27	2018/12/4	
2018/12/4	2018/12/11	try to train the NPC brain
2018/12/11	2018/12/18	
2018/12/18	2018/12/25	Study for other module, it time for final exam
2018/12/25	2019/1/1	
2019/1/1	2019/1/8	interim report
2019/1/8	2019/1/15	
2019/1/15	2019/1/22	
2019/1/22	2019/1/29	
2019/1/29	2019/2/5	Winter break, if I have time I might start next semester work early.
2019/2/5	2019/2/12	
2019/2/12	2019/2/19	
2019/2/19	2019/2/26	
2019/2/26	2019/3/5	
2019/3/5	2019/3/12	try to build second demo and research how to make the brain tool adaptable
2019/3/12	2019/3/19	
2019/3/19	2019/3/26	
2019/3/26	2019/4/2	
2019/4/2	2019/4/9	If the previously plan did not went well, here some time for more working, but every single work cannot delay more than one week, so that I have time to consider how to write the final dissertation.
2019/4/9	2019/4/16	
2019/4/16	2019/4/23	
2019/4/23	2019/4/30	
2019/4/30	2019/5/6	

Reference

[1] Jay Collins; William Hisrt; Wen Tang; Colin Luu; Peter Smith; Andrew Watson; Reza Sahandi. "EDTree: Emotional Dialogue Trees for Game Based Training". Spirits.bournemouth.ac.uk. Retrieved 10 October 2017.

[2] Juliani A, Berges V P, Vckay E, et al. Unity: A General Platform for Intelligent Agents[J]. arXiv preprint arXiv:1809.02627, 2018.