Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

Maze GAME

Final year project

Lokesh Panti

Software and Electronic Engineering (Hons)

Technologies Used

- Unity
- Blender
- C#
- ML-Agents
- Jira
- GitHub
- Python
- Pytorch
- TensorFlow
- Microsoft Visual Studio

About Maze Game

Maze game is an interactive experience with machine learning agents and a user interface, challenging players to navigate through a randomly generated maze to reach the flag.

Using Unity Gaming Engine, maze game focuses around the observation of machine learning agents, and their efficiency to navigate mazes.

Why Unity ?

- Build in support for ML-Agents, TensorFlow and Python.
- Unitys Machine learning tools are more beginner-friendly.
- Unity also has a large community.

C#

- Primary scripting language for Unity, for game logic, behaviour and gameplay mechanics.
- Popular and more widely used programming language.
- Unity uses C# as it is an OOP language.

What are ML-Agents ?

- ML-Agents are machine learning tools developed by Unity Technologies, for game developers to create intelligent agents in their games.
- They allow developers to create and train agents that can interact with the game environment and learn from their experiences using different reinforcement learning algorithms.

What is TensorFlow?

- Tensorflow is an open-source software library developed by Google.
- Uses a wide range of features and tools for developers and researchers.

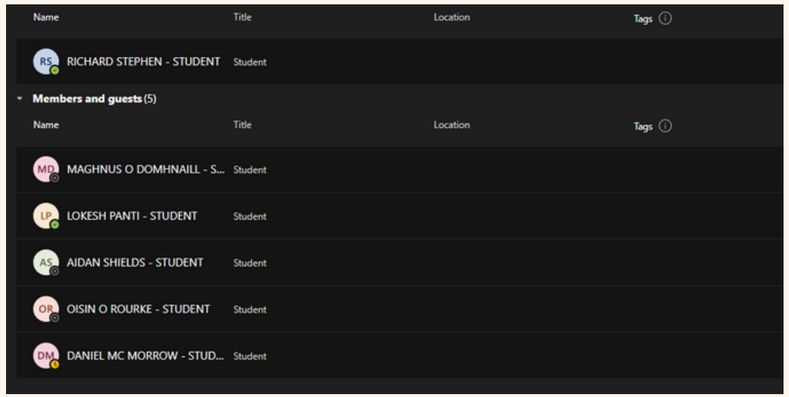
Blender

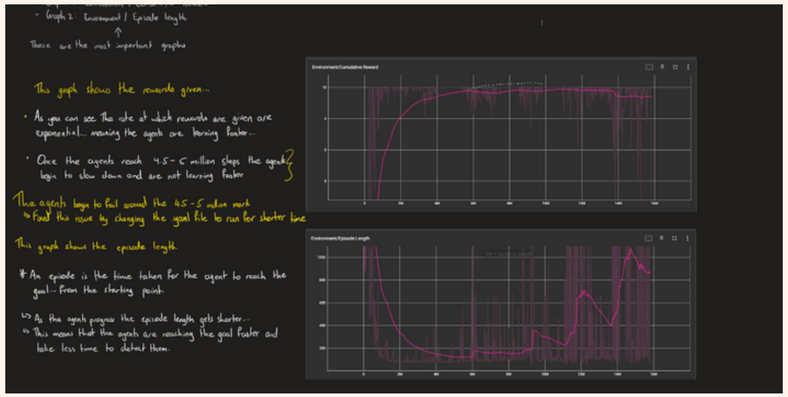
- Blender is a free and open-source 3D creation software used for creating and designing 3D models, animations, and visual effects.
- Supported by a large community, so there wide range of information and tutorials.

