**Assessment 2 – Planning for drunks**

This assessment is an agent based model (ABM) which aims to show where drunks walk when leaving a pub and trying to get home.

The file used for this project is a single 300x300 raster file which contains a point for the pub and points for each drunk’s home.

The first necessary step was to bring in the raster file into my python file in order to find out the pub point and the home points for each drunk. Bringing in the txt file was relatively straight forward, but retrieving the relevant information from the file took some time. After some trial and error, I managed to convert the raster file into a 2D list which then made it possible to loop through each row individually, meaning it was possible to identify which row had the values of interest in, such as the pub, which was denoted by ones. Once the appropriate row and value were found, it was then possible to create a variable for the pub based on the appropriate row and values. This technique however to loop through the 2D list to find the values of interest was time consuming and not very efficient, as the values of interest, such as the 1’s were present in more than just one row. It is likely that there is a more time efficient way to do this, however I was uncertain how to do it.

Once the pub had been identified it was necessary to create a blank drunk list and create a loop appending each drunk to the pub value as a starting position for each drunk. These drunks were then plotted onto a scatter graph, shown in Figure 1 below.

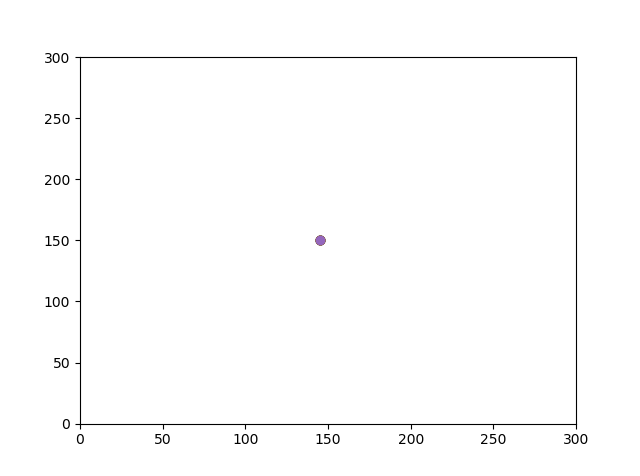


Figure 1 – Scatterplot showing drunks

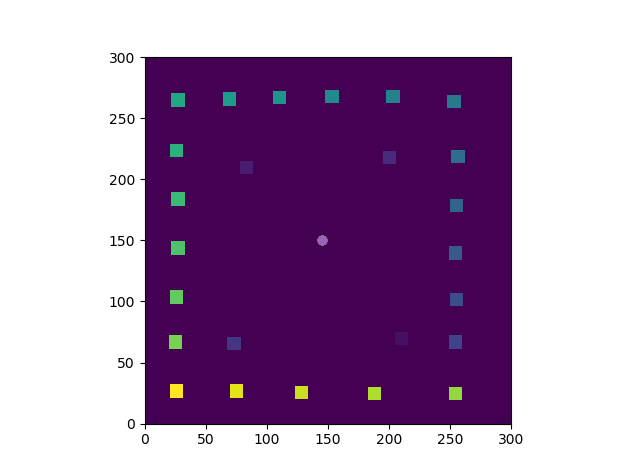
Figure 1 shows the drunks in a 300x300 grid, however what would be much more useful would be to see the drunks in the area file which contains points for the pub and each drunks house. Therefore, a new scatterplot was printed showing the environment. 

Figure 2 – Scatterplot showing the environment

Figure 2 shows where the drunks are starting from in relation to each drunk’s home.

The next step was to see if I could make each drunk take a random step, for this I used the random function and then printed each drunk once again as well as plotting them once again on the scatterplot to check the drunks had moved, as shown in Figure 3.

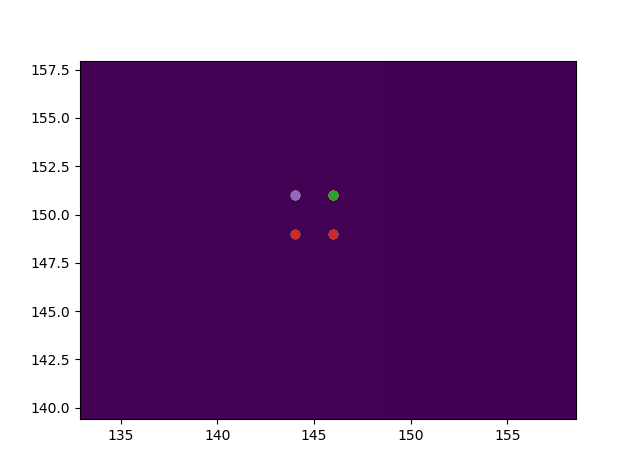


Figure 3 – Drunks taken an individual step.

