

Robot Control Game

DeNA wants to find the very best and brightest to join our fast growing team in Vancouver. We're after talented, imaginative, passionate and motivated individuals with a deep interest in gaming and we're sweeping aside those dry interview test questions and offering you a chance to impress us with a full range of your abilities.

While the challenge presented here might serve to power the mechanics of a real game, you are not expected to develop an actual game. No complex view layers are required. We just want to see the core logic work.

About Us

Our stack uses javascript from top to bottom. We use modern software methodologies to ensure our products are the very best in the industry. We are confident in our products. They are robust, easy to maintain, refactor and deploy. Our customers love our games because they are fun and challenging, but easy to play.

Submission

We expect the qualities described above from you. We encourage you to show us your stuff in javascript; however, you may submit in a language of your choice. Please include all your work including notes, etc as well as a readme indicating how we run your project.

The Task

We want you to implement remotely controlled robots. The robots are very simplistic, but there are two of them, a ground vehicle, and an airborne vehicle.

Ground robots can do the following:

- Right turn: 90 degrees.
- Left turn: 90 degrees.
- Move forward 1 meter.
- Move backward 1 meter.

Airborne robots can do the following:

- Turn right 90 degrees.
- Turn left 90 degrees.
- Move forward 1 meter.
- Move higher by 10 meters.
- Move lower by 10 meters.

Robots of any type can move around on a grid in 1m increments, where each grid is 1m x 1m.

We like to work in retro-ways and feed our robots tickertapes of command sequences - the individual commands follow:

- R: turn right.
- L: turn left.
- F: move forward.
- B: move backward.
- U: move higher.
- D: move lower.

What we want to do is control to a robot

```
ground RFFFFLLFFFFRFFFF
```

or

```
air URFFFLLLUUDDDD
```

This moves the robots around the grid.

Finally, we can request that the robot tell us the direction it is facing, and which part of the grid it is on. It will tell us the x,y,z position of the coordinate, and whether it is facing N,E,S,W (north, east, south, or west)

For example, if we position the robot at the starting point on the bottom left hand corner of the grid (0,0) facing North, and issue the following command FFR, then the robot will report that its position is: 0,2,0 and its direction is E.

Implement a solution that allows us to instantiate a robot, move it around and ask it for its position and direction. Input and output can be very basic, and no visualization is needed. You can even hard code some input and output for various cases.