

Evolució, organització i responsabilitats del projecte

ALEXANDER GIMÉNEZ ORTEGA (alexander.gimenez)

GENÍS RIERA PÉREZ (genis.riera.perez)

MAURICIO IGNACIO CONTRERAS PINILLA (mauricio.ignaci.contreras)

Codi de projecte: 7.2

Versió del lliurament: 3.0

Índex

1. Evolució de les funcionalitats del projecte	2
2. Organització de les classes	7
3. Responsabilitats del projecte	14

1. Evolució de les funcionalitats del projecte

Durant tota l'etapa de desenvolupament del projecte, des del seu plantejament fins a la fase de proves, passant per l'especificació, l'anàlisi, el disseny i la codificació, s'han anat fent modificacions sobre la marxa relacionades amb les funcionalitats que en un principi es volien implementar al projecte i les que finalment s'han implementat i integrat. Aquests canvis de direcció i decisions han estat consensuades per tots els membres del grup. Els motius principals que han portat a fer aquests canvis han estat fruit de la falta de temps (per garantir tenir acabat el projecte en la data acordada) i la falta de recursos humans (el projecte es va iniciar amb un grup de quatre persones i enmig del procés una d'aquestes va abandonar el projecte). No es pretén utilitzar aquest diàleg com a pretext per la no implementació de tot el que s'havia promès des d'un bon començament, però sí que pretén demostrar que en tot desenvolupament de projectes sorgeixen dificultats que s'han d'anar superant al mateix temps que s'ha d'intentar avançar el projecte, i aconseguir aquesta fita és realment un repte per l'equip que està al darrere.

Seguidament s'ensenyen les diferències entre les funcionalitats (obligades i opcionals) proposades a l'inici del desenvolupament del projecte i les que finalment s'han implementat (obligades i opcionals):

Funcionalitats proposades inicialment	Funcionalitats implementades finalment
Identificar Usuari	Identificar Usuari
Crear Usuari	Crear Usuari
Tancar Sessió	Tancar Sessió
Iniciar Mode Partida Ràpida	Iniciar Mode Partida Ràpida
Iniciar Mode Entrenament	Iniciar Mode Entrenament
Iniciar Mode Arcade *	-
Iniciar Mode Supervivència *	-
Iniciar Mode Contrarellotge *	-
Jugar Partida	Jugar Partida
Guardar Partida	Guardar Partida
Sortir Partida	Sortir Partida
Carregar Partida	Carregar Partida
Reanomenar Partida	Reanomenar Partida
Esborrar Partida	Esborrar Partida
Importar Partida *	-
Exportar Partida *	-
Consultar Rànquings i Rècords	Consultar Rànquings i Rècords
Actualitzar Rànquings i Rècords	Actualitzar Rànquings i Rècords
Consultar Manual Usuari	Consultar Manual Usuari

*Nota: les funcionalitats amb asterisc són les considerades com a opcionals

Com es pot apreciar en l'anterior taula comparativa totes les funcionalitats obligades que inicialment s'havien proposat s'han acabat implementant. També cal indicar que s'han implementat tres estratègies de joc diferents per a tres nivells de dificultat (fàcil, mitjà i difícil), com estava proposat inicialment. Per altra banda, totes les funcionalitats

proposades inicialment que es van considerar com a opcionals no s'han dut a terme. Tampoc s'han integrat totes les variants de joc proposades a l'inici del projecte (Gomoku estàndard, Renju, Gomoku japonès i *freestyle* Gomoku), implementant només la variant del Gomoku estàndard (les altres es van considerar en el seu moment com a opcionals).

No obstant, algunes funcionalitats finalment implementades també han sofert alguns canvis respecte el seu plantejament inicial. Aquests canvis es citen a continuació:

