



AGE ROYALE

Grup C6 – Mentor Jordi Malé

DESCRIPCIÓ DEL PROJECTE

Projecte basat en el famós joc Clash Royale realitzat durant el curs 2020/21 pels següents alumnes: Inés Graells Sanz, Pol Piñol Castuera, Marta Hernández Jiménez, Giovanni Maurizio Vecchies i Oriol Guimó Morell

Object Oriented Programming and Design

Índex

1	Resum de les especificacions del projecte	2
2	Disseny de la interfície gràfica	4
2.1	Interfície d'entrada al programa – LogoPanel.....	5
2.2	Interfície d'inici de sessió de l'usuari - UserPanel.....	6
2.3	Interfície de registre d'un nou usuari - RegisterPanel.....	8
2.4	Interfície del menú del joc – MenuPanel.....	10
2.5	Interfície de l'historial – HistoryPanel.....	11
2.6	Interfície dels rànquings – RankingPanel.....	12
2.7	Interfície del Dialog Configuració – SettingsDialog	13
2.8	Interfície del Dialog d'instruccions del joc – InstructionsDialog	14
2.9	Interfície del joc – GameView (JFrame).....	15
2.10	Interfície del joc en repetició – ReplayView (JFrame).....	18
2.11	Interfície del Dialog per guardar el joc – SaveGameDialog	20
3	Diagrama de classes	21
3.1	Capa PresentationLayer	22
3.2	Capa BusinessLayer	26
3.3	Capa PersistanceLayer.....	28
4	Metodologia de desenvolupament.....	29
4.1	Age Royale Sprint 1	29
4.2	Age Royale Sprint 2.....	29
4.3	Age Royale Sprint 3.....	30
4.4	Age Royale Sprint 4.....	30
5	Dedicació.....	31
6	Conclusions	32
7	Bibliografia	33

1 Resum de les especificacions del projecte

En aquest projecte hem creat una versió del popular joc Clash Royale. El conjunt total del programa consta d'un total de 7 pantalles.

El programa consta d'una pantalla de benvinguda, inici de sessió, registre de nou participant, menú amb les opcions del joc, rànding amb opció a reproduir partides d'altres usuaris, historial de partides pròpies, i finalment, la pantalla del joc.

