Enrico Blackwell

I want to create a little game about buying and selling stocks on a very fluid stock market while traveling to many different cities. I want to use a graph to create a map of the cities and connect them with weighted edges and use a vector to sort the player's stocks. The city will be a class with variables like where they are connected and the certain stocks they sell. The stocks will most likely be structs with the price of buying/selling and how many the player has.

City struct

- Has a vector of stocks that they are currently selling
- Has an adjacency list of nearby cities the player can travel to

Stock struct

- Name and abbreviation
- How many the player has bought
- Price of individual stock
- Queue of 5 day forecast

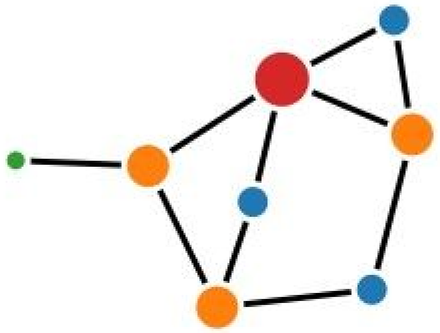
Player class

- Vector of stocks
- How much money the player has
- Functions to travel, buy, sell, and wait

Game class

- Keeps track of how many days have passed
- Generates the stock numbers for the next 5 days
- Has all the other classes inside of it
- Also keeps track of the map and cities using a weighted, undirected graph

The game class is a personal choice because I could just make all of those functions within the other classes, but they don't really fit anywhere else. I also would like a general place to store all the other variables that I don't want to be floating around main(). I will be using dijkstra's algorithm to find the best way for the player to travel to a city using the shortest path by days required.



8 cities with 10 roads:

The player wins by earning $1,000,000 through buying and selling the stocks. This game will be really hard, but very rewarding once you get the hang of it.