Project Planning & Teamwork

Project Planning

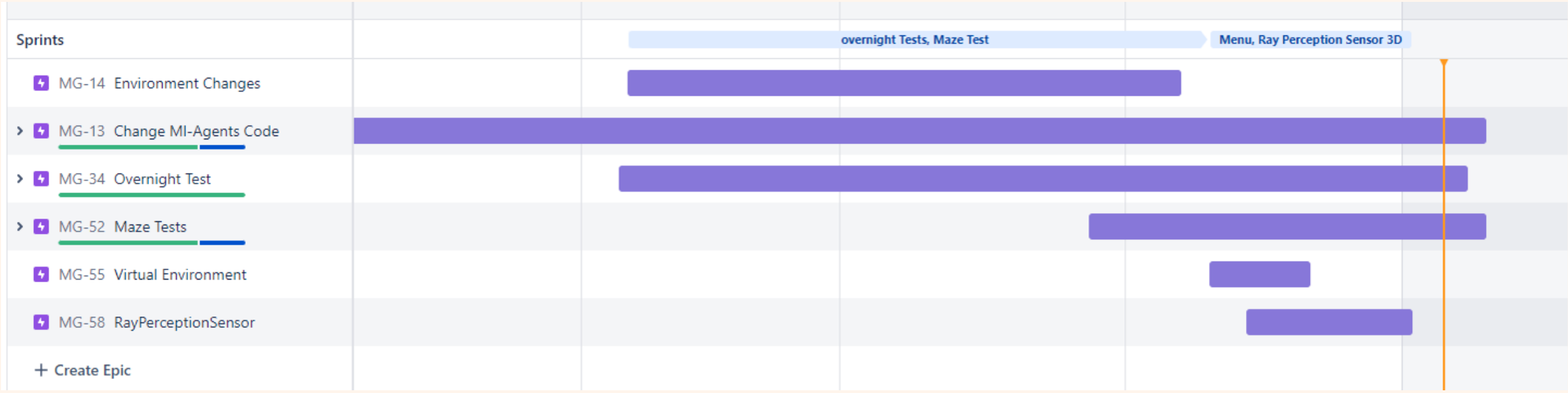
- My project planning was done on Jira.
- I created sprints for ML-Agent test cases, environmental changes, code changes, and any virtual environment changes.
- I also kept a log of the works i have done on my OneNote





Teamwork

- I was part of a stand-up group where we discussed each others projects.
- We also showed each other out timelines on jira.
- Gained feedback on project deliverables. Report, presentation, poster and video.



Maze Generation & Algorithm

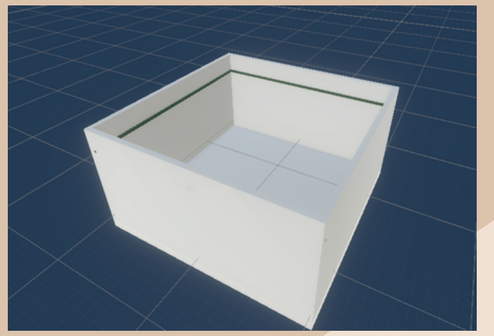
The environment maze is designed using a node, which consists of four walls and a floor as well as a C# algorithm.

The algorithm starts by placing a grid of nodes of a size selected by the player, e.g a 7x7. It then selects a node at random, then it moves to an adjacent node that has not yet been visited. If all adjacent nodes have been visited, it backtracks to the previous node until it finds an unvisited node.

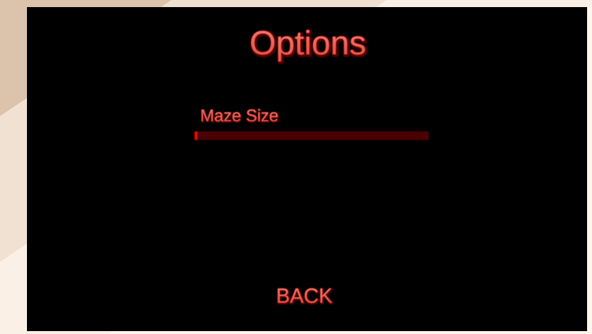
In the process of moving between nodes, the walls connecting these nodes are removed.

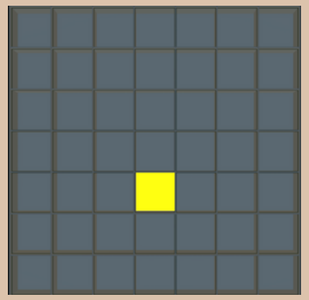
The algorithm is randomizes, which means that the order in which it visits the ndoes and removes walls is random. This results in a unique maze every time it generates.

