


FARTOS 2086 V2.0




David García Alonso
2DAM-A
2021-2022

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	FARTOS 2086 v2.0 <u>M03 PROGRAMACIÓ UF5</u> DAVID GARCÍA ALONSO	Departament d'informàtica Curs: 2021-22
--	---	---

ÍNDEX

1. Funcionalitat pròpia.....	3
2. Framework Collections	4
2.1. Classe Carta.....	4
2.2. Classe Baralla	5
2.3. Classe Jugador.....	5
2.4. Classe Casella.....	6
2.5. Classe Cementiri.....	6
2.6. Classe Taulell	7
3. Funcionalitat del programa.....	8
3.1. Package menu.....	8
3.2. Package musica	8
3.3. Package sistemaJoc	9
3.3.1. Classe Carta	9
3.3.2. Classe Baralla	10
3.3.3. Classe Jugador	11
3.3.4. Classe Casella.....	12
3.3.5. Classe Cementiri.....	13
3.3.6. Classe Taulell	14
3.3.7. Classe Joc	17
4. Conclusions	19

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

1. FUNCIONALITAT PRÒPIA

Com a funcionalitat pròpia del desenvolupador s'ha agafat una idea basada en el joc de Màgic The Gathering.

Consisteix en crear un cementiri de cartes on s'apilen les cartes que els jugadors van utilitzant.

S'introdueix una nova carta (4 per cada baralla) anomenada Retorn.

L'efecte de la carta és que el jugador pot escollir retornar una carta del cementiri a la mà del jugador objectiu.

Una vegada aquesta carta es torna a fer servir no va al cementiri sinó que va a l'exili i no es pot tornar a recuperar amb un altre carta Retorn.


El cementiri es buida a cada ronda ja que les cartes es tornen a repartir entre tots els jugadors.

La implementació de la funcionalitat està a la classe Cementiri i la de la carta a la funció retorn() (línies 197 a 200) de la classe Taulell.

```

197 @      public void retorn(Jugador jugador, Cementiri cementiri) {
198         cementiri.mostraCementiri();
199         cementiri.treuCartaCementiri(jugador);
200     }

```

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

2. FRAMEWORK COLLECTIONS

S'han fet servir diverses classes del Framework Collections en les diferents classes de les que consta el programa.

La majoria d'ArrayList són heretades de la primera versió del programa. La resta s'han implementat en funció del que més o menys hem trobat necessari pel programa.

Tot i que hi han algunes implementacions que estan més fetes per fer servir el framework més que per necessitat estricta de la gestió del programa.

A continuació s'expliquen els usos de les mateixes.

Classe Carta:


Generem un Map on guardem dos Strings i l'implementem en un TreeMap per que estigui ordenat alfabèticament.

El que fem amb això és guardar el tipus de carta (key) i la seva descripció (value).

```
14      private static Map<String, String> descripcioCartes = new TreeMap<>();
```

A partir d'aquí fem dos mètodes. Un per guardar les dades (generaTipus()) i un altre per imprimir-les (imprimeixTipus()).

```
47      public static void generaTipus() {
48          descripcioCartes.put("/- 1", "Mov al jugador objectiu una casella davant o darrera\n");
49          descripcioCartes.put("/- 2", "Mov al jugador objectiu dues caselles davant o darrera\n");
50          descripcioCartes.put("/- 3", "Mov al jugador objectiu tres caselles davant o darrera\n");
51          descripcioCartes.put("Teleport", "Intercanvia les posicions de dos jugadors\n");
52          descripcioCartes.put("Zancadilla", "El Jugador objectiu perd una carta de la seva mà\n");
53          descripcioCartes.put("Patada", "Durant una ronda el jugador objectiu veu minoritzada en 1 l'efecte de les seves cartes\n");
54          descripcioCartes.put("Hundimiento", "El jugador objectiu retorna a la casella de sortida\n");
55          descripcioCartes.put("Broma", "Intercanvia les cartes amb el jugador objectiu\n");
56          descripcioCartes.put("Retorn", "Retorna una carta del cementiri a la mà del jugador\n");
57      }
58
59      public static void imprimeixTipus() {
60          generaTipus();
61          for (Map.Entry<String, String> entrada : descripcioCartes.entrySet()) {
62              String nom = entrada.getKey();
63              String descripcio = entrada.getValue();
64              System.out.println("Tipus carta:\n" + nom);
65              System.out.println("Descripció:\n" + descripcio);
66          }
67      }
```