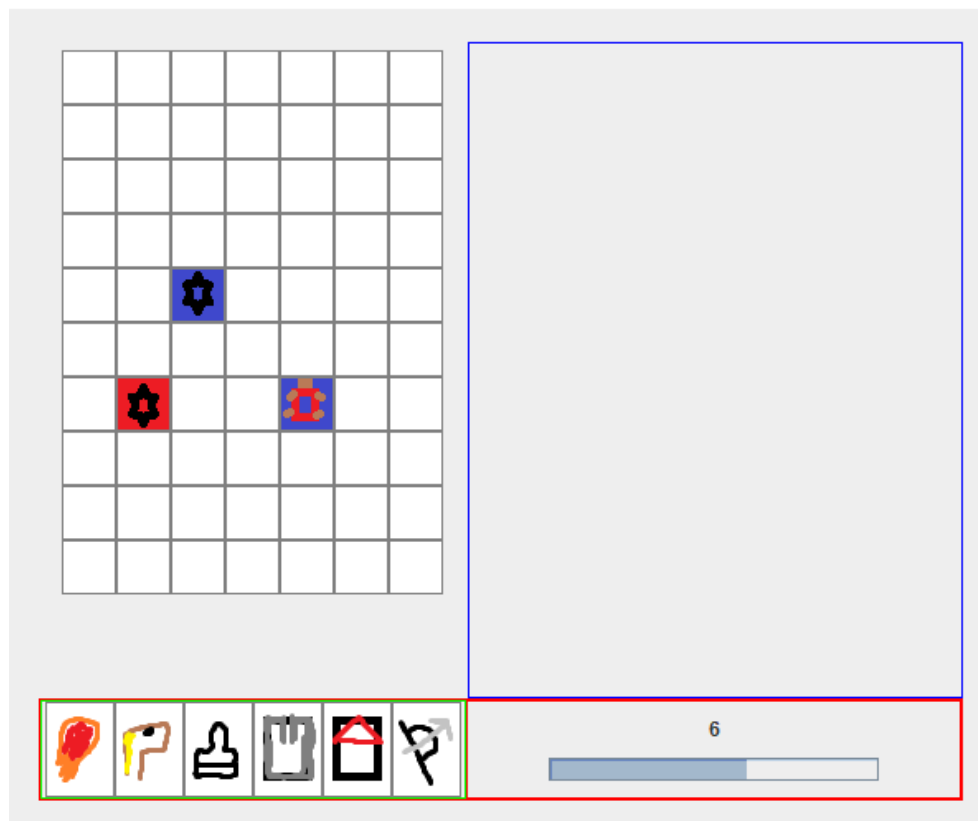
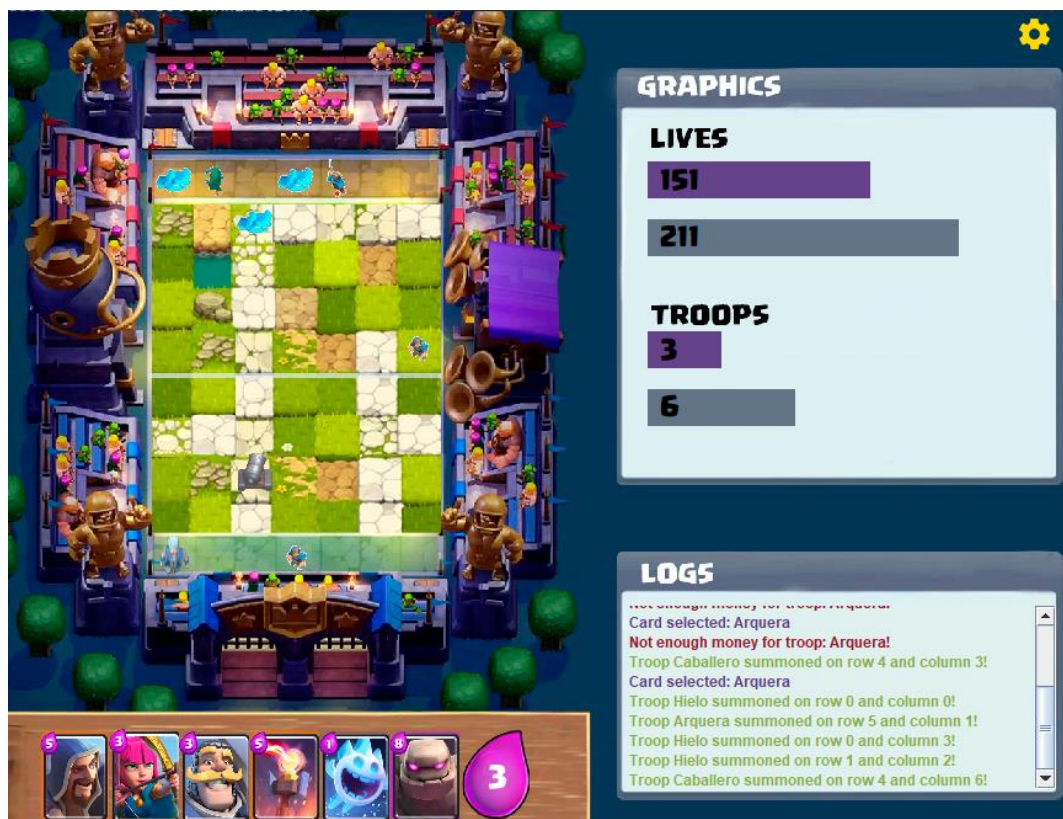
En iniciar el programa entrem al panell de benvinguda, seguit d'aquest es mostra el panell d'iniciar sessió amb l'opció de registrar un nou usuari (en el cas d'iniciar sessió s'accedeix al menú, mentre que si es registra un nou participant es torna al panell d'inici de sessió), un cop iniciada la sessió anem a parar al menú, a on ens trobem amb tres opcions (jugar partida, veure el rànding o anar al nostre historial de partides).

