Assignment 2 Report

Code Summary

`part1.unity` has the solution to Part #1 of the assignment, which is a basic "Seek" behavior.

`part2.unity` uses the Unity NavMesh to implement A* pathfinding. Click once to place the player object (the skeleton), and then the second click will draw a line with the shortest path between the player and where you clicked.

`part3.unity` builds upon `part2` but makes it so the player will now actually follow the shortest path to the point you clicked.

Part 1

In part one, I had to do some math to compute the angle that the player should be facing whenever the user designates a new position for the player to move to. I used `Quaternion.Lerp` to smoothly rotate the player as he progresses towards the destination. Then I used `Vector3.MoveTowards` to move the player in the direction of the clicked point progressively each frame.

Part 2

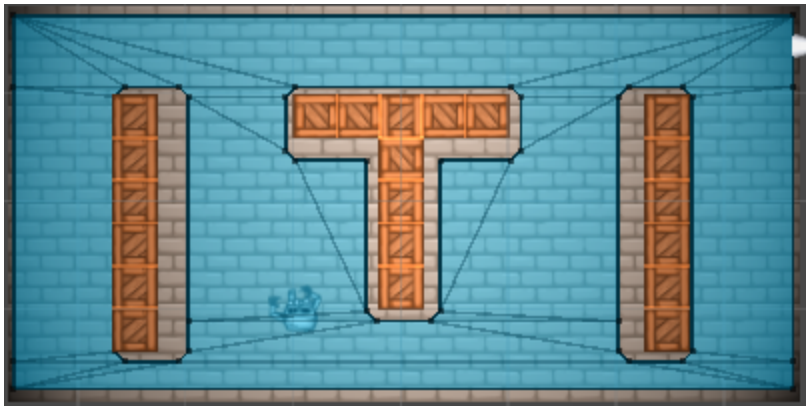
For the A* pathfinding in Parts 2 and 3, I used the Unity NavMesh feature which makes A* pathfinding much easier. I had to convert the 2D sprite world to a partially 3D world, as NavMesh is only supported for 3D games. I used 3D cubes underneath the sprites for the walls, and a 3D plane object underneath the floor which allowed me to bake a NavMesh for the player to traverse.

Once I had the NavMesh set up, I used the LineRenderer to actually draw the line that represents the path between the source (the player) and the destination (the clicked point). NavMesh makes this easy, as I just had to use `.CalculatePath` and it allowed me to get an array of the points along the path from the player to the clicked point. I just had to loop through this array and add these points to the LineRenderer.

