AI Tag Game with Unity

Matthew Waters
BEng Software & Electronic Engineering

GMIT
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY

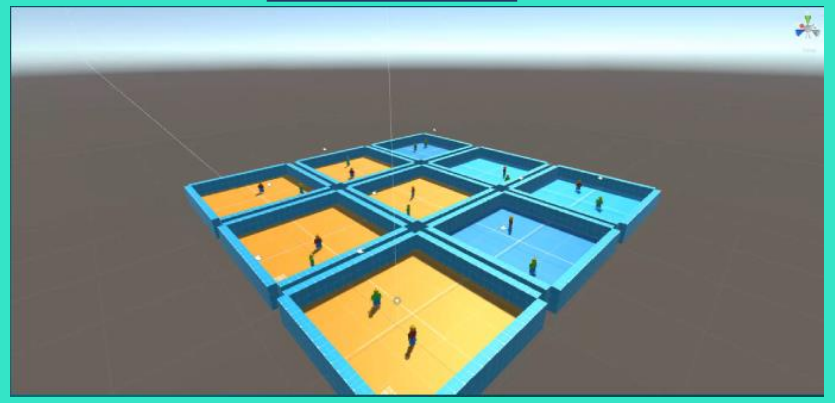
Summary

My plan for my project is to create a tag game with Unity which I will use as a base to train two AI agents to play tag with each other. I have a fascination with games, and I like the idea of AI characters playing against each other, so I want to bring the two ideas together.

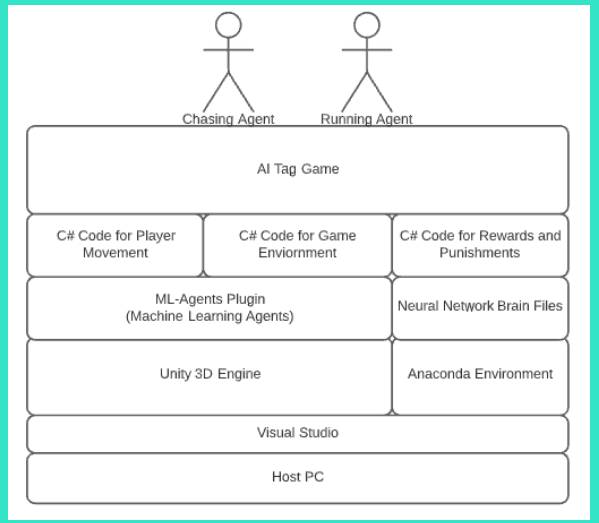
I plan to use Unity's ML-Agents plugin which uses reinforcement learning to teach the AI agents. I think this is where the challenge will be. To successfully program the agents so they can learn easily and hopefully play tag with each other.

This will be challenging because I will be learning a new language C# and learning how reinforcement learning works along with teaching a neural network for the agents.
The basic game of tag starts off simple with two characters chasing each other.

Agents Learning Environment



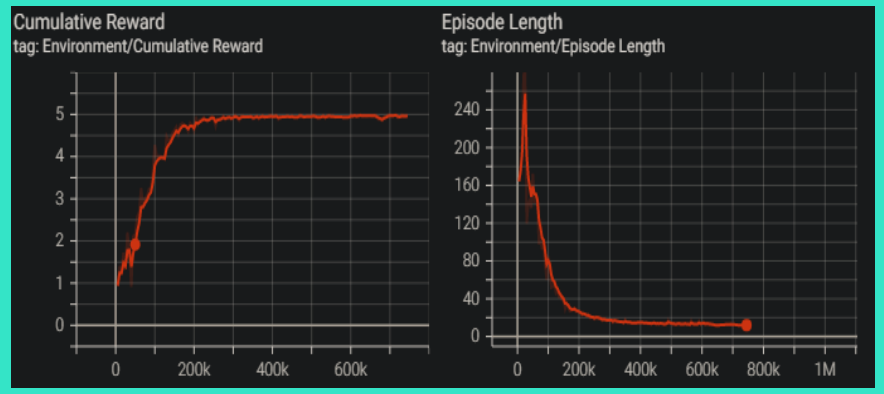
Architectural Diagram



Unity

Unity is a cross platform game development engine which was developed with ease of use in mind which is why it was my chosen development platform.
Unity has a plugin called ML-Agents which allows you to train AI Agents within Unity
It is developed in a way where it can create extremely complex games. There is many corporations which use Unity as their game engine and use it to create industry standard games which do very well. Having this diversity and support in a game engine is great for modern game development.

Agent learning to how to get a reward faster



Reinforced Learning

Reinforced Learning works very similar to how humans and animals use their brain to learn new things.
Reinforced learning uses its own brain and given abilities to make observations about its surroundings and use these observations to make a decision.
The brain then gets a reward depending on its decision. If it is correct it gets a big reward and learns that they did something correct. If it is incorrect it gets a negative reward and then learns from this mistake for the next time it tries.