En tot moment del programa tenim un botó de configuració que ens permet sortir del programa o esborrar la partida. En alguns panells (rànding i historial) també tenim l'opció de tornar al menú principal i així no haver d'estar iniciant sessió tota l'estona, en el cas del joc només pots tornar al menú quan acaba la partida, si vols sortir abans es fa un "logout".

El joc consta d'un taulell de 7 columnes i 10 files, a on cada quadrat del taulell tenim un JPanel. L'objectiu d'aquest és d'eliminar totes les vides del contrincant a través de llençar unes tropes que aconseguim per mitjà de monedes. El nombre de monedes augmenta cada 2 segons i cada vegada que mates una tropa.

A part de les cartes de les tropes i el nombre de monedes, el joc consta d'un panell informatiu amb missatges importants del joc, que poden ajudar a l'usuari, i un altre panell amb dos gràfics que mostren el nombre de vides del mateix usuari i de la IA, d'aquesta manera sabem en tot moment l'evolució de la partida.

A continuació es mostra un petit resum del treball inicial i final, amb totes les seves millores i especificacions de l'enunciat.

Versió inicial del joc:**Versió final del joc:**

2 Disseny de la interfície gràfica

La pràctica consta de tres JFrame, dos JDialog.

Un JFrame que conté tots els panels (pantalla d'inici, login, register, menu, ranking, historial). Per aconseguir tenir tots aquests panels en un mateix JFrame hem emprat un CardLayout. Al principi mostrem la pantalla de benvinguda, i a partir d'aquí anem fent swapCardLayout als corresponents controllers de cada panel per anar passant d'un panel a un altre, d'aquesta manera no creem tants JFrames.

El segon JFrame conté la nova partida del joc i el tercer conté les repeticions dels jocs, que en definitiva s'ha reutilitzat tot el que es tenia creat al joc, amb l'excepció que també s'ha afegit l'opció de reproduir o parar la reproducció d'aquest.

2.1 Interfície d'entrada al programa – LogoPanel

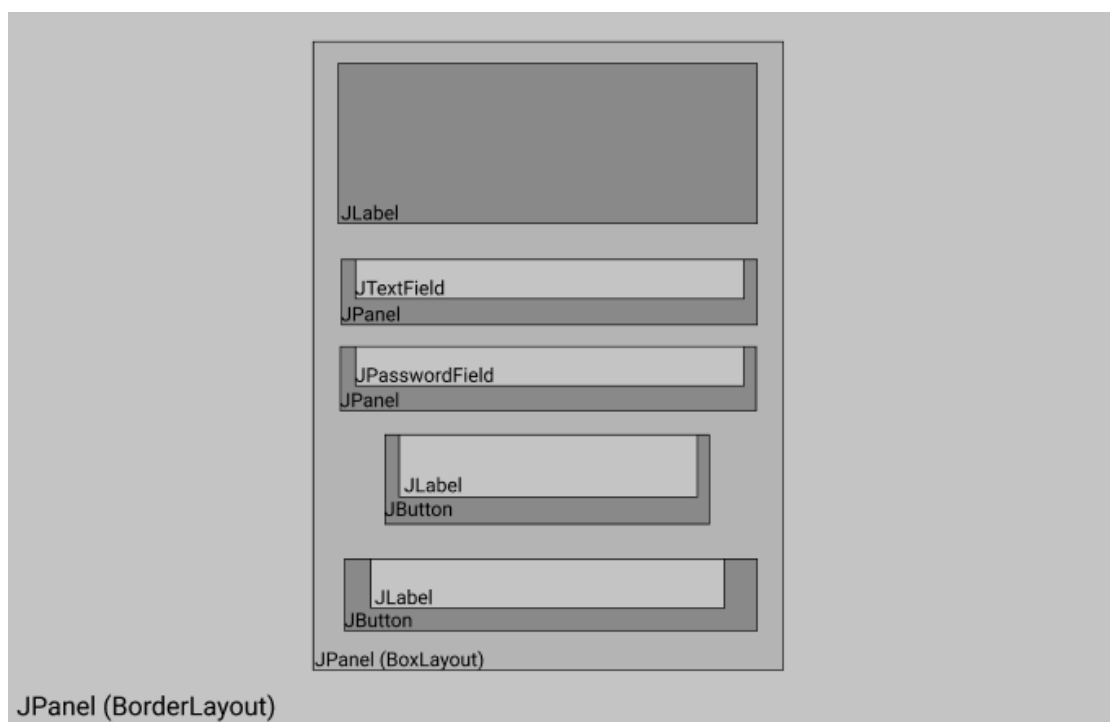


En aquest card del JFrame hem emprat només un JPanel amb una imatge i li hem posat un listener al mateix panel per a que passi al següent.

