Jacob Curtis

09/13/2025

DS 5010

1. The assignment was fairly straightforward. The challenge from the entire assignment was understanding how methods from Agent interact in the City class. After a few hours of reading and trial and error, I got the hang of the interaction. Agent class is primarily for establishing the “agent” through random selection. Also, determining the threshold and how that correlates with the is_satisfied method. Both are instrumental in the simulation process as shown in City.py.
2. The assignment serves as a purpose as it demonstrates how individual agents or in our case, persons, interact with those around them. The neighbors can represent others in groups, and agents are the individuals. Those that choose to move, relate to those that are similar to them, in whichever similarity may be the case. This could relate to culture, race, gender, or even likes and dislikes.
3. With the self grade, after a solid effort, I believe a grade of 90 would be an accurate representation. I did struggle slightly with the move_agent and get_neighbors, therefore, there may exist errors that I have yet to pick up on. Both programs, city.py and agent,py, execute without errors.
4. For the creative portion of this assignment, I chose to implement an option for the agent to choose a cell based on corresponding neighbors. For example, if agent X would like to move, the agent can “look” at surrounding neighborhoods and choose whether or not to move. This will remove the randomness component to the assignment of agents to cells. This new method is choose_neighbor(), and update_move_agent is an updated

method from the original move_agent. In this method, a new variable is defined as new_spot and utilizes choose_neighbor.

5. The output from the creative portion, is a more structured movement of the agents. In other words, agents are nearly evenly spread in the grid. Before, in the original program, it was a random assortment, the sense that agents would be chosen at random. The set_up method does the assignment of agents via appending cells to a row in a nested for loop. In the original program of Agent and City, the agents would stabilize before the agents would become as structured as they would in this current change.
6. On average, the simulation stabilized at around round 52. I ran the program a few times, and this was the average round of stabilization. The rounds per simulation, I set to 100. The total number of simulations are 1000. The formula for average simulation is the total sum of the number of rounds divided by the length of the stabilized rounds list. In other words the total number of rounds over the length or number of round occurrences. See updated_simulate for this formula.
7. The frequency, or the number of simulations that stabilize is ~60%. This is calculated via the number of stabilized rounds divided by the total number of simulations. Both the average number of rounds and frequency are close in calculation, which would make sense as if ~50% of all rounds are stabilized, so will the total overall frequency.