- Com a conseqüència de la no implementació de tots els modes de joc proposats inicialment, l'estructura final dels rècords individuals i els rànquings globals han sofert canvis. Ara només es tenen en compte els criteris relacionats amb la modalitat de Partida Ràpida, i els criteris contemplats per a la modalitat d'Entrenament es fusionen amb els criteris de la Partida Ràpida. Aquest canvi provoca que es comptabilitzi en el mateix criteri els resultats obtinguts tant si es disputen partides en una o altra modalitat. En contraposició, ara s'afegeixen nous criteris en funció de si l'usuari a jugat contra un altre jugador humà o contra un jugador màquina, i en funció del nivell de dificultat d'aquest. Per tant, els criteris finals que es tenen en compte per comptabilitzar els rècords individuals són els següents (sumen un total de 35 criteris a contemplar):
 - Nombre de victòries contra el jugador màquina en nivell fàcil.
 - Nombre de victòries contra el jugador màquina en nivell mitjà.
 - Nombre de victòries contra el jugador màquina en nivell difícil.
 - Nombre de victòries contra un altre jugador humà.
 - Nombre total de victòries.
 - Nombre d'empats contra el jugador màquina en nivell fàcil.
 - Nombre d'empats contra el jugador màquina en nivell mitjà.
 - Nombre d'empats contra el jugador màquina en nivell difícil.
 - Nombre total d'empats.
 - Nombre de derrotes contra un altre jugador humà.
 - Nombre de derrotes contra el jugador màquina en nivell fàcil.
 - Nombre de derrotes contra el jugador màquina en nivell mitjà.

- Nombre de derrotes contra el jugador màquina en nivell difícil.
- Nombre de derrotes contra un altre jugador humà.
- Nombre total de derrotes.
- Percentatge de victòries contra el jugador màquina en nivell fàcil.
- Percentatge de victòries contra el jugador màquina en nivell mitjà.
- Percentatge de victòries contra el jugador màquina en nivell difícil.
- Percentatge de victòries contra un altre jugador humà.
- Percentatge total de victòries.
- Percentatge d'empats contra el jugador màquina en nivell fàcil.
- Percentatge d'empats contra el jugador màquina en nivell mitjà.
- Percentatge d'empats contra el jugador màquina en nivell difícil.
- Percentatge total d'empats.
- Percentatge de derrotes contra un altre jugador humà.
- Percentatge de derrotes contra el jugador màquina en nivell fàcil.
- Percentatge de derrotes contra el jugador màquina en nivell mitjà.
- Percentatge de derrotes contra el jugador màquina en nivell difícil.
- Percentatge de derrotes contra un altre jugador humà.
- Percentatge total de derrotes.
- Nombre total de partides disputades contra el jugador màquina en nivell fàcil.
- Nombre total de partides disputades contra el jugador màquina en nivell mitjà.
- Nombre total de partides disputades contra el jugador màquina en nivell difícil.
- Nombre total de partides disputades contra un altre jugador humà.
- Nombre total de partides disputades al sistema.

I els criteris finals que es tenen en compte per comptabilitzar els rànquings globals són els mateixos que els descrits per als rècords individuals, però per cadascun dels 35 criteris es mostraran els cinc millors usuaris registrats al sistema. És a dir, per al criteri “Nombre de victòries contra el jugador màquina en nivell fàcil” es mostraran els cinc usuaris registrats al sistema amb més

victòries contra aquest jugador en el nivell fàcil, i així successivament amb la resta de criteris.

- Com a feina extra no proposada durant les primeres fases de desenvolupament, però que s'ha dut a terme de cara al projecte final, és la incorporació d'un mecanisme de seguretat pel registre i identificació dels usuaris en el sistema. Aquest mecanisme és el xifrat de les seves contrasenyes, per evitar que possibles atacs contra el sistema accedeixin a les dades dels usuaris per apropiar-se-les, modificar-les o esborrar-les. Per a conèixer millor com funciona aquest mecanisme es pot consultar del document `Descripcio_algorismes_i_estructures_de_dades.pdf` l'apartat 6.1.
- Segons el plantejament inicial, durant el transcurs d'una partida aquesta seria cronometrada, i en el mode Entrenament es podria establir un límit de temps per a les partides durant la configuració d'aquestes. Aquestes petites funcionalitats al final no s'han dut a la pràctica.
- L'esquema inicial per a les finestres del programa pertanyents als rècords individuals i als rànquings globals han sofert lleugeres modificacions visuals i d'estructura dels criteris. El canvi més rellevant ha estat en la mostra dels rànquings globals, els quals s'havien d'estructurar per filtres i consultar-los individualment. Aquesta era la idea principal, però finalment s'ha decidit mostrar-los tots de cop en forma de llista organitzada, tot a dins d'una àrea de text que permet consultar-la tota gràcies a la seva barra de desplaçament. Aquest mètode de mostreig també s'ha aplicat per als rècords individuals.
- Quan disputen la partida dos jugadors màquina, inicialment s'havia proposat en què hi hauria un botó per avançar un sol torn de la partida. Finalment, aquest botó avançarà dos torns a la vegada, per agilitzar una mica més el seu transcurs i evitar que l'usuari hagi d'efectuar excessius clics al botó per prosseguir en la partida.