NODE



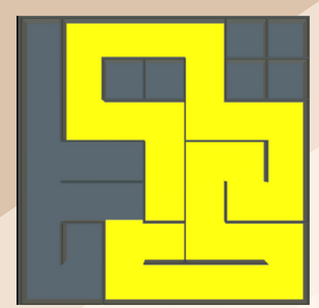
MAZE SIZE SELECT





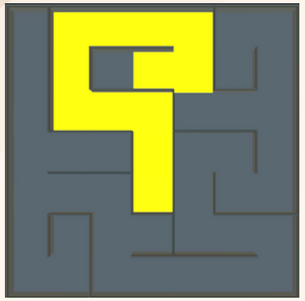
STEP 1

Random node selected as starting node.



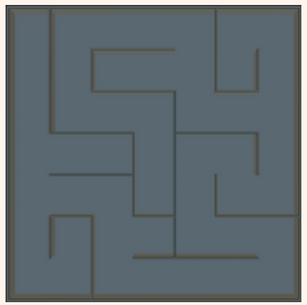
STEP 2

Algorithm removing walls in the grid to make maze path.



STEP 3

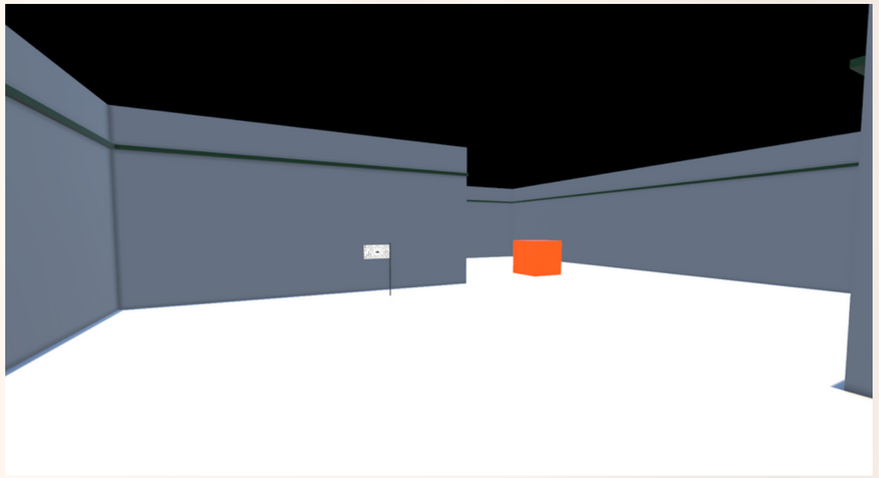
Algorithm backtracking to find available nodes.

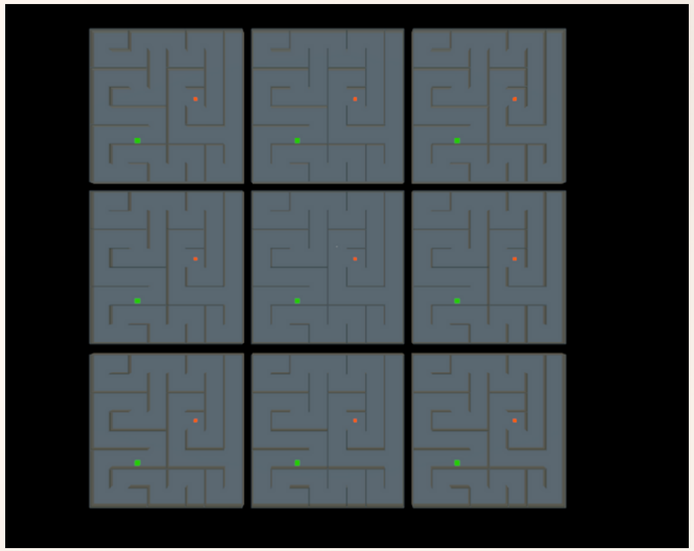


STEP 4

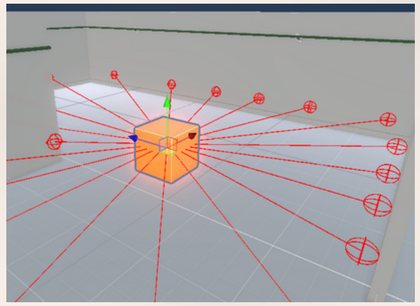
No more available nodes, results in final maze. Random every time.

