THE PACMAN RETURNS

DOCUMENT DE ESPECIFICACIÓ

Laberint

Descripció general: Laberint “quadrat” i connex format per elements estàtics “monedes, paret ...” i elements dinàmics “pacman, fantasmes items...”

Operacions:

```
//Pre: el fitxer existeix, és de text i conté un laberint NxN
//Post: em creat un laberint a partir de un fitxer;
Laberint(String fitxer)

//Pre: – –;
//Post: em creat un laberint a partir del motor de generació
especificat per paràmetre;
Laberint(IGeneradorLaberint)

//Pre: la matriu elements és quadrada;
//Post: em creat un laberint a partir de elements;
Laberint(EElementLaberint elements[][])

//Pre: posicio és valida dins del laberint;
//Post: em retornat l'item de la posició;
EElementLaberint obtenirObjecte(Posicio posicio)

//Pre: el moviment pOrigen i pDesti són valids dins del laberint;
//Post: en pDesti ara hi ha l'element que hi havia en pOrigen i
ara en pOrigen no hi ha res;
void anotarMoviment (Posicio pOrigen, Posicio pDesti)

//Pre: no queden monedes en el laberint;
//Post: em sortejat una porta en el laberint;
void sortejarSortida()

//Pre: – –;
//Post: em sortejat un element “patins, monedes X2 o la mongeta
màgica de en kain” en una casella        //acessible del laberint;
void sortejarItem()

//Pre: – –;
//Post: retornem si posicio és valida, entenem per valida que està
dins del laberint i no és paret;
boolean posicioValida(Posicio posicio)
```

Partida

Descripció general: representació de una partida on intervenen dos personatges “pacman controlat per l'usuari i fantasma controlat per la màquina” i on han de obtenir el màxim nombre de punts per guanyar.

Operacions:

```
//Pre: nivell >= 1;
//Post: em iniciat una partida en un nivell i dificultat
especificada;
Partida(EEnumDificultat dificultat, int nivell)


//Pre: la partida no ha sigut finalitzada anteriorment;
//Post: em finalitzat la partida;
void finalitzarPartida()
```

ItemMovible

Descripció general: Ens representa qualsevol objecte del laberint que té l'habilitat de moure's;

Operacions:

```
//Pre: millis >= 1000
//Post: em creat un ItemMovible amb els paràmetres especificats;
ItemMovible(Laberint laberint, Posicio inici, long millis)

//Pre:--;
//Post: aplicar el moviment precalculat.
void realitzarMoviment()

//Pre:--;
//Post: em precalculat el següent moviment “valid” per el laberint
void calcularMoviment()
```

Item

Descripció general: Especialització de ItemMovible el qual ja sap calcular el proxim moviment, la funcionalitat bàsica del item serà aportar “poders temporals” a qui l'agafi i el seu comportament serà fujir dels personatges .

Operacions:

```
//Pre: millis >= 1000;
//Post: em creat el nou item amb els paràmetres especificats;
Item(Laberint laberint, Posicio inici, long millis, ETipusItem
tipusItem)
```

Personatge

Descripció general: Especialització de ItemMovible, encara no sap com actuar a cada moviment “ja que dependrà del tipus de personatge que sigui”, tot personatge conté un cúmul de punts acumulats, i té altres fucionalitats com la capacitat de canviar la seva freqüència de moviment;

Operacions:

```
//Pre: millis >= 1000;
//Post: em creat el nou personatge amb els parametres especificats
Personatge(Laberint laberint, Posicio inici, long millis)

//Pre: millis >= 1000;
//Post: em canviat el temporitzador de moviment del personatge;
```

```
void canviarTimer(long millis)

//Pre: – –;
//Post: retornem el nombre de punts que porta acumulats el
personatge;
int obtenirPunts();
```

Pacman

Descripció general: Generalització de personatge que té la seva pròpia manera de calcular la següent posició segons el seus interessos.

Fantasma1

Descripció general: Generalització de personatge "controlat per la màquina" que calcularà de forma aleatoria el següent moviment.

