Geography Programming Courses

Programming for Social Science

Assignment 2: Planning for drunks

Submitted by: Tran Thi Thanh Huong

Student ID: 201376144

Project on Github: https://huongtran-3.github.io/Assignment-2/

ID: 201376144

Planning for drunks

This project is going to build an agent-based model that shows how drunks people in town find their home during walking around after getting out of pubs.

There are twenty-five agents (drunks) in the model. The agents (drunks) are moving randomly in the environment (their town) and trying to find their home after leaving pubs. The drunks have two functions: moving and steps.

There are two assumptions that the town is under control of planning the town is full of drunks people at night. There is a map of the area in a file drunk.txt, that contains pubs from which all the town's drunks spew out at night, and a map of all home of drunks. Fundamentally, drunks will leave the pub and wander randomly around until they hit their home.

This report includes four main parts: data, the algorithm for the model, the programming, conclusion, and appendix.

The data part will introduce the raster file and drunk.txt which data I used in this project. The algorithm for the model will briefly describe the algorithm in the drunk project. The programming part provides two sections of code which are drunkframework.py and drunk_model.py. In the appendix part, the UML class diagram and UML flow diagram were prepared, and coding of drunkframework.py and drunk_model.py was shown.

Data

The raster file named drunk.txt representing the pub points (start points) and drunks homes (end points) is 300 x 300 pixel raster file were used as input data in this project. In the raster file, each line in the file is a line in the raster image, starting at the top left corner. The background is denoted by the number zero, the pub is denoted by 1s, and the drunk home is designed by the numbers 10-250. There are 25 drunks (house numbers 250, 240, 230...etc...10). Each drunk should be given a number before leaving the pub, and they will need to find the home with the same number in the town plan.

The algorithm for the model:

1. Set up the environment
   1.1 Create the list
   1.2 Read the raster data

1.3 Put drunk.plan file csv into rows.

1.4 Locate the pubs and returns its position.

1.5 Check how the environment looks

2. Create drunk model and name drunks

2.1 Create the list of drunks.

2.2 Create the number of drunks and iterations.

2.3 Code to size the plot and animating.

2.4 Create and name drunks

3. Move the drunks and draw the density

3.1 Moving the drunks

3.2 Creating a list with the coordinates of every drunk (no. drunks, x, y)

3.3 Creating a text file with the coordinates of every drunk

3.4 Plot to check if the model works

4. Save the density map to a file as text

The programming

1. drunkframework.py

First of all, creating the agent framework includes twenty-five agents (drunks) which have two main functions: moving and steps.

These sets of rules were then used to build the agent framework. This can be found at **drunkframework.py**. The agent framework consists of the creation of the agent itself containing four attributes: coordinate x, coordinate y, drunks and name. These four attributes will be taken from specific variables in the model. The goal is to create a collection of all drunken people with a specific number (name) and number of their houses before the algorithm is run. The drunk was created and followed the randomly moving and monitoring density.

The move method will create a local variable that generates a random decimal number between 0 and 1 and the direction will be given depending on the value of such number.

The monitoring density method will let the drunk move around the town with step equal to 1 until they hit their home.

The raster data file provided would be used as the environment where our agents will interact. The drunk framework was built including the definition of drunks, randomly moving, and monitoring density. This environment comprised a grid of home numbers of drunks from 0 to 250.

Secondly, the agents (or drunks) was built that will interact with the environment. There is no specification for drunks to interact with each other. Consequently, each drunk will represent an agent with an independent route set by the random.

2. Agent-based model: drunk_model.py

There are some libraries were used to develop the code:

csv: the csv library was used to read (open) the raster file and write the outputs from the model.

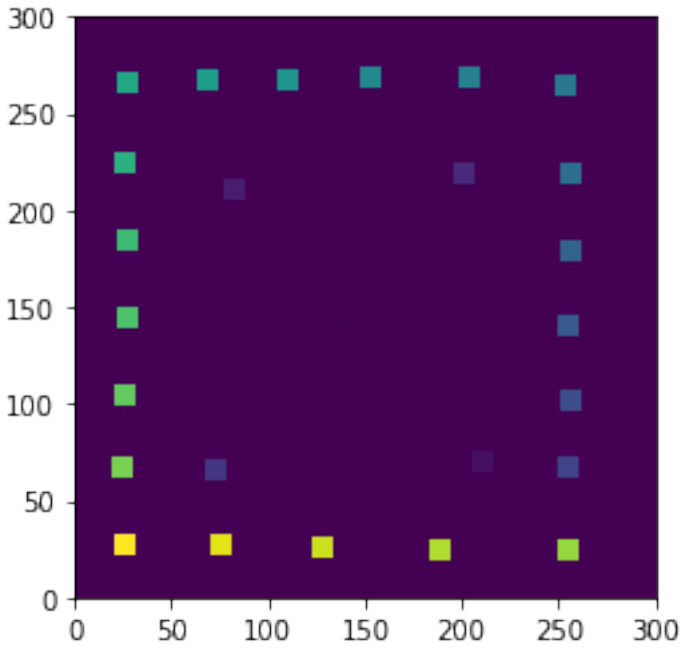
random: that used to generate random (decimal or float) numbers between 0 and 1.

