Josh Grazda

EGP-410

Timanus, Jeff

October 23, 2018

Assignment 3

Solution Architecture:

The architecture of my code follows many of the previously implemented code in the assignment with some additions to increase efficiency. To achieve the container, I needed for both Dijkstra and A*, I took the built-in priority queue and added some additional functions to it to increases it efficiency over a normal list solution. This not increased the performance of my pathfinding, but also structured my code in a easy to read format, using my function calls as a way to follow what the code is doing.

Additionally, I structured many of my classes for searching algorithms similarly, so they could be easily understood and see the differences in each without having to look to hard. This made it easier to implement as well as debug later during the projects cycle. Many of the necessary operator overloads for my NodeRecord class also allowed my code to remain clean and easy to understand.

I implemented a similar messaging system from the previous assignment to create messages based off input from the user. Using this same structure, I was able to allow clean communication between scripts without causing to much coupling. Many of the classes built in have also been given class guards, which were previously missing from my solution, which allowed me to fix some issues which were previously hidden due to lack of protection.

Overall the state of the architecture tries to maintain a clean set of code to read, which came from the AI for Games textbook used in class. This book provided a structure to create both searching algorithms, and with the addition the containers created, I was able to follow it very closely to come up with my solution to this assignment.