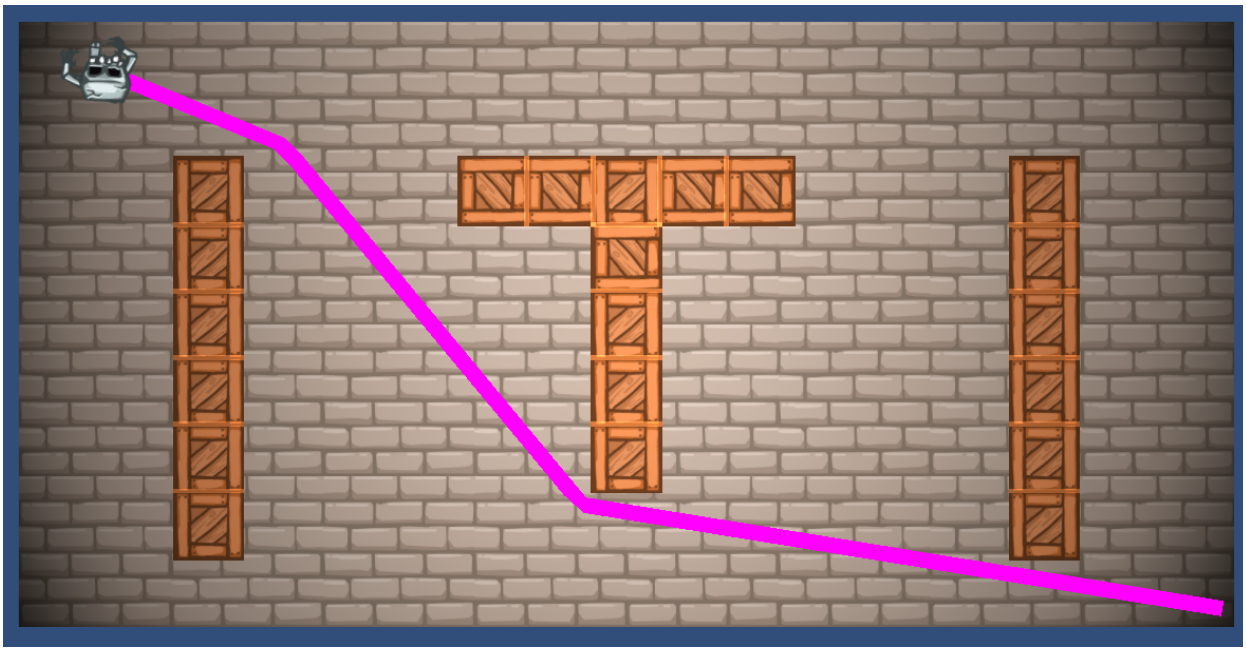
Part 3

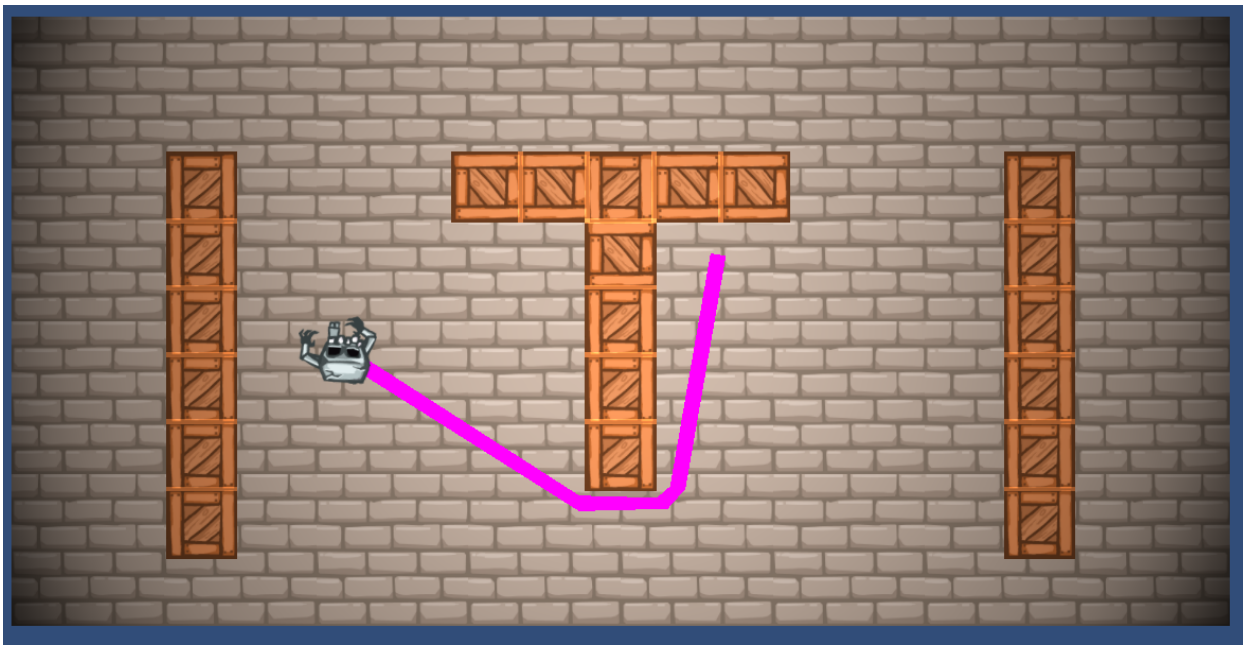
I didn't have to use to much more code than in Part 2 to implement the actual path following. Unity has a `.SetDestination` method which moves the player to the clicked destination. I used this to allow the player to find the shortest path between the start and end nodes.

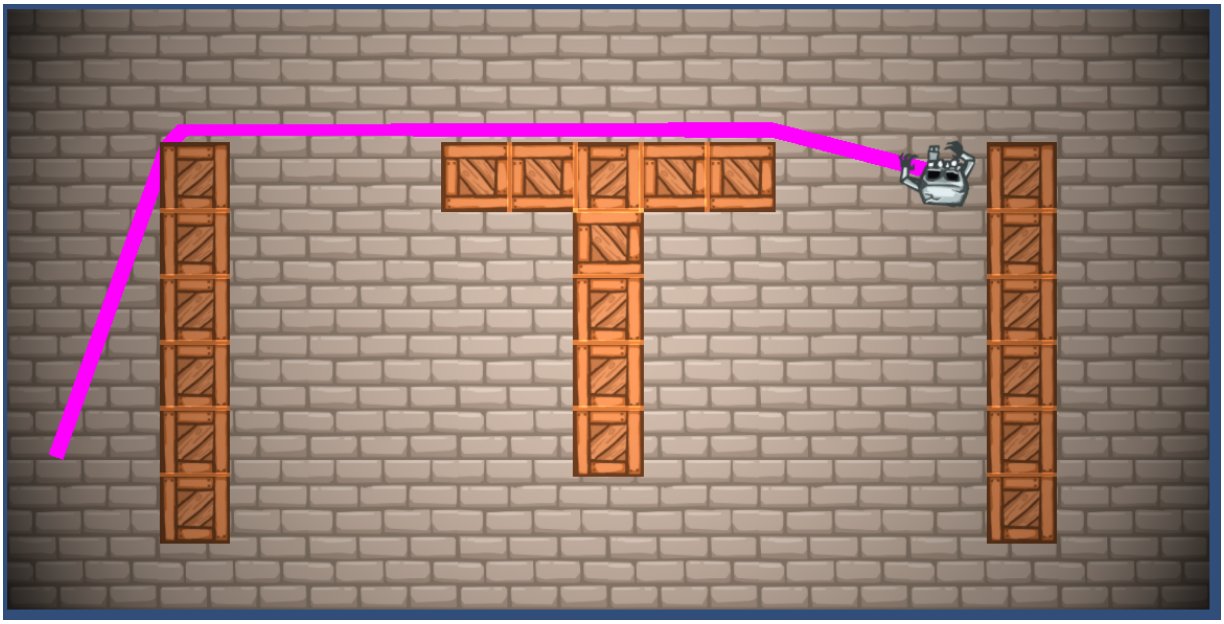
Navigation Map



Example Paths







Lessons Learned

I learned a lot about some of the unique and powerful features of Unity, and also a lot about how A* and pathfinding in general is implemented. I never knew that games often use a pre-baked “Navigation Mesh” that determines which part of the world are walkable or not. It seems like it makes A* search much more manageable, as it does not have to worry about looking for obstacles if the walkable area is predetermined. I also found out how to essentially rig Unity 3D features to work with a 2D game, and I found a lot of the limitations of Unity as a 2D engine.

If I were to complete this assignment again, I would have started earlier, and attempted to implement my own version of A*, without using the NavMesh.