matplotlib.pyplot ; matplotlib.animation: this library was used for the graphical representations of the raster file.

drunkframework: a set function of agents (drunks)

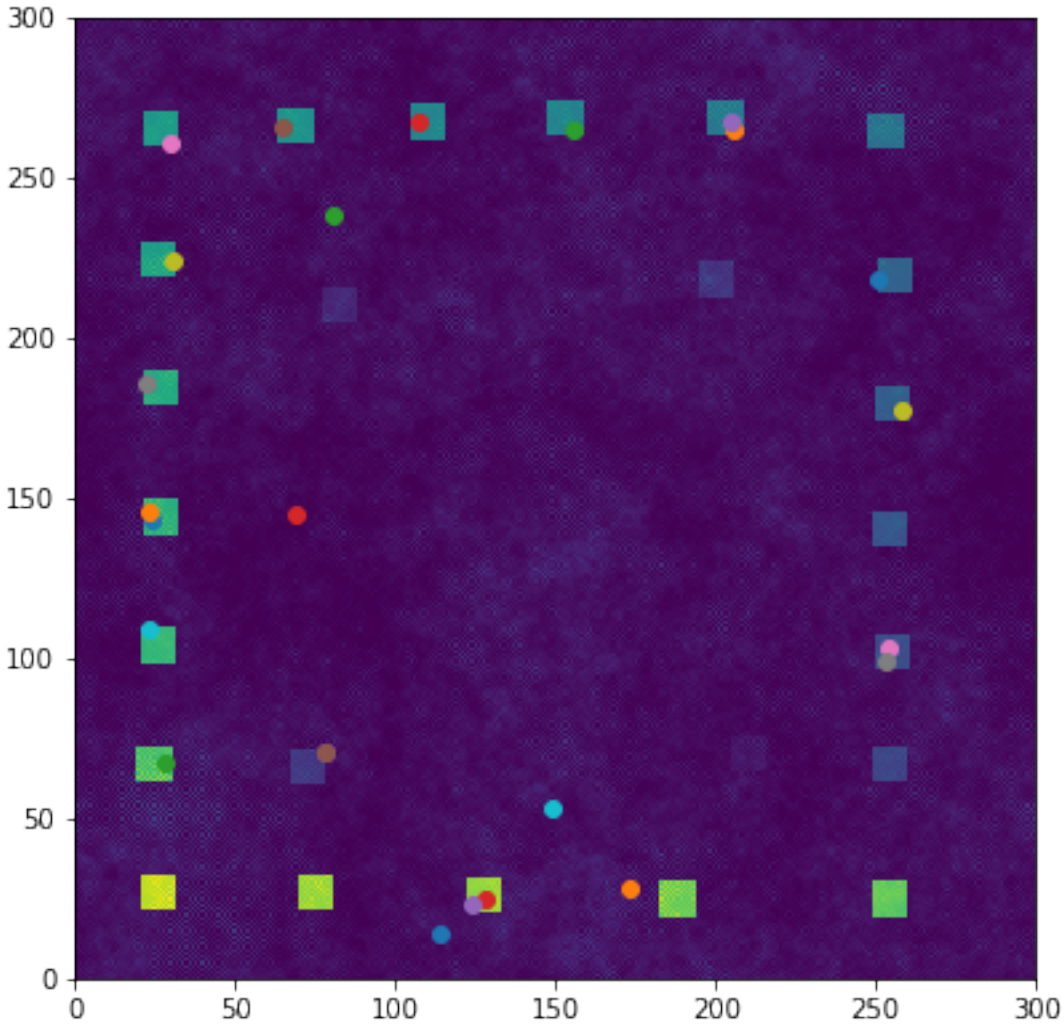
The drunk model saved as **drunkmodel.py** includes four main sections which presented step by step from the set up environment; the second section is the creating drunk model and name drunks; the third section is running the move function of drunks and draw the density map; the final section is about export the density map and save a file as text.