- Inicialment es va plantejar que, quan un usuari volgués carregar una partida guardada l'oponent del qual també fos un usuari existent al sistema, el sistema demanés que aquest últim s'identifiqués per evitar conflictes de suplantació d'identitat. Finalment aquest pas extra en la càrrega de partides quan es dona aquest cas no s'ha implementat, i la partida es carrega directament.

2. Organització de les classes

Tot el desenvolupament del projecte ha estat regit per una organització consensuada tant a nivell intern de grup com a nivell de clúster pel que fa a l'agrupació de classes. Mitjançant els *packages* de Java s'ha esquematitzat una bona estructura de l'organització de totes les classes que s'han implementat durant tot el desenvolupament. Aquest apartat pretén explicar detalladament quina ha estat l'agrupació escollida per a les classes.

El següent diagrama mostra l'estructura dels *packages* de tot el projecte:



L'element `src` és el directori on estan contingudes totes les classes i fitxers necessaris del projecte agrupades en *packages*. Una primera agrupació de les classes s'ha definit segons si aquestes són compartides a nivell de clúster o són internes a nivell de grup. Per tant, en primer lloc tenim la següent agrupació:

- Tots els *packages* que comencen el seu nom pel prefix compost `prop.cluster` són els acordats a nivell de clúster, i agrupen totes les classes compartides.
- En canvi, tots els *packages* que comencen pel prefix `prop.gomoku` són totes les classes implementades a nivell intern del nostre grup en particular.

Una vegada definida aquesta primera classificació de classes, la següent classificació es basa en el model de l'arquitectura en tres capes, segons si les classes pertanyen a la capa de presentació, a la capa del domini o a la capa de gestió de dades. Aleshores la segona agrupació és la següent:

- Tots els *packages* que en el seu nom complet contenen el subnom `.presentacio` agrupen les classes que pertanyen a la capa de presentació, és a dir, les que figuren al diagrama de classes UML de la capa de presentació. Aquest diagrama es pot consultar en el fitxer `Diagrama_Presentacio.png` dins del directori DOCS del directori arrel del lliurament.
- Tots els *packages* que en el seu nom complet contenen el subnom `.domini` agrupen les classes que pertanyen a la capa del domini, és a dir, les que figuren al diagrama de classes UML de la capa del domini. Aquest diagrama es pot consultar en el fitxer `Diagrama_Domini.png` dins del directori DOCS del directori arrel del lliurament.
- Tots els *packages* que en el seu nom complet contenen el subnom `.gestors` agrupen les classes que pertanyen a la capa de gestió de dades, és a dir, les que figuren al diagrama de classes UML de la capa de gestió de dades. Aquest diagrama es pot consultar en el fitxer `Diagrama_Gestio_Dades.png` dins del directori DOCS del directori arrel del lliurament.

Una menció a part tenen les classes agrupades en el package `prop.gomoku.auxiliars`. Aquestes són classes necessàries per al bon funcionament del projecte, però que conceptualment no apareixen en cap dels diagrames UML que modelen les tres capes de l'arquitectura.

Una altra menció a part té el *package* `prop.gomoku.recursos`. Aquest conté fitxers que no són classes de Java, però són imprescindibles per al bon funcionament del programa, tals com imatges que apareixen impreses en les finestres visuals d'aquest.

Una vegada les classes estan agrupades segons la capa de l'arquitectura a la que pertanyen, es fa una nova distinció segons la funció que cada classe exerceix dins el projecte. Aquesta agrupació està definida de la següent manera:

- Tots els *packages* que en el seu nom complet contenen el subnom `.models` agrupen les classes que representen el conjunt de dades que volem representar. Són aquelles classes que intenten modelar la realitat que en aquest projecte hem volgut representar, i bàsicament aporten funcionalitats molt bàsiques per a que altres classes puguin encarregar-se de la gestió i control de el funcionalitats del programa pròpiament dites.
- Tots els *packages* que en el seu nom complet contenen el subnom `.controladors` agrupen les classes que s'encarreguen d'aglutinar classes més bàsiques (en especial les prèviament descrites) per coordinar les seves funcionalitats i descarregar-les de funcionalitats més específiques inherents al projecte i no pas a la realitat que volem representar. Aquestes classes són les que realment implementen les funcionalitats pròpies del programa i la seva lògica.