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Classe Baralla:

Es crea un ArrayList de Carta a on guardem totes les cartes que formen la baralla.

```
29      private ArrayList<Carta> baralla = new ArrayList<>();
```

Fem servir diferents mètodes com add(), remove() i get().

Per barrejar-les fem servir un mètode on implementem el mètode static shuffle de classe Collections.

```
81      //Mètode per barrejar les cartes
82      public void barrejaBaralla() {
83          Collections.shuffle(baralla);
84      }
```


Classe Jugador:

Un dels atributs de cada jugador és un ArrayList de Carta on es guarden les seves cartes.

```
13      private ArrayList<Carta> cartes;
```

Es fan servir els mètodes size() per saber si un jugador té cartes i el mètode add()

```
64      //Mètode per saber si el jugador té cartes (l'array de cartes és major que 0)
65      public boolean teCartes() {
66          if (this.cartes.size() > 0) {
67              return true;
68          }
69          return false;
70      }
71
72      //Mètode per afegir una carta al jugador
73      public void afegirCarta(Carta carta) { this.getCartes().add(carta); }
```

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Classe Casella:

Un dels seus atributs és un ArrayList de Jugador. D'aquesta manera sabem els jugadors que hi ha a una casella determinada.

```
8      private ArrayList<Jugador> jugadors;
```

Es fan servir els mètodes add() i remove().

```
66      public void afegirJugadorArray(Jugador jugador) { this.jugadors.add(jugador); }
69
70      public void eliminaJugadorArray(Jugador jugador) { this.jugadors.remove(jugador); }
73
```

Classe Cementiri:


El seu únic atribut és un List de Carta on es guarden les cartes que formen el cementiri.

```
11     private List<Carta> cementiri = new ArrayList<>();
```

Fem servir els mètodes add() i size().

Finalment, per buidar la llista fem servir el mètode removeAll().

```
60      public void buidaCementiri() {
61          cementiri.removeAll(cementiri);
62      }
```

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Classe Taulell:

Un dels seus atributs és un ArrayList de Jugador. Aquí guardem els jugadors que juguen en el taulell.


```
18      private ArrayList<Jugador> jugadors;
```

Creem un mètode anomenat informaBloquejos() per obtenir les caselles que estan bloquejades (hi ha dos jugadors a la mateixa).

Aquestes posicions les guardem en un objecte de la interfície Set de Strings. I l'implementem en un TreeSet perquè estiguin ordenades de menor a major.

Per recórrer el Set i imprimir els seus elements fem servir la interfície Iterator.

```
219      //Mètode que ens dona les posicions que estan bloquejades
220      public void informaBloquejos() {
221          Set <String> casellesBloquejades = new TreeSet<>();
222          for (int i = 0; i < this.caselles.length; i++) {
223              if (this.caselles[i].isBloquejada()) {
224                  casellesBloquejades.add(Integer.toString(i));
225              }
226          }
227          if (casellesBloquejades.isEmpty()) {
228              System.out.println("No hi han caselles bloquejades");
229          } else {
230              System.out.print("Posicions de les caselles bloquejades: ");
231              for (Iterator<String> it = casellesBloquejades.iterator(); it.hasNext();) {
232                  System.out.print(it.next());
233                  if (it.hasNext()) {
234                      System.out.print(", ".concat(it.next()));
235                  }
236              }
237              System.out.println();
238          }
239      }
```

