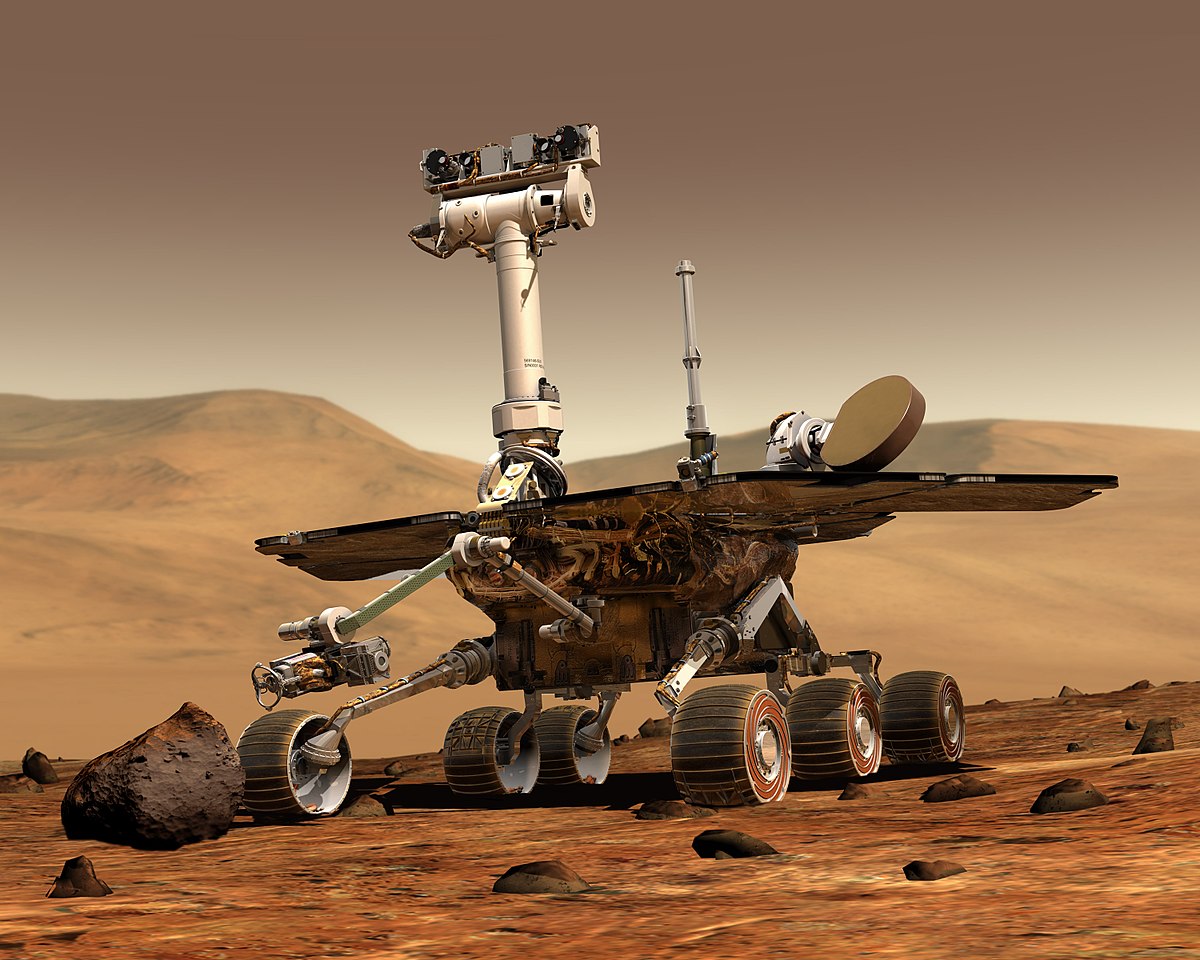
 **MARS ROVERS** 

1. **Introduction**

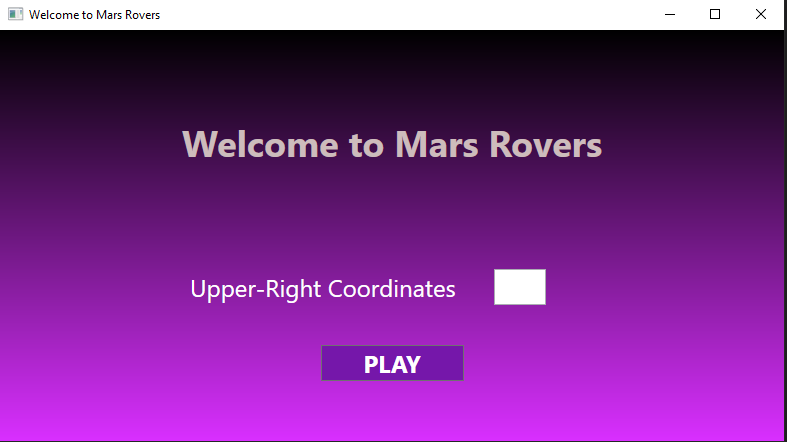
Mars Rover is a mini-game simulating real rovers. Rovers have wheels and specialize in moving around. They land on the surface of Mars and drive around to different spots. Rovers help scientists in their quest to understand what different parts of the planet are made of. The robots also have camera on the top of their heads. The cameras have extension to show the direction at that they are looking at.

