DOCUMENTACIÓ

1a ENTREGA

Identificador de l'equip: 43.1

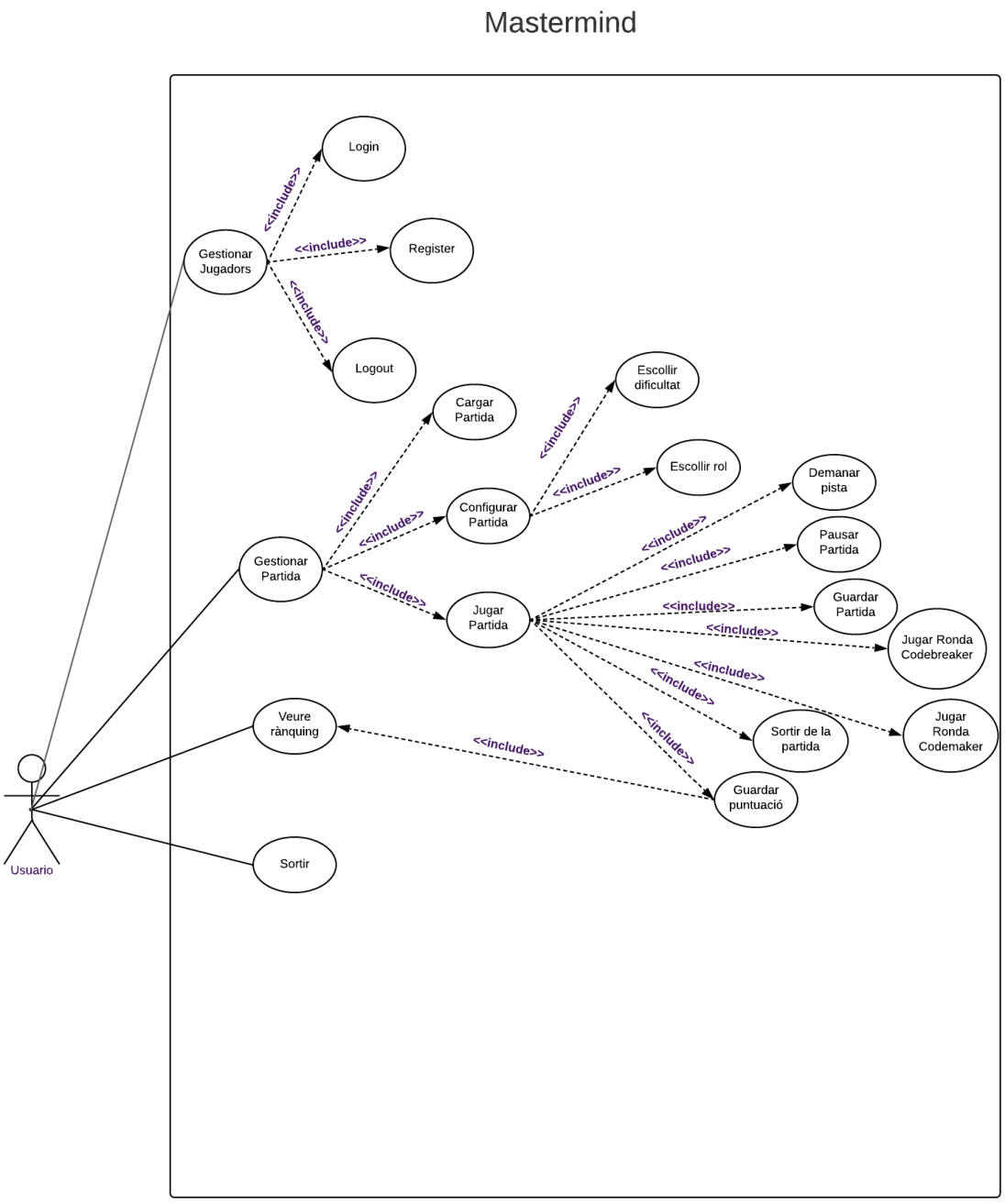
barranquero.morata.david: david.barranquero@estudiantat.upc.edu
domenech.bravin.arnau: arnau.domenech@estudiantat.upc.edu
ortiz.lopez.marc: marc.ortiz.lopez@estudiantat.upc.edu
ye.sheng: sheng.ye@estudiantat.upc.edu

