School of Sciences and Engineering
Computer Science and Engineering Department

Fall 2020
November 23rd, 2020
Programming in Java

Assignment 4 – COVID Tracer Simulation

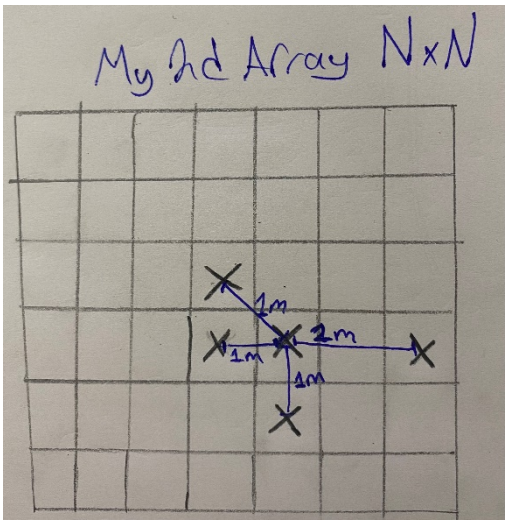
Name: Chris Amgad
ID: 900170819

Name: Haroon Tharwat
ID: 900152625

Submitted to:
Prof. Sherif Aly

Assumptions

- The screen size of the simulation is always fixed with a size of 500x500
- Each Human Node ,either carrying the COVID-19 or neutral, is assigned a Thread, where each thread has a unique ID that is printed at the beginning of the simulation to indicate that the threads were spawned successfully.
- No Node can get out of the screen
- A 2D array of a defined size by the user has every index has a node that is mapped to its corresponding location on the screen using a ratio formula.
- The default value of the size of the 2d Array is 30x30 and its size X equal size Y as we see; This means that the 1 meter distance that every node moves horizontally = the 1 meter distance that the node moves vertically = the 1 meter that the node moves diagonally and anti-diagonally.

