CA Assignment 0

Gurjaipal Singh, Tanishque Soni, Animesh Jain

January 2024

1 Introduction

The Infiltrator Simulation Project is a *Java*-based program that simulates an infiltrator navigating through a border of sensors to get into the defending country. The project employs a 2D array to represent the border, where each element of the border is a sensor. The movement of the infiltrator is determined by the state of the sensors, and a time counter is used to control the timing of the simulation.

The key components of the project include

- **my_Sensor.java:** class representing individual sensors
- **my_Border.java:** class managing the 2D array of sensors
- **my_Infiltrator.java:** class modeling the infiltrator's movement
- **Simulation.java:** class orchestrating the simulation
- **Simulation_Plot.py:** python file used for plotting the outcome of simulation. Running this will result in a plot of *Probabilities* vs *Average Time* for the all widths in the interval that will be provided by the command line.

2 How to run the code

To run Infiltrator Simulation code follow these steps:

1. **Download the code:** Download all the files from the moodle. Make sure you have all the required files namely `my_Sensor.java`, `my_Border.java`, `my_Infiltrator.java`, `my_Simulation.java`.
2. **Compile:** Open a terminal or command prompt and navigate to the directory containing the Java files. Compile the code using the following command:

   ```
   javac my_Sensor.java my_Border.java my_Infiltrator.java Simulation.java
   ```

3. **Run the Program:** After successful compilation, in order to run the `Simulation.java` program successfully, it is essential to provide six command-line arguments.
   These arguments specify the required parameters for the simulation.

   ```
   java Simulation arg1 arg2 arg3 arg4 arg5 arg6
   ```

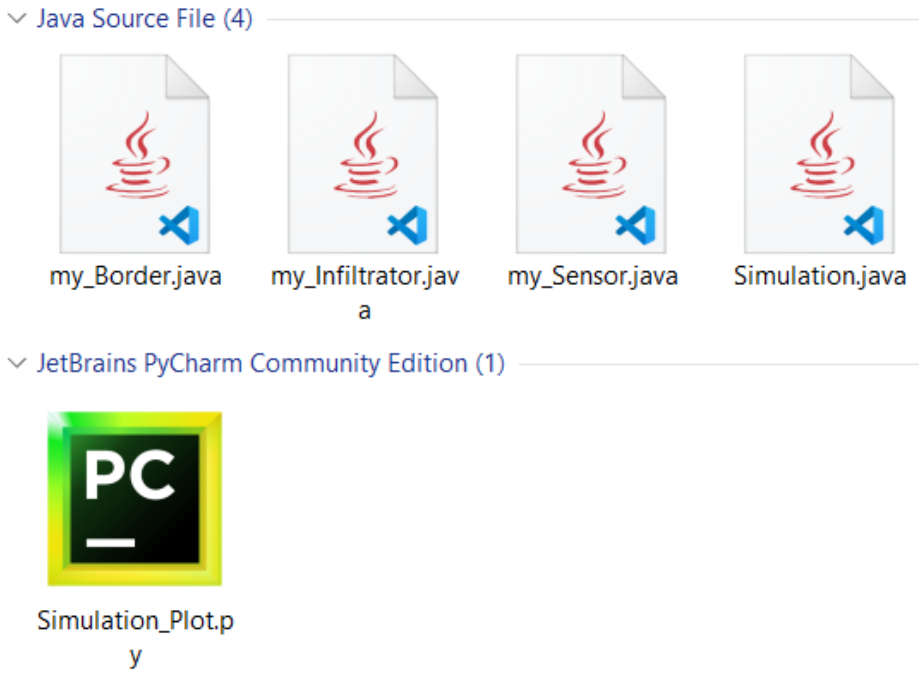
   Here,
   `args1` is the minimum depth of the border
   `args2` is the maximum depth of the border
   `args3` is the minimum probability of the sensor
   `args4` is the maximum probability of the sensor
   `args5` is the interval size of probability
   `args6` is the number of times the simulation will run for each depth and on-probability combination

4. **Observe Output:** The program will execute the `Simulation.java`, and you will observe the output in the newly created file `Simulation_Result.txt` and `Simulation_Result_Py.txt`.

5. **Plot the output:** Run this to plot the output *Probabilities* vs *Average Time* for each depth.