This confirmed that each agent had moved a step and the random function was working as intended. The next step was calculating the distance between each drunk, allowing agents to interact with each other, although at present each drunk has only moved one step so at the moment the distance has been calculated for only two drunks, to check the coding works as expected.

The program aims to move each individual drunk so many times until they reach their home, therefore to check I could move an individual drunk I moved one drunk 1000 times, then plotted the results in the scatterplot, as shown in Figure 4.

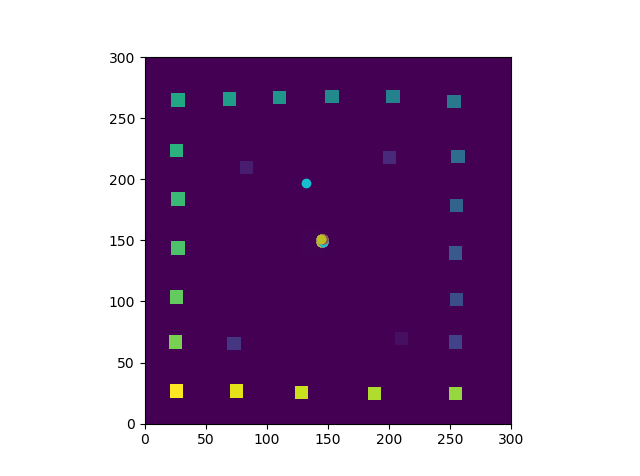
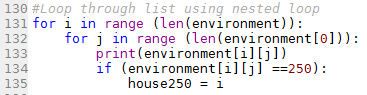


Figure 4 – Individual drunk moving 1000 random steps.

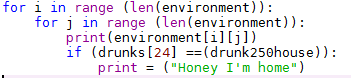
Figure 4 shows that one drunk has successfully moved 1000 random steps. It should also be noted that I didn’t want any drunks to leave the environment and go off the grid, as I wanted each drunk to interact with the environment, therefore I used the common solution ‘Torus’ which allows the drunks to come back into the environment from the opposite side, meaning they will always be interacting with the environment.

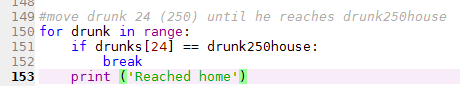
Next, I wanted to create a variable for each drunk’s home, however this resulted in the same issue which I encountered when creating a variable for the pub values, with the values for each home (10-250) being present in multiple rows. I tried some other methods which I learned through the lectures/practical, but was unable to get the result I wanted. One technique I tried was looping through the 2D dataset using a nested loop, and trying the “if” statement, to get the value of interest, however this was unsuccessful. The code is shown below:



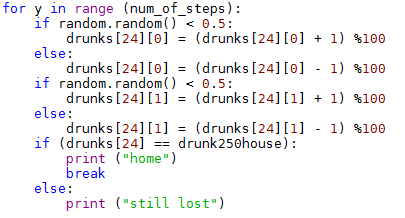
Due to this, I had to assign the values for that particular home manually. I firstly created a variable for drunk250 home to test I could get the first drunk to walk to this house rather than just walking a certain number of steps. This proved to be difficult as I struggled to workout how to tell the program to move a particular drunk until it reached the drunk250 home variable.

I tried several pieces of code, including:

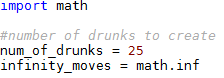




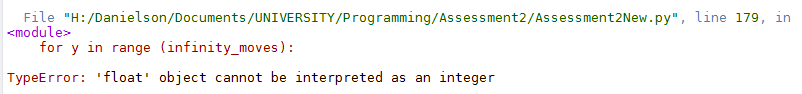
I realised that this would not work as I wasn’t actually moving the agents in the above code, so I tried the following to attempt to move the drunk so many times until he reached the right home:



This worked in a sense that the console displayed “still lost”, but it never showed “home”, so I attempted to use an infinite number, rather than number\_of\_steps, which was set to 1000.



This however did not work:



So I increased the num\_of\_steps variable to 10000, then to 100000, but this still didn’t work, with only “still lost” being displayed.

This issue restricted me from taking the program any further, therefore I was forced to leave the program as it was.

**The following shows the algorithm I was attempting to do.**

#Read in the drunk raster file

#Convert the raster file into a 2D list

#Print the raster file, displaying the pub and houses

#Loop through the environment to find the values of interest

#Build a drunks list and populate it

#Move a drunk randomly from the pub

#Move a drunk from the pub to their house, recording the environment values as it moves based on the distance I define

#Do this for all agents, storing how many drunks pass through each point on the map

#Plot the drunk’s movement using animation, to show the density of drunks passing through each point on the map