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u> DAVID GARCÍA ALONSO	Departament d'informàtica Curs: 2021-22
--	--	---

3. **FUNCIONALITAT DEL PROGRAMA**

Package “menu”:

Classe Menu

Ha canviat respecte a l'anterior versió del programa. Inclou cinc mètodes statics:

- dibuixa(): Fa un dibuix en codi ascii.
- introducció(): Escriu la introducció que hi ha al pdf de FARTOS2086 proporcionat pel professor. Fa servir un parell de mètodes sleep per fer que dos frases s'imprimeixin amb un cert delay.
- menu(): Mostra el menú principal del joc i retorna un enter amb l'opció escollida per l'usuari.
- escriuInstruccions(): Imprimeix les instruccions del programa fent servir el pdf de FARTOS2086. Dins del mètode crida al mètode static de la classe Carta imprimeixTipus() per imprimir cada tipus de carta i la seva descripció.

Package “musica”:


Classe Reproductor:

Canvia molt poc respecte a la primera versió del programa.

S'ha afegit la funcionalitat que la cançó reproduïda es repeteixi de forma indefinida fent servir el mètode loop(Clip.LOOP_CONTINUOUSLY).

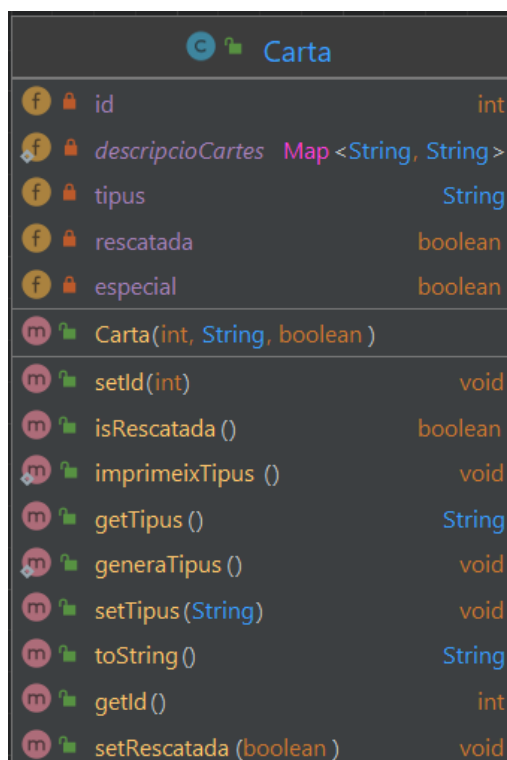
Aquesta modificació ha fet que l'anterior mètode reproduceixTema() s'ha duplicat fent un mètode que reproduceixi les cançons en loop i un altre que no ho faci.

Això s'ha fet així perquè la cançó que sonava a l'hora de mostrar un guanyador es quedava constantment sonant en bucle.

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Package “sistemaJoc”:

Classe Carta:




L'atribut `rescatada` de tipus `boolean` ens permet saber si la carta ja s'ha agafat del cementiri amb la carta `Retorn`.

Els mètodes per generar i imprimir els tipus de carta estan explicats a l'apartat del framework `Collections`.

El mètode `toString` imprimeix la carta d'una forma “elegant” depenent de quin sigui el tipus de la mateixa.

Per la resta, poc canvi respecte a la versió anterior del joc.