Machine Learning Agents

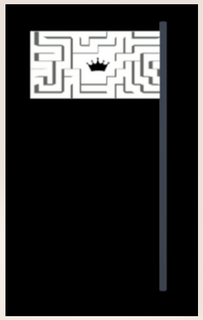




AGENT



FLAG



1 The machine learning agent used in this project are Unity ML-Agents, which utilize deep reinforcement learning.

2 The agents were trained using the exploitation and exploration trade-off method and the Proximal Policy Optimization (PPO) training algorithm, with hyperparameters to optimize performance.

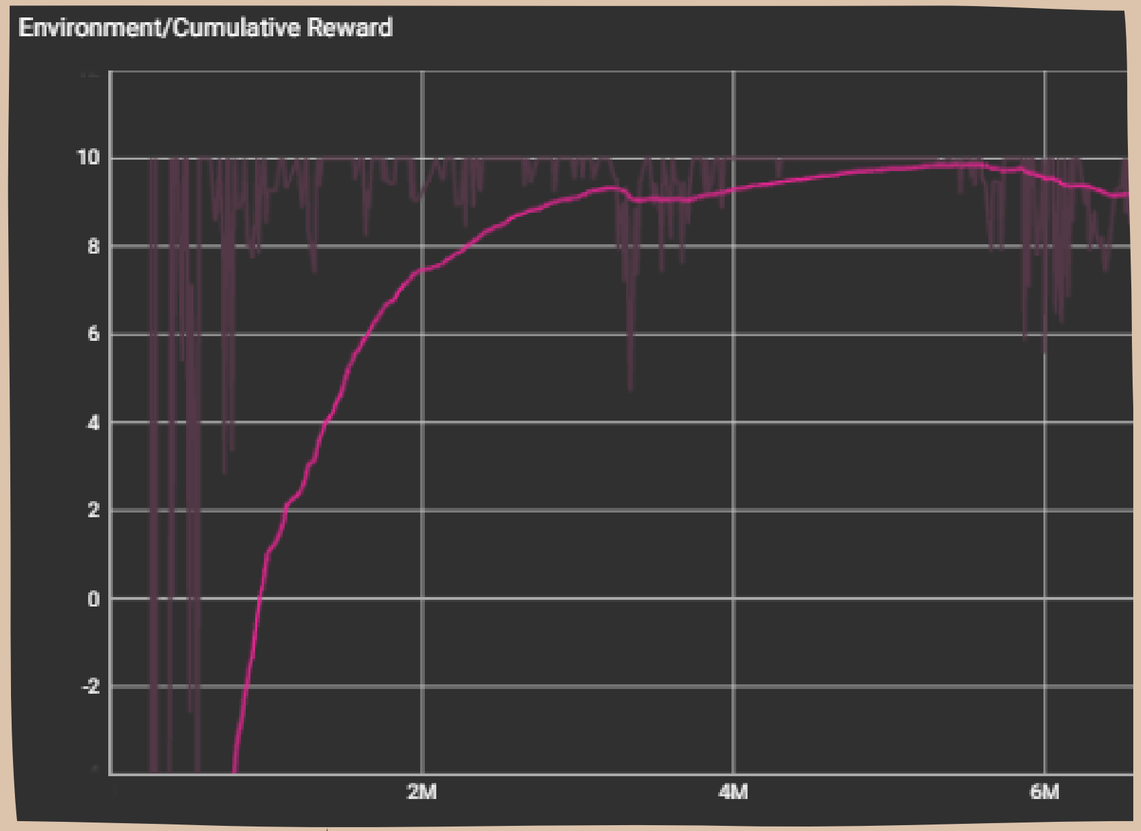
3 They start training on a smaller maze of size 3x3, and as they improved, the difficulty of the maze increased gradually up to a 7x7 size. With every increase in maze size, the mazes became more complex.

