

## Soft Assignment

### KU 1

**A\* PathFinding avoidance:** This features is set up with every individual Ai bot and its purpose is to avoid other AI bots.

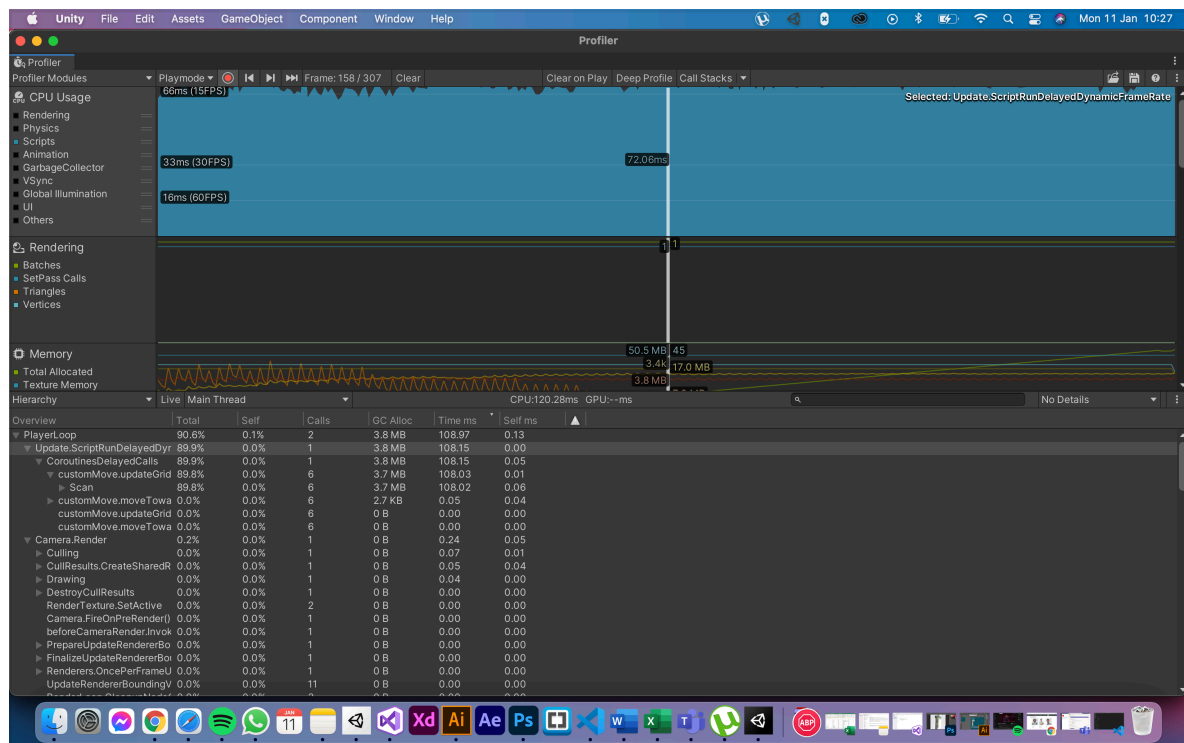
**A Grid Graph:** Examines the entire scene and creates a grid of possible paths that the ai can take. It generates nodes in a grid pattern, the entire grid is walkable and considered. This can be a slow since the pathfinder has to scan the entire scene to create the graph, this can be especially slow when you have a large scene/map.

**Point Graph:** The point graph allows the user to place sever points(waypoints) on a scene and the algorithm will only follow those nodes i.e: Only the selected points are walkable. This type of graph is much faster than a grid graph as it follows just the selected points however if the points aren't place strategically the closes point might be far away from the target or unreachable.

**Navmesh:** This type of graph works in a similar way to the grid graph however instead of generating nodes in a grid(i.e:squares) it generates the grid made out of triangles (i.e triangle mesh). All of the different nodes generated by the triangle mesh are the walkable area for the algorithm. It is usually faster than a grid graph since it creates less nodes since it uses triangles.

### KU 3

According to the unity profilers the larger spikes during the test occurred while the A\* pathfinding allotted was creating the scanning of the graph. This causes more resources to be used as the game engine has to calculate every square in the grid.



## KU5

A search tree is a data structure which has the form of a tree with a root on the top and branches on the left and right of the root node. The values on the left of the tree have to be less than the values of the root node while the values on the right are larger than the root node. The tree keeps on working in the same way for all the values below i.e if the number is smaller than the node it is placed on the left of said node and if the number is larger it is placed on the right of the node.

The Monte Carlo search tree is known as an ai which has the capability of solving open ended games the Monte Carlo search tree also can also learn how to master a task without guidance by programmers. The Monte Carlo search tree was used famously used with reinforcement learning and neural networks to beat the best Chinese GO player. The difference between this algorithm and older algorithms is that this algorithm doesn't analysis every option inside a game tree since the games are open ended . The way the algorithm works is that it chooses a node and creates multiple simulations for every node randomly selecting where it thinks the player will play his next move. After analysing the tree it will check which path is more likely to lead to a win using the Monte Carlo simulation which uses randomness for deterministic problems which are

difficult or impossible to solve using other approaches.

## KU6

- Unity Pro pathfinder makes use of the one graph the navmesh while the A\* pathfinding while offers three different types of graphs.
- A\* pathfinding is used for 2D games while unity pro is mainly used for 3D games however can be adjusted to work with 2D games.
- A\* pathfinding makes use of pre-processing which speeds up the pathfinding algorithm however this does not necessarily mean that the A\* pathfinding algorithm is always faster from the Unity Pro
- A\* gives you the possibility for the programmer to access to pathfinding data such as the nodes and the path, meanwhile unity pro can provide some data using api's however it isn't as accessible as the data provided by A\*
- The A\* pathfinding algorithm also offers Multiple Heuristics which helps the ai to solve a problem quickly while the unity pro doesn't provide any heuristics.
- The A\* pathfinder also offers a path smoother which brings points closer together to create a smooth path there are different smoothing paths available. Unity Pro doesn't provide the path smoother however it does have a funnel modifier which can find the shortest path within the path to make the path cleaner/smoothed.