A part d'això hem creat diverses classes per acomodar certes accions, com es el cas de posar una imatge de fons a un JPanel (degut a que no hem trobat una manera més senzilla de fer-ho).

Com que només era la pantalla del títol no necessitàvem molts elements més que una imatge de benvinguda així que un JPanel senzill amb la imatge i un listener a aquest ja hem considerat que eren suficients.

2.2 Interfície d'inici de sessió de l'usuari - UserPanel



En aquest cas hem emprat un JPanel bàsic amb la imatge de fons (una altra vegada utilitzant la classe JPanel que hem creat per pintar jPanels), a partir d'aquest ens trobem un altre JPanel que ja conté tots els elements més importants d'aquesta pantalla en aquest JPanel hem afegit un BorderLayout per tal de que ens afegeixi els elements un sota de l'altre de manera automàtica (tot i que hem afegit un element anomenat "verticalGlue" entre cada element afegit per a ajustar la distància entre cadascun dels elements posats).

Com hem escrit a la segona imatge, el títol està format per un JDialog (elements com el JDialog o el JButton ja permeten posar imatges per mitjà de la comanda setIcon, per tant no hem hagut de crear res nou, tot i que per millorar l'aspecte dels botons hem creat un parell de classes).

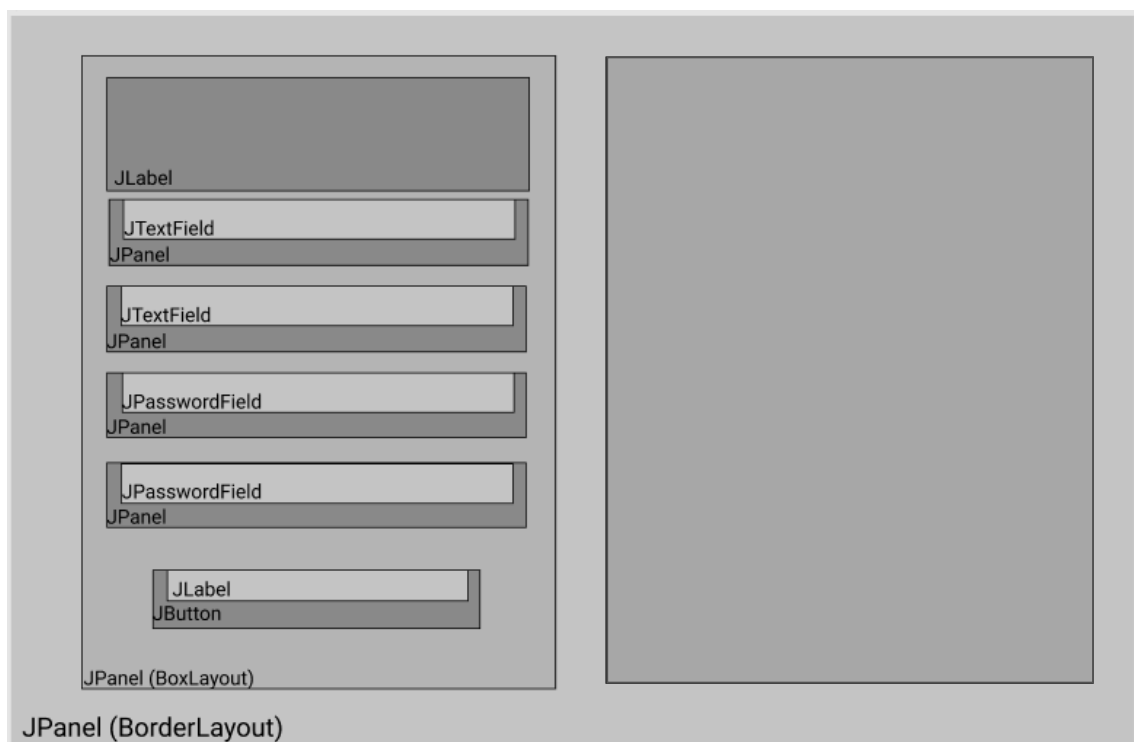
A continuació tenim dos camps de text que hem modificat amb dues classes que hem creat (RoundJText i RoundJPassword) el primer text field permet inserir el nom d'usuari o email, i el segon es per la contrasenya, tots dos han sigut modificats per a que es comenci a escriure en un punt determinat d'aquests i per a arrodonir les seves cantonades (en el cas de la contrasenya també s'ha modificat per a poder escriure-la sense que es vegi el que tenim escrit).

