// rover.spec.js

const assert = require('assert');

const Message = require('../message.js');

const Command = require('../command.js');

const Rover = require('../rover.js');

const commands = [new Command('MODE\_CHANGE', 'LOW\_POWER'), new Command('STATUS\_CHECK')];

let message = new Message('e1', commands);

const rover = new Rover(88929237);

let response = rover.receiveMessage(message);

console.log("response top: ", response);

// Test 5 - constructor sets position and default values for mode and generatorWatts

describe("constructor sets position and default values for mode and generatorWatts", function() {

it("constructor sets position and default values for mode and generatorWatts", function() {

assert.strictEqual(rover.mode, 'NORMAL');assert.strictEqual(rover.generatorWatts, 110);

assert.strictEqual(rover.position, 88929237);

});

});

// Test 6 - response returned by receiveMessage contains name of message

describe("response returned by receiveMessage contains name of message", function() {

let getName = true;

let nameVar = rover.receiveMessage(message,getName);

it("return name of message", function() {

assert.strictEqual(message.name, nameVar);

});

});

// Test 7 - response returned by receiveMessage includes two results, if two commands are sent in message

describe("response returned by receiveMessage includes two results, if two commands are sent in message", function() {

let numCommands = message.commands.length;

let checkLength = 'length';

// console.log("message: ", message);

twoCommands = rover.receiveMessage(message, numCommands, checkLength);

console.log(twoCommands);

it("check for 2 results returned if two commands sent", function() {

assert.strictEqual(numCommands, twoCommands.commands.length);

});

});

// Test 8 - responds correctly to status check - STATUS\_CHECK command then receiveMessage(message) returns an object with all 4 properties

// describe("responds correctly to status check", function() {

// it("responds correctly to status check", function() {

// let someVar = rover.checkStatus(message);

// assert.strictEqual(someVar.completed, true);

// assert.strictEqual(someVar.mode, 'NORMAL');

// assert.strictEqual(someVar.generatorWatts, 110);

// assert.strictEqual(rover.position, 88929237);

// });

// });

// Test 9 - responds with correct status after MODE\_CHANGE

// describe("responds with correct status after MODE\_CHANGE", function() {

// it("responds with correct status after MODE\_CHANGE", function() {

// let otherVar = rover.receiveMessage(message);

// console.log("response: ", response);

// console.log(otherVar);

// assert.strictEqual(otherVar.completed, true);

// // Is this supposed to compare the rover.mode to NORMAL or the new status to LOW\_POWER?

// assert.strictEqual(otherVar.mode, 'LOW\_POWER');

// });

// });

// Test 10 - responds with false completed value, if attempt to move while in LOW\_POWER mode

// console.log("commands: ", commands);

// console.log("\ntype of commands: ", typeof commands);

// console.log("\ncommands[0]: ", commands[0]);

// console.log("\nrover: ", rover);

// describe("responds with false completed value, if attempt to move while in LOW\_POWER mode", function() {

// let roversPosition = rover.position;

// rover.checkMove(message);

// console.log("rover after checkMove: ", rover);

// it("responds with false completed value, if attempt to move while in LOW\_POWER mode", function() {

// assert.strictEqual(rover.completed, false) && assert.strictEqual(rover.position, roversPosition);

// });

// });

// Test 11 - responds with position for move command