**Programming for Geographical Information Analysis: Core Skills**

Assessment 2 – Planning for drunks

**1.) Intention of the software**

Imagine being part of a drunken crowd spilling out of a pub late at night. Your initial thoughts will be of drunk people staggering around trying to find their way home. I am right…...?

This model attempts to mimic the movements of these drunk people, as they leave a pub in the centre of a town (environment) and attempt to navigate their way home. At a basic level, the model does the following:

1. Pulls in a (‘drunk.plan’) 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 the 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.

Essentially the model puts you in the shoes of a town planner. Your aim is to help the drunks reach their homes, and map the routes taken by each individual (i.e. the density of drunks in the environment).

Files in the model:

‘drunk.model.py’ – the main python programme

‘drunkframework.py’ – a secondary file containing an agent (drunk) class.

‘drunk.plan.txt’- the model pulls in a 300x300 raster data file which represents the pub point and houses. The pub is denoted by 1s, the houses by numbers 10-250, and the empty walking spaces by 0’s. Essentially 25 drunks will each be assigned to an individual house. They will move around randomly (drunkenly) and attempt to find their specific house within the environment.

‘environment\_density.txt’ – a file showing the density of drunks passing through each point on a map. Any spaces a drunk has walked over are denoted by numbers > 0 and any spaces a drunk has not walked over are denoted by numbers <= 0.

**2.) Issues and thoughts during software development**

The first step was to read in the ‘drunk.plan’ data file and display it as a graphic within the python console. This step was relatively easy as I had done something similar in Assessment 1 (Agent Based Model) to read in and display a raster environment. Hence, the model initially presented me with this graphic:

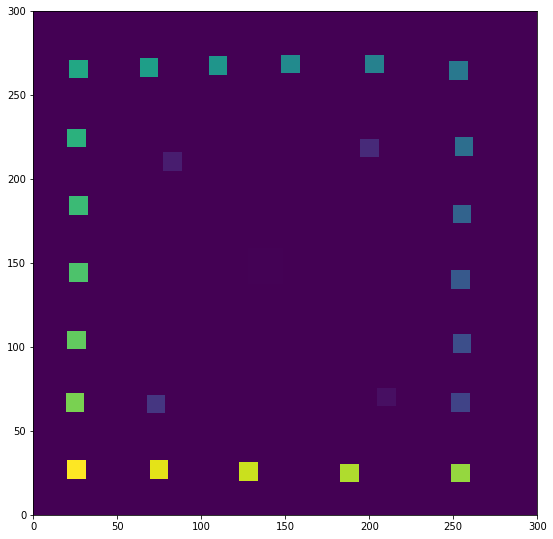


Figure 1: The ‘drunk.plan’ data file displayed as a graphic.

One problem was identifying where the pub was on the graphic. For example, you can clearly see the houses along the periphery of the graphic, but the pub was nowhere to be seen. I therefore had to examine the ‘drunk.plan.txt’ file to find the x and y coordinates (i.e. the group of 1’s) of the pub. I found that the pub was roughly in the centre of the ‘drunk.plan.txt’ file, between the ranges of 128-148 (x-coordinates) and 138-159 (y-coordinates) respectively. These x and y coordinates became the starting point for my agents (drunks). Hence, I set the self.x and self.y variables in the ‘drunkframework.py’ file to 138 and 145 respectively (as they were roughly in the centre of the group of 1’s in the ‘drunk.plan.txt’ file). While it was now clear (to me) that the agents were starting in the pub, it remained unclear (to the reader) where the pub was located on the graphic. This primarily seemed to be an issue with the colour scheme of the graphic. If I was to improve the model, I would edit the colour scheme of the environment and perhaps mark the pub (e.g. with a cross) on the graphic.

The next step was to assign a house number to each individual agent (drunk) and animate these agents in a GUI. The first phase was relatively easy as it involved looping through the agents, adding 1 and multiplying by 10, to give each agent a specific house number (i.e. between 10-250). This produced 25 agents, each of which had an individual house assigned to them. The second phase involved creating a GUI to animate the agents. Using my knowledge from the previous Assessment, I created a fully functioning GUI with an associated menu (e.g. ‘Start’ and ‘Quit’ buttons). I also created a ‘move’ function in the ‘drunkframework.py’ file. This made the agents move around the environment in a drunkenly manner (i.e. random directions). One issue I found with the GUI was that it took a very long time for the agents to reach their homes. To speed up the process, I chose to implement a ‘drunk\_speed’ variable so that the user can easily increase (or decrease) the speed of the agents. This will allow the drunks to find their homes more quickly. As a side note, if you find this process takes too long please feel free to uncomment line 82 in the ‘drunk.model.py’ file. This will set all drunks (agents) to be at their assigned homes, when you run the model.

The final step/issue involved displaying the density graphic (i.e. where the drunks had walked to find their homes) within the python console. I wanted the GUI to finish (showing the drunks moving around the environment) and then an associated density map to appear within the python console (showing the routes taken by the drunks). This proved to be an issue as the GUI could only display one graphic within its pop-up window. I therefore had to create a second ‘matplotlib.pyplot.figure’ at the end of the ‘drunk.model.py’ file, which was outside of the GUI functions. To get the figure to appear ‘inline’, I had to explore the range of IPython interactive features, that I found on the [Python Central](https://www.pythoncentral.io/) website. Here, I came across the IPython Magic commands (sets up matlpotlib’s interactive mode) which helped me plot the graphic inline. Previously, I had the second ‘matplotlib.pyplot.figure’ within the GUI function, and the programme struggled to identify which matplotlib.pyplot to show in the pop-up window.

To improve the model, I would like to explore the possibility of displaying two GUI windows next to each other, both showing the drunks moving around the environment (i.e. the house map and the density map simultaneously animating). This would be a much simpler model to visualise for the reader.

As a side note, I have also taken onboard the feedback from Assessment 1 (Agent Based Model) and have attempted to incorporate a lot more testing and commenting within the programme. This will hopefully make the programme a lot clearer to the reader.

**3.) General sources used**

The model has been created and developed from my own knowledge and from the skills acquired in Assessment 1 (Agent Based Model). The first assessment was particularly helpful in developing the GUI and reading in the txt files.

It was also helpful to visit sites such as [Python Central](https://www.pythoncentral.io/) to brush up on a few things I may have forgotten or struggled with. This became particularly useful when I attempted to use IPython Magic commands to plot the density graphic inline (i.e. in the python console).

The only source utilised from outside the programme was Andy Turner’s ‘drunk.plan.txt’ file which became the starting point for the project. You can find the instructions and associated data file [here](https://www.geog.leeds.ac.uk/courses/computing/study/core-python/assessment2/drunk.html).