També em treballat amb listeners per a que quan l'usuari no hagi escrit res pugui veure el que li toca posar a cada apartat i que en cas de que cliqui a un desaparegui i el text i quedi només el que ha escrit ell.

Seguidament tenim el JButton d'iniciar sessió que també ha estat modificat per a que canviï de forma i d'opacitat, amb aquest accedirem al programa.

Per últim es troba un JButton amb un JLabel que mostra el cas per registrar-se, a partir d'aquest s'accedeix a la pantalla de registre, s'ha fet d'aquesta manera per a que es cliqui a on es cliqui de la frase, sempre s'accedeixi al registre, sense haver de donar-li a un lloc en concret.

2.3 Interficie de registre d'un nou usuari - RegisterPanel



Aquest card està format per un JPanel principal que conté dos panels més que divideixen cadascun el contingut total del card, aquest panel té assignat un BorderLayout en X_AXIS d'aquesta manera els elements es posen un al costat de l'altre en comptes d'un sobre de l'altre segons anem afegint elements.

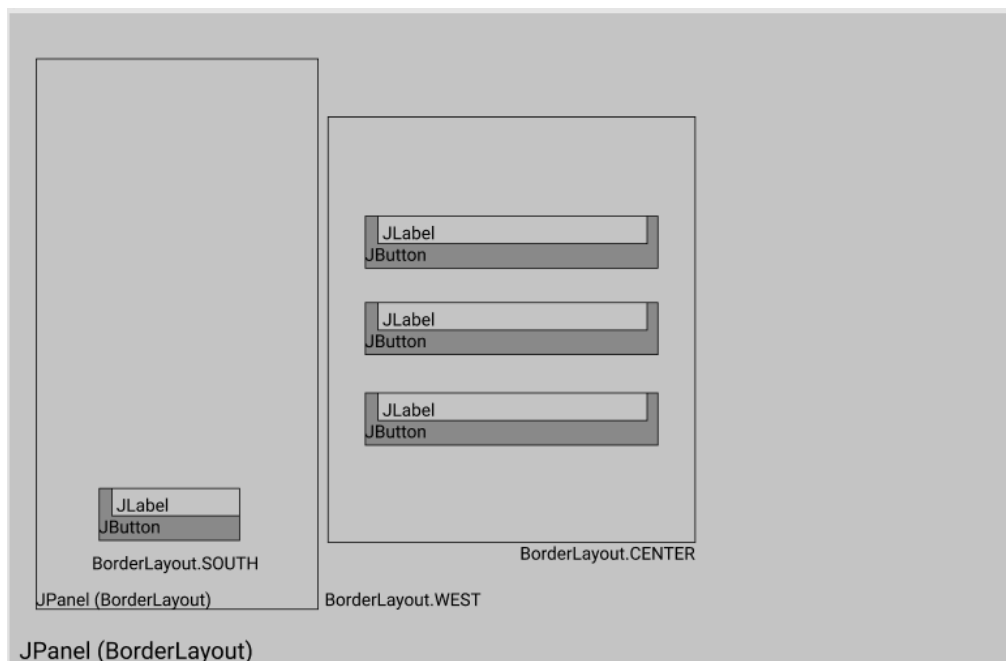
Al JPanel de l'esquerra trobem tots els elements per escriure la informació del registre, aquests són els mateixos que en el cas de l'inici de sessió. Aquest panel està format també per un BorderLayout per tal de que cada element afegit quedi un sota l'altre.

