

Maze Game

El meu primer joc en java

By Lua Trevin Hernaiz

M03 Programació & M05 Entorns de desenvolupament
Institut TIC de Barcelona

Abril 2024



Descripció del projecte

En aquest projecte he desenvolupat un joc sense UI, s'executa i s'utilitza a un Terminal.

El joc consisteix en sortir d'un laberint:

Al jugador se li presenta un laberint amb diverses sortides i una única entrada. La complexitat del joc neix en que el laberint també compta amb parets que no són visibles fins que el jugador hi colisiona.

El transcurs del joc es basa en un prompt que recull el moviment que vol fer el jugador i posteriorment imprimeix el resultat del mapa tras fer el moviment.

Si el jugador es xoca, es suma un intent.

El joc acaba quan el jugador arriba a una sortida o decideix deixar de jugar.

Cada laberint compta amb un rècord d'intents. Aquestos estan enmagatzemats en un fitxer csv.

Quan el jugador guanya es compara la puntuació obtinguda amb el rècord, si la superat s'actualitza

Funcionalitats base

Inici de joc

Al jugador se li mostra un header amb info del joc (quin laberint s'està jugant, si existeix un rècord i com es mostra el menú d'ajuda)

```
Joc del laberint
=====
H: mostra ajuda

Laberint: laberint02
Rècord actual: lua en 1 intents
```

Prompt

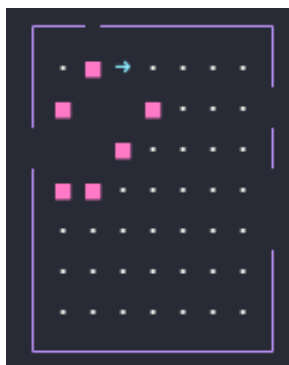
És l'encarregat de recollir el que vol fer el jugador. Compta amb info sobre el nom del joc, quin laberint s'està jugant i quants intents porta fins al moment el jugador.

```
MazeGame ~ laberint02 1 tries
```

Menú d'ajuda

S'ha implementat un menú d'ajuda per orientar al jugador en la partida. Aquest surt quan el jugador introdueix caràcters invàlids o quan s'utilitza la opció h

```
MazeGame ~ laberint02 1 tries
h
Les opcions disponibles són:
H: Mostra aquest text d'ajuda
L: gira a l'esquerra
R: gira a la dreta
F: mou una passa endavant
nF: mou n passes endavant
Q: Sortir
```



Mapa

El mapa compta amb diverses icones escollides per a que resultin intuïtives per al jugador. Compta amb parets, blocs, el jugador, caselles per a on ha passat el jugador i caselles per descobrir.

Aquest es mostra cada cop que el jugador realitza un moviment vàlid.

Per cada tipus de caràcter s'han implementat color per a que resulti més senzill i agradable visualment.

Moviment Jugador

El prompt recull el que vol fer el jugador, els casos vàlids són q (sortit), h (menú d'ajuda) o un moviment. En cas de ser un moviment vàlid (més info sobre els invàlids a la secció de control d'errors) S'actualitza la posició del jugador i es torna a imprimir el mapa.

Les quatre casolístiques després de moure's són:

- El jugador **arriba a una porta** per tant guanya
- El jugador **es xoca amb un bloc**, es suma un intent i es retorna a la posició inicial
- El jugador **gira**, no s'ha de comprovar la col·lisió, es torna a mostrar el mapa i s'espera el següent moviment
- El jugador **arriba a una posició lliure**, es torna a mostrar el mapa i s'espera el següent moviment

Rècord

Si el jugador guanya, es comprova si ha superat el rècord d'intents. Si l'ha superat se li pregunta el nom per guardar un nou rècord, si no se li dona ànims per intentar superar-lo.

