

Touchlab Programming Task

As part of your application process, please complete the following programming task. Comprehending the task as written, good coding style, as well as communication are more important than getting everything exactly right, so stress what's important (quality code, etc). In this zip file is a gradle project. See the program input sections for how to invoke the task. If you would like to use a different project structure or build system that's fine as long as it is structured well.

Grid animation

A grid animation system. The program should receive four arguments. The first argument is a file containing descriptions of "Actors" to be included in the animation. The second and third are the width and height of the grid. The fourth and final argument is the number of "frames" to run.

After initializing the actors, the program should run through the specified number of frames. Moving the actors according to the behavior defined below, and printing their status.

The data file for the actors will include one actor per line. Each actor is described by up to 4 parameters which are "type", "row", "column", "direction". All actors will have a type, and based on the type they may or may not use the other parameters. If a parameter is not used by a given actor, it may or may not be included in the input.

row, column

Zero based. Describe the position on the grid that the actor starts at

direction

The direction an actor will move. A number 0-7, indicating up, up-right, right, down-right, down, down-left, left, up-left.

type

One of the following:

- 'L' (Line) - Moves in the specified direction, 1 move per frame
- 'S' (Still) - Ignore direction. Don't move.
- 'VL' (Veer Left) - Turns **counter-clockwise** in an ever widening fashion. On the first frame it will turn one direction counter-clockwise (decrease direction by one) before moving forward. The second frame it will stay the same direction and just move forward. It will turn again on the third frame then stay the same direction for BOTH the 4th and 5th frame before turning again on the 6th. Stay the same direction for the next 3 frames,

then turn again on the 10th. Add one more frame to skip each time (skip 4, then skip 5, then skip 6, etc).

- **'VR'** (Veer Right) - Same thing as Veer Left, but turn **clockwise** (increase direction by one) instead.
- **'R'** (Random) - Pick a new spot randomly anywhere on the grid for each frame.

Notes

The most complicated parts of this are really the 'Veer' types, so if things are running long, you can make those simpler (maybe just have them turn each frame, or every other frame, rather than changing their values). Getting the output EXACTLY right is less important than understanding the basic idea of the specs and seeing how you approach the problem.

Here we are looking for spec comprehension and OOP skill.

Program Inputs:

`./gradlew run -PappArgs="testgrid.txt 10 10 3"`

Arguments:

1. Filename for actor descriptions
2. int width of grid
3. int height of grid
4. int number of frames to run

Sample actor file: you can find this data in the file **testgrid.txt** in the zip file.

L,2,5,3
VR,4,2,5
S,1,1
R
L,3,3,2

Program Output:

The program should output the frame, type, and position of each actor, to the standard output Stream.

For the data set above, with a width and height of 10, and 3 frames:

0,L,2,5
0,VR,4,2
0,S,1,1

0,R,5,9
0,L,3,3
1,L,3,6
1,VR,4,1
1,S,1,1
1,R,3,2
1,L,3,4
2,L,4,7
2,VR,4,0
2,S,1,1
2,R,5,0
2,L,3,5

