Computer Architecture Laboratory
Assignment 0

Consider the scenario where one country, called the defending country (DC), wishes to defend its border against another country, called the attacking country (AC), whose aim is to send an infiltrator to cross the border and enter DC's land. DC decides to deploy a wireless sensor network along the border. If a sensor detects an infiltration attempt, DC can then send its troops to counter the infiltration with "fire and fury" (Trump style!). Quite obviously, the infiltrator would like to enter DC's land without triggering any sensors.

Let's define the model of the different elements:

Border

The border is a long rectangular strip of land as described in Figure 1. The rectangle is discretized into a grid as shown. The length of the rectangle is infinity, while the width is W cells.

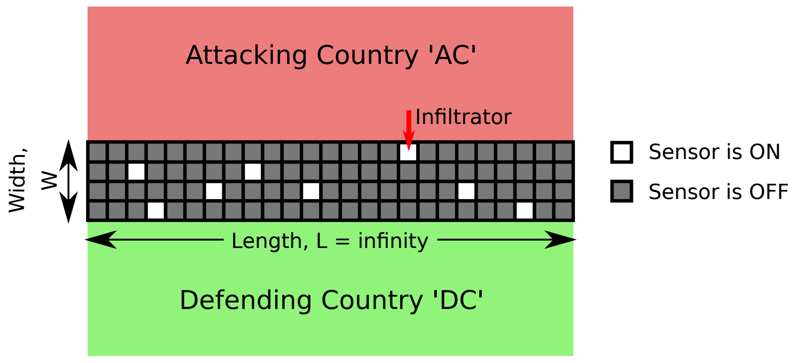

Figure 1: Illustration of the scenario

Sensors

Every cell in the border grid has exactly one sensor. The range of its sensing is limited to the cell. Each sensor is a motion sensor, that is, it can detect an infiltrator who is moving. If the infiltrator is stationary, the sensor *does not* trigger.

The sensors have a fixed battery life. DC wishes to extend their lifetime. To do, it follows a policy of duty cycling. Duty cycling means that the sensor is switched OFF for some periods of time in order to save energy. A sensor takes a decision every 10 seconds, whether or not to stay ON for the next 10 seconds. This decision is taken randomly – a coin, that has probability p of falling heads, is flipped. If heads, the sensor is switched ON for the next 10 seconds. If tails, the sensor it switched OFF for the next 10 seconds. Each sensor takes the decision completely independently, with no communication with another sensor. Every sensor takes its first decision at time '0'.

Infiltrator

The infiltrator moves in steps. In each step, he may move to any of the 8 cells around him. It takes him 9 seconds to move to another cell. So every 10 seconds, the infiltrator spends the first 1 second studying the cells around him, and the next 9 seconds moving (if he decides to move at all).

Remember the motion sensor model. So if the infiltrator decides to move from cell A to cell B, both cells should have their sensor OFF. If even one of them are ON, the infiltrator is caught.

Software Simulation

Your task is to simulate the described border crossing scenario in software. Define classes for each element: Border (you can consider the length of the border to be some large value, say 1000), Sensor and Infiltrator. Also, define a class Clock that captures the time in our simulated world.

Your driving loop would look something like this:

```
t = 0           // refers to time
while (infiltrator has not succeeded and infiltrator has not been caught)
{
        for each sensor s:
                s does some work as determined by time t
        infiltrator does some work as determined by time t
        increment time by 1 second
}
```

Implementation

- Implement the simulator using Java.
- Use Eclipse environment to help with coding and debugging.
- Use Git for version controlling. We have a Git server at https://gitea.iitdh.ac.in/ . Log in using your LDAP credentials.
- Vary p and W, and study how much time it takes for the infiltrator to cross.
- Prepare a document describing your observations. It is recommended you use Latex to prepare the document, and python matplotlib to plot the graphs.

Submit a single zip file named <roll-number-1>_<roll-number-2>_assignment0.zip , that contains the document, as well as a zip file containing your code. Submit only the source files, and not the object (`.class` extension) files.

- Document your code sufficiently. A stranger should be able to pick up your code and immediately start working on improvements!