

PRIMER LLIURAMENT

PROJECTE DE PROGRAMACIO 2021 | MONOPOLY

GERARD ROVELLAT CARBÓ

MARC GOT CRITG

Tipus Sortida refina Casella

Descripció general: Casella de sortida on començaran tots els jugadors. Despres de haver començat la partida, cada vegada que un jugador passi per la casella, cobrarà la recompensa

Operacions

Sortida(terreny recompensa)

Pre: recompensa != NULL

Post: Crea una casella amb un terreny com a recompensa

Sortida(int recompensa)

Pre: recompensa != NULL

Post: Crea una casella de Sortida amb una quantitat fixe com a recompensa

Terreny recompensa()

Pre: --

Post: Retorna la recompensa amb terreny de passar per Sortida

int recompensa()

Pre: --

Post: Retorna la recompensa amb diners de passar per Sortida

Tipus Terreny refina Casella

Descripció general: Casella de terreny. Correspon a un terreny de la ciutat o territori representats al tauler.

Operacions

Terreny ()

Pre: ---

Post: Crea un Terreny buit

Terreny (string nom, int preu_venda, int preu_hipoteca, string color, boolean estat, Set lloguer, int lloguer_hotel, boolean construible, int max_cases, int cost_apartament, int cost_hotel, int edificacions)

Pre: tots els parametres valids

Post: Crea un Terreny amb els parametres entrats

int lloguer ()

Pre: ---

Post: Retorna el preu del lloguer segons els apartaments o hotel construïts

void edificar()

Pre: --

Post: Aumenta el contador de edificacions de la casella.

Tipus Aposta refina Casella

Descripció general: Casella de aposta. Calcula el resultat de les apostes realitzades

Operacions

Aposta()

Pre: ---

Post: Crea una Aposta buida

int resultatAposta(int dauX, int dauY)

Pre: dauX i dauY $\geq 1 \leq 6$

Post: Retorna la quantitat (positiva o negativa) del resultat de l'aposta

Tipus Sort refina Casella

Descripció general: Casella de Sort. Executa la següent targeta d'acció

Operacions

Sort()

Pre: ---

Post: Crea una Cassella de Sort buida

void targetaAccio(Jugador actual)

Pre: acutal != NULL

Post: Executa la targeta següent de la pila

Tipus ComandaDirecte refina Casella

Descripció general: Casella de Comanda Directe. Fa executar una acció determinada de Targetes

Operacions

ComandaDirecte()

Pre: ---

Post: Crea una ComandaDirecte buida

void accioComandaDirecte()

Pre: --

Post: segons la comanda Directa crida a la funció adient de Targetes

Tipus Casella

Descripció general: Informacio del tipus de casella i gestio de les funcions de cada tipus

Operacions

Casella(string tipus_casella)

Pre: ---

Post: Crea la casella segons el argument

Tipus Tirada

Descripció general: Executa una accio possible en una tirada d'un Jugador.

Operacions

Tirada()

Pre: ---

Post: Crea una Tirada buida

void sortida(Jugador actual)

Pre: acutal != NULL

Post: Afageix la recompensa al passar per sortida a el jugador actual

void terreny(Jugador actual)

Pre: acutal != NULL

Post: Treu la informació del terreny on ha caigut i gestiona la compra o no de aquest

void aposta(Jugador actual)

Pre: acutal != NULL

Post: Crida a funcio aposta de caselles per
saver el canvi pagament o cobrament del jugador
amb la seva aposta

void preso(Jugador actual)

Pre: acutal != NULL

Post: Si es la primera vegada que entres a la
preso, posa el bolea a true. Comprova si portes
tres torns a presó i si es aixis, canvia el estat
a no presoner. sino suma 1 a les tirades a la
preso del jugador

void comandaDirecta(Jugador actual)

Pre: acutal != NULL

Post: crida a la funcio adient de la comanda
directe de la casella on es troba el jugador

void sort(Stack<Targetes> pila_targetes)

Pre: pila_targetes != NULL

Post: crida a la funcio adient de la comanda
directe de Caselles

void vendre(Jugador actual)

Pre: acutal != NULL

Post: Realitza la venda d'una propietat del Jugador actual

void comprar(Jugador actual)

Pre: acutal != NULL

Post: Realitza la compra d'una propietat del Jugador actual

void prestec(Jugador actual)

Pre: acutal != NULL

Post: S'afegeix diners al Jugador actual que haurà de tornar.

void cartaSort(Jugador actual)

Pre: acutal != NULL

Post: Executa una funcio depenent de la carta de sort que hagi obtingut el Jugador actual.

