



Backend Engineer - Senior

Daniel Santos Silva Júnior

daniel.junior.ifpb@gmail.com

Test Rover

O objetivo central do desafio é desenvolver uma aplicação capaz de receber uma série de instruções a serem movimentadas por um Rover, e após realizar essas instruções exibir a posição final dele.

Solução

Para solucionar o problema, desenvolvi uma classe chamada Rover que armazena os dados de posicionamento e direção de um Rover. É nela onde são implementados os métodos de comportamento de um Rover, como : rotacionar para direita ou esquerda assim como se mover para frente. O método StartMoving é responsável por receber a string com a sequência de ações a serem tomadas e de fato executar. A seguir, é possível visualizar a implementação desse método que é o centro da aplicação.

```

66   StartMoving(xLimit, yLimit, moves) {
67
68       for (let move of moves) {
69           switch (move) {
70               case 'M':
71                   this.MoveInSameDirection();
72                   break;
73               case 'L':
74                   this.Rotate90Left();
75                   break;
76               case 'R':
77                   this.Rotate90Right();
78                   break;
79               default:
80                   console.log("Invalid command");
81                   break;
82           }
83       }
84
85       if (this.X < 0 || this.X > xLimit || this.Y < 0 || this.Y > yLimit) {
86           console.log("Position out of boundaries: ", this.X, this.Y, xLimit, yLimit);
87       }
88   }
89 }

```

Teste

Com o objetivo de verificar se os métodos comportamentais da classe estavam funcionando como o esperado, implementei no arquivo `indexTest` a realização de testes unitários como pode ser observado abaixo.

```

1   const Rover = require('./Rover')
2
3   function moving(xLimit, yLimit, startX, startY, startDir, directions) {
4       const rover = new Rover(startX, startY, startDir);
5       rover.StartMoving(xLimit, yLimit, directions);
6       const {X, Y, Direction} = rover;
7       return [X, Y, Direction]
8   }
9
10  test(`Rover starting at "1 2 N" moving with LMLMLMLMM expected: 1, 3, N`, () => {
11      expect(moving(5, 5, 1, 2, 'N', 'LMLMLMLMM')).toEqual([1, 3, 'N']);
12  });
13
14  test(`Rover starting at "3 3 E" moving with MMRMMRMRRM expected: 5, 1, E`, () => {
15      expect(moving(5, 5, 3, 3, 'E', 'MMRMMRMRRM')).toEqual([5, 1, 'E']);
16  });

```

Execução da aplicação

Para executar a aplicação, é necessário adicionar um arquivo txt, nomeado de inputFile na pasta /tests, é necessário ter o nodejs instalado e o npm. Ao executar o comando npm install serão baixadas as dependências necessárias. Feito isso ao executar o comando npm run start será executada a aplicação e exibida na saída padrão o resultado esperado.

Para execução dos testes é necessário executar o comando npm run test.