THE UNIVERSITY OF SYDNEY

INFO1103 Assignment 3

Due: 9:00pm Sunday, 5 June 2016

This assignment is worth 10% of your final assessment

Task description

Rural firefighters regularly backburn bushland in a controlled manner to reduce the risk of hazardous bushfires. In this assignment you will implement a simulation of a controlled burn within a region of land. Trees of varying heights are scattered throughout the region and fires are started at particular locations. The simulation operates on a daily time schedule where the fire spreads to neighbouring trees as each day passes according to the wind direction and keeps track of the amount of air pollution.

You are encouraged to ask questions on Ed using the assignments category. As with any assignment, make sure that your work is your own, and you do not share your code or solutions with other students.

Working on your assignment

You can work on this assignment on your own computer or the lab machines. It is important that you continually back up your assignment files onto your own machine, external drives, and in the cloud.

You are encouraged to submit your assignment on Ed while you are in the process of completing it. By submitting you will obtain some feedback of your progress on the sample test cases provided.

Academic declaration

By submitting this assignment you declare the following:

I declare that I have read and understood the University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy, and except where specifically acknowledged, the work contained in this assignment or project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the the Academic Dishonesty and Plagiarism in Coursework Policy, can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

I acknowledge that the School of Information Technologies, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and or communicate a copy of this assignment to a plagiarism checking service or in house computer program, and that a copy of the assignment may be maintained by the service or the School of IT for the purpose of future plagiarism checking.

Implementation details

Your task is to implement a controlled fire simulation based on the descriptions and examples below.

Your program must be contained in `Firebot.java`, `Simulation.java`, `Tree.java` and `Perlin.java` source files and produce no errors when compiled on the lab machines and Ed. The program will run with given arguments, reading from standard input and writing to standard output.

Important – your program will be marked automatically, so make sure that you follow the assignment specifications carefully. Your program output must match the exact output shown in the examples.

Tree attributes

Each tree has the following attributes:

- Fire – Fire intensity between 0 and 9. Tree with fire 0 indicates there is no fire at position.
- Height – Tree height between 0 and 9. Tree with height 0 indicates there is no tree at position.
- Burnt down – Whether or not the tree has completely burnt down from a fire.

Command line arguments

Your program should read in three positive non zero integer command line arguments that specify the random number generator seed, width and height of the land. If any given arguments are missing or invalid then output the following error message and immediately terminate the program. For example:

```
Usage: java Firebot <seed> <width> <height>
```

The simulation starts at day 1 with no wind and no pollution.

Given this information, the program generates a rectangular region of land with trees of varying heights. If you change the `seed` value then a different set of trees will be generated.

You interact with the simulation by entering commands, e.g. displaying the heights of the trees, displaying the fires, starting a fire within a region, advancing the day, setting the wind direction, etc.

The program terminates when the bye command is issued or there is no more input.

Displaying the simulation

There are two ways to display the current state of the simulation. You can show the height of each tree with the `show height`, or the fire intensity of each tree with the `show fire`. In either case, the trees are displayed in a rectangular fashion with a border around the entire region, like so:

```
$ java Firebot 0 8 5
Day: 1
Wind: none

> show height
+---------------+
|    2 2 1 3   2|
|6   3       2  |
|5 4   1 3   1 5|
|3 3       1 3  |
|5   2 2 1   2 4|
+---------------+
```

```
> show fire
+---------------+
|    . . . .   .|
|.   .       .  |
|. .   . .   . .|
|. .       . .  |
|.   . . .   . .|
+---------------+

> fire 4 0
Started a fire
```

```
> show fire
+---------------+
|    . . 1 .   .|
|.   .       .  |
|. .   . .   . .|
|. .       . .  |
|.   . . .   . .|
+---------------+
```

- Trees that have been burnt down will display `x` for both height and fire views.
- Trees with height 0 indicates there is no tree at that location, so be sure not to display 0.
- For the fire view, if there is a tree at that location and no fire then display `.` otherwise display a number indicating the intensity of the fire. Do not display anything when the tree height is 0.