Una menció a part tenen les classes agrupades en el *package* `prop.cluster.domini.models.estats`. Aquest *package* agrupa unes classes especials en Java anomenades enumeracions, les quals defineixen una sèrie de constants útils a nivell de clúster per implementar la lògica de la funcionalitat principal, que és la de jugar una partida.

Seguint tot aquest consens, les classes del projecte queden agrupades en *packages* de la següent manera:

- **`prop.cluster.domini.controladors:`**
 - `InteligenciaArtificial`

- **prop.cluster.domini.models:**
 - Partida
 - Tauler
 - Usuari
- **prop.cluster.domini.models.estats:**
 - EstatCasella
 - EstatPartida
- **prop.gomoku.auxiliars:**
 - Lectura
 - LecturaBuffers
 - LecturaScanners
 - ProvaJugadorsMaquina
- **prop.gomoku.domini.controladors:**
 - ControladorDocumentacio
 - ControladorPartida
 - ControladorPartidaEnJoc
 - ControladorPartidesGuardades
 - ControladorPreparacioPartida
 - ControladorUsuari
 - IAGomoku
 - IAGomokuAlternativa
 - IAGomokuOptimitzada
 - InteligenciaCPU
 - Vigenere
- **prop.gomoku.domini.controladors.excpecions:**
 - ContrasenyaIncorrecta
 - ContrasenyaInvalida

- **prop.gomoku.domini.controladors.records:**
 - AccesRecords
 - ControladorRecordsGlobals
 - ControladorRecordsIndividuals
 - CriteriRecords

- **prop.gomoku.domini.models:**
 - EstadistiquesPartides
 - LlistaRecordsIndividuals
 - PartidaGomoku
 - ResumResultats
 - TaulerGomoku
 - TipusUsuari
 - UsuariGomoku

- **prop.gomoku.gestors:**
 - GestorDocumentacio
 - GestorPartidesGuardades
 - GestorUsuaris

- **prop.gomoku.gestors.excepcions:**
 - UsuariJaExisteix
 - UsuariNoExisteix

- **prop.gomoku.presentacio:**
 - CasellaGUI
 - ControladorPresentacio
 - FrameAccedirEstadistiques
 - FrameBenvingut
 - FrameCarregaPartides
 - FrameConfiguracioPartida1

- o FrameConfiguracioPartida2
- o FrameConfiguracioPartida2CPU
- o FrameConfiguracioPartida2Persones
- o FrameConfiguracioPartida3
- o FrameError
- o FrameEstadistiquesGlobals
- o FrameEstadistiquesIndividuals
- o FrameIdentificacio
- o FrameMenuPrincipal
- o FrameNovaPartida
- o FramePartida
- o FrameRegistrar
- o FrameTaulerGUI
- o FrameTaulerGUIEntrenament
- o ProgramaPrincipalGomoku72
- o TaulerGUI

- **prop.gomoku.recursos:**

- o casella_buida.gif (és una imatge, no és una classe).
- o fitxa_blanca_alt.gif (és una imatge, no és una classe).
- o fitxa_blanca.png (és una imatge, no és una classe).
- o fitxa_negra_alt.gif (és una imatge, no és una classe)
- o fitxa_negra.png (és una imatge, no és una classe).
- o tauler_fons_no_centrat.JPG (és una imatge, no és una classe).
- o tauler_fins.JPG (és una imatge, no és una classe).

3. Responsabilitats del projecte

En aquest apartat es descriuen les relacions de responsabilitat dins del projecte entre les tasques que s'han dut a terme per aquesta tercera entrega i els seus autors, que són els membre del grup (excepte per a les classes compartides, on els autors que hi figuren pertanyen a altres grups dins del mateix clúster).