Ampliacions

Colors

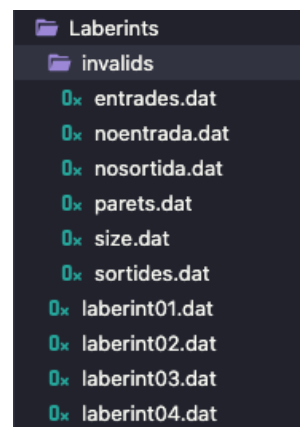
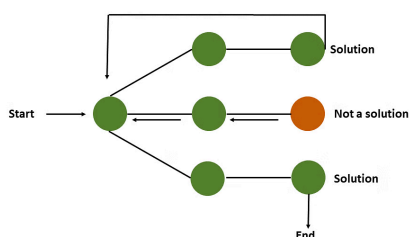
Per a facilitar la jugabilitat i fer més el joc més agradable de jugar s'han implementat Colors. Aixó es fa gràcies a la classe Log.

La classe log compta amb una sèrie de mètodes que simplement imprimeixen les dades rebudes amb els codis de color de terminal (guardats en variables).

Implementació detallada a l'esquema UML de classes.

Laberint solucionable

Mitjançant backtracking, s'ha implementat una funció recursiva que comprova si el laberint té solució. Des de l'inici comproba tots els camins possibles desde les posicions adjacents. Si a la crida inclou una trucada que acaba en una posició no vàlida (fora el mapa, parets o blocs) torna a l'últim punt vàlid.



Makefile

El makefile es un Arxiu que compta amb una sèrie de regles i comandaments a executar. En aquest cas s'ha definit una regla per compilar, per borrar els arxius temporals i per executar (amb argument de quin laberint es vol jugar).

He decidit implementar-ho perquè:

A l'hora de desenvolupar facilita molt la feina ja que **només compila els arxius modificats des de l'últim cop**. Aixó ho fa comparant les dates de modificació dels objectes i el codi.

L'usuari no necessita saber compilar, només executar. Aixó es degut a que les regles tenen dependències. Quan s'indica que es vol executar el projecte, abans comprova si ha de compilar. En cas de necessitar-ho compila i després executa el joc.

La neteja del projecte. En aquest cas s'ha establert que els arxius objecte vagin a una carpeta bin. Aixó es molt útil perquè aquest arxiu només ocupen espai innecessari un cop acabat de jugar i amb la regla de neteja simplement es borra el directori.

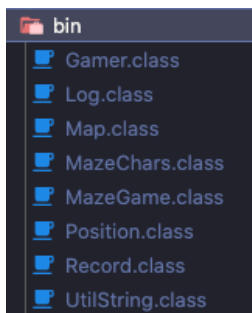
Múltiples girs

Al igual que quan avances, també pots girar diverses vegades indicant el número davant de la lletra.

Control d'errors

Estructura del projecte

A l'arrel projecte comptem amb diversos archius i directoris:



Bin

Es crea quan s'executa el joc i es borra quan s'executa la regla de neteja del makeFile. Conté tots els fitxers class del projecte.

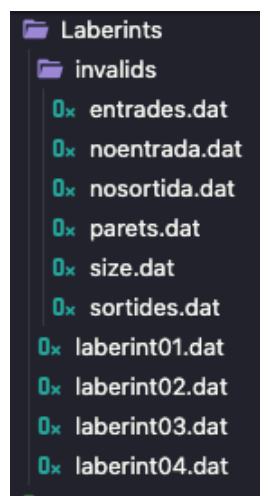
La decisió de deixar-ho tot a una carpeta fa que l'arrel del projecte no s'ompli d'archius temporals.

Laberints

Són tots els fitxers amb les dades dels laberints. Compta amb un subdirector i a on estan classificats els archius invàlids.

Cada laberint conté una primera línia amb el tamany i després un seguit de caràcters:

- X - Blocs i parets
- G - Sortides
- E - Entrada



src

Carpeta amb tot el codi del projecte.

Cada classe està a un fitxer individual.

Altres fitxers

També comptem amb:

- Gitignore:** que s'encarrega de que no es pugin els .class al repositori
- Makefile:** Més info a les [ampliacions](#)
- readme.md:** Per donar info del projecte al repositori
- records.csv:** Per enmagatzemar els records dels laberints
- docs.pdf:** Aquest propi document