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---


Classe Baralla:

Baralla		
f	HUNDIMIENTOS	int
f	CARTESESPECIALS	int
f	BROMES	int
f	CARTESMESQUA	int
f	NCARTES1	int
f	TOTALCARTES	int
f	CARTESMOVIMENT	int
f	RETURNS	int
f	ZANCADILLES	int
f	CARTESMENYSQUA	int
f	TELEPORTS	int
f	baralla	ArrayList<Carta>
f	NCARTES2	int
f	NCARTES3	int
f	PATADES	int
m	Baralla()	
m	imprimeixBaralla()	void
m	getCarta(int)	Carta
m	creaBaralla()	void
m	reparteixCartes(int)	ArrayList<Carta>
m	barrejaBaralla()	void
m	getBaralla()	ArrayList<Carta>

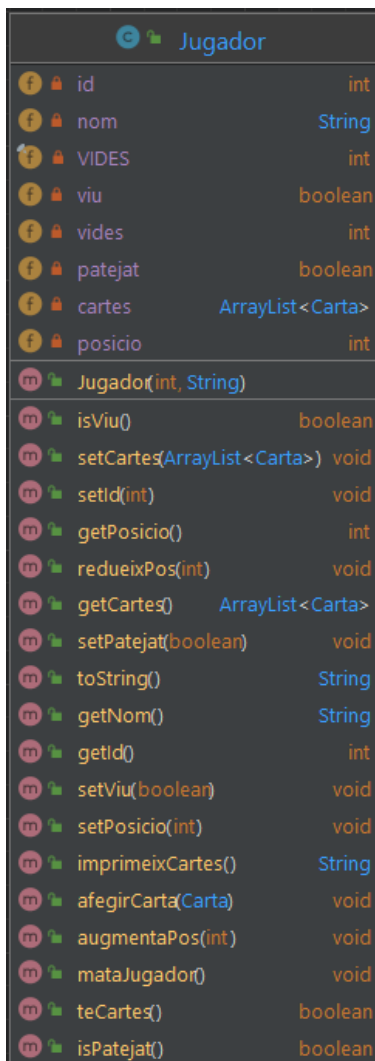
Diverses constants per determinar la quantitat de cartes de cada tipus que té el joc.

Mateixos mètodes que la versió anterior.

Els canvis més important és dins del mètode creaBaralla on construïm la baralla segons la quantitat de cartes de cada tipus que hi ha.

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Classe Jugador:




Jugador		
f	id	int
f	nom	String
f	VIDES	int
f	viu	boolean
f	vides	int
f	patejat	boolean
f	cartes	ArrayList<Carta>
f	posicio	int
m	Jugador(int, String)	
m	isViu()	boolean
m	setCartes(ArrayList<Carta>)	void
m	setId(int)	void
m	getPosicio()	int
m	redueixPos(int)	void
m	getCartes()	ArrayList<Carta>
m	setPatejat(boolean)	void
m	toString()	String
m	getNom()	String
m	getId()	int
m	setViu(boolean)	void
m	setPosicio(int)	void
m	imprimeixCartes()	String
m	afegirCarta(Carta)	void
m	augmentaPos(int)	void
m	mataJugador()	void
m	teCartes()	boolean
m	isPatejat()	boolean

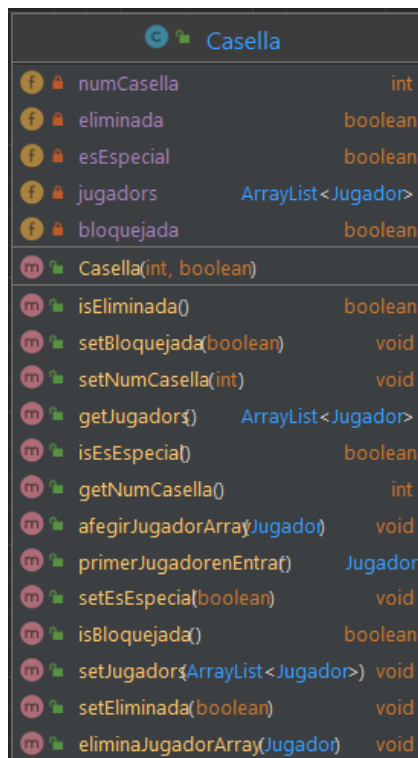
Definim un atribut de tipus booleà anomenat patejat. Així sabrem si el jugador ha estat afectat per la carta patada.

