

Basic Threshold 75 pts.	
Total:	75
Begin Default Map (3 defaults)	2
Final Values F(x), H(x), and G(x)	6
Final Path Drawn	2
Final Length & #Steps Displayed	2
Efficency Smooth Anim	2
Control Speed of Animation	2
At Least Size of 500 cells support (size configurable)	4
Configurable Environmental Obstacles placement	4
Goal (x,y) Configurable placement	3
Start (x, y) Configurable placement	3
Provide a Stepped Speed	0
Report w/ screen shots or Video	6
Quality Craftmanship	6
Grading.txt	4
3 Tests for Correctness	3
Readme.txt	2
Demo (MH) + Q/A + youtube video (quality)	10
Game like look (you'll need to craft Sprites & Tiles)	10
Larger than 1k cells	10
Different shaped tiles (line path)	10
Embellishment – capsule moves to target enemy	?
TOTAL	91

This works by implementing a wire frame grid over the map size. The wire frame is then divided into graph nodes. Then the objects on the terrain are placed into a node based off of transform size. The A star algorithm then calculates based on starting size and the current grid world. The a star also has a heap optimization that reduced compute time by up to 100%. There is then a path request manager that will handle all of the astar path requests from the objects so that all path requests are not trying to be computed simultaneously. There is a unit class that is attached to all seeker objects that has a field for the target, then the unit class makes a path to the path request manager class which in turn will request a computation from the a star class. The result of the program is a fully implementable a star path planning for enemies in a unity game. The user is able to modify the speed of the objects and the size of the terrain and any obstacles to any number and any size.

The game will run on play. To modify the size the plane size needs to be changed and the empty game object grid object size needs to be modified to the plane size. Objects can be moved and added. New seeker objects or targets can also be added.