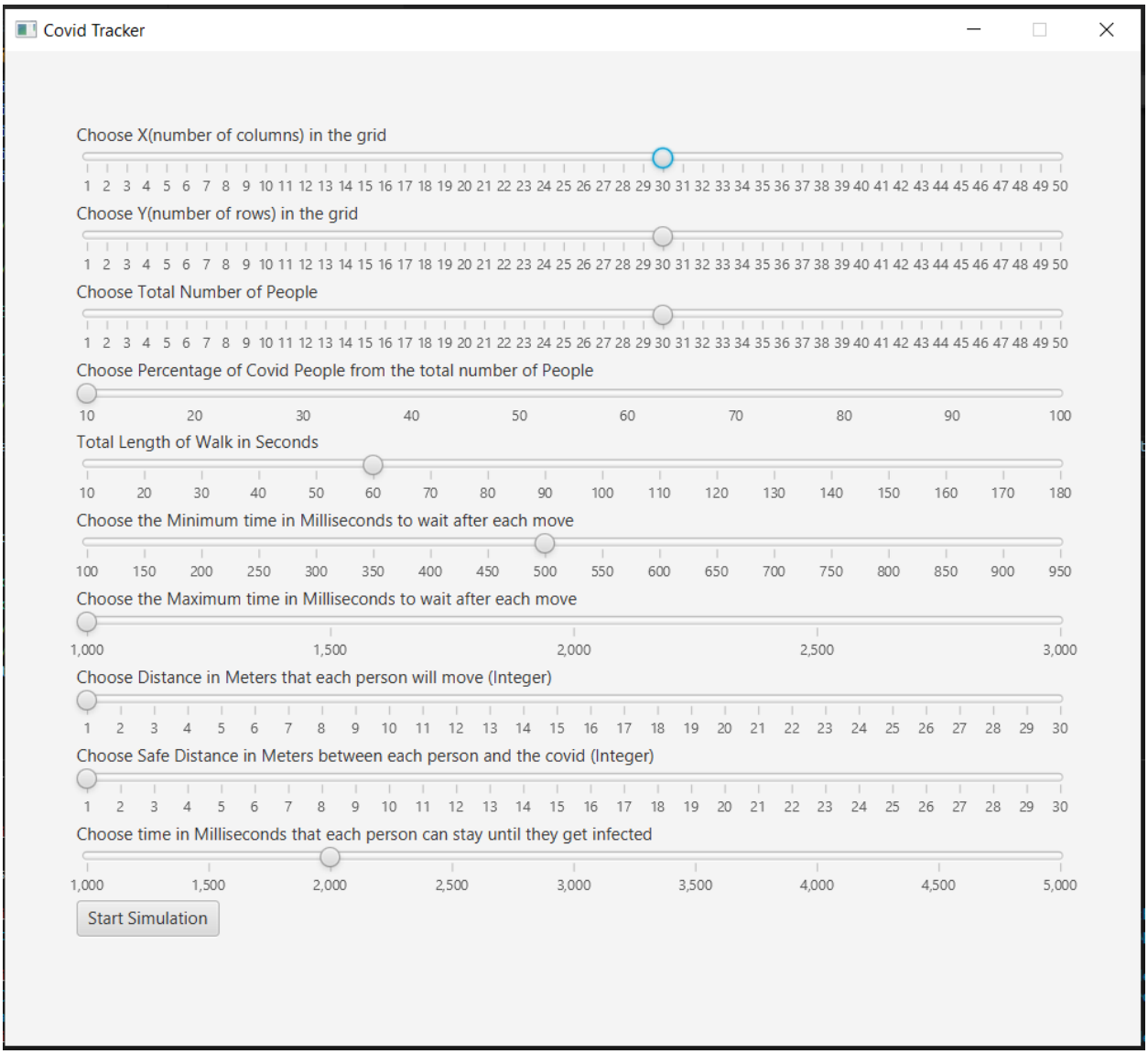


- In case the X size of the 2d Array is not equal the Y size of the array (ex: 20x30) , then here the 1 meter distance moved horizontally != the 1 meter distance that the node moves vertically. This is customizable by the user but it makes sense and more logical to choose the 2d Array size as NxN in order to maintain the ratio of distances between the X and Y on the screen as it is not normal in a real life scenario that a 1 meter on Y is not equal to a 1 meter on X.
- **Important assumption:** If a node spends a TOTAL of 2 seconds (Default value and can be changed) within the “unsafe social distance” of a COVID positive, they are flagged for possible infection. This means that a node can be in the Unsafe area for 1.5 seconds then leaves the unsafe area, then returns back to an Unsafe area and stays for another 0.5 seconds; We now have a total of 2 seconds spent near a COVID node so it gets infected. In other words, a node doesn’t have to stay the entire 2 seconds once it is in the unsafe area but instead this total of 2 seconds could be spent on different time occasions.

Note: The reason we did it this way is due to the fact that the nodes are moving too fast, and they can keep going in and out of the areas constantly before completing the 2 seconds once, so the simulation life could end and only a few nodes will be marked as infected due to the rarity of the condition that should be met; therefore, we added this feature of allowing the nodes to spend a total of 2 seconds in multiple of times instead of having it just to be spent it once inside the unsafe area .

Inputs

A welcome screen will ask the user to choose an input from 10 sliders, where each slider represents a parameter.



As shown in the above screenshots, the controlled parameters in order are

- X, which represents the number of columns in the grid*, in integer*

- Y, which represents the number of rows in the grid*, in integer*
- The Total number of People nodes, *in integer*
- The percentage of COVID People nodes of the total number of people,
- The Total length of walk, *in seconds* (Program pauses after this time)
- The Minimum limit of Milliseconds, which represents the minimum time at which a random time will generated between this boundary and another Maximum parameter in the next slider

  →*random(THIS_VALUE, max)*
- The Maximum limit of Milliseconds, which represents the maximum time at which a random time will generated between this boundary and the minimum limit

  →*random(min, THIS_VALUE)*
- The Distance that each node will move each time in meters, *in integer meters ex: 1 meter, or 2 meters, etc..,*
- The Safe Distance at which each node if stayed within this distance near another COVID node for a certain period of time, it gets infected. *in integer meters ex: 1 meter, or 2 meters, etc..*
- The amount of Milliseconds time to remain in the unsafe social distance to be infected.

*In case, there an invalid input within one of the sliders when the Start Simulation button is pressed, then a message will be printed with the reason of the invalid input and the program will wait till another valid input is selected.

For example, a user chooses the number of people being more than the grid size(X*Y) and presses the Start simulation button, a message is printed and we wont move to the simulation screen until another valid input is selected of the number of people is re-chosen.

*Default values are set already on the slider at the beginning of the game.

* At the end of the simulation, The Thread IDs of the nodes that were infected will be printed out, and if no nodes were infected, a print message will indicate that as well.

Screenshots of Simulation