3D Game: Enhancing Game AI with Machine Learning

Nicholas English, Dr. Byron DeVries

Grand Valley State University
College of Computing

Abstract

Bots/NPCs in video games do not generally implement Artificial Intelligence (AI) or Machine Learning (ML). The goal of this project is to implement ML so that bots react and use strategies as a game progresses.

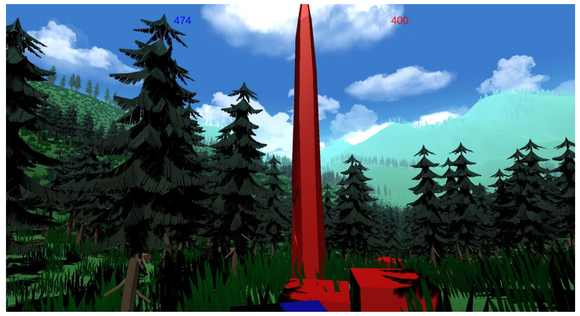
Introduction

Video game companies often refer to bots in their games as AI. However, these bots do not actually implement any form of AI or ML. The terms AI and ML are often used synonymously, but they are ever-so slightly different. ML is a subset of AI and involves creating an algorithm that computers use to perform pattern recognition.

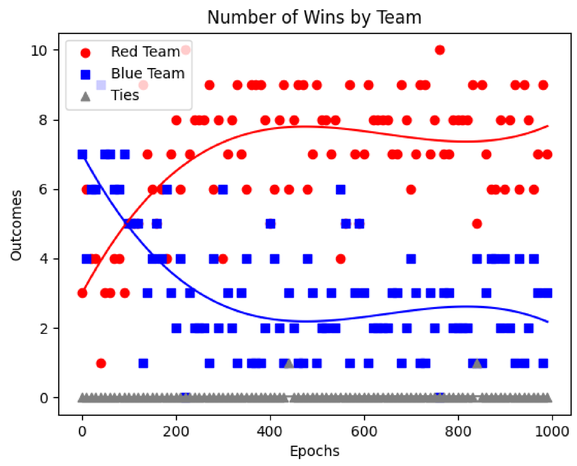
The goal of this project is to create a 3D game from scratch in Unity, and to create smarter bots that implement ML so that a team's bots can react as the state of a game progresses. The only game mode that exists is a conquest inspired mode. Teams must capture points on a map, which allows for the team to accrue tickets. First team to 500 tickets wins the game. The smart team's bots are expected to learn which points are most important to capture and hold for any given state.

Results/Implementation

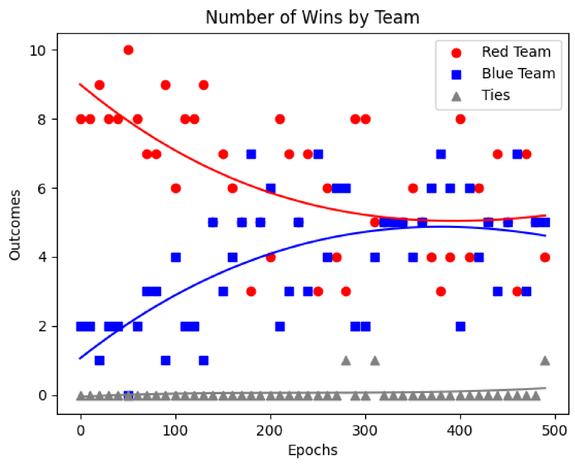
The game has five obelisks that act as point that can be captured. The bots on the teams are comprised of cubes that run round the map. The player can join either team, and the player can set if a team is basic or smart. Basic bots use randomization to pick a point to attack. Smart bots are controlled by a ML orchestrator, which will determine the resource requirement of every point for the given state, and then distribute bots as needed.



The first team trained was the Red team, which played against a Blue team that was comprised of basic bots. As would be expected, the Red team needed some time to learn the game, but by roughly epoch 130, they were winning more consistently. By about epoch 250, the Red team was winning consistently at just under 80% of the time.



The next round of games placed a smart Red team against a now smart Blue team. Red team retained its knowledge from the games it previously played, which placed it as an initial advantage. This advantage did not last long as the Blue team was able to figure out consistent tactics for winning. By epoch 300, Red team and Blue team reach convergence of a 50% win rate.



Technical Details

The game was built using the Unity Game Engine, and it was developed using C#. Art and music assets are appropriately attributed in the game and in the GitHub repository.

Learning occurs by implementing the Q-Learning algorithm, which is a reinforced learning algorithm. The bots are directed via an orchestrator, which is the object that implements the knowledge. The orchestrator uses Boosting to help determine which obelisks are most important to capture. All next states that have a positive value are considered.

The scatter plot visualizations were generated through Python. The script developed utilizes the Matplotlib library.

Future Work

It would be beneficial to have a layer of AI on the individual bots so they can make real-time discissions as they encounter enemy bots. An additional beneficial modification would be to create a setting that increases the speed of the game, as this would allow for training sessions to be completed in quicker succession.