1. **Development**

The Project is made by using the UI FrameworkWindows Presentation Foundation (**WPF**) for the frontend and C# for the backend.

2.1 ) Welcome to Mars Rover

**Start Window**

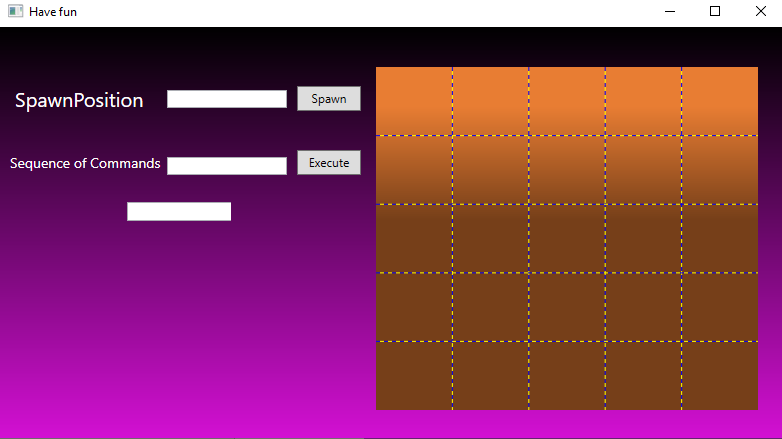


(1)

There is in an input field for Upper-Right Coordinates of the plateau on which robots land. Upper-Right Coordinates are the maximum X and Y of the plateau which actually defines the size of the plateau. The input should be in type "{integer}{integer}". The first integer is the X of the grid and the second is the Y.

2.2 ) Start exploring

**Playground**



(2)

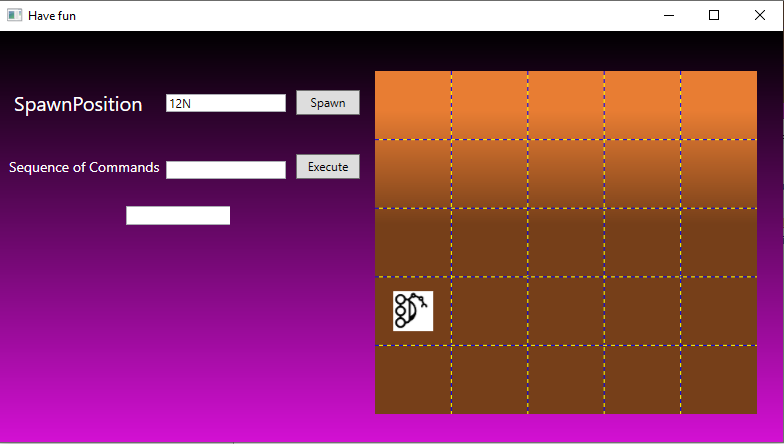
The Grid on the right is the plateau. It is X:5 by Y:5 which means the user input is "55".

After we have the plateau it is time to start spawning Mars Rovers. We have input field for "SpawnPosition". It takes as input "{integer}{integer}{char}". The first integer is the X and the second is Y at which the first rover should be spawned. The char stays for the direction(North = N,South = S,East = E, West = W) of the camera. There are how the rover image looks accordingly to its direction.

N E S W



**We input "12N" and press Spawn.**



(3)

The "sequence of commands" input is sequence of moves sent to the Rover on Mars so it can move around.

**Possible moves**

'L' and 'R' make the rover spin 90 degrees left or right respectively, without moving from its

current spot.

'M' means move forward one grid point, and maintain the same heading. Assume that the square

directly North from (x, y) is (x, y+1).

**Test input: LMLMLMLMM**

**Output**



(4)

After that you can keep spawning Rovers if you want. The previous one disappears when you spawn a new one.

1. **User Friendly Explanation**

You have to start Rover.exe. Choose x and y for your plateau and type them in the input form.(1) It should be in format "{x}{y}". Press play. Choose x,y and direction for your Rover's spawn.(2) It should be in format "{x}{y}{direction}". Make up sequence of moves and pass them to the input form and press Execute. You get X:Y:Direction of the rover after the moves. (4)

1. **Future Updates**

I am working on controller which makes the simulator much more efficient, because it shows the result and changes the image after every move. It is almost ready but I met one bug and unfortunately it is still not available.