BDA450 W23 Assignment 3

Due date: March 29, 11:59pm

This is an INDIVIDUAL assignment. The work that you complete and submit must be your own, not the work of others, nor the work of a team, nor material taken from other sources, nor making use any tools that generate material (including, but not limited to, AI tools like ChatGPT). It is permissible to discuss the assignment in general terms, discuss general approaches to problems, help someone to understand a concept or give them a bit of advice if they have a problem, etc. However, you are not permitted to 'work together', nor to give/receive solutions (whether partial or complete) to/from other students.

Clear indicators that you are violating the rules include (but are not limited to):

- Looking at someone else's work and using that to come up with your own submission
- Sharing/sending/receiving/copying files to/from other students
- Using cut-and-paste or copy-and-paste (if both the source and destination are not your own creations)
- Etc.

If you are unclear at all about what is permissible, please ask your professor.

Submit your work in Learn. Include both your code, and the required output.
Late submissions <u>will not be accepted</u>.

Scenario

You will be developing an agent-based simulation of people walking on a sidewalk, to study possible transmission of a virus. (Please note that the characteristics of this virus are not intended to represent COVID-19!) You have been provided with most of the infrastructure for this simulation; your job will mostly entail:

- Generating new agent arrivals appropriately;
- Implementing the behaviour of the agents; and
- Aggregating data.

An examination of the code will give you more insight into the simulation itself, but a brief outline will be provided here. The sidewalk is a large rectangular Cartesian plane. People will arrive at one end of the sidewalk, with the goal to reach the other end and leave the sidewalk. (Agents enter the sidewalk via an enter method, and leave via a leave method.)

Arrivals have not been implemented for you (rather, a number of agents have been dumped into the environment simply to demonstrate trivial random movement). Arrivals of agents are determined independently for each end of the sidewalk; the arrivals should be random (converted to integer, of course), with the average interval specified in the code. Agents must enter at the very ends (in terms of x-coordinates), but may enter at any y-coordinate.

Each turn, the step() method of each (active) agent is called. This is where you should implement agent actions. To be clear: The step() method should implement the behaviour for a single agent. At each time step, each agent will have its step() method called—by triggering step() individually for each agent, the behaviour of all agents will be determined.

To move (or enter/leave the sidewalk), the agent must make a request to the sidewalk (via the attemptmove()/enter_sidewalk()/leave_sidewalk() methods). The sidewalk will determine if the move can be taken, and will update the state accordingly (including the agent's x-y coordinates). In particular, moves will not be permitted if the desired square is already occupied. You may not have your agent modify its coordinates directly.

(Note that for simplicity, in each time step we do not strictly track agents' positions at time t-1 and then move them all simultaneously at time t. (This requires complexity of managing agents' desired moves and resolving conflicts.) Instead, agents' moves are processed in arbitrary order in each time step.)

Some agents are infected with a virus. (Agents cannot see whether agents are infected (including themselves), so this should not be a basis for behaviour.) When agents are close to an infected agent, they are infected as well with some probability. This probability at distance 1, for a duration of 1 time step, is set as a simulation parameter, and decreases with distance. (See the code for details.) Agents track both 'infected' and 'newly infected' status; in the graphical display, uninfected agents are green or purple (as explained below), originally infected agents are red, and newly infected agents are yellow. (Infection, tracking and coloring is handled for you—you do not need to implement this.)

You must implement the behaviour of your agents. In fact, **you must develop two versions of your simulation:**

- **Part 1:** Recall that agents cannot move into an occupied square; this means that agents travelling in opposite directions can block one another. **In the first version of your simulation, agents simply move (to either their left or their right) when they are obstructed so that they can pass by one another.**
- **Part 2: Most agents (who will be identified in green) continue to move as described for Part 1. However, a small percentage of agents (those with self.careful set to True, and identified in purple) are aware of the importance of distancing, and actively seek to keep at least 1 full square away from other agents, and further if possible.**

Important notes:

- **There is no single right answer, particularly for Part 2.** In fact, there is a wide variety of possible strategies/implementations for Part 2, and some will work better than others. (You will find it harder for your agents to keep their distance when only *some* of the agents are seeking to keep their distance—it will require more thought to develop good strategies.) Do your best to try to generate ‘good’ behaviour! Better behaviour will earn higher marks.

- Agents can use the isoccupied() method of the sidewalk to determine if other agents stand at particular coordinates.
- It is likely easiest to consider relative coordinates when ‘looking’ or moving. For example, if your agent is at coordinates (x, y), and you want to check the square to the left, you can us isoccupied() on coordinates (x-1, y).
- Your agent should likely be making its decisions using if statements, based on whether particular squares are occupied. For example, if the square to the left is available, you might want to try to move to the left. (It might be better to think about things first in terms like ‘if the square to the left is unoccupied, move to the left’, then translate these to coordinates.)
- In Part 2, you will need to implement two different behaviours in the step() method, selecting the appropriate instructions with an if statement: the behaviour if the agent is not careful (which will be the same as part 1), and the behaviour if the agent is careful.
- For the careful agents, you will need to consider whether a number of squares are occupied before deciding where to move. I would strongly encourage you to think about how to do this with loops, rather than writing if statements with many hard-coded coordinates/conditions.

The behaviour of your agents should not result in strange activity (e.g., moving ‘sideways’ repeatedly for no reason without progressing forward), nor should it require an unrealistic degree of coordination (e.g., ‘when meeting directly, people always divert to the right’). Different people will do different things at different times, and your simulation should reflect this. (On the other hand, you are not expected to model different agents with different habits, tendencies, etc. You are simply expected to include some randomness in the behaviours of your agents.)

As with the Boids code, the graphical routines are responsible for executing each ‘clock tick’. At each clock tick, the sidewalk’s run_step() method is called; please implement time step activities here, rather than modifying the update_figure() function.

Most parameters are set at the top of the code provided. For both versions of your simulation, you should track (at minimum) initial infection rate, new infection rate, and total infection rate. It would also be interesting to track infection rates over time (using some time interval).

Please submit your code for both simulations, your statistical output, and any other output you feel is of interest.