INDEX

1. Diagrama de casos d'ús..... 3
1.1. Descripció Casos d'ús..... 4
2. Diagrama del model conceptual..... 8
2.1. Disseny del diagrama de model conceptual..... 8
2.2. Descripció de les classes..... 9
2.2.1. Jugador..... 9
2.2.2. Partida..... 9
2.2.3. Codebreaker..... 9
2.2.4. Codemaker..... 9
2.2.5. EstadistiquesPartida..... 9
2.2.6. Rànquing..... 9
2.2.7. Ronda..... 9
2.2.8. Interface Màquina..... 9
2.2.9. FiveGuess..... 10
2.2.10. Genetic..... 10
2.2.11. CtrlPartida..... 10
2.2.12. CtrlJugador..... 10
2.2.13. CtrlEstadistiquesPartida..... 10
2.2.14. CtrlAlgorisme..... 10
2.2.15. CtrlRanquing..... 10
2.2.16. CtrlDomini..... 10
3. Relació de les classes implementades per membre de l'equip..... 11
4. Estructures de dades i algorismes utilitzats..... 12
4.1. Estructura de dades..... 12
4.1.1. HashMap..... 12
4.1.2. ArrayList..... 12
4.2. Algorisme FiveGuess:..... 13

1. Diagrama de casos d’ús

Diagrama de casos d’ús que podeu trobar dins el directori DOCS.



1.1. Descripció Casos d'ús

Nom: Login
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

1. L'usuari farà clic al botó de login
2. El sistema li demanarà les credencials (username i contrasenya)
3. L'usuari indica el username i contrasenya
4. El sistema procedeix a fer el login

Errors possibles i cursos alternatius:

1a. No existeix el username. (Sortirà un avís indicant que el username no existeix).
2a. Username o contrasenya incorrecte. (Sortirà un missatge indicant que el username o la contrasenya estan malament).

Nom: Register
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

1. L'usuari farà clic al botó de register
2. El sistema li demanarà les credencials (username, contrasenya i Confirma contrasenya)
3. L'usuari indica el username i la contrasenya que vol dos cops
4. El sistema procedeix al registre

Errors possibles i cursos alternatius:

1a. Les contrasenyes no coincideixen. (Sortirà un avís indicant que les contrasenyes no coincideixen).
2a. El usuari username existeix. (Sortirà un avís indicant que el usuari username ja existeix).

Nom: Logout
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

1. L'usuari farà clic al botó de Tancar Sessió
2. El sistema tancarà la sessió de l'usuari.

Errors possibles i cursos alternatius: -

Nom: Escollir rol
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

1. El sistema mostra les dues opcions de partida (codebreaker i codemaker)
2. L'usuari indica si vol jugar la partida codebreaker o codemaker

Errors possibles i cursos alternatius:

Nom: Escollir dificultat
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

1. L'usuari clica el botó Jugar
2. El sistema mostra les dues opcions de partida (codebreaker i codemaker)
   Escollir Codemaker:
   1. El sistema demana el codi solució.
   2. L'usuari indica el codi solució que vol.
   Escollit Codebreaker:
   1. El sistema demana el número de intents, número de colors i longitud de la combinació.
   2. L'usuari indica el el número de intents, número de colors i longitud de la combinació.

Errors possibles i cursos alternatius:

1a. Número de intents fora de rang (Sortirà un avís indicant que el número de intents està fora de rang).
2a. Número de colors fora de rang (Sortirà un avís indicant que el número de colors està fora de rang).
3a. Longitud de la combinació fora de rang (Sortirà un avís indicant que la longitud està fora de rang).

Nom: Demanar pista
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

1. L'usuari clica el botó Pista
2. El sistema mostra la pista corresponent aleatoria

Errors possibles i cursos alternatius: -

Nom: Jugar ronda Codebreaker
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

Fins que guanyi o perdi:

1. L'usuari dona una combinació
2. El sistema mostra la resposta corresponent

Errors possibles i cursos alternatius:

1a. Longitud de la combinació fora de rang (Sortirà un avís indicant que la longitud de la combinació està fora de rang).
2a. Número de colors fora de rang (Sortirà un avís indicant que el número de colors està fora de rang).

