1. (20%) Create a design **before** you start coding that describes or shows how a graph structure could be used to store some kinds of data and attempt to solve some kind of problem (yes, this can be a game that needs a graph to represent a map!),

A screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer program

Description automatically generated with medium confidence

1. (20%) Create some tests (at least **two** for each piece of functionality) **before** you start coding...

Tests were all written in design.md as seen above. Tests implemented in test.cpp- here is a snip of some of them:

A picture containing text, screenshot

Description automatically generated

1. (40%) Implement a graph class with at least (this category effectively combines implementation and specification, partly to emphasize getting the algorithms working!):
   1. (5%) a function to add a new vertex to the graph (perhaps add\_vertex(vertex\_name)),

A picture containing text, screenshot, software

Description automatically generated

* 1. (5%) a function to add a new edge between two vertices of the graph (perhaps add\_edge(source, destination) or source.add\_edge(destination)),

A picture containing text, screenshot, software

Description automatically generated

* 1. (15%) a function for a shortest path algorithm (perhaps shortest\_path(source, destination)),

A screen shot of a computer code

Description automatically generated with low confidence

Shortest path helpers:

A screen shot of a computer program

Description automatically generated with low confidence

* 1. (15%) a function for a minimum spanning tree algorithm (example min\_span\_tree()).

A screen shot of a computer code

Description automatically generated with low confidence

Minimum spanning tree helpers

A screenshot of a computer program

Description automatically generated with low confidence

1. (10%) Analyze the complexity of all of your graph behaviors (effectively a part of our documentation for grading purposes),

I put a comment next to each function declaration in my header files. For example

A screen shot of a computer program

Description automatically generated with low confidence

The same is true for both graph.h and graph\_node.h.