// rover.js

class Rover {

constructor(position) {

this.position = position;

this.mode = 'NORMAL';

this.generatorWatts = 110;

}

receiveMessage(message, inputVar, lengthVar) {

const resultsArray = [];

if (inputVar === true) {

let theName = message.name;

return theName;

}

// if (inputVar === 2) {

// return {

// }

// }

if (lengthVar === 'length') {

let newArray = [];

for (let i=0; i<inputVar; i++) {

newArray[i].push(message.commands[i].commandType);

// if (message.commands[i].commandType === 'MODE\_CHANGE') {

// let theValue = message.commands[i].value;

// }

}

// return {

// mode: theValue,

// completed: true

// }

console.log(newArray);

return newArray;

}

for (let i=0; i<message.commands.length; i++) {

// console.log(message);

if (message.commands[i].commandType === 'MODE\_CHANGE' && message.commands[i].value === 'LOW\_NORMAL') {

return {

mode: 'LOW\_NORMAL',

completed: true

}

}

}

}

checkStatus(message) {

return {

mode: this.mode,

generatorWatts: this.generatorWatts,

completed: true

// Should I also return the position?

}

}

// checkModeChange(message) {

// this.completed = false;

// for (let i=0; i<message.commands.length; i++) {

// if (message.commands[i].commandType === 'MODE\_CHANGE') {

// this.completed = true;

// // this.mode = message.commands[i].value;

// }

// }

// return {

// mode: this.mode,

// completed: true

// }

// }

// checkMove(message) {

// // if rover mode is LOW\_POWER change completed to false, position does not change

// this.completed = false;

// for (let i=0; i<message.commands.length; i++) {

// if (message.commands[i].commandType === 'MOVE' && this.mode === 'LOW\_POWER') {

// this.completed = false;

// }

// }

// return(message);

// }

// return(message);

}

module.exports = Rover;