4 The agents collected observations about the maze and their surroundings using ray perception sensors.

Agents Results

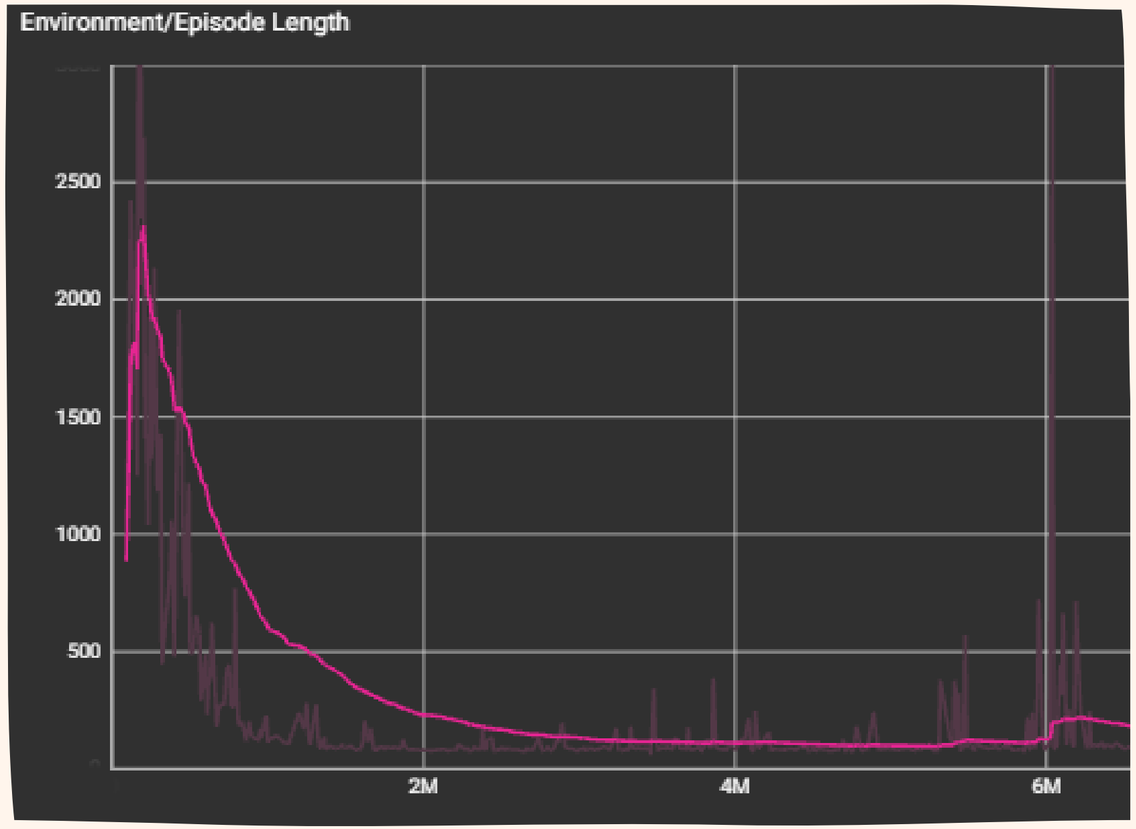
Cumilative Reward

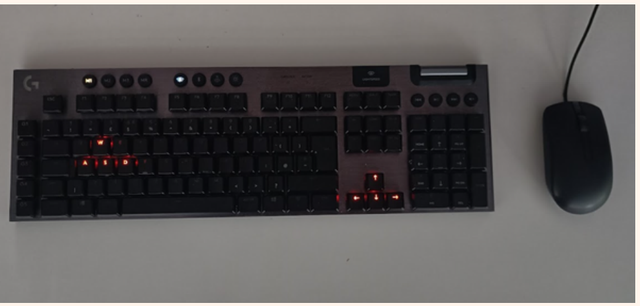
This graph shows the rewards gained by the agents. On the x-axis is the steps taken by the agent, and on the 4-axis the steps taken, and on the y-axis is the rewards.