Fire spreading simulation

Initially there will be no fires, but you can start a fire in a particular region by using the `fire` command. This will start a fire with intensity 1 at each tree within the specified rectangular region. The top left tree is at `(0, 0)`, and the bottom right tree is at `(width - 1, height - 1)`.

Each day that progresses, two things will happen: existing fires will continue to burn down the trees, and the fires will spread onto neighbouring trees in the direction of the wind.

Fire spreading simulation

- Initially the tree will not be on fire (`fire = 0`).
- The tree can be lit on fire (`fire = 1`) in two ways: by the `fire` command, or from spreading from a neighbouring tree. If a tree has height 0, it cannot be lit on fire.
- Each day that passes, the fire intensity will increase by 1 until it reaches 9 (thus trees that have been burning for longer will have a higher `fire` value). At that time, the fire intensity cannot increase any more, so the tree will start to burn down by decreasing the height of the tree by 1 for each subsequent day.
- As soon as the height of the tree reaches 0, the tree has completely burnt down and so the fire exists no longer (`fire = 0`, `height = 0` and `isBurntDown = true`).

Fire spreading process

The fire will spread each day according to the current wind direction.
There are 6 different wind directions: all, none, north, south, east and west.

A tree with height > 0 can be lit on fire the next day by spreading from a neighbouring tree. If the wind is in the east direction and a tree at location `(5, 5)` is on fire (`fire > 0`), then the next day the tree (if any) at location `(6, 5)` will be lit on fire (`fire = 1`).

The fires will only spread to neighbouring trees if the wind is in any direction of: all, north, south, east or west. Fires cannot spread to diagonal trees, trees already on fire, or burnt down trees. If the wind direction is all then the fires will spread in all four directions each day.

When applying these rules you should create a copy of the grid that contains the next state.
The next state of the grid should be calculated based on the previous state of the grid.

```
$ java Firebot 2 8 5
Day: 1
Wind: none

> fire 1 3
Started a fire

> wind east
Set wind to east

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      . .     .|
|. 1 . . . .    |
|  .       . .  |
+---------------+

> next
Day: 2
Wind: east

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      . .     .|
|. 2 1 . . .    |
|  .       . .  |
+---------------+

> next
Day: 3
Wind: east

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      . .     .|
|. 3 2 1 . .    |
|  .       . .  |
+---------------+

> wind north
Set wind to north
```

```
> next
Day: 4
Wind: north

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      1 .     .|
|. 4 3 2 . .    |
|  .       . .  |
+---------------+

> wind all
Set wind to all

> next
Day: 5
Wind: all

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      2 1     .|
|1 5 4 3 1 .    |
|  1       . .  |
+---------------+

> wind none
Set wind to none

> next 8
Day: 13
Wind: none

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      9 9     .|
|9 x 9 9 9 .    |
|  9       . .  |
+---------------+
```

```
> show height
+---------------+
|7 6 4   1   1 1|
|  1 6   1      |
|      3 7     6|
|3 x 1 1 2 2    |
|  3       3 2  |
+---------------+

> next
Day: 14
Wind: none

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      9 9     .|
|9 x x x 9 .    |
|  9       . .  |
+---------------+

> show height
+---------------+
|7 6 4   1   1 1|
|  1 6   1      |
|      2 6     6|
|2 x x x 1 2    |
|  2       3 2  |
+---------------+
```

Pollution

The simulation will keep track of the amount of pollution in the air each day. Pollution is initially 0. After the fire burning and spreading procedures are carried out, the pollution level is updated.

For each tree:

- The pollution is lowered by an amount equal to the height of the tree.
- The pollution is raised by an amount equal to the fire intensity of the tree.

After processing all trees, if the pollution level is negative, then set it to 0.

Commands

Your program should implement the following commands, that are all case insensitive.

