**Cops and Robbers Write-up**

1. Our cops and robbers game requires a graph to connect all the neighboring cities that the character can move to. We have many instances of two-way connections, as well as a few one-way connections. Each location holds a dollar amount available for the robber to steal each time he moves to that specific location. We use breadth-first search in our game to find where the robber is because he is hidden from the user. It is important to use breadth-first searching because it is the most efficient way to find where the robber is from the cop’s current location in the network. When implementing this method, we check to see if the cop is at the same location as the robber. If they are, we know that the game is over, and the user has won the game. If not, we know that the user must keep trying to move around the graph to find the robber.
2. We both worked on the backend implementation of the project. We met together for the entirety of creating the backend and collaborated for everything non-swing related. Andrew developed a stronger understanding for how swing works and implemented more of that portion. We both collaborated on the design on how the game screen should look and operated.

Andrew – 60% Grant – 40%

Part 3

* We implemented the basic functionality for all requirements within the driver class.
* We implemented real game functionality for the following:
  + When the user moves cop using the commands the robber also is moved randomly
  + When the cop moved to a new location we use Breadth-first search to check if the cop has found the robber.
  + The driver exits if the robber is found.
  + If the user decides to send their cop to jail, then they will be stuck in that location because jail is a dead end with no outward connections.

Part 4

* The purpose of our breadth-First search is to find where the robber’s location is because he randomly moves locations and is hidden from the user.
* When implementing this method, we used a queue and a linked list to track the seen locations.
* We deque if the queue is not empty and return the location.
* While iterating through the currentConnections list we see if the location has already been seen and if it hasn’t, we add it to the queue and add it to the seen list.
* We use this method in our isRobberFound method to see if the cop is at the same location as the robber.
* Using breadth-first search is most efficient way of finding where the robber’s location is based off of the cop’s location.

Part 5

* We implemented swing into our project to make is more user friendly and more visually appealing.
* We created a starting interface where the user can start a new game or resume the last saved game.
* After pressing the start or resume button a new window appears which puts the user in the game.
* The main game screen has an image of Minnesota as the background to show the relative distance between the various locations which are buttons on the screen.
* This screen has updating text fields for the current location of the cop, the current clue if there is one, and the turn number.
* In the top right-hand corner, there is a save button. When clicked, the game will be saved to a file and available to be resumed by pressing the resume button on the start screen the next time the user runs the game.
* From the current location of the cop, the buttons for the connected locations and the current location are activated and are available for the user to press to move the cop.
* When the cop moves to the same location as the robber the cop has won the game and the user is directed to an ending screen.
* The end screen shows the amount of turn the user took to find the robber. It shows the dollar amount the robber stole. It also has a restart button that allows the user to set up a new game to play.

1. Some limitations of our game are that we wanted to store a high score list where the user could compare themself to past games and see if they caught the robber in a lower number of turns or the lowest amount of money stolen. Another limitation is that our game only has one robber. We were originally going to add multiple robbers and have the cop catch all the robbers. Also, we were planning on adding more pictures throughout the gui; however, due to time constraints and swing glitches we decided to only add the MN Map.