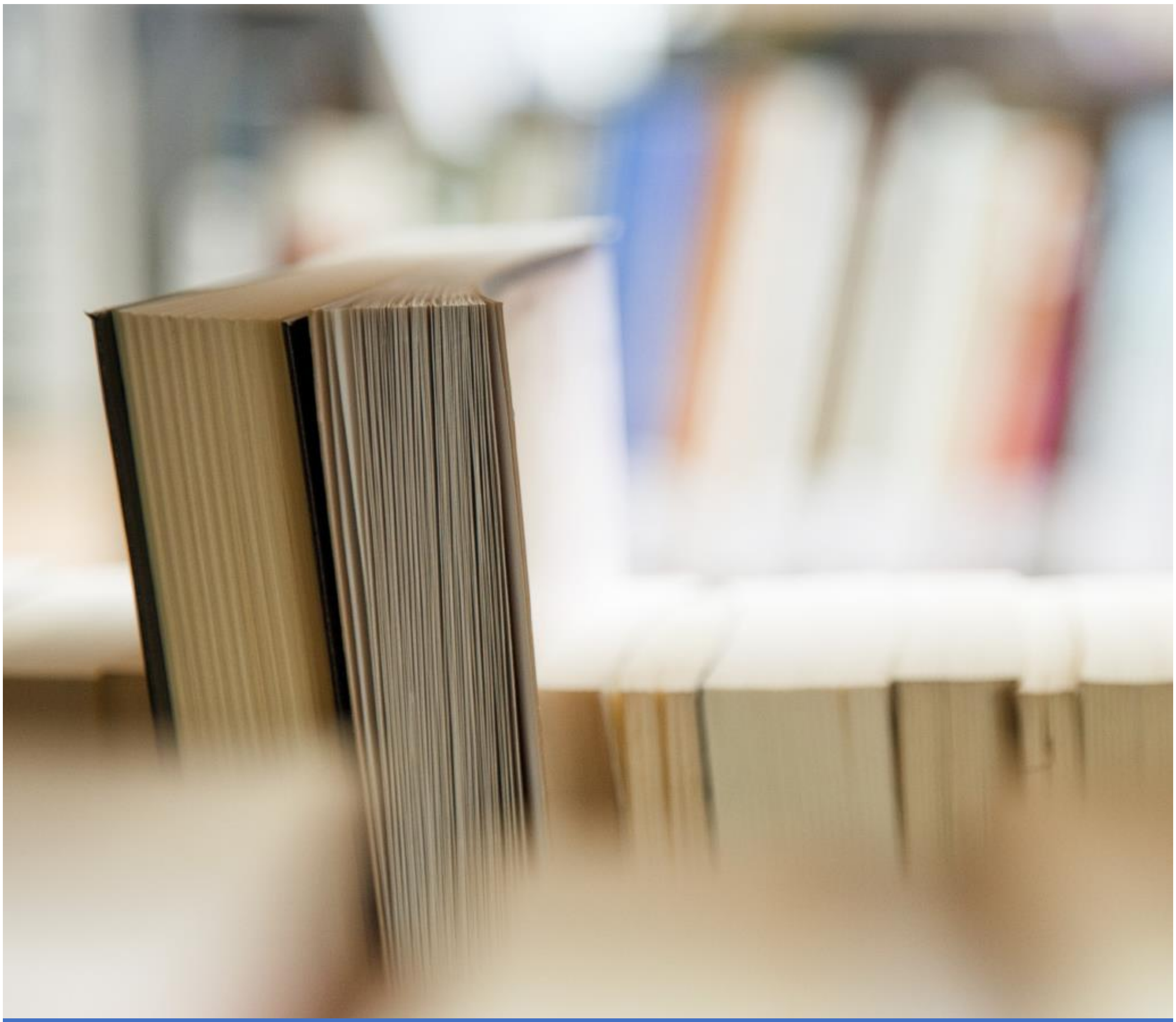

DOCUMENTATION PLAN

FOP1005 – Fundamentals of Programming

Student Name:Dhrubo Jouti Das Troyee 9/19/23 [redacted]

OVERVIEW:

In this assignment we will be able to be simulating the infection of human which was transmitted to a group of people commonly known as “vampire”. This simulation will show a scatter plot containing humans, vampires, food, water, and garlic using 2D map. Moreover, on the 2d map we will see how c the vampires and humans will move and interact with one another, as well as with the food, water, and garlic .. All the program are store in a animation.py file. We also need various types of plotting packages such as matplotlib.pyplot , random, sys for running the bash script file.

USER GUIDE : HOW TO USE THIS PROGRAM :

In the “Animation.py’ python file , we have created so many functions classes for the programming purposes . Here, we have created three main classes which are “human”, “Vampire” and for the “item” “food” class , “Water” class “Garlic” class are created for the better understanding of interaction the humans and the vampires. Here, in the assignment 2 guideline, there are specific things are mentioned that we need to follow and make sure our code work maintains the instruction and our code should work. According to the assignmnt2 , the interaction between the human and human, and human with vampire, and human must interact with the items such as water, food, garlic. To perform the action, we have created objects for doing every specific task. This specific task upholds the methods of the classes that we have mentioned above. Animation.py contains utility functions and classes necessary for the simulation to run.

