Benjamin Schreyer Project 6 Writeup 12/13/2020

**Design and Planning:**

Graph Structure:

The graph structure will be designed to be easily useable for use in Dijkstra’s Algorithm and the required interfacing methods required by the project. An array list will hold references to vertices and edges that are also linked by reference to other cities in a graph.

Dijkstra’s Algorithm:

This will be implemented in the manager class and will be similar to the Java Dijkstra’s Algorithm I wrote for the lab we completed recently. Extra code will need to be written to specifically target locations as specified in the project. The algorithm will also need to be adapted to the data structures created for this lab. The algorithm will also possibly be faster given that the requirements only require the shortest distance between two points, not the shortest distance from one point to all other points.

**Test Plan:**

Testing the basic graph elements Road and Town or vertex and edge is just constructing objects and testing get and set methods.

The Graph class needs to be tested for construction, vertex and edge removal/addition, finding edges and vertices by name etc, and the shortest path algorithm implementation. A graph will be constructed at test start with vertices and edges and it will be modified as needed to test methods like addEdge.

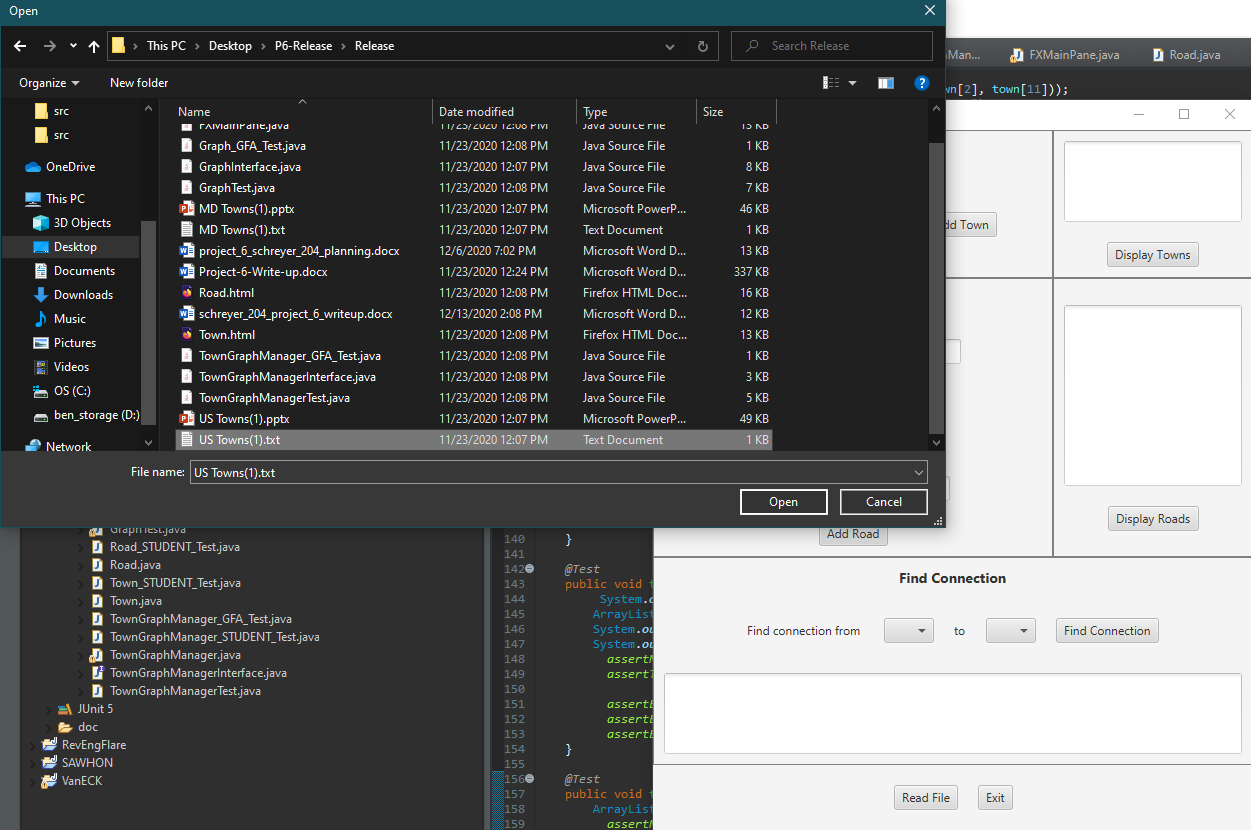
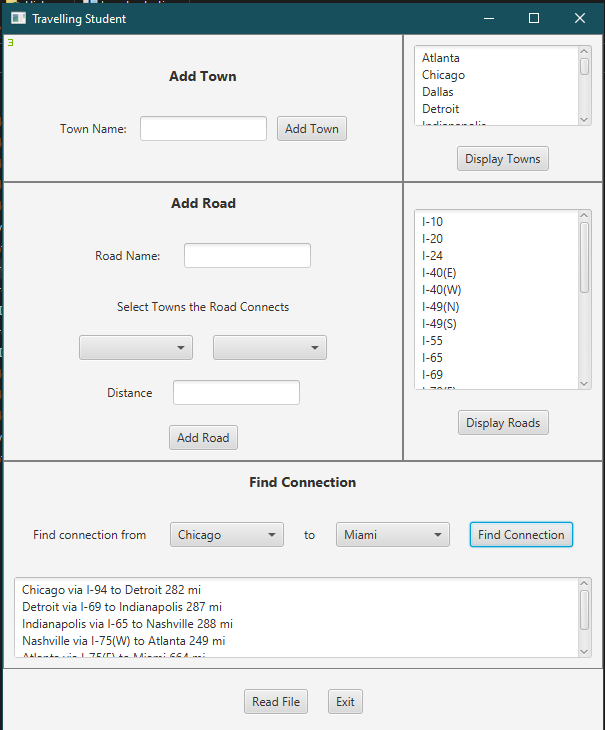
The Town Graph Manager class will be tested in a similar way to the Graph class. The main difference will be testing access of nodes by string name instead of object reference.

All class testing will be written as JUnit tests along with the provided JUnit tests.

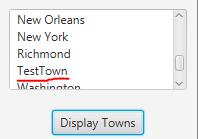
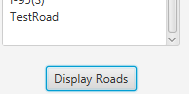
The GUI needs to be tested to make sure all the button elements are correctly linked to functions that work.

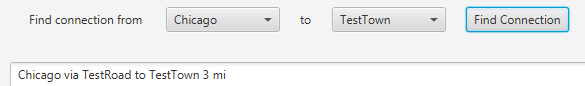
**GUI Testing:**

Testing loading a map from a file and using the data loaded, the results indicate it works.



Adding a Town using the interface works. Adding a Road between Chicago and TestTown works.



**What I Learned:**

To complete this project I had to learn how to connect an abstract graph class with a manager class that deals with real objects (roads and towns).

I also had to deepen my understanding of Dijkstra’s algorithm because my implementation that I brought over from a previous lab was not perfect. Adding one line to this implementation fixed it but it was not a trivial line of code, removing a vertex from the visited set if a new shorter path to it was found fixed the failure of my code on one of the shortest path test case.