Tenim el títol un JLabel altre cop, un missatge a sota que explica l'objectiu del panell en concret i a continuació els panells de text (que tornen a ser JTextArea per a seguir amb el mateix estil de la pantalla anterior i així aconseguir un disseny unificat).

Per últim trobem el boto de "següent" per a guardar la informació (en aquest cas també es canvia la opacitat al passar el ratolí per sobre i així donar un efecte al botó).

El JPanel de la dreta està pintat per a que quedi igual que la resta del fons del panel principal, aquest conté un JLabel amb la imatge del personatge, es una manera senzilla d'afegir imatges.

2.4 Interfície del menú del joc – MenuPanel



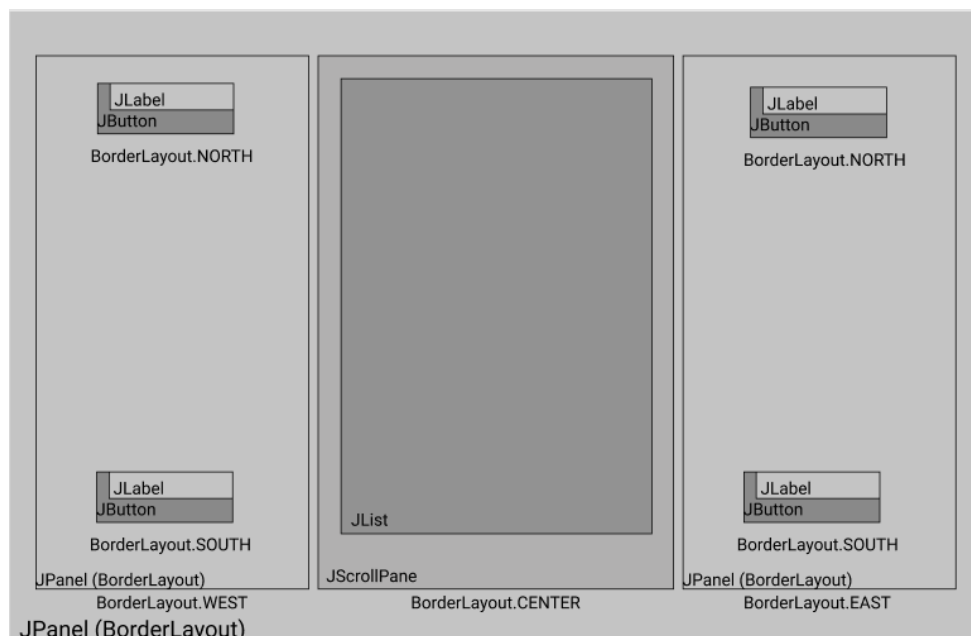
Aquest panel està format per un JPanel principal que té un BorderLayout assignat per tal de posicionar els elements de manera senzilla assignant les posicions per a que ocupin el seu lloc designat directament. Aquest mateix JPanel conté la imatge de fons, la resta de JPanels tenen posat una imatge transparent per així unificar tot el panell i que quedi una mateixa imatge.

Començant per l'esquerra es troba un JPanel al que li hem assignat també un BorderLayout, aquest només conté el boto de Settings al peu del panel.

Al centre de la pantalla trobem el JPanel que conté el títol i els tres botons amb les opcions del menú, aquest està ordenat així amb un BoxLayout, entre cada element tornem a trobar un verticalGlue per separar-los a certa distància.

En aquest cas els hi hem afegit un gif al fons dels botons per donar la sensació de que estan animats.

2.5 Interfície de l'historial – HistoryPanel



En aquest card tenim el JPanel principal que està format per la imatge del fons de pantalla i un BorderLayout per posicionar els tres JPanels corresponents que hem afegit, en pràcticament tots els cards podríem fer servir BorderLayout o BoxLayout sense molta diferència.

Al panel de l'esquerra trobem dos JButtons modificats, un per tornar al menú i un per anar als Settings, aquest panel es distribueix amb un BorderLayout, per tal de posar ràpidament un botó a South i un a North.