For better understanding I am providing a brief knowledge about it :

```
class Vampire:
            def __init__(self):
           def movement(self)
            def interaction(self, other)


class Human:
          def __init__(self):
         def movement(self)
         def interaction(self, other)


class Water:


class Food:
```

class Garlic:

These are some function we have used to simulate the program and the attribute.

def interactions_of_performance(humans, vampires, items):

def (initial_human_count, initial_vampire_count):

def update_positions(Human):

def visualize_simulation(humans, vampires, items, timestep):

def simulation_data_save(filename, initial_human_count, initial_vampire_count, final_human_count, final_vampire_count):

 def main(initial_human_count, initial_vampire_count, num_timesteps):

1.TIMESTEPS:

The program should accept 3 command line parameters which are

- Initial number of humans
- Initial number of vampires
- Number of timesteps in the simulation

These are used to control all the other functionality of the classses. Basically, 'Command-Line Parameters is designed to get user input on the amount of time steps and the initial value .

2.OBJECTS:

In the “**human: class** and the “ **vampire class**, there are attributes like age, health and position for human and vampire on the map. As it mentiond, the human can only have health variable and the value for it will be 100,so inside the **class “human”** we have created def_int_ function to set this. This means at the ver begening of the simulation human will start with a 100 health . Moreover, like this, the age attribute for the human class are also set up which is a random number between 10 to 50. Again, in the **“vampire class”** we have set the health attribute .For the vampire. Moreover, we have set up the location of the human ad vampire class at the very beginning for the simulation project. The initial locations of food, water and garlic are also done in their specific class.

3. MOVEMENT:

For the movement purposes, we have created the movement function for both the **human class** and the " **vampire" class**. Human can move up to 4 spaces (steps) in any direction. Which is a random number, that's why random.randint was used inside the function. For the vampire class , Vampires can move up to 8 spaces (steps) in any direction, on the movement function . As food, water, and garlic don't move, we did not use any movement function for this. Here, **"human**" also lose 1 health point for every step when they move, this is also set in the movement function both for the '**human class** " and "**vampire class"**. Random initialization' is to randomly initialize attributes such as health, age, and positions for humans, vampires, and things.

4.INTERACTION:

The use of **‘interactions_simulation ’** function is to simulate interactions between humans, vampires, and other items over a specified number of time steps. Which is 8. the **‘initialize_simulation “function** is used to set all items at the initial positions with attribute like,initial_human_count, initial_vampire_count . And here, the each occurred will be printed by the printout function both for the interaction by the human, food, the vampire on the terminal. Furthermore, we also created **“ interactions_of_performance”** function for the interaction of human, vampire and the other items. . here we have created code for the Implication of interactions between humans and items, implication of interaction between vampires & item,, vampire to vampire..

5. VISUALIZATION OF RESULT :

The function **“ updates_positions “**are used to update the locations of humans and vampires at each time step. There is also a command line for the game's settings, produces interactions between people, vampires, and items.

6.CSV DATA:

The most important component of this simulation is the **"main simulation loop "**, when the function has finished running for the specified number of steps of time, it saves it into a csv file. Moreover, The purposes of the **"save_simulation_data "**which is a function save simulation data to a CSV file. It is also check if a CSV file has records of both the initial and final numbers of humans and vampires.

6.PARAMEYER SWEEPS:

As we know, Bash script that performs a parameter sweep and saves the results in a CSV file. In order to doing that, we have to put all the value according to the variable name and write a code for this and put all the data in a csv file which is **"save_simulation_data"** . In order to run this we have to import sys, and os packages.

EXPLANING FEATURES & CODE:

To run the Animation.py program, run the Python script in the terminal, and the user will be asked to enter the mentioned three parameters.. Those parameters are asked to determine how the simulation will run. Following that, during the specified amount of time steps, the program's Simulation Execution will simulate interactions between individuals, vampires, and things. The loop will run in a certain way that it will update and position , simulate the interactions between the human , vampire and food, water, garlic.   The initial and last numbers of humans and vampire data in a csv fie which is **"simulation_data_save** " I initialize matplotlib.pyplot as plt in the time steps programming portion to start the representation of data I also include random packages because we need to initialize attributes like health, age, and positions for humans, vampires, and other entities at random. I also activated the map size (map_width and map_height) which is 100, to demonstrate the data. After calling e initialize simulation method to set the initial population of humans and vampires in your model, the simulation will run until the loops go fulfil the requirement. We will see, all the function being use for interacting, moving purposes, and impacting each other health ,and by the end of the time, it will include the initial and final counts of humans and vampires in the CSV file.