També creem un mètode teCartes() per saber si aquest jugador té cartes.

La resta de mètodes són iguals que els d l'anterior versió del programa.

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Classe Casella:




Casella		
f	numCasella	int
f	eliminada	boolean
f	esEspecial	boolean
f	jugadors	ArrayList<Jugador>
f	bloquejada	boolean
m	Casella(int, boolean)	
m	isEliminada()	boolean
m	setBloquejada(boolean)	void
m	setNumCasella(int)	void
m	getJugador()	ArrayList<Jugador>
m	isEsEspecial()	boolean
m	getNumCasella()	int
m	afegirJugadorArray(Jugador)	void
m	primerJugadorentre()	Jugador
m	setEsEspecial(boolean)	void
m	isBloquejada()	boolean
m	setJugador(ArrayList<Jugador>)	void
m	setEliminada(boolean)	void
m	eliminaJugadorArray(Jugador)	void

Afegim dos atributs de tipus booleà:

- Eliminada. Així sabem si aquesta casella ha estat eliminada a la “muerte súbita”.
- Bloquejada. Així sabem si en aquesta casella hi ha un bloqueig perquè hi ha dos jugadors a la mateixa.

La resta de mètodes i atributs són els mateixos que a l'anterior programa.

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Classe Cementiri:

Cementiri		
f	cementiri	List<Carta>
m	Cementiri()	
m	treuCartaCementiri(Jugador)	void
m	afegeixCartaCementiri(Carta)	void
m	buidaCementiri()	void
m	mostraCementiri()	void

Classe implementada per primera vegada en aquesta versió del programa.


Consta d'un únic atribut. Una llista de cartes (les que els jugadors van fent servir).

Quatre mètodes:

- treuCartaCementiri(Jugador). Pregunta al jugador quina carta vol treure del cementiri (primer comprova si el cementiri té cartes). Controla que no es produeixi una excepció a l'hora d'introduir el número de carta i afegeix la carta escollida a la ma del jugador fent servir el mètode afegirCarta() de la classe Jugador.
- afegeixCartaCementiri(Carta). Comprova que la carta no estigui rescatada, si no ho està afegeix aquesta carta a la llista de cartes del cementiri.
- buidaCementiri(). Elimina totes les cartes de la llista cementiri.
- mostraCementiri(). Mètode toString. Imprimeix totes les cartes del cementiri.

Classe Taulell:

Taulell		
MAXJUGPERCASELLA	int	
cartesJugades	int	
baralla	Baralla	
jugadors	ArrayList <Jugador >	
NCASELLES	int	
torn	int	
ronda	int	
MOVIMENTCASELLAESPECIAL	int	
CASELLAESPECIAL	int	
caselles	Casella[]	
Taulell(ArrayList <Jugador >, Baralla)		
pateja(Jugador)	void	
reiniciaTorn()	void	
getRonda()	int	
barrejaJugadors(ArrayList <Jugador >)	void	
jugadorAHangar(ArrayList <Jugador >)	boolean	
setJugadors(ArrayList <Jugador >)	void	
setRonda(int)	void	
teleport(Jugador, Jugador)	void	
retrocedir(Jugador, int)	void	
retorn(Jugador, Cementiri)	void	
informaBloquejos()	void	
setTorn(int)	void	
avancar(Jugador, int)	void	
primeraCasellaNoEliminada()	int	
inicialitzaTaulell()	void	
zancadilla(Jugador)	void	
unicJugadorViu(ArrayList <Jugador >)	boolean	
bloquejaCasella(int)	boolean	
respostaCasellaEsp(Jugador)	void	
comprovaCasellaEliminada(Jugador)	void	
getJugadors()	ArrayList <Jugador >	
jugadorMesAvancat(ArrayList <Jugador >)	void	
casellaEspecial(Jugador)	void	
eliminaCasella(int)	void	
broma(Jugador, Jugador)	void	
statusJugadors()	void	
augmentaRonda()	void	
hundimiento(Jugador)	void	
getTorn()	int	
hihaGuanyador(ArrayList <Jugador >)	boolean	
augmentaTorn()	void	


 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Atributs i estructura semblant a l'anterior versió. Consta d'un ArrayList de jugadors i un array de objectes de la classe Casella.