Al panell central es troba el JList amb un llistat de JDialogs interactius que obren la partida corresponent, així com el títol del card al principi de tot.

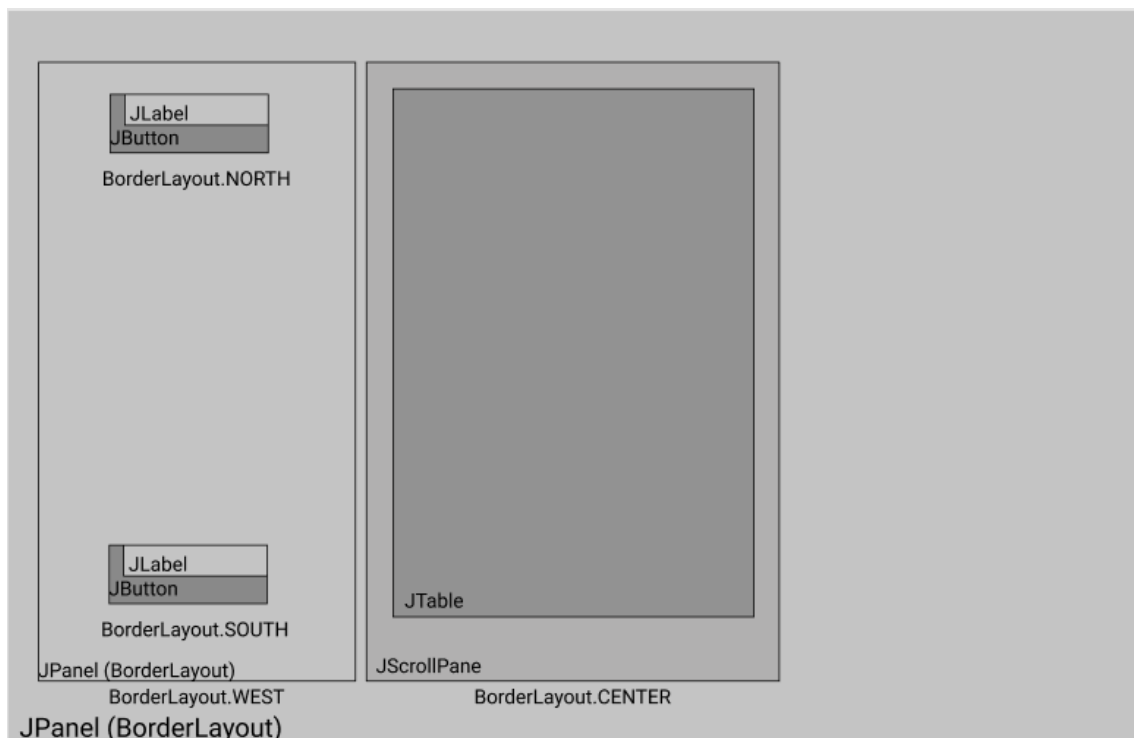
El JList es troba a dins d'un JScrollPane per tal de que si tenim més partides que espai a la pantalla les podem visualitzar totes sense cap problema.

Per últim al JPanel de la dreta hem aplicat el mateix que al de l'esquerra BorderLayout per posicionar els dos botons i posar-los a SOUT i NORTH.

2.6 Interfície dels rànquings – RankingPanel



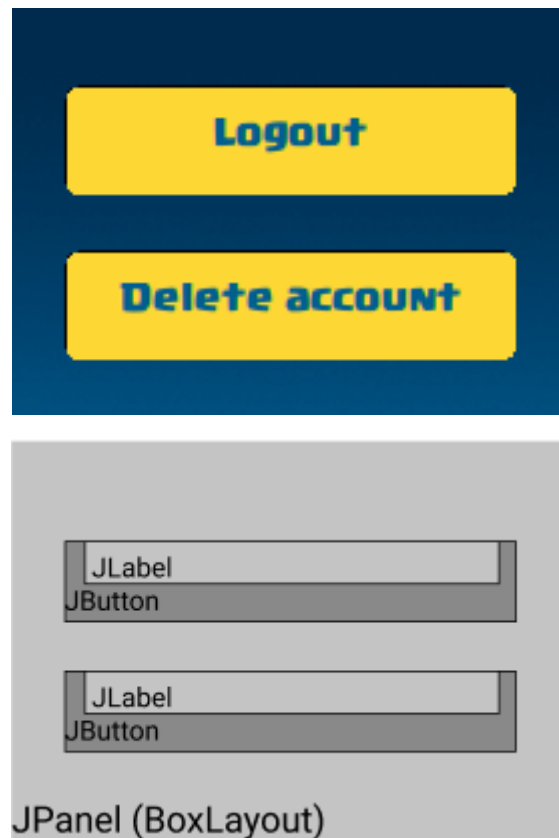
Player	Wins	Win Ratio
MAMMA	31	100.0%
HACE	20	66.0%
HENRY	12	80.0%
donald	10	66.0%
pistol	9	60.0%
wonka	5	33.0%
marcelo	5	33.0%
umpa	2	13.0%
pato	0	0.0%
505	0	0.0%