Nom: Jugar ronda Codemaker
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

Fins que guanyi o perdi:

1. El sistema mostra la seva combinació
2. L'usuari indica la resposta
3. Si la resposta és correcte, el sistema mostra la seva següent combinació.

Errors possibles i cursos alternatius:

1a. L'usuari indica una resposta errònia.
1aa. El sistema mostra que s'ha efectuat una resposta incorrecte.
1ab. L'usuari torna a indicar una resposta.

Nom: Veure Rànquing
Actor: Usuari
Comportament (diàleg entre els actors i el sistema):

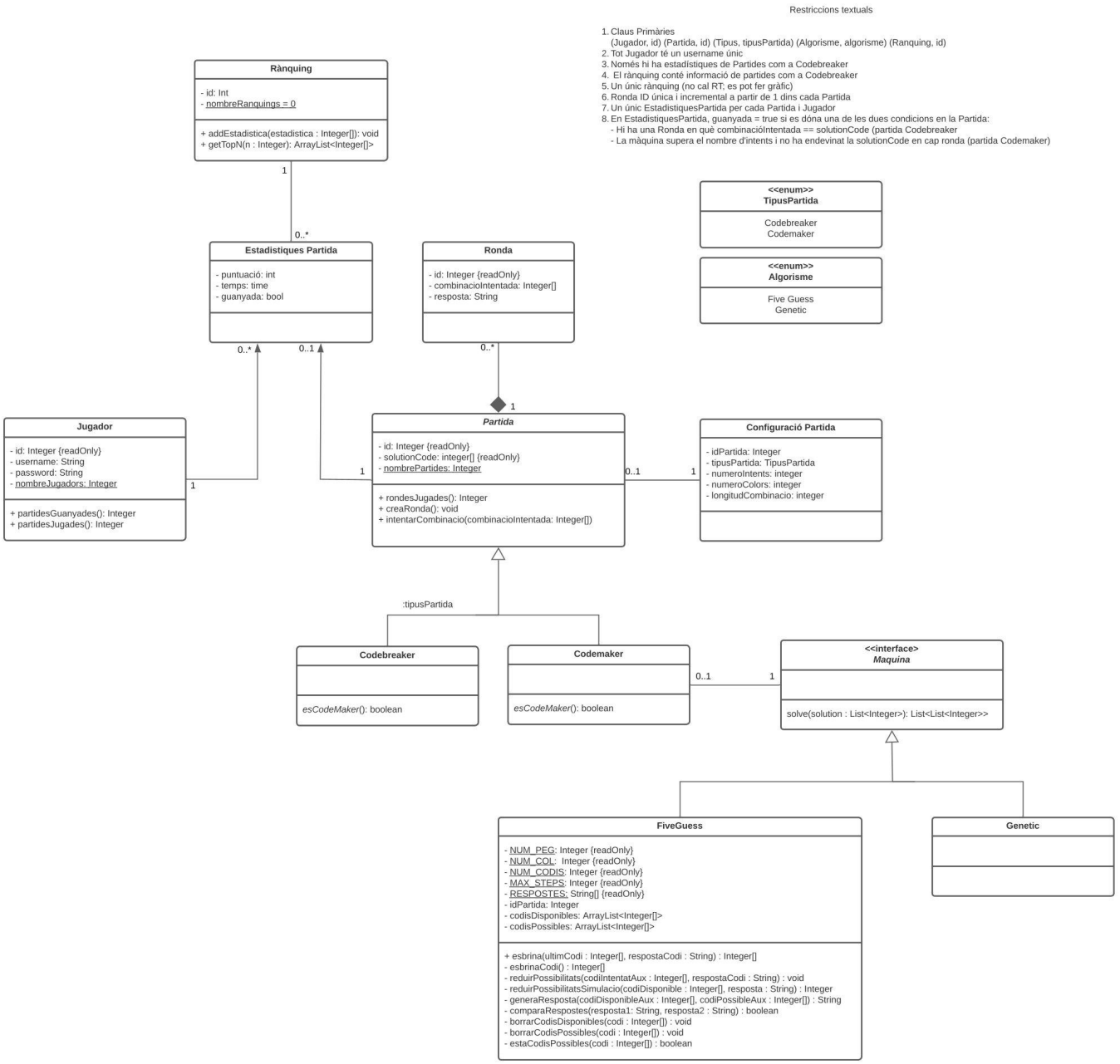
1. L'usuari clica el botó Rànquing
2. El sistema mostra el Rànquing actual