In the first section, the aim is showing all elements in the environment including the data in the raster file which describes the town, set down this data by row, pub locations in the town, and then check how the environment looks like. The code is written respectively creating the environment, reading the raster file, putting drunk.plan file csv into rows, and checking how the environment looks in matplotlib.pyplot.



In the second section, there are four supplement steps to create a drunk model for this project. Firstly, it is creating a list of drunks in town in which we have many drunks in town and will pew out from pubs after drunks at night. Secondly, the number of drunks and iterations was specific, in which we have 25 drunks and 100000 iterations. Then, it is a coding to size the plot and animating for our model. Finally, every drunk will have a house in the town, in order to find the house, drunks will be marked the name from 1 to 25, to fit with the house number from 10 to 250.

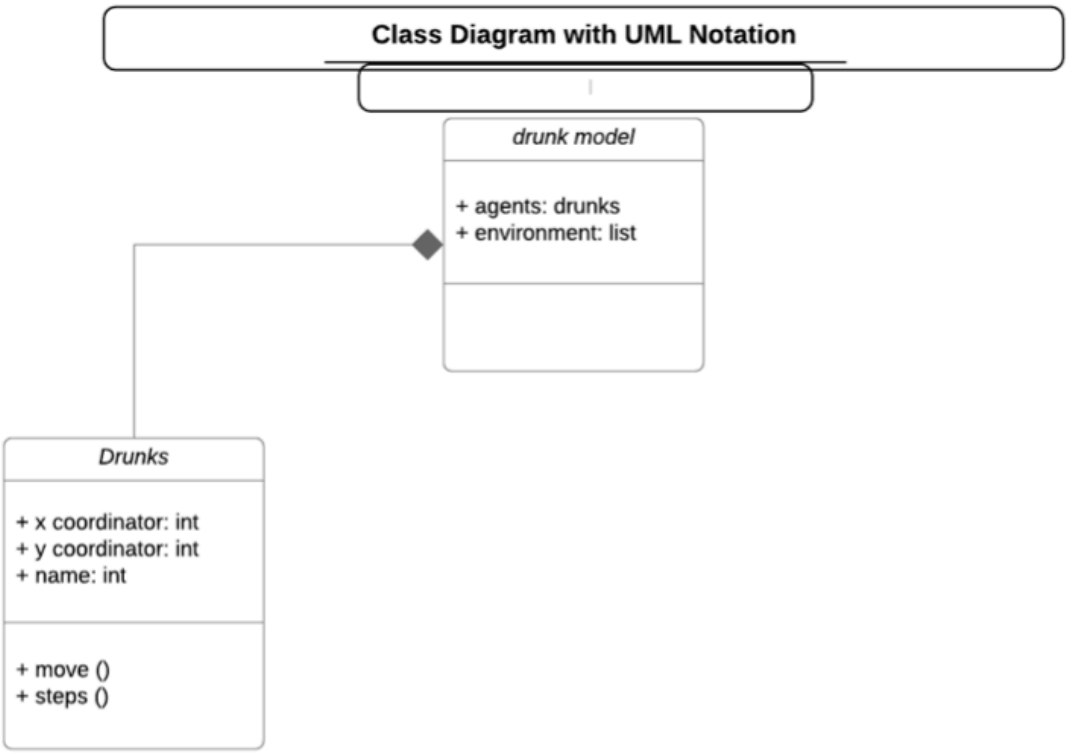
In the third section, the function moving by step of drunks from pubs at night, randomly walk in town to find their house with 100000 iterations show in 3.1 section. Next, the code in 3.2 shows the coordinates of every drunk (number of drunks, x, y) in the list. Then, we create a text file with the coordinates of every drunk to know the location of drunks after moving in 3.3, the text file saved as 'planningfordrunk.txt'. Finally, this is a checking step, the program is to plot data to check if the model works in 3.4 section.



