Voronoi Graph Path Planning

To: Dr. Beard
From: Derek Knowles
Date: April 16th, 2019
Subject: Project Description
Appendix A: Project Source Code

For my project, I implemented the Voronoi Graph path planner from Chapter 12.

In order to make my project relevant in simulation, I created a world with pyqtgraph that randomly placed buildings in a larger area than the template files. Voronoi graphs aren't very interesting if the obstacles are in a perfect grid.

First for the Voronoi graph, I calculated the vanilla Voronoi graph with the center of the buildings as the obstacles. Next, I added edges to the three closest vertices to the start and end nodes. I then computed the cost of each line which takes into account the length of the path (fuel efficiency) and how close the path comes to a building (obstacle avoidance cost). I used a heap structure to Dijkstra's search the possible paths and return the least cost path.

I found this algorithm to work well to avoid obstacles as much as possible, but fails when there are few obstacles, a path more optimized for fuel efficiency is desired, or obstacle sizes vary.

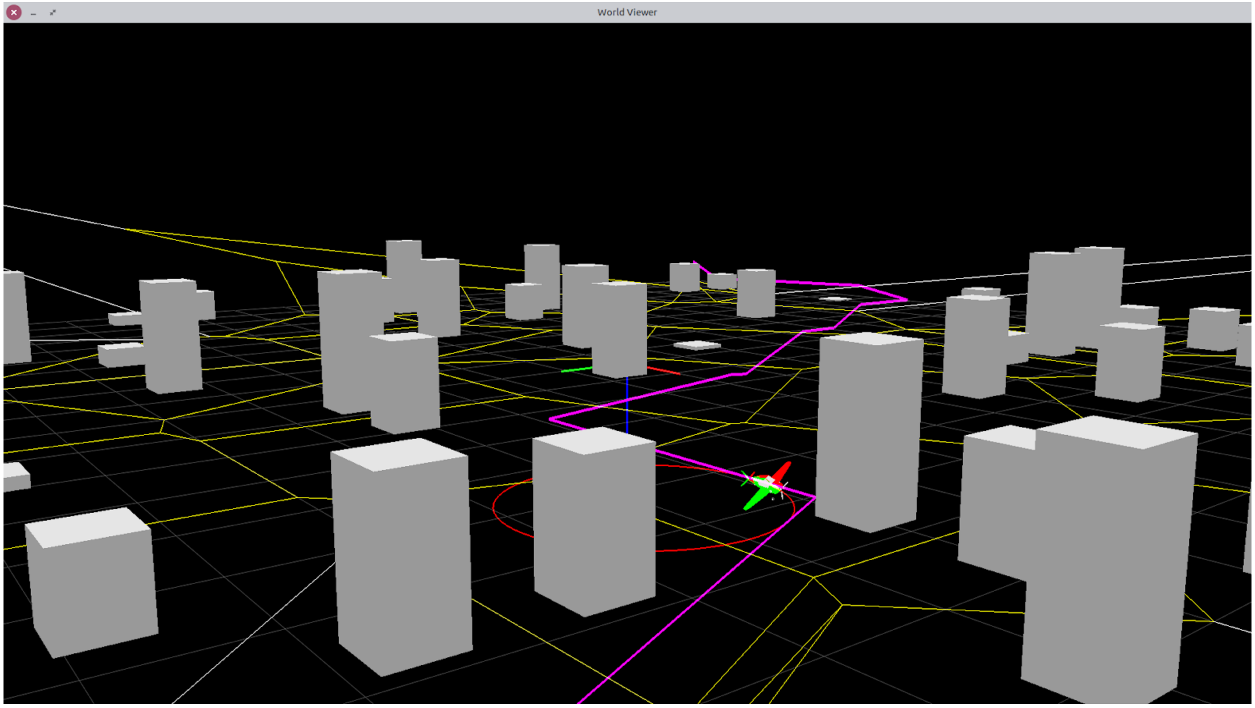

Figure 1. UAV flying through city on Voronoi graph planned path

Appendix A: Project Source Code

https://github.com/betaBison/EC-EN-674-Flight-Dynamics-Controls/tree/master/project