Errors possibles i cursos alternatius: -

Els casos d'ús no descrits aquí queden per implementar-se i documentar en properes entregues.

2. Diagrama del model conceptual

2.1. Disseny del diagrama de model conceptual



2.2. Descripció de les classes

2.2.1. Jugador

La classe Jugador és la classe que representa un Jugador. Aquesta classe s'encarrega d'identificar un jugador, de la seva creació i modificació.

2.2.2. Partida

La classe Partida és la classe que representa una Partida. Aquesta classe emmagatzema les seves rondes i s'encarrega d'identificar una partida, de la seva creació i modificació. També té algunes funcionalitats del joc com *intentarCombinació* o *getCodiMaquina.*

2.2.3. Codebreaker

La classe Codebreaker és un dels fills de la classe Partida que representa una partida Codebreaker. Aquesta classe s'encarrega de crear partida Codebreaker.

2.2.4. Codemaker

La classe Codemaker és un dels fills de la classe Partida que representa una partida Codemaker. Aquesta classe s'encarrega de crear una partida Codemaker i conté una funció de jugabilitat Codemaker anomenada *getCodiMaquina.*

2.2.5. EstadistiquesPartida

La classe EstadistiquesPartida és la classe que representa un Jugador i una Partida. Aquesta classe emmagatzema informació útil de la partida que ha jugat un jugador.

2.2.6. Rànquing

La classe Rànquing és la classe que representa un Rànquing. Aquesta classe s'encarrega de guardar la idJugador i la seva puntuació corresponent. També permet retornar un conjunt de mida N de idJugadors i puntuació amb la funció *getTopN.*

2.2.7. Ronda

La classe Ronda és la classe que representa una Ronda. Aquesta classe s'encarrega de guardar la informació vital per la jugabilitat del joc, com pot ser *combinacióIntentada i resposta.*

2.2.8. Interface Màquina

Aquesta classe és una interfície que ens han donat de la qual només té una funció per realitzar les proves de l'algorisme pertinent. Té dos fills: FiveGuess i Genetic.

2.2.9. FiveGuess

La classe FiveGuess és la classe que s'encarrega d'implementar l'algoritme fiveguess. Aquesta classe conté totes les funcionalitats del algoritme per tal de que es pugui jugar la partida Codemaker.

2.2.10. Genetic

La classe Genetic encara no està implementada. Contindrà les funcionalitats per tal de poder jugar una partida Codemaker però usant l'algoritme genètic.

2.2.11. CtrlPartida

Aquesta classe és el controlador de la classe Partida. S'encarrega de fer les funcionalitats generals de totes les funcions relacionades amb la jugabilitat d'una partida.

2.2.12. CtrlJugador

La classe CtrlJugador és la classe que s'encarrega de gestionar totes les operacions de la classe Jugador. Es comunica amb Jugador per tal de poder realitzar les funcionalitats.

2.2.13. CtrlEstadistiquesPartida

La classe CtrlEstadistiquesPartida és la classe que s'encarrega de gestionar totes les operacions de la classe EstadistiquesPartida.

2.2.14. CtrlAlgorisme

Aquesta classe s'encarrega de gestionar les funcionalitats dels algorismes i fa de pont entre els algorismes i les altres classes.

2.2.15. CtrlRanquing

La classe CtrlRanquing és la classe que s'encarrega de gestionar totes les operacions de la classe Rànquing.

2.2.16. CtrlDomini

És la classe Controlador que es comunica amb tots els altres controladors.

3. Relació de les classes implementades per membre de l'equip

Cada membre de l'equip s'ha encarregat de les classes següents. Els commits fets per persones externes als encarregats de cada classe han tingut el vistiplau del responsable.