- If any command that is inputted is unknown or invalid then output: `Invalid command`
- The `fire` and `extinguish` commands accept a region like so:
  - `<x> <y> <width> <height>` – Rectangular region with top left at `(x, y)` and size `(width, height)`. `width` and `height` must be positive and the region must not extend outside the bounds of the land.
  - `<x> <y>` – Single tree at location `(x, y)`.
  - `all` – All trees.

bye

Displays `bye` and terminates the program.

```
> bye
bye
```

help

Displays the help message.

```
> help
BYE
HELP

DATA
STATUS

NEXT <days>
SHOW <attribute>

FIRE <region>
WIND <direction>
EXTINGUISH <region>
```

status

Displays the current day and wind direction.

```
> status
Day: 1
Wind: none
```

data

Displays the fire damage and air pollution statistics.

- Fire damage is the percentage of trees that have burnt down, rounded to 2 decimal places.
- Pollution is a non negative integer value and is updated after every day as explained above.

```
> data
Damage: 12.34%
Pollution: 420
```

next [<days>]

Advances to the next day and carries out the fire burning and pollution rules described above. It can optionally accept a positive integer argument which indicates the number of days to advance by.

The status current day and wind direction are displayed after the next command is run.

```
> next
Day: 2
Wind: none

> next 5
Day: 7
Wind: none
```

show <attribute>

Displays the state of the trees.
A border is displayed around the land and each tree in each row is separated by a space.

If a tree is burnt down, it is displayed as `x`, otherwise depending on the attribute:

- `fire` – shows the fire intensity level of each tree:
  - Trees with height 0 should display nothing.
  - If `fire = 0` then display `.`
  - Otherwise display the fire intensity level.
- `height` – shows the height level of each tree:
  - Trees with height 0 should display a space instead of 0.

wind <direction>

Sets the wind direction. The `direction` can be set to: all, none, north, south, east or west.

```
> wind north
Set wind to north
```

fire <region>

Ignites every tree witin the given region by setting `fire` to 1 for those trees. Only trees that are not already on fire can be lit. Display `Started a fire` or `No fires were started` depending on the outcome.

See above for an explanation of how the region can be specified.

```
Day: 1
Wind: none

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      . .     .|
|. . . . . .    |
|  .       . .  |
+---------------+

> fire 2 2 3 3
Started a fire

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      1 1     .|
|. . 1 1 1 .    |
|  .       . .  |
+---------------+

> fire 2 2 3 3
No fires were started
```

extinguish <region>

Extinguishes all fires in the given region.

If there are no fires within the region, then output `No fires to extinguish`, otherwise extinguish all fires in the region by setting fire to and then display `Extinguished fires`.

```
> fire 0 0
Started a fire

> show fire
+---------------+
|1 . .   .   . .|
|  . .   .      |
|      . .     .|
|. . . . . .    |
|  .       . .  |
+---------------+

> extinguish 1 0
No fires to extinguish

> extinguish 0 0 1 1
Extinguished fires

> extinguish all
No fires to extinguish

> show fire
+---------------+
|. . .   .   . .|
|  . .   .      |
|      . .     .|
|. . . . . .    |
|  .       . .  |
+---------------+
```

Examples

Burning one tree

```
$ java Firebot 0 4 3
Day: 1
Wind: none

> show height
+-------+
|  1   3|
|2 4   1|
|2   2 4|
+-------+

> fire 1 1
Started a fire

> show fire
+-------+
|  .   .|
|. 1   .|
|.   . .|
+-------+

> next 8
Day: 9
Wind: none

> show fire
+-------+
|  .   .|
|. 9   .|
|.   . .|
+-------+
```

```
> show height
+-------+
|  1   3|
|2 4   1|
|2   2 4|
+-------+

> next
Day: 10
Wind: none

> show fire
+-------+
|  .   .|
|. 9   .|
|.   . .|
+-------+

> show height
+-------+
|  1   3|
|2 3   1|
|2   2 4|
+-------+
```

```
> next 3
Day: 13
Wind: none

> show fire
+-------+
|  .   .|
|. x   .|
|.   . .|
+-------+

> show height
+-------+
|  1   3|
|2 x   1|
|2   2 4|
+-------+

> bye
bye
```