També es defineix una casella especial. En aquest cas continua sent la número 8.

S'han implementat nous mètodes per fer les funcionalitats noves que es demanaven per aquesta versió 2.0

- avançar(Jugador, int). S'ha modificat per poder comprovar el tema dels bloquejos de caselles. En comptes d'avançar i sumar la magnitud a les posicions, ara es fa d'una en una. Així cada vegada que s'avança una posició es comprova si la casella on s'avança està bloquejada. En cas que ho estigui, s'informa de la circumstància i es retrocedeix una posició al jugador. Una vegada es troba un bloqueig, surt del bucle i no avança més. Quan tot queda resolt, s'afegeix al jugador a l'ArrayList de la casella. Si la casella és bloqueja al guardar la posició, s'informa a través d'un missatge.
- retrocedir(Jugador, int). Funcionament igual al del mètode avançar(Jugador, int).
- pateja(Jugador). Canvia l'estat del jugador objectiu a patejat.
- hundimiento(Jugador). Funcionament semblant a avançar i retrocedir. Elimina al jugador de l'arrayList de la casella i el guarda a la primera casella que no estigui eliminada. Comprova si hi havia un bloqueig a la posició anterior del jugador i si hi havia, l'elimina.
- teleport(Jugador, Jugador). Semblant a hundimiento en funcionament. Intercanvia les posicions dels jugadors fent servir una variable auxiliar. També comprova si les caselles estaven bloquejades i intercanvia els jugadors dels arrayLists de les caselles.
- zancadilla(Jugador). Comprovem si el jugador té cartes i escollim una carta de la seva mà aleatòriament. Eliminem aquesta carta. Si el jugador no té cartes, s'informa que la carta no ha tingut efecte.

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

- `broma(Jugador, Jugador)`. Intercanvia les cartes fent servir una variable auxiliar. Una vegada intercanviades, eliminem la carta Broma de la ma del jugador objectiu.
- `retorn(Jugador, Cementiri)`. Cridem els mètodes `mostraCementiri()` i `treuCartaCementiri()`.


S'han creat dos mètodes per gestionar el bloquejos de les caselles.

- `bloquejaCasella(int)`. Passem un número de casella i comprova si hi ha dos jugadors a la mateixa i que la casella no sigui 0 (Sortida). Si és correcte canvia l'estat de la casella a true. Sinó, a false.
- `informaBloquejos()`. Es crea un Set de String per guardar els números de les caselles que estan bloquejades. Es recorre tot l'array i si està bloquejada es guarda al Set. Si el Set està buit s'informa que no hi ha bloquejos. En cas contrari, es mostres fent servir la interfície Iterator.

Finalment, s'han creat dos mètodes per gestionar les caselles eliminades:

- `comprovaCasellaEliminada(Jugador)`. Comprova si el jugador està a una casella eliminada del taulell durant la “muerte súbita” de les rondes 4 i 5. Si ho està i el jugador encara està viu (sinó, informa a cada torn que el jugador ha mort), el mata i informa.
- `primeraCasellaNoEliminada()`. Retorna la primera casella que no ha estat eliminada. El necessitem per gestionar la carta Hundimiento.

La resta de mètodes de la classe Taulell són iguals que en l'anterior versió del programa (gestió de la casella especial, gestió de guanyadors...).

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

Classe Joc:

Funcionament molt semblant a l'anterior versió del programa.

Instanciem els objectes baralla, cementiri i taulell.