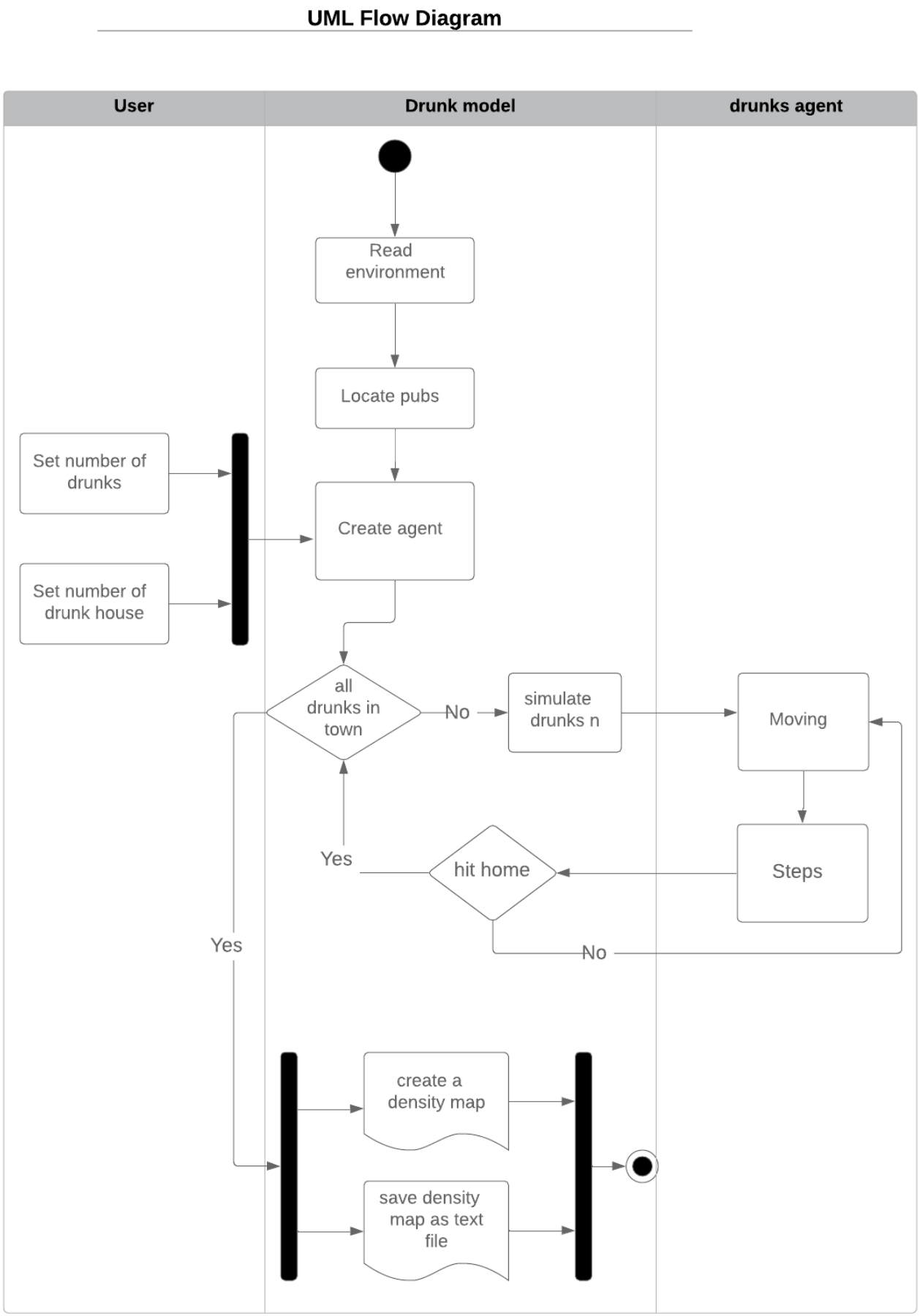
In fourth section, the density map which shows the distribution of values in a dataset over the range of quantitative variables, was saved to a file as text, as name 'drunks_environment.density.txt'.

Appendix

Appendix 1: UML class diagram



Appendix 2: UML flow diagram



Appendix 3: The Code programming of agent framework

drunkframework.py (agent framework)

```
# Create the environment
# Agent Framework

import random

# Defining agents (drunk)
class Drunk():
    def __init__(self, environment, drunks, name):
        self.drunks = drunks
        self.environment = environment
        self.x = random.randint(0,300)
        self.y = random.randint(0,300)
        self.name = name

# Moving randomly
    def move(self):
        if random.random() < 0.5:
            self.y = (self.y + 1) % 300
        else:
            self.y = (self.y - 1) % 300

        if random.random() < 0.5:
            self.x = (self.x + 1) % 300
        else:
            self.x = (self.x - 1) % 300

# Method to monitor density.
    def steps(self):
        self.environment[self.y][self.x] += 1
```