Sheng Ye	Arnau Domènech	David Barranquero	Marc Ortiz
Jugador	Partida	Fiveguess	Ranquing
CtrlDomini	Codebreaker	ConfiguracioPartida	Ronda
CtrlJugador	Codemaker	CtrlAlgorisme	EstadistiquesPartida
TestJugador	CtrlPartida	DriverFiveGuess	Driver
TestRonda	CtrlEstadistiquesPartida	DriverConfiguracioPartida	CtrlRanquing
TestRanquing			DriverEstadisticaPartida
TestConfiguracioPartida			DriverRanquing
			DriverRonda

4. Estructures de dades i algorismes utilitzats

4.1. Estructures de dades

4.1.1. HashMap

Amb diferència, l'estructura de dades que més hem utilitzat. Aquesta, se'ns ha fet molt útil gràcies a la seva funcionalitat de guardar Keys i Values. Gràcies a aquesta hem pogut emmagatzemar les ids d'un Objecte amb l'Objecte en qüestió. Ens ha suposat un avantatge ja que HashMap conté mètodes propis per tractar amb ella mateix com get…

Com a punts positius pot ser molt eficient en termes de temps per a les operacions d'inserció, cerca i eliminació d'elements, si s'utilitza correctament. També, ens permet accedir als elements mitjançant una clau, la qual cosa és útil si s'està cercant un element específic i es coneix la seva clau.

Com a contres l'ordre en què es recorren els elements no està garantit (tot i que a nosaltres no ens ha afectat molt). Si es necessita un ordre específic, cal utilitzar altres estructures de dades, com ara TreeMap o LinkedHashMap. Pot ser difícil de llegir i comprendre si la taula de dispersió està molt plena. A més, l'ús excessiu de la memòria pot ser un problema si s'emmagatzemen molts elements.

4.1.2. ArrayList

Hem utilitzat bastant també aquesta estructura de dades per tal de guardar conjunts d'objectes. Hem optat per utilitzar la ArrayList en els casos en què no era tan crític cercar a partir de la clau. L'avantatge principal que té ArrayList sobre usar una Array normal és, principalment, el fet que la seva mida no està limitada. Això és especialment útil per a desar grans quantitats de dades. Per exemple, quan tinguem capa de persistència de dades, guardarem les dades de moltes partides a la llarga, i podria ser perillós fer servir una estructura de dades amb mida limitada ja que ens arrisquem a emplenar-la i a haver de sobreescriure dades.

Dit això, en casos en què hem pogut fer servir arrays normals de Java (com per exemple per als intents de combinacions, que tenen mida definida) els hem preferit per sobre del ArrayList. Això és perquè ArrayList en general és menys eficient que una array normal. A més, ArrayList ve d'una llibreria, i corre perill de patir algun canvi en futures edicions de Java.

4.2. Algorisme FiveGuess

L'algorisme Five Guess retorna en 5 intents o menys el codi solució de la partida amb una configuració de 4 fitxes i 6 colors. Es tenen dos conjunts de totes les combinacions possibles, és a dir, 1296, de la forma 1111, 1112, … , 6665, 6666. Un dels conjunts s'utilitza per gestionar els codis utilitzats i l'altre per gestionar els codis que tenen possibilitat de ser solució.

Es comença al primer torn amb un codi intent de la forma AABB. Donada la resposta, recorrem el conjunt de codis possibles de tal manera que els codis que no donarien la mateixa resposta són eliminats.

Seguidament realitzant la tècnica del Minmax (assegura el millor dels casos pitjors), recorrem tots els codis no intentats i per cadascun d'ells es simula per cada tipus de resposta possible, quants codis es mantindrien al conjunt de codis possibles després de l'eliminació d'aquells que no donarien la mateixa resposta, i ens quedem amb el més gran de totes les respostes (el pitjor cas). Quan hem obtingut aquest còmput de tots els codis no intentats, ens quedem amb el valor mínim (millor cas dels pitjors). Aquest és el codi que es realitza al següent torn. Donada la nova resposta es repeteixen els passos, sense mai trigar més de 5 torns.