Planning Document

We will be using test-driven coding for this project. Our first set of tests will be over loading the map and location files. These will test the basic file-not-found errors, as well as testing that the correct information was retrieved, and in the correct format. Our next set of tests will be for the searchers. These will test that each searcher is classified in the correct way, that they are all loaded properly, and that the given identifying information has been properly stored for each one. The next tests will be over the grid creation. These will check to make sure that the grid is created properly, and that every searcher is put into the inputted starting place. The next set of tests will be over the image creation. These will make sure the image loads correctly, that the grid and map match up, and that each searcher is given the correct icon. The next set of tests will be over the searcher movement. These will test that each searcher moves in the predicted fashion, based on default directions and speeds. They will also test the manual movement inputs for the dogs, as well as the manual override for all the other inputs. The final set of tests will test that all the visited places are recorded properly.

Our development strategy is to start with the basics. For part one, we plan on first writing all of the basic tests. Then, we will be creating the main class, followed by the children classes that hold the different types of classes. These will include their symbols and outlines for predicted movements and the manual movements. This will be followed by create the operating class, with just enough function to create the map in code. We will also load the map and grid dimensions. Finally, we will create the GUI image, with all of the pieces in their correct starting locations. Part two will have us updating the tests to check for case specific problems and issues. Then we will create the timer function, and make sure it updates correctly. Implementing the predicted motions for each searcher will be next, each one syncing with the timer function. We will then code in the manual input functionality for each of the searcher types, and code in the recording system for the previously visited spaces. Making the GUI refresh after each movement, including shading previously visited spaces will come next. We will finally check the while thing, making sure that there are no missed issues and that everything functions with a variety of starting conditions.