Fantasma2

Descripció general: Generalització de personatge "controlat per la màquina" que calcularà el següent moviment a partir dels items que té en línia recta;

Fantasma3

Descripció general: Generalització de personatge "controlat per la màquina" que calcularà el seguent moviment de forma "intel·ligent" tinguent en compte diferents factors com ara:
- Benefici al seguir una ruta concreta.
…

IGeneradorLaberint

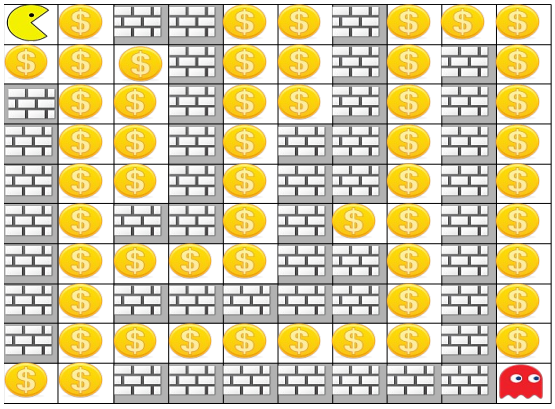
Descripció general: Abstracció de com crear un laberint, ens permet separar la lògica de un laberint de com s'ha de crear, tothom que implmenti aquesta interficie es compromet a generar un laberint que compleix les especificacions del joc;

Operacions:

```
//Pre: costat >= 0 i EElementLaberint és un fantasma;
//Post: Retornem una laberint en format de matriu "quadrada i
connexa" amb un pacman en la primera posició i un fantasma en
l'última;
EElementLaberint[][] generarLaberint(int costat, EelementLaberint
fantasma);
```

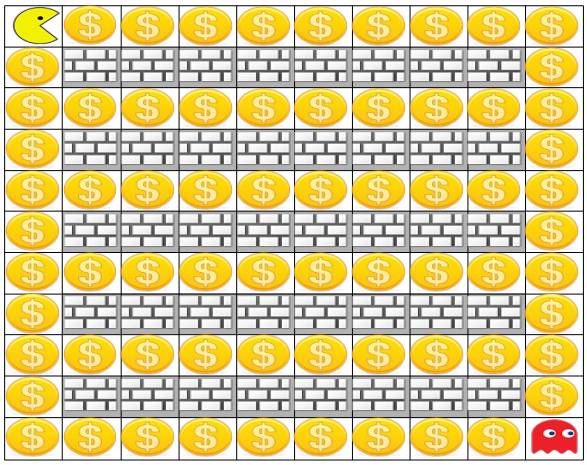
GeneradorLaberintAleatori

Descripció general: Implementa l'interficie IgneradorLaberint i té com a finalitat generar un laberint de forma aleatòria.



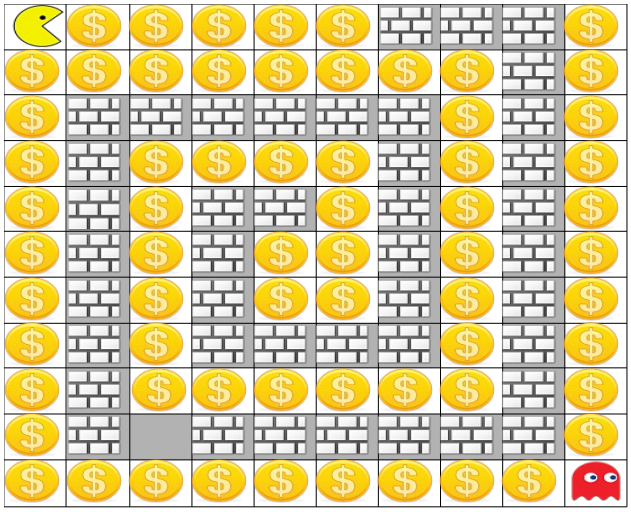
GeneradorLaberintLineal

Descripció general: Implementa l'interficie IgeneradorLaberint I té com a finalitat generar un laberint on en les files parelles hi ha monedes i en les imparelles paret;



GeneradorLaberintEspiral

Descripció general: Implementa l'interficie IgeneradorLaberint I té com a finalitat generar un laberint en forma d'espiral



HistoricPosicions

Descripció general: ens representarà una estructura de dades per guardar el camí que va realitzant tot item que es mou pel laberint.

