Planning for Drunks – Python Model

*Intention of the software*

The model that has been built for this project has the aim of monitoring how a set number of drunk people in one town environment navigate their way to their individual homes from the one pub in the town. When fully functioning, and run numerous times, the model will be used to monitor the town’s planning by analysing the routes that are generally taken by the drunks when the model is run. This will be calculated by looking at the density of each point of the environment as 25 people pass through the town on their way home. As well as a monitoring town planning, the intention of this model is to help control crowds on a busy night out within this environment, by seeing what the most common routes that are taken by the drunks. This is important because it will help to ensure that people are kept safe within vulnerable situation, such as when they are drunk and leaving the pub to go home.

Although this is the main intention of the model that is being created, there are further additions of code that could be later be added that would extend its use. For example, more elements of the town environment could be included such as other buildings, as this would improve how realistic the model is. Also, road networks could be added, as in reality the drunk people would not just walk anywhere within the town, but would instead follow specific paths, such as pavements or roads. Furthermore, the agents that are representing the drunk people could be given further instructions that would detail their behaviour, thus adding an element of complexity to their movement around the environment as they try to find their way home. This could incorporate them not retracing their steps, or having a knowledge of other landmarks within the town that could help them direct themselves.

Once the initial model is built, and is successfully running, the code can be modified in places so that it suits the aim of the user. For example, a different town environment could be run into the model by using a different raster file, or by changing how many iterations the model is run for.

*Sources used in the software*

The code that has been created has not been taken from any other authors, but instead has developed on skills learnt in lectures and online documentation, as well as the previous practical portfolio that was developed, which created an ABM, and thus had similar aspects to this model.

The environment that is incorporated into this model has been created using a text file, which represents a raster file, using numbers to show the points of the pub and the houses in the town. The pub is denoted by ones, and the houses by numbers 10-250, which are shown within the model as different coloured squares. The empty spaces in the town environment are the zeros, and these are the points where the drunk agents will move as they navigate between the pub and their own house. This file, along with the idea for this project, was provided by Dr Andy Evans at the University of Leeds, as part of the GEOG5990 Core Programming for GIS module within the School of Geography.

If this project was developed further, then more sources would be utilised within the model building process, as they could provide additional information about exactly how drunk people behave when they are trying to reach a certain destination, and thus help conclude what routes they would take.

*The process of the software design*

When initially planning the design of the model, it was important to keep it relatively simple, as this would ensure that it would run correctly, as well as meaning it can be understood by a wide audience. Alongside it being simple but effective, the design process had to make sure that the main aim of the software was addressed – analysing the main routes in the town centre when drunk people leave the pub to return to their own house. Once this initial question is answered, then the model can be further developed to add more complexity and ask more questions about the town environment.

As each drunk agent in the model knows that they have their own house, and thus have the knowledge about where they live within the environment, during planning an emphasis was placed on each drunk reaching their own specific location. This meant that much of the model and the code works on the movement of the agents around the town, and how they respond to the environment of the model.

There are two main elements of the model – the agent framework code for the drunk people of the town, and the code that creates the model itself. The agent framework creates the drunk agents, and tells them that they each have their own house. This code also sets their initial location to the coordinates of the pub, which is located in the centre of the environment, and is represented by the ones within the raster file that is included. This framework then tells the agents to move randomly around the environment, which will allow them to find the correct house. The other part of the software is the main model code, which links to the agent framework and assigns the drunks using the class that was created. It then develops on from the original agent framework to tell the agents to stop moving when they reach the correct location. Moreover, the code attempts to animate the movement of each individual agent as they navigate themselves around the model environment.

After designing the software, a model has been built that does the following:

1. Creates the ‘drunks’ class, where 25 drunk agents start at the pub, and move randomly around the environment until they reach their assigned house number.
2. Creates an environment within the model to represent the town; this was done by reading in a csv file into the code that shows the location of the pubs and houses, meaning that the town could be visually mapped using matplotlib to plot these buildings onto the screen.
3. Creates a new map environment, so that the density plot of each map point can be seen.
4. Assign the drunks class to the model, and tell the drunks the number of their house. This is then used to stop them randomly moving when they reach the correct location.
5. Show the animation of the routes that are taken by the drunk agents whilst they are trying to find their correct house. This animation allows the user to see the main congestion routes.
6. Saves the new map environment as a csv file, so that the user can see how many times a drunk agent has passed through each point within that environment, creating a density point map.

*Issues during development*

One of the first problems that was encountered when starting the coding process was ensuring that the drunk agents started at the pub, as this was one of the requirements to answer the main aim of the software. To begin with, the agents were situated at the corners of the environment. To overcome this issue, when creating the drunks class, their location was set to (140,150), as when the town environment is mapped, this is the location of the ones, and thus the location of the pub. If this project was to be developed further, the colour of the pub could be changed when the environment is mapped, so that it is more clearly seen where the agents start from within the town.