Burning east

```
$ java Firebot 0 8 2
Day: 1
Wind: none

> show height
+---------------+
|  4 3 3 2 3 4 3|
|3 6 1   6 2 5  |
+---------------+

> fire 2 0
Started a fire

> show fire
+---------------+
|  . 1 . . . . .|
|. . .   . . .  |
+---------------+

> wind east
Set wind to east

> next 5
Day: 6
Wind: east
```

```
> show fire
+---------------+
|  . 6 5 4 3 2 1|
|. . .   . . .  |
+---------------+

> show height
+---------------+
|  4 3 3 2 3 4 3|
|3 6 1   6 2 5  |
+---------------+

> next 6
Day: 12
Wind: east

> show fire
+---------------+
|  . x 9 9 9 8 7|
|. . .   . . .  |
+---------------+
```

```
> show height
+---------------+
|  4 x 1 1 3 4 3|
|3 6 1   6 2 5  |
+---------------+

> next 9
Day: 21
Wind: east

> show fire
+---------------+
|  . x x x x x x|
|. . .   . . .  |
+---------------+

> show height
+---------------+
|  4 x x x x x x|
|3 6 1   6 2 5  |
+---------------+

> bye
bye
```

Big fire

```
$ java Firebot 1 8 8
Day: 1
Wind: none

> show height
+---------------+
|         3 1   4|
|       2   5   3|
|2 1 1 2   1    |
|3 1 3     3 1  |
|  2     1   3 1|
|5 2     1 4    |
|5       1 1 3 3|
|5   1 1 2 3 2 1|
+---------------+

> fire 0 2
Started a fire

> wind all
Set wind to all

> next 9
Day: 10
Wind: all
```

```
> show fire
+---------------+
|         . .   .|
|       6   .   .|
|9 9 8 7   .    |
|9 8 7     . .  |
|  7     .   . .|
|5 6     . .    |
|4       . . . .|
|3   . . . . . .|
+---------------+

> next 4
Day: 14
Wind: all

> show fire
+---------------+
|         . .   .|
|       9   .   .|
|x x x x   .    |
|x x 9     . .  |
|  x     .   . .|
|9 9     . .    |
|8       . . . .|
|7   . . . . . .|
+---------------+
```

```
> extinguish all
Extinguished fires

> show fire
+---------------+
|         . .   .|
|       .   .   .|
|x x x x   .    |
|x x .     . .  |
|  x     .   . .|
|. .     . .    |
|.       . . . .|
|.   . . . . . .|
+---------------+

> show height
+---------------+
|         3 1   4|
|       1   5   3|
|x x x x   1    |
|x x 1     3 1  |
|  x     1   3 1|
|5 1     1 4    |
|5       1 1 3 3|
|5   1 1 2 3 2 1|
+---------------+

> bye
bye
```

Writing your own testcases

We have provided you with some sample testcases but these do not test all the functionality described in the assignment. It is important that you thoroughly test your code by writing your own testcases.

You should place all of your test cases in a `tests/` directory. Ensure that each test case has the `.in` input file along with a corresponding `.out` output file. We recommend that the names of your test cases are descriptive so that you know what each is testing, e.g. `fire.in` and `westwind.in`

Submission details

Final deliverable for the correctness is due in week 13. Further details will be announced on Ed.

You must submit your code in the assignment page on Ed. To submit, simply place your code into the workspace, then click the run button to check your program works and then click the submit button.

You are encouraged to submit multiple times, but only your last submission will be marked.

Marking

10 **marks** are assigned based on automatic tests for the *correctness* of your program.
This component will use our own hidden test cases that cover every aspect of the specification.

Your program will be marked automatically, so make sure that you carefully follow the assignment specifications. Your program output must match the exact output shown in the examples. You are encouraged to submit your assignment while you are working on it, so you can obtain some feedback.

Warning: Any attempts to deceive or disrupt the marking system will result in an immediate zero for the entire assignment. Negative marks can be assigned if you do not properly follow the assignment specification, or your code is unnecessarily or deliberately obfuscated.