

Mars Rover Mission

Aquest projecte s'ha creat amb Laravel per a la lògica del codi i amb html, css i amb les plantilles Blade per al Frontend.

Primerament quan entrem a l'aplicació ens mostra un missatge de benvinguda mentre en segon pla ens genera la Matriu que formarà el nostre planeta a explorar. Com deia a l'enunciat del problema, les seves dimensions són de 200x200, encara que es podria modificar aquest valor pel que es volgués. Els valors de la matriu seran aleatoris i poden tenir 2 valors: 0 si no hi ha cap obstacle i 1 si hi ha obstacle. Aquesta matriu es guarda en una variable de sessió per a poder obtenir la seva informació més endavant.

Un cop li dones al botó de la pàgina inicial, et porta a la pàgina del formulari. Aquesta demana que s'introdueixi la coordenada X i Y inicial i la Direcció cap a on està mirant el Rover inicialment.

Adicionalment, també es demana que s'introdueixin les ordres de moviment del Rover:

- F: moviment endavant.
- R: Gir a la dreta i moviment endavant.
- L: Gir a l'esquerra i moviment endavant.

En el formulari s'ha posat un filtre per tal de que les dades introduïdes siguin correctes:

- S'ha establert que les coordenades inicials estiguin entre el 0 i la coordenada màxima de la matriu.
- S'han limitat els caràcters de la direcció inicial a les lletres N,E,S,W.
- S'accepten tant en minúscules com en majúscules, al backend es comprova que siguin majúscules i sinó es transformen.
- Les ordres estan limitades als caràcters F,R,L tan en majúscula com en minúscula i com en la direcció ens assegurarem que siguin admeses les dues opcions.

Un cop introduïdes les dades, son enviades al backend. La cadena de caràcters que seran les ordres donades es divideixen de forma individual perquè sigui més fàcil accedir a aquesta informació.

Dins del backend, cridem la funció calculaDireccio on li passarem la direcció, i l'ordre actual. Aquesta funció determina quina serà la nova direcció a la qual estarà mirant el Rover.

Tot seguit es fa la crida a la funció calculaPosició a la que li passarem la direcció, la posició actual (X i Y) i les dimensions de la Matriu (maxx i maxy). Així aquesta funció ens retornarà la nova posició on s'hauria de desplaçar el Rover. Per a això modificarem la posició en l'eix X o Y sigui sumant o restant una posició en un dels eixos.

Un cop tenim aquesta informació, recuperarem la informació de la matriu, que tenim guardada en una variable de sessió. En aquest punt es busca la posició inicial en la matriu i ens

assegurem que estigui inicialitzat a 0 (si el Rover es troba en aquesta posició no hi hauria d'haver obstacle).

En el cas de que sigui la primera vegada que l'usuari introdueix ordres, s'inicialitzaran unes arrays on guardarem la informació de les caselles on s'ha desplaçat el Rover i les que caselles on s'han trobat obstacles.

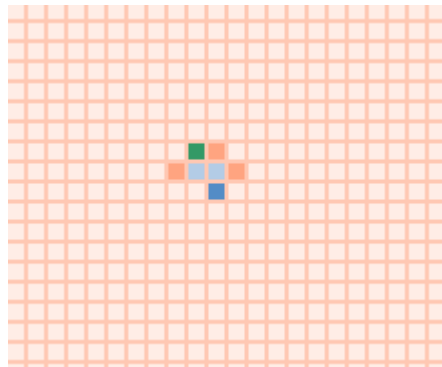
En el cas que sigui una ordre posterior recuperarem la informació emmagatzemada en una variable de sessió.

Ara per a cada ordre que s'hagi donat, haurem de fer els següents passos:

- Es comprova aquesta nova posició en la matriu si és un 1 o un 0
- En el cas que sigui un 0 es segueix amb la següent ordre
- En el cas que sigui un 1 s'anul·len la resta d'ordres
- Un cop ha acabat de fer les ordres o s'han anul·lat, es mostra per pantalla els moviments realitzats, la nova posició i en cas d'obstacle, la posició d'aquest i l'ordre realitzada que provocava la col·lisió.

Per a fer l'apartat de resultats molt més visual, s'ha fet una representació en forma de graella dels resultats. S'ha establert una llegenda de colors per a representar les 4 tipologies de caselles:

- Blau: la posició inicial del Rover
- Verd: Posició actual del Rover
- Blau cel: Caselles per on ha passat el Rover
- Taronja: Obstacles que s'ha trobat el Rover



S'ha establert que la representació gràfica de la matriu sigui d'una de 200x200. Si deixéssim que el programa mostrés una matriu més gran podria portar problemes de rendiment i fins i tot col·lapsar el servidor.

Un cop obtinguts els resultats, just a sota de la graella tenim un altre formulari per a continuar donant ordres al Rover, en aquesta nova iteració, la posició inicial i la direcció inicial seran definides per la direcció i posició que tingui el Rover en aquell moment.

Finalment, a sota de tot de la pàgina de resultats tenim un botó que ens permet canviar de planeta. Això ens torna a la pàgina d'inici, que ens generarà una nova matriu aleatòria.

A causa de l'aleatorietat de la Matriu, és possible que el Rover es trobi en una posició totalment bloquejada i que no es pugui desplaçar ni cap a la dreta, ni cap a l'esquerra ni recte. Per a aquest motiu, s'ha programat perquè primer realitzi el gir, i després intenti el desplaçament. Si ho fes tot en el mateix pas, el Rover es podria quedar bloquejat, ja que no disposa de marxa enrere, fent fracassar la missió d'explorar un planeta remot.

També s'ha tingut en compte un detall i és que si el Rover es troba en un dels laterals de la matriu, per exemple en la posició (0,56) i s'intenta moure's cap a la posició (-1,56) aquest sortiria de la matriu donant error. Aquí s'ha fet la interpretació que com els planetes són esfèrics, si intentés moure's cap a la posició (-1,56) en realitat ho farà a la posició de la matriu (199,56), que és a l'última columna de la matriu donant així la volta al planeta.