Rover On Mars

Design

1. Command Stream:

It is given for a purpose, although it’s not the shortest path to the destination, but it is necessary since Mars grids would have different altitude, obstacles that Rover should circumvent to save energy or to overcome.

Hence directly shorting the command stream with the shortest path to destination is not practical, since I assume the command stream assigned to Rover is already in general pre-optimized to save energy or overcome obstacles.

We calculate the shortest path only when it comes to rover-out-of-boundary and another-deployed-rover-in-the-way scenario, we would use the valid shortest path to update the command stream to get to our destination.

1. Input:

The input would always be validated. It will keep prompting user to re-enter until the input is valid.

1. Grid

Since it’s more practical to have Mars consist of multiple groups of grid to more easily command the Rovers, the X, Y dimension of the given Mars grid would not be taken into consideration as rounded in this design, i.e. (if Rover wants to Go Right on Grid X Max, it would be out of bound, instead of Go to Grid X 0)

1. Solution:

As stated in Command Stream, we would only calculate the shortest path to circumvent only when necessary. i.e. when rover-out-of-boundary and another-deployed-rover-in-the-way scenario.

**rover-out-of-boundary**

Grid

**another-deployed-rover-in-the-way**

Grid

When those scenario occurs, we would stop the command, calculate the next command which Rover would have accumulated valid position.

And calculate the needed commands using shortest path from current last valid command position to next valid command position.

After calculation, we would insert the needed additional commands into command stream to circumvent the obstacles using the shortest path (BFS)

And then we would resume to executing the updated command stream.

1. Design Overview
2. Rover

The Rover class would have basic information of Rover status, like position, direction.

And it would also have basic move method, but how to move is the job of the Command Center.

On initialization, it would be assigned to a grid, and a command center.

It also has a class property of a list rovers, so we could easily fetch the Rover with a rover Id

1. Grid

The Grid class would have basic information of the Grid. It also has knowledge and notify Command Center that whether each position in a grid is vacant or taken by a Rover with rover Id.

1. Command Center

The Singleton Class Command Center is responsible of parsing incoming commands, inserting additional commands as needed using shortest path, executing commands, and consuming Grid and Rover Information, and notify Grid and Rover to update their information.

1. Input

The input is designed in such a way that it is always require a valid entry, otherwise it would keep prompting user to re-enter

1. Test

All class and helper methods should be unit tested, and each scenario of rover-deployment should be tested (in RoverTests.cs)