Appendix 4: The Code programming of drunk model

drunk_model.py

```
# The algorithm for the model:

# 1. Set up the environment
#    1.1 Create the list
#    1.2 Read the raster data
#    1.3 Put drunk.plan file csv into rows.
#    1.4 Locate the pubs and returns its position.
#    1.5 Check how the environment looks

# 2. Create drunk model and name drunks
#    2.1 Create the list of drunks.
#    2.2 Create the number of drunks and iterations.
#    2.3 Code to size the plot and animating.
#    2.4 Create and name drunks

# 3. Move the drunks and draw the density
#    3.1 Creating a list with the coordinates of every drunks (no.
drunks, x, y)
#    3.2 Creating a text file with the coordinates of evety drunks
#    3.3 Plot to check if the model works

# 4. Save the density map to a file as text


# First, we import all packages we will need for the model.
import csv
import matplotlib.pyplot
import matplotlib.animation
import drunkframework

# 1. Set up the environment.
# 1.1 Create the list.
environment = []

# 1.2 Read the raster data.
f = open('drunk.txt', newline='')
reader = csv.reader(f, quoting=csv.QUOTE_NONNUMERIC)

# 1.3 Put drunk.plan file csv into rows.
for row in reader:
    rowlist =[]
    for value in row:
        rowlist.append(value)
    environment.append(rowlist)
f.close()
```

```
# 1.4 Locate the pubs and returns its position.
row_counter = 0
for row in environment:
    for value in row:
        if value != 0:
            x = row.index(value)
            y = row_counter
            #print("pub at coordinates","x =", x, "y =", y)

    row_counter += 1


# 1.5 Check how the environment looks.

matplotlib.pyplot.xlim(0, 300)
matplotlib.pyplot.ylim(0, 300)
matplotlib.pyplot.imshow(environment)
matplotlib.pyplot.show()


# 2. Create drunk model and name drunks

# 2.1 Create the list of drunks.
drunks = []

# 2.2 Create the number of drunks and iterations.
num_of_drunks = 25; # There are 25 drunks in the town name from 1
to 25, and their home from 100 to 250 repectively.
num_of_iterations = 100000 # There are unlimited step unti drunks
reach their home, I assmue drunks do 100000 iterations

# 2.3 Code to size the plot and animating.
fig = matplotlib.pyplot.figure(figsize=(5, 5))
ax = fig.add_axes([0, 0, 1, 1])

# 2.4 Create and name drunks
for i in range(num_of_drunks):
    name = ((1+i)*10) # Python starts in 0 we have to add 1 first
and then just multiply by 10.
#   print(name) # test names are correct.
    drunks.append(drunkframework.Drunk(environment, drunks, name))


# 3. Move the drunks and draw the density.

# 3.1 Moving the drunks
for i in range (num_of_drunks):
    drunk = drunks[i]
    for j in range(num_of_iterations):
        if environment [drunk.y][drunk.x] != drunk.name:
```

```
            drunks[i].move()
            drunks[i].steps()


# 3.2 Creating a list with the coordinates of every drunks
(no.drunks, x, y)
drunks_location = []
for p in range(num_of_drunks):
    no_drunks = []
    no_drunks.append(p + 1)
    no_drunks.append(drunks[p].x)
    no_drunks.append(drunks[p].y)
    drunks_location.append(no_drunks)

# 3.3 Creating a text file with the coordinates of every drunks
with open('planningfordrunk.txt', 'w', newline='') as f:
    csvwriter = csv.writer(f, delimiter=',',
quoting=csv.QUOTE_NONNUMERIC)
    csvwriter.writerow(["drunks","x","y"])
    for row in drunks_location:
        csvwriter.writerow(row)


# 3.4 Plot to check if the model works
matplotlib.pyplot.xlim(0,300)
matplotlib.pyplot.ylim(0,300)
matplotlib.pyplot.imshow(environment)
for i in range(num_of_drunks):
    matplotlib.pyplot.scatter(drunks[i].x,drunks[i].y)

matplotlib.pyplot.show()

# 4. Save the density map to a file as text.
with open('drunks_environment.density.txt', 'w', newline='') as f:
    csvwriter = csv.writer(f, delimiter=',',
quoting=csv.QUOTE_NONNUMERIC)
    for row in environment:
        csvwriter.writerow(row)
```