

Abdallah Said

40099027

COMP-476 Assignment 2 write-up

Introduction

The assignment required us to submit a grid graph with no less than 300 units, it also required us to get a fully animated humanoid model and make the humanoid model use steering behaviors to move from its position to a target position using A* pathfinding algorithm. A* algorithm is implemented with 2 clusters, Manhattan distance heuristic and cluster heuristic.

In my implementation, I made a graph consisting of 2200 grid nodes with equal distance between each of the nodes. I also successfully implemented a sum of steering behaviors required for the assignment which is “path following” delegating to “arrive” and “looking where you’re going”. I successfully implemented the A* algorithm with Manhattan distance heuristic and cluster heuristic with a total of 7 heuristics with a generated table calculating the Manhattan distance between the centers of the heuristics.

Implementation

Assets

For the assignment the humanoid model along with its animations from lab 01 were used for this assignment. The scripts present are 5:

1. Node: which stores the information for each individual node such as position, cost, parent node, and cluster.
2. Cluster: which stores information for each individual cluster such as its number, the distance from it to all the other clusters and its world location.

3. GridManhattan: contains information about the graph such as its size and number of nodes and their sizes. It also contains the functions that initialize the graph, get information about the grid, and world position of nodes. It also uses OnDrawGizmos() to visualize the graph.
4. GridCluster: contains all the information and functions such as the script GridManhattan, but it also contains information about the number of clusters, their layers it also visualizes the graph by giving each cluster a different color using OnDrawGizmos().
5. Pathfinding: is where everything else happens it contains information about the steering behaviours such as velocity, acceleration, etc., it also gets information about the target, NPC, and animator. It also has a boolean called isGridManhattan which if is true initializes gridManhattan and if false initializes gridCluster. It contains the function SteeringBehaviours() which is responsible for the movement of the NPC, it controls the animations based on the speed of the character from walking to running to idle. It contains the 2 functions GetDistanceManhattan() and GetDistanceCluster() which get the distance based on preselected heuristic. It also has the retracePath() which stores the path, openList and closedList in lists of nodes to pass to the OnDrawGizmos() in the preselected grid and display it.

Inspector

To chooses the heuristic:

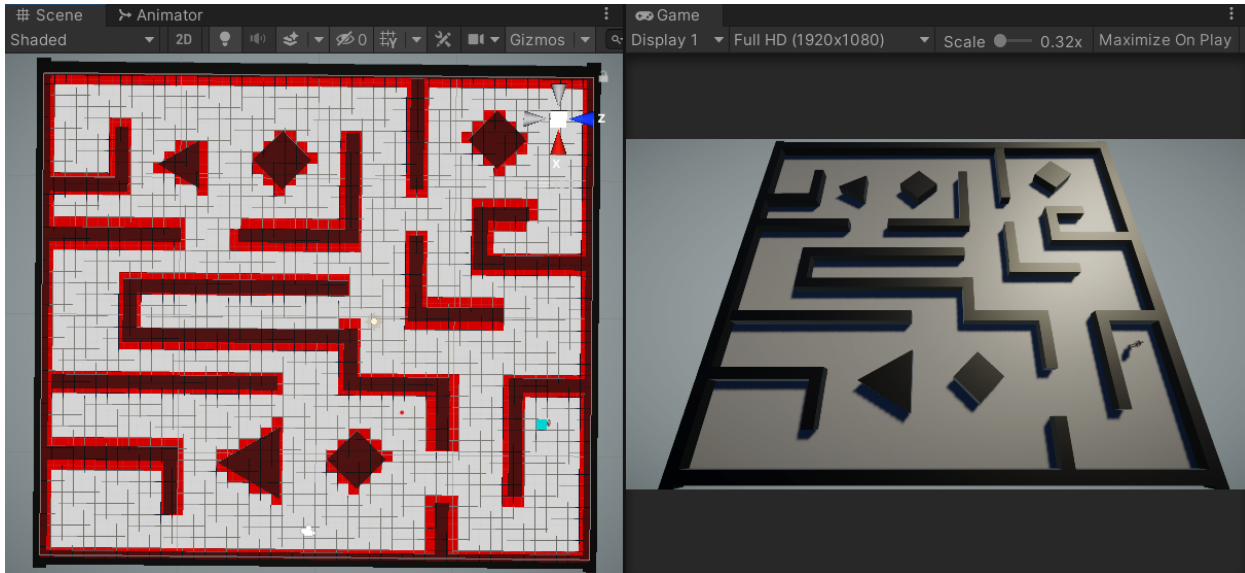
1. Go to the A* object
2. In the pathfinding script attached to the object, enable or disable Is Grid Manhattan depending on whether you want to apply the Manhattan heuristic or cluster heuristic path finding.
3. Depending on which heuristic is chosen, disable the script for the other heuristic's graph script. For example, if you choose Manhattan heuristic disable the Grid cluster script and vice versa.