L'autoria relacionada amb els fitxers que contenen la implementació de les **classes compartides** és la següent (nom del fitxer que conté la implementació de la classe compartida: nom i cognoms de l'autor, grup dins el clúster al qual pertany (projecte associat)):

- `EstatCasella.java`: Grup 7.3 (Hex)
- `EstatPartida.java`: Mauricio Ignacio Contreras Pinilla, grup 7.2 (Gomoku)
- `InteligenciaArtificial.java`: Genís Riera Pérez, grup 7.2 (Gomoku)
- `Partida.java`: Mauricio Ignacio Contreras Pinilla, grup 7.2 (Gomoku)
- `Tauler.java`: Grup 7.3 (Hex)
- `Usuari.java`: Grup 7.1 (Othello)

L'autoria relacionada amb els fitxers que contenen la implementació de les **classes a nivell intern de grup** és la següent, classificada en ordre alfabètic pel nom i cognoms dels autors:

- Alexander Giménez Ortega:
 - `EstadistiquesPartides.java`
 - `LlistaRecordsIndividuals.java`
 - `ResumResultats.java`
 - `UsuariGomoku.java`
 - `ControladorPresentacio.java`
 - `ControladorPresentacio.java`
 - `FrameAccedirEstadistiques.java`

- o FrameBenvingut.java
- o FrameCarregaPartides.java
- o FrameConfiguracioPartida1.java
- o FrameConfiguracioPartida2.java
- o FrameConfiguracioPartida2CPU.java
- o FrameConfiguracioPartida2Persones.java
- o FrameConfiguracioPartida3.java
- o FrameError.java
- o FrameEstadistiquesGlobals.java
- o FrameEstadistiquesIndividuals.java
- o FrameIdentificacio.java
- o FrameMenuPrincipal.java
- o FrameNovaPartida.java
- o FramePartida.java
- o FrameRegistrar.java
- o ProgramaPrincipalGomoku72.java

- **Genís Riera Pérez:**

- o TaulerGomoku.java
- o IAGomoku.java
- o IAGomokuOptimitzada.java
- o Vigenere.java
- o ProvaJugadorsMaquina.java
- o CasellaGUI.java
- o FrameTaulerGUI.java
- o FrameTaulerGUIEntrenament.java
- o TaulerGUI.java
- o LecturaBuffers.java

- **Mauricio Ignacio Contreras Pinilla:**
 - o `ControladorPartidaEnJoc.java`
 - o `ControladorDocumentacio.java`
 - o `PartidaGomoku.java`
 - o `ControladorPartida.java`
 - o `ControladorPartidesGuardades.java`
 - o `ControladorPreparacioPartida.java`
 - o `ControladorUsuari.java`
 - o `InteligenciaCPU.java`
 - o `UsuariJaExisteix.java`
 - o `UsuariNoExisteix.java`
 - o `GestorDocumentacio.java`
 - o `GestorPartidesGuardades.java`
 - o `GestorUsuaris.java`
 - o `TipusUsuari.java`
 - o `AccesRecords.java`
 - o `ControladorRecordsGlobals.java`
 - o `ControladorRecordsIndividuals.java`
 - o `CriteriRecords.java`
 - o `ContrasenyaIncorrecta.java`
 - o `ContrasenyaInvalida.java`
 - o `IAGomokuAlternativa.java`
 - o `Lectura.java`
 - o `LecturaScanners.java`

L'autoria relacionada amb **la resta de fitxers que conformen el projecte** (documentació, manuals, diagrames, etc.) és la següent (nom del fitxer: nom i cognoms de l'autor):

- `Descripcio_algorismes_i_estructures_de_dades.pdf`: Genís Riera Pérez
- `Diagrama_Presentacio.png`: Mauricio Ignacio Contreras Pinilla
- `Diagrama_Domini.png`: Mauricio Ignacio Contreras Pinilla
- `Diagrama_Gestio_Dades.png`: Mauricio Ignacio Contreras Pinilla
- `Manual_usuari.pdf`: Genís Riera Pérez
- `Evolucio_organitzacio_i_responsabilitats_del_projecte.pdf` (document actual): Genís Riera Pérez

L'autoria relacionada amb el **disseny dels fitxers de recursos** presents al *package* `prop.gomoku.recursos` corre a càrrec del membre del grup Genís Riera Pérez (responsable de dissenyar les imatges emprant software extern de disseny fotogràfic bàsic).

L'autoria relacionada amb la **cerca bibliogràfica d'informació** per al desenvolupament dels algorismes emprats en el projecte (els descrits en el fitxer PDF `Descripcio_algorismes_i_estructures_de_dades.pdf`) corre a càrrec del membre del grup Genís Riera Pérez (responsable de cercar, estriar i llegir tot el contingut del directori *Bibliografia* preparat expressament per aquesta tercera entrega).