   ```
   python Simulation_Plot.py
   ```

2.1 File Structure

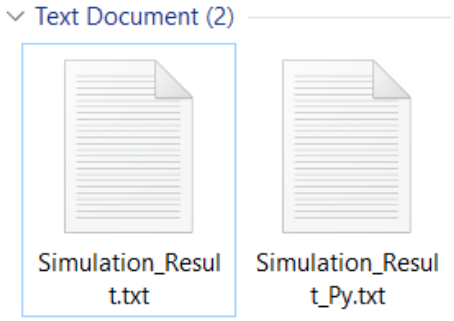


Submission contain 4 .java files and 1 .py file. The java files contain the code for simulating the border the infiltrator

To run the program first test by running the code provided below in the terminal

1. javac .\Simulation.java
2. java Simulation 1 5 5 95 5 100

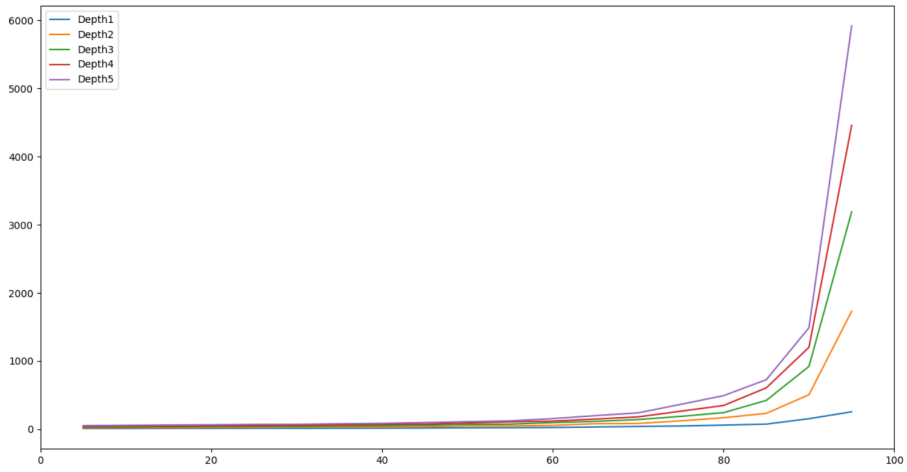
This will create 2 new files named Simulation_Result.txt and Simulation_Result_py.txt



Simulation_Result.txt is meant to a human readable log for logging and testing purpose

Simulation_Result_py.txt is meant to be parsed by python script provided to draw the plot

To see the plot simply run python .\Simulation_Plot.py and plot similar to provided below must appear



3 Explanation

3.1 Code explanation

DC

$\cdots$	0	1	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\cdots$	0	0	0	$\cdots$
$\cdots$	1	0	1	$\cdots$

I

AC

Start State

At the start of simulation the infiltrator checks the 3 sensors in from of him. And decide which are On and which are Off

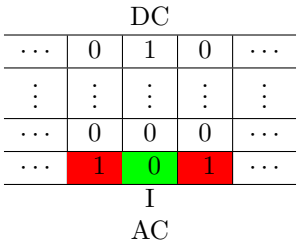
DC

$\cdots$	0	1	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\cdots$	0	0	0	$\cdots$
$\cdots$	1	0	1	$\cdots$

I

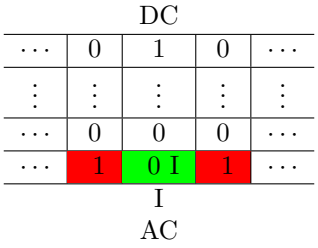
AC

Infiltrator checks the cells



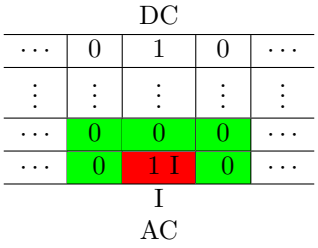
Infiltrator decide which cell to move in

Then move to the cell in which the sensor is off. If none of the three sensor the on it wait for that turn



Infiltrator move into the cell

Then the border refresh to new random values. This time the Infiltrator checks the sensor in its own cell and the 3 cells in front or it. If both his sensor and the one of the sensor in front of him is off then he move to the cell where the sensor is off. If not then he doesn't move.



Infiltrator checks his cell and cells in front

This cycle continues till Infiltrator reach the end if the border where he only checks his own cell and if off move into the DC

DC

$\cdots$	0	1 I	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\cdots$	0	1	0	$\cdots$
$\cdots$	0	1	0	$\cdots$

I

AC

Infiltrator checks his cell and cells in front

3.2 Argument explanation

So the arguments passed in the command " java Simulation 1 5 5 95 5 100 " corresponds to in order of appearance

1. args[0] (1 in this case) is the minimum depth of the border
2. args[1] (5 in this case) is the maximum depth of the border
3. args[2] and args[3] corresponds to the min and max probability (5 and 95 in this case) of the sensor to be on and must of type integer and between 0 and 100 inclusive (although 100 will cause the program to never halt)
4. args[4] is the step size the program will take in calculating the probability as in this case 5 means that it will simulate the border with on-probability of 5 10 15 and so on till 95 as change in probability by one doesn't increase the time significantly so to speed up computation a step can be given the probability
5. args[5] (100 in this case) is the number of the times the simulation will for each depth and on-probability combination as it is probabilistic process there exist large variation in the time to cross hence the simulation is ran multiple times to average out the variation