En aquest card ens trobem amb un JPanel principal que conté la imatge de fons i que està format per un BorderLayout, aquest conté un JPanel posicionat a WEST que conté els botons per tornar al menú o obrir el JDialog dels Settings i un JPanel que conté el JTable i el títol del panel.

La JTable es troba a dins d'un JScrollPane que a la vegada està a dins d'un altre JPanel. El center JPanel està format per un BorderLayout així que queda el JTable sota el JLabel del títol.

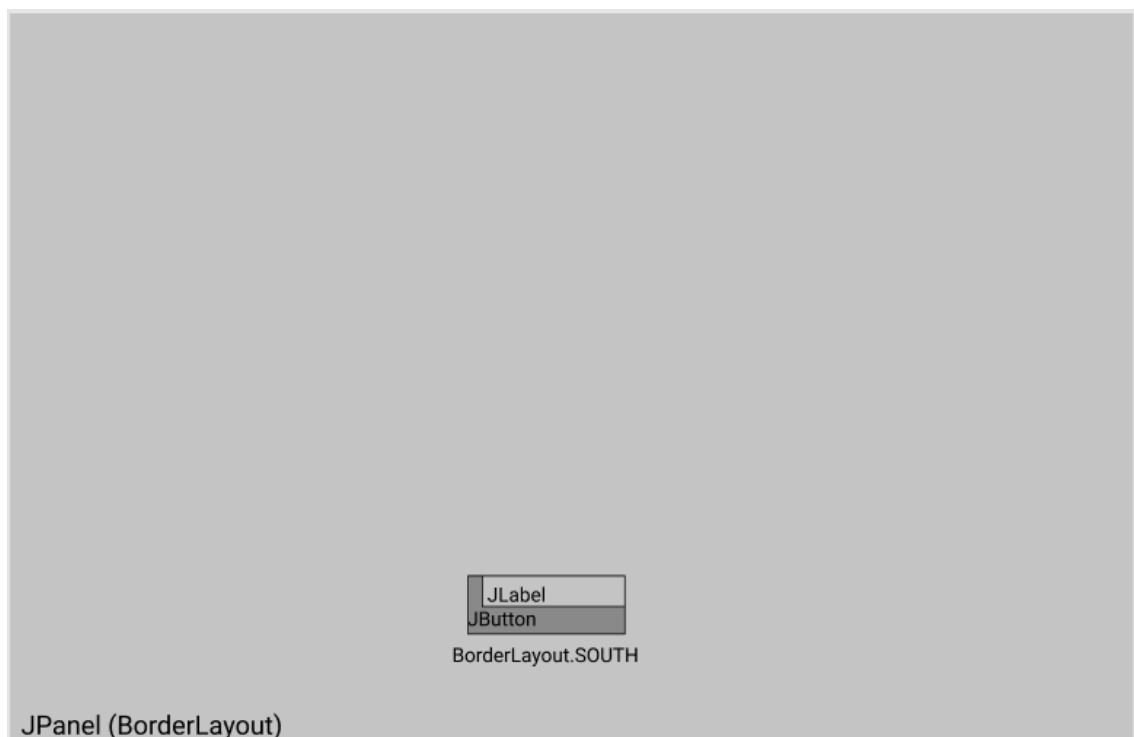