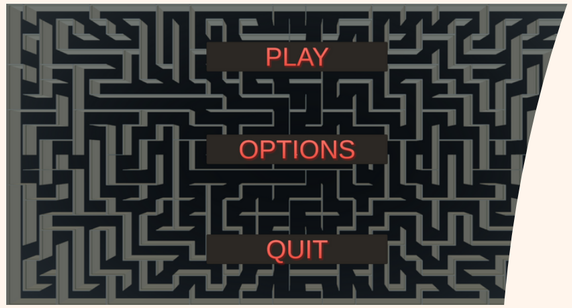


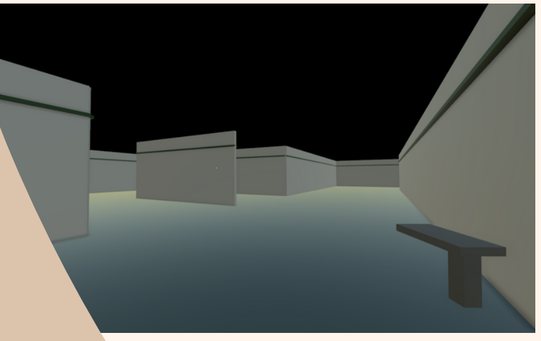
Episode Length

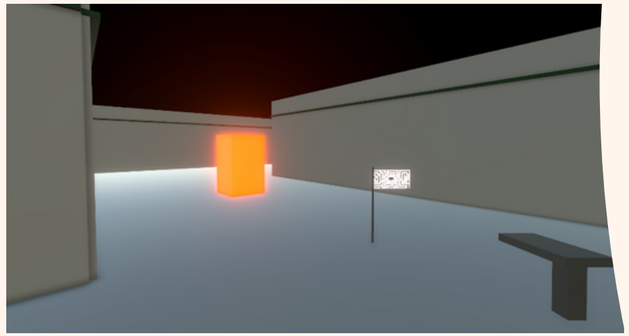
This graph plots the episode length. As you can see the episode length is gradually getting smaller. This means that the agents are taking less time to reach the goal.

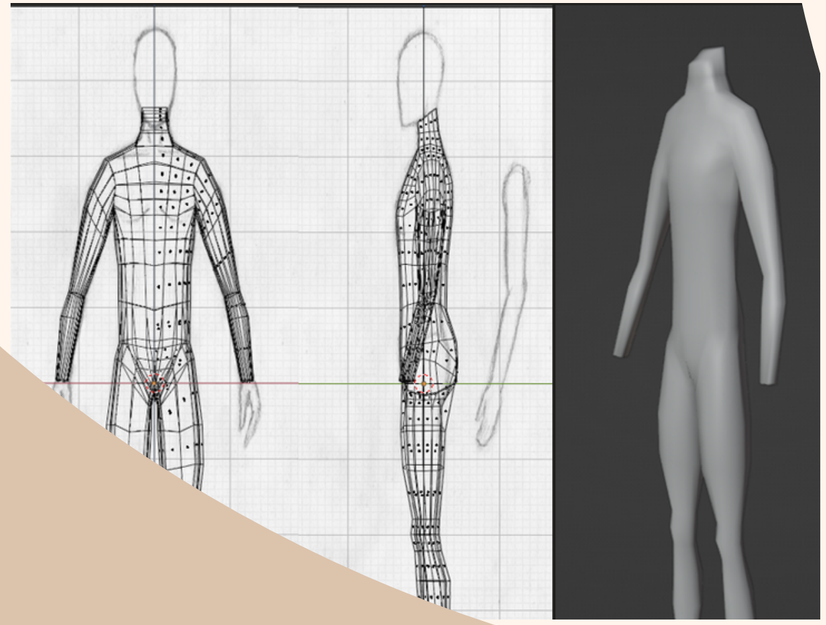












User Interface

The user interface for this project allows users to play the game using a keyboard and mouse to control a player in Unity.

The user also has a gun, which they can use to fight against the agent.

Both the user and the agent are trying to reach the flag first. This allow for competition between them.

Blender

This is a player model that I created in blender to beused in my game.

Using a premade sketch of the human body, I sculpted this model. The hands are replaced by the weapon in game.

The head is the position of the user camera.

THANK YOU

QUESTIONS