



Maze Game

El meu primer joc en java

By Lua Trevin Hernaiz

CFGs Desenvolupament d'aplicacions multiplataforma

Institut TIC de Barcelona

M03 Programació | M05 Entorns de desenvolupament

Abril 2024





Descripció del projecte	3
Funcionalitats base	4
Inici de joc	4
Prompt	4
Menú d'ajuda	4
Mapa	4
Moviment Jugador	4
Rècord	5
Ampliacions	5
Colors	5
Laberint solucionable	5
Makefile	5
Múltiples girs	5
Control d'errors	6
Laberints	6
Moviments	6
Estructura del projecte	7
Bin	7
Laberints	7
src	7
Altres fitxers	7
Conclusions	8



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 llaberint:

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 r  cord i com es mostra el men   d'ajuda)

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

Laberint: laberint02
R  cord actual: lua en 1 intents
```

Prompt

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

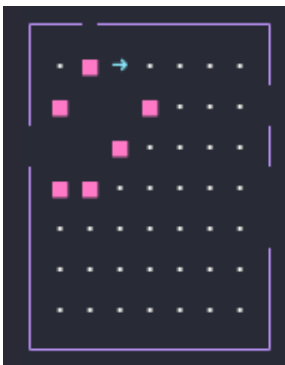
```
MazeGame ~ laberint02 1 tries
|
```

Men   d'ajuda

S'ha implementat un men   d'ajuda per orientar al jugador en la partida. Aquest surt Quan el jugador introdueix car  cters inv  lids o quan s'utilitza la opci   h

```
MazeGame ~ laberint02 1 tries
h

Les opcions disponibles s  n:
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. Compa amb parets, blocs, el jugador, caselles per a   n ha passat el jugador i caselles per descobrir.

Aquest es mostra cada cop que el jugador realitza un moviment v  lid.

Per cada tipus de car  cter s'han implementat color per a que resulti m  s senzill i agradable visualment.

Moviment Jugador

El prompt recull el que vol fer el jugador, els casos v  lids s  n q (sortit), h (men   d'ajuda) o un moviment. En cas de ser un moviment v  lid (m  s info sobre els inv  lids a la secci   de control d'errors) S'actualitza la posici   del jugador i es torna a imprimir el mapa.

Les quatre cas  listiques despr  s de moure's s  n:

- 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 posici   inicial
- El jugador **gira**, no s'ha de comprovar la col  lisi  , es torna a mostrar el mapa i s'espera el seg  ent moviment



- El jugador **arriba a una posició lliure**, es torna a mostrar el mapa i s'espera el següent moviment

Récord

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

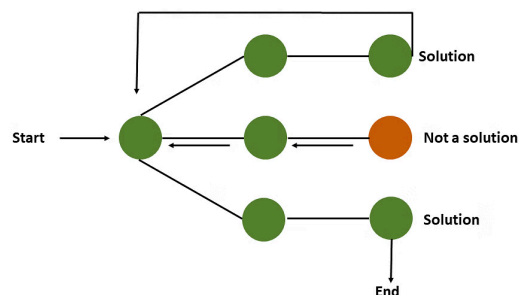
Per a facilitar la jugabilitat i fer més el joc més agradable de jugar s'han implementat Colors. Això es fa gràcies a la classe Log.

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 comprova 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 comandes 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**. Això ho fa comparant les dates de modificació dels objectes i el codi.

L'usuari no necessita saber compilar, només executar. Això 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. Això és molt útil perquè aquest arxius 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

El principal repte d'aquest projecte ha sigut controlar totes les casolístiques de fitxers o moviments invàlids. Aquí una llista del que s'ha controlat.

Laberints

Ha d'existir el fitxer

Ha de tenir una capçalera amb format NxN indicant el tamany del laberint

El laberint ha de complir les dimensions establertes a la capçalera

Les sortides han d'estar a les parets i mínim n'hi ha d'haver una.

Ha d'haver una única sortida ubicada en una de les parets

No pot haver-hi espais buits a les parets

Moviments

Ha d'estar format per caràcters vàlids

En cas d'haver-hi números, han d'estar seguits d'un caràcter vàlid

No pot moure's a una paret

S'ha de comprovar colisions pas per pas. Si no, en cas de que la posició final sigui vàlida però el camí contingui blocs, simplement els saltaria.

Per cada cas d'error hi ha un missatge personalitzat.

Gràcies a la classe Log ha facilitat la implementació perquè:

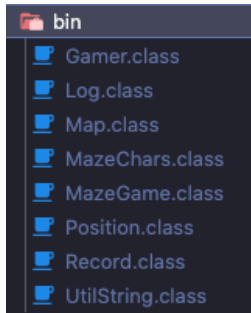
Depenent de la gravetat, segueixen un codi de colors

Els errors están classificats en tipus, les funcions de Log recullen un primer string per el tipus d'error i un altre del error en sí.



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.

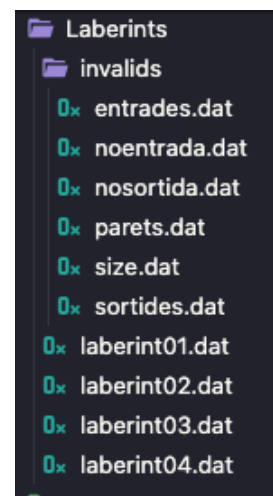
La decisió de deixar-ho tot a una carpeta fa que l'arrel del projecte no s'ompli d'archius temporals.

Laberints

Són tots els fitxers amb les dades dels laberints. Compta amb un subdirectori 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

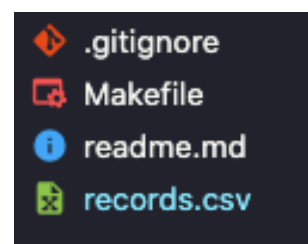
Carpeta amb tot el codi del projecte.

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





Conclusions

En aquest projecte, he pogut aplicar els conceptes que hem après a classe, els quals ha estat fonamental tenir clarament integrats i relacionats per poder afrontar aquest repte.

Fer aquest treball de manera individual ha requerit una gran dosi de concentració i resolució de problemes. He hagut d'afrontar diverses complexitats durant el desenvolupament del projecte, des de la generació del laberint fins a la implementació de les mecàniques del joc.

No obstant això, el que realment ha enriquit la experiència és la col·laboració amb els meus companys. L'intercanvi d'idees ens ha permès resoldre problemes de maneres que no hauriem pensat sols i plantejar coses de maneres diferents ha millorat significativament el joc.

A través d'aquesta experiència, he après que la combinació del treball individual i la col·laboració és fonamental per aconseguir l'èxit en projectes.

La interconnexió de coneixements de diverses àrees ha estat crucial en aquest projecte de programació del joc per terminal. He integrat principis de programació, mecàniques de joc i conceptes d'eficiència en l'algorítmia i el rendiment del codi. També he barrejat altres conceptes que no havia utilitzat a classe, com per exemple el Makefile, el que ha fet que treballi la meua adaptabilitat i creativitat per modificar coses que ja sabia en altres contextos.

Aquesta combinació ha resultat fonamental per crear un joc funcional i atractiu.

En resum, aquest projecte no només m'ha proporcionat una comprensió més profunda de la programació i el desenvolupament de jocs, sinó que també m'ha ensenyat la importància de l'intercanvi d'idees amb els meus iguals i en veure una visió global dels projectes.