Delta College

CST 283
Programming Assignment 4

Winter 2024
Instructor: T. Klingler

Objective

This program provides an opportunity to practice use of two-dimensional arrays with a graphical simulation.

Overview & Instructions

The U.S. Forest Service has contracted you to write a program that will simulate the behavior of a forest fire.

Your program should include a two-dimensional grid (array) of forest zones. Your simulation should allow you to see the progression of the fire as it spreads over time.

For the general grid, define a reasonably large drawing window and with that a "substantial" number of rows and columns: At least 40-50 suggested. An odd number of rows and columns would also be suggested to allow there to be an exact middle position.

Variables to define the simulation include:

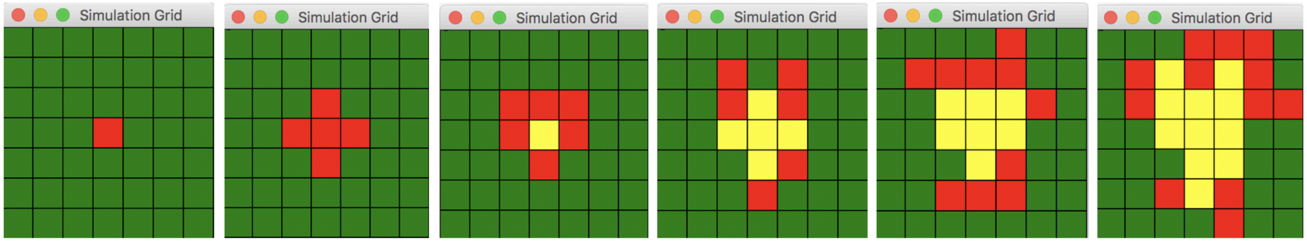
- Probability of fire spreading from a zone on fire to an adjacent zone not yet burning (0.0...1.0)
- Wind direction (from N, S, E, or W). Note: wind directions are always referenced as direction wind is blowing *from*

Create a basic **JavaFX** user interface that will include the ability for the user to set the fire probability and the wind direction, a "go" button to start the simulation, as well as define the simulation drawing area.

The fire will start in the center grid. This will define the first time unit in the simulation cycle. As the simulation progresses, each step should include the following actions for each grid:

- If a grid is on fire (red), and zones adjacent are untouched (green) use a random number generator value to determine if the adjacent zone should catch fire. For example, if a random number (0.0...1.0) is less than the burn probability, then the adjacent grid will be on fire for the next cycle.
- Once a grid catches fire, it will burn for two simulation time cycles. Afterwards, it is considered scorched and should be given another color (yellow as seen below). This area will then not burn again.
- The fire will only spread to the left, right, up, or down (i.e. no diagonals).
- The probability of burning downstream from the wind will increase 10% and upstream from the wind will decrease 10% based on that wind direction. For example, if the general probability of burning is 30% and the wind is from the north, then the simulation probability for burning should be north (up) 20%, east (right) 30%, south (down) 40%, and west (30%). Be sure that the wind adjustment does not push the probability number for a direction outside of the 0.0...1.0 range.

As an example, this could be a snapshot of the behavior of the simulation.



Similar to what is depicted above, your program sould include a visual simulation of the burn event. Utilize a JavaFX drawing Pane (constructed from tools and examples provided in class) to animate the simulation. Each "step" would potentially advance fire, updated the burning grids, and change colors as needed.

The simulation should start at the button click. Retrieve the burn probability and wind direction from the interface, and start the simulation animation. Stop the program simulation when the fire is out (i.e. no "red" grids). The forest service personal (you instructor?) can then examine the extent of the burn (the yellow grids) for their analysis.

The simulation would continue until all cells are visited and then should be terminated. After this completion, use a JavaFX `Alert` dialog box to report the number of simulation cycle time units taken from the start through the last one when the fire is out. Please avoid use of `JOptionPane` dialogs.

Please utilize traditional two-dimensional array(s) for the solution and avoid use of `ArrayList`, `Map`, or other Java Collections classes. You are encouraged to make use of 2-D array(s) of objects, as necessary.

Deliverables

Demonstrate the development steps of your program with at least two version commits to the assignment Git repository.

Deliver the following to the eLearning system **Assignment Dropbox** as your final product:

- **Upload**
  - your **source code** (.java) file(s); preferably zipped in more than one file
  - the **data file** used for testing