Tipus Jugador

Descripció general: Informació dels Jugadors que jugaran al Monopoly. Conté els diners, propietats, targetes de sort entre altres.

Operacions

`Jugador()`

Pre: ---

Post: Crea un Jugador buit

`Jugador(string nom, int caixa, int tirades_a_preso = 0, boolean a_preso = false, int posicio)`

Pre: Tots els arguments vàlids

Post: Crea un Jugador amb els paràmetres entrats

`void cobrar(int quantitat)`

Pre: ---

Post: S'afegeix una quantitat de diners al Jugador

`void pagar(int quantitat)`

Pre: ---

Post: Es treu una quantitat de diners al Jugador

```
void activarPreso()
```

```
Pre: ---
```

```
Post: Posa a true a_preso
```

```
void canviarPosicio(int posicio)
```

```
Pre: ---
```

```
Post: Canvia la posicio del Jugador
```

```
void guardarTargetaSort(int indexTargeta)
```

```
Pre: ---
```

```
Post: Guarda una nova targeta a la ma del  
jugador
```

Tipus Targeta

Descripció general: Accions que el Jugador haura de executar quan tingui/surti d'una targeta de Sort

Targeta()

Pre: ---

Post: Creta una Targeta per defecte

void donar(Jugador actual)

Pre: actual != NULL

Post: Dona una propietat de Jugador actual a un altre jugador a eleccio

void rebre(Jugador actual)

Pre: actual != NULL

Post: Jugador actual rep una propietat d'un altre Jugador a elecció

void multa(Jugador actual, int quantitat)

Pre: actual != NULL

Post: El Jugador actual se li retira una quantitat de diners per la multa

void cobrar(Jugador actual, int quantitat)

Pre: actual != NULL

Post: El Jugador se li suma una quantitat com a premi de diners

void pagar(Jugador actual, int quantitat)

Pre: ---

Post: El Jugador actual paga una quantiat de diners a un altre Jugador a elecció

void preso(Jugador actual)

Pre: actual != NULL

Post: Mou el Jugador actual a la preso

void sortirPreso(Jugador actual)

Pre: actual != NULL

Post: El Jugador actual surt de la preso

void anar(Jugador actual, int num_casella)

Pre: actual != NULL

Post: El Jugador actual es mogut a una casella

Tipus Monopoly

Descripció general: Guarda la pos de cada jugador i un boolea si la casella esta comprada i qui la te comprada

Operacions

Monopoly()

Pre: ---

Post: Crea un Monopoly per defecte

Monopoly(set<Casella> Caselles, set<Jugador> Jugadors, Stack<Targeta> Targeta)

Pre: Tots els arguments valids

Post: Crea el joc de Monopoly amb les caselles, jugadors i targetes de la sort entrats

pair<int,int> tirarDaus()

Pre: ---

Post: Retorna dos numeros aleatoris entre 1-6

Casella moureJugador(Jugador actual, pair<int,int> resultat_daus);

Pre: Tots els arguments valids

Post: Guarda la nova posicio del jugador i retorna la casella on ha caigut

```
void tirada(Jugador actual, Casella casella);
```

```
Pre: jugador and casella != NULL
```

```
Post: Segons la casella, crida a la funcio de  
Tirades corresponent ( aquesta casella es pot  
passar com a referencia o amb un index  
identificatiu )
```

```
Jugador seguentJugador(Jugador actual);
```

```
Pre: actual != NULL
```

```
Post: Retorna el seguent Jugador al actual
```

```
Jugador iniciPartida();
```

```
Pre: ---
```

```
Post: Posa tots els jugadors al inici i  
retorna el primer a jugar
```

Tipus jsonManager

Descripció general: Gestor de accions sobre els fitxers JSON

```
jsonManager(string nom_fitxer)
```

Pre: ---

Post: Crea jsonManager amb el nom del fitxer JSON

```
void llegir(Monopoly monopoly)
```

Pre: monopoly != NULL

Post: Llegeix i escriu a monopoly el fitxer json de configuracio

Diagrama de moduls

