Assignment 2

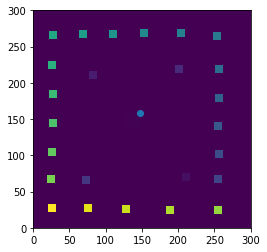
In this assignment the objective is to produce an ABM program in Python which in summary, models 25 drunks leaving a pub and finding their way home. The IDE Spyder was used to compile and debug the code, all outputs ans Python code is loaded to Git Hub at [**cman2000.github.io**](https://github.com/cman2000/cman2000.github.io).

Core objectives I have tried to achieve with the program include:

1. Pull in the data file and finds out the pub point and the home points.
2. Draws the pub and homes on the screen.
3. Models the drunks leaving their pub and reaching their homes, and stores how many drunks pass through each point on the map.
4. Draws the density of drunks passing through each point on a map.
5. Saves the density map to a file as text.

The first stage required data, provided, to be imported into the program. The file called ‘drunk.plan’ was imported using a csv reader. The data import was then inserted into a new list named ‘drunkworld’, this is the primary environment in which agents (drunks) move around in (Figure 1)

Figure 1 House and pub locations



The environment is 300 by 300 cells/pixels and contains the house and pub locations. (Figure 1)

Also presented, on Figure 1, as a scatter plot point, is a specific cell location for the pub – represented by cells containing 1’s. For this assignment I decided to make the drunks start/exit point of the pub the last coordinate in the range of values – (148,158). This was obtained using a for loop with enumerate.

I’m sure the first few elements here, import and pub coordinate value could have been consolidated into a single function, however I have chosen to use separate functions, as I could get them to work.

The program then defines how to move each drunk, when they reach home, and how to loop through all 25 drunks.

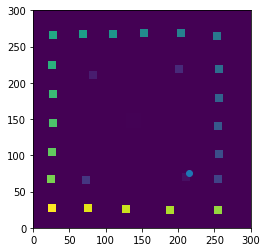
The ‘move\_drunk’ functions asks for a single random move in x and y direction, while also ensuring movement off the end of the grid results in a flow back on at a different end.

The key piece of code here is in identifying when the drink is home. An if else statement is used in the ‘is\_drunk\_home function’ to match the house number and movement coordinates of the drunk to ascertain if condition is true.

Using both functions above, taking the number of drunks, a ‘for loop’ and ‘while not’ will continue to move a drunk until the drunk world value equals the house number value.

Figure 2 Drunk 10 arrival home in drunkworld

Drunk with house number [10] Has arrived at home on coordinate [215, 75]



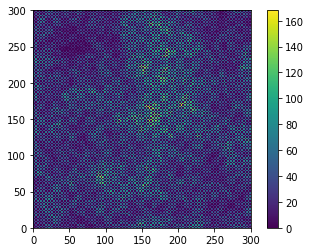
Drunk with house number [10] Has arrived at home on coordinate [215, 75]

The final part is getting the program to provide information on where drunks have been, both as data and presented as a density map.

An initial issue that came up was how to avoid changing values in drunkworld cells without affecting ABM calculations and movements by changing the source values. The answer was to make a mirror environment of 0’s (in the first function) which can be used to separately present the added values from drunks.

‘Drunk density.png’ (Figure 3) - image output showing colour shaded density of number of drunk interactions at any given cell.

Figure 3 Drunk travel history density



‘Drunk density.txt’ – a text file (csv) output of numbers of drunk interactions with any given cell, on route to finding home.

These outputs provide the main evidence of drunk movements and the model, however reference Figure 2 I would ideally want to place all final drunk coordinates on the same scatter plot, unfortunately I could not get the syntax or code just right to achieve this, therefor I had set the code to present a plot for each drunk endpoint with reference to house umber and final coordinate.