Operacions:

```
//Pre: – –;
//Post: em afegit una posició a l'estructura;
void afegiPosicio(Posicio posicio);


//Pre: l'historic de posicions no està buit;
//Post: em retornat l'última posició afegida a l'estructura;
Posicio obtenirUltimaPosició()

//Pre: l'històric de posicions no està buit;
//Post: em tret I retornat l'última posició afegit a l'estructura;
Posicio treureUltimaPosicio();

//Pre: –;
//Post: retornem si l'estructura està buida;
boolean esBuit();
```

Posició

Descripció general: ens representa un punt en dos dimensions (eix de les ordenades i eix de les absices).

Operacions:

```
//Pre:  – –;
//Post: inciem el punt en el pla a partir de la x i la y;
Posicio(int x, int y)

//Pre: –;
//Post: retornem la coordenada x del punt;
int obtenirX();

//Pre: –;
//Post: retornem la coordenada y del punt;
int obtenirY();

//Pre: –;
//Post: retornem els 4 punts del voltant en ordre (esquerra,
adalt, dreta i abaix);
Posicio[] obtenirPosicionsDelVoltant();
```

SISTEMA DE LOG

EL SISTEMA DE LOG UTILITZA EL PATRÒ DE DISSENY SINGLETON, D'AQUESTA MANERA ENS ASSEGUREM QUE NOMÉS HI HAURÀ UNA INSTANCIA DE AQUESTA CLASSE PER TOTA L'APLICACIÓ, I AIXÍ, CENTRALITZEM EL LOG.

Log

Descripció general: Ens representa un sistema per anar anotant tot el que succeeix al llarg de l'execució de l'aplicació, l'utilitzarem principalment durant la fase de desenvolupament;

Operacions:

```
//Pre: -;
//Post: em creat una nova instancia de log;
Log();

//Pre:  solicitant no és null;
//Post: retornem l'instancia del log, tot el que s'afegeixi en el
log es farà amb el nom del solicitant;
Log getInstance(Class solicitant)

//Pre: -;
//Post: em afegir un nou error al log;
void afegirError(String registre)

//Pre: -;
//Post: em afegir un nou warning al log;
void afegirWarning(String registre)

//Pre: -;
//Post: em afegir un nou missatge de debug al log;
void afegirDebug(String registre)

//Pre: -;
//Post: em retornat el contingut complet del log formatat per
colors
//   -Vermell → error
//   -Groc → warning
//   -Verd → debug
String obtenirContingutCompletDelLogAmbColor()

//Pre: -;
//Post: em retornat el contingut complet del log;
String obtenirContingutCompletDelLog()


//Pre: -;
//Post: em retornat tots els registres del log que són de tipus
error;
String obtenirErrorsLog()

//Pre: -;
//Post: em retornat tots els registres del log que són de tipus
warning;
String obtenirWarningsLog()
//Pre: -;
```

```
//Post: em retornat tots els registres del log que són de tipus
debug;
String obtenirDebugsLog()

//Pre: –;
//Post: em retornat el últim registre d'error registrat en el log;
String obtenirUltimError()

//Pre: –;
//Post: em retornat el últim registre de warning registrat en el
log;
String obtenirUltimWarning()

//Pre: –;
//Post: em retornat el últim registre de debug registrat en el
log;
String obtenirUltimDebug()

//Pre: –;
//Post: em retornat el últim registre del log de una prioritat
especifica;
String obtenirUltimRegistreDeUnTipus(Prioritat prioritat)

//Pre: –;
//Post: em retornat tots els registres de log de una prioritat
especifica;
String obtenirRegistreDeUnaPrioritat(Prioritat prioritat)

//Pre: el path del fitxer és valid;
//Post: em exportat tot el contingut del log al fitxer
especificat;
void exportarLogAFitxer(String fitxer)
```

ParellaPrioritatMissatge

Descripció general: Ens encapsula un missatge amb una prioritat

Operacions:

```
//Pre: –;
//Post: em creeat un missatge amb una prioritat associada;
ParellaPrioritatMissatge(String missatge, Prioritat prioritat)

//Pre: –;
//Post: em retornat el missatge;
String getMissatge()

//Pre: –;
//Post: em retornat la prioritat;
Prioritat getPrioritat()
```

Prioritat

Descripció general: Ens representa un nivell de prioritat que pot ser: ERROR, WARNING o DEBUG;

Oscar Galera I Alfaro
Moises Saus Ten