Cridem a les funcions estàtiques de menú i mostrem el menú principal mentre no sortim del joc o completem una partida.


L'opció 1 és jugar una partida, l'opció 2 mostra les instruccions del joc i l'opció 3 surt del programa.

El sistema de Joc es fa de la següent forma:


La idea principal és que els jugadors han de jugar mentre mínim un d'ells tingui cartes. Això es fa controlant a cada torn que la variable "jugadorsAmbCartes" (inicialitzada a la quantitat de jugadors que hi ha -> jugadors.size()) sigui més gran que 0.

Estructura:

1. Bucle que es repeteix la quantitat de rondes que indiquem a la constant que ho controla.
2. Per cada volta fem diverses accions abans de que els jugadors juguin:
 - a. Establim quants jugadors tenen cartes (jugadors.size()).
 - b. Fem que els jugadors deixin d'estar patejats.
 - c. Buidem el cementiri.
 - d. Escollim el tema musical de la ronda que toca.
 - e. Reiniciem el torn a 1
 - f. Barregem els jugadors.
 - g. Creem les cartes.
 - h. Barregem les cartes.
 - i. Repartim les cartes.
 - j. Informem que comença la ronda i l'ordre de joc de la mateixa.

 <p>INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona</p>	<p><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p>DAVID GARCÍA ALONSO</p>	<p>Departament d'informàtica</p> <p>Curs: 2021-22</p>
---	---	--

3. Fem les següents accions mentre algun jugador tingui cartes i no hi hagi guanyador:
 - a. Iterem per cada jugador i cridem les següents funcions sempre que el jugador estigui viu, tingui cartes i no hi hagi un guanyador:
 - Mostrem les cartes del jugador i demanem que esculli una.
 - Mostrem l'estat dels jugadors i demanem que esculli un.
 - Cridem al mètode triaAccioJugador que segons quina carta hem escollit farà una acció o un altre.
 - Mostrem l'estat dels jugadors després de fer servir la carta.
 - Informem dels bloquejos.
 - Comprovem si el jugador objectiu ha arribat a la casella especial.
 - Afegim la carta utilitzada al cementiri.
 - Eliminem la carta utilitzada de la mà del jugador.
 - Comprovem si hi ha un guanyador.
 - b. Quan tots els jugadors que compleixen les condicions han jugat una carta, comprovem quants jugadors tenen cartes.
 - c. Si la ronda és major que tres, comença la “muerte súbita” i eliminem una casella començant per la 0 (Utilitzem la variable casellaAeliminar i augmentem el seu valor en 1). Comprovem si algun dels jugadors està en aquesta casella eliminada.
 - d. Comprovem si hi ha un guanyador.
 - e. Augmentem la ronda
4. Una vegada acabades totes les rondes, si no hi ha guanyador comprovem quin jugador està més avançat.

 INSTITUT AUSIÀS MARCH Consorci d'Educació de Barcelona Generalitat de Catalunya Ajuntament de Barcelona	<p style="text-align: center;"><u>FARTOS 2086 v2.0</u> <u>M03 PROGRAMACIÓ UF5</u></p> <p style="text-align: center;">DAVID GARCÍA ALONSO</p>	<p style="text-align: center;">Departament d'informàtica</p> <p style="text-align: center;">Curs: 2021-22</p>
--	---	---

4. CONCLUSIONS

Força content del resultat obtingut.

He intentat imprimir el taulell, però ja vaig dir que el tema del disseny no és el meu fort i no ho he pogut implementar.

M'hauria agradat disposar de més temps per fer comprovacions i test del programa. És possible que algunes funcionalitats no estiguin del tot comprovades (bloquejos, alguna excepció out of bounds...), però m'ha estat impossible.

De tota manera, crec que el programa és bastant solvent, implementa totes les noves funcions, unes quantes classes del framework Collections i una nova funció molt reshulona!