To fully see the visualization of the algorithm and the game itself put the game and scene view side by side to see both. The only debug shown in the console is the cluster distance table used in the heuristic.

Screenshots

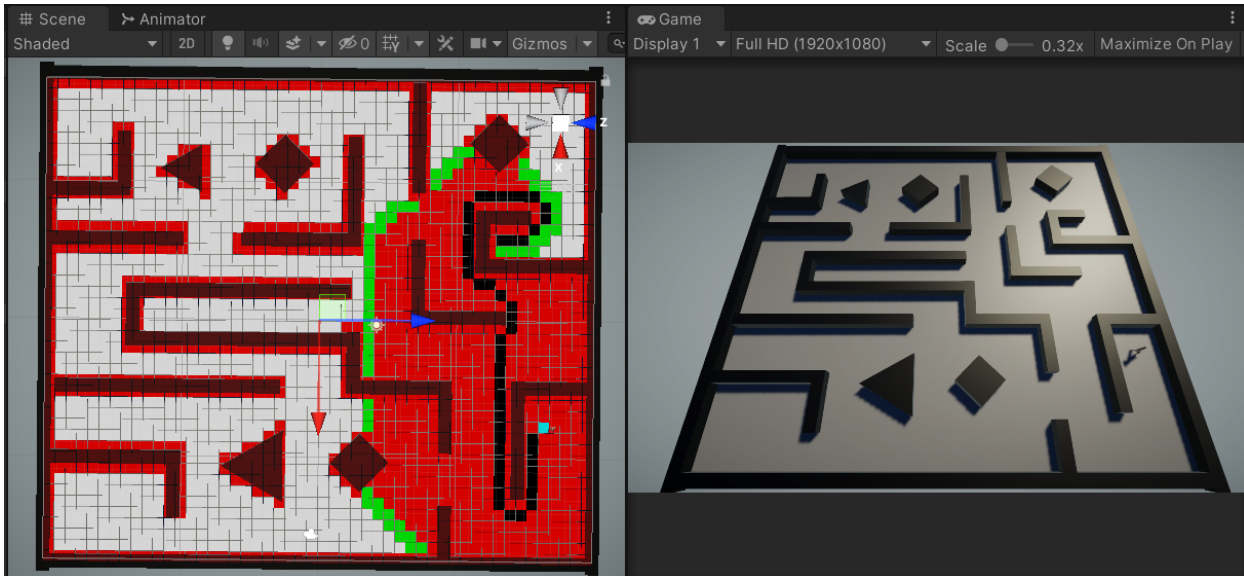
Screenshot of an empty and full Manhattan heuristic visualization.

Manhattan grid with no path:



NPC node color is cyan, unwalkable nodes are red, walkable nodes are white.

Manhattan grid with path:

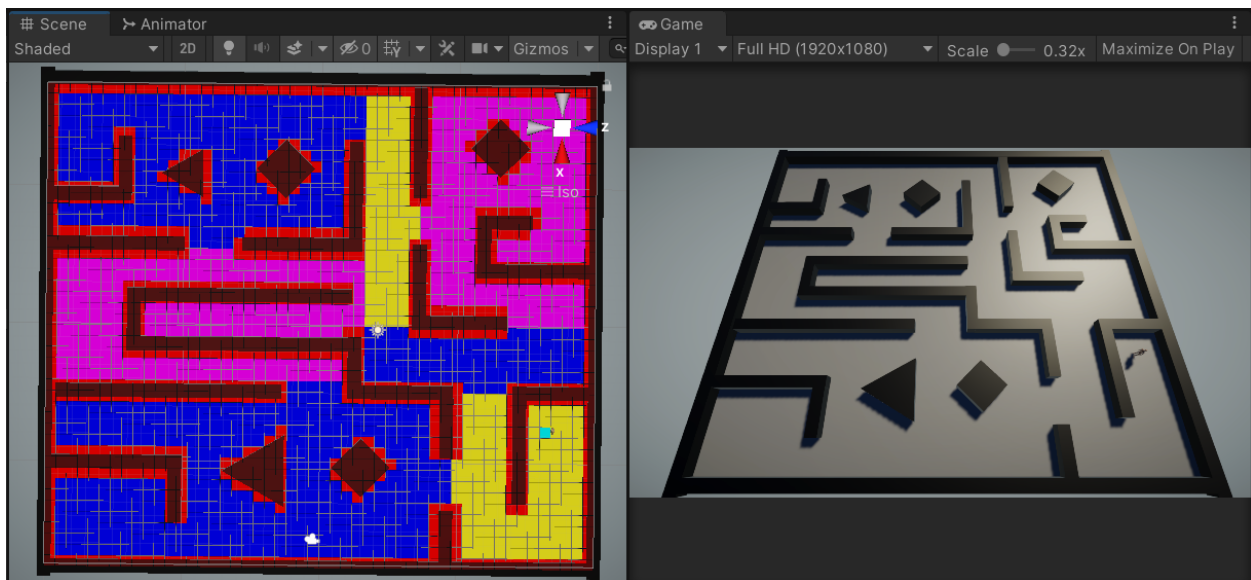


NPC node color is cyan, unwalkable nodes are red, walkable nodes are white, closed nodes are red, open nodes are green and path is black.

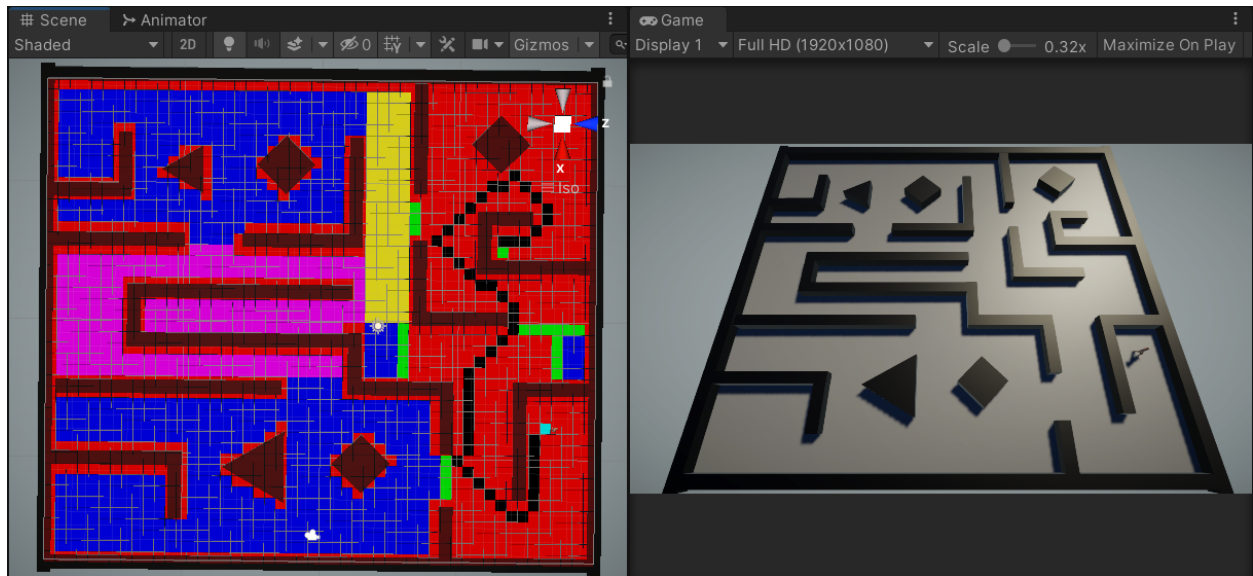
Screenshot of an empty and full cluster heuristic visualization.

Cluster colors are as follows:

1. Blue
2. Yellow
3. Magenta
4. Blue
5. Magenta
6. Yellow
7. Blue



Cluster grid with no path:



Cluster grid with path:

NPC node color is cyan, unwalkable nodes are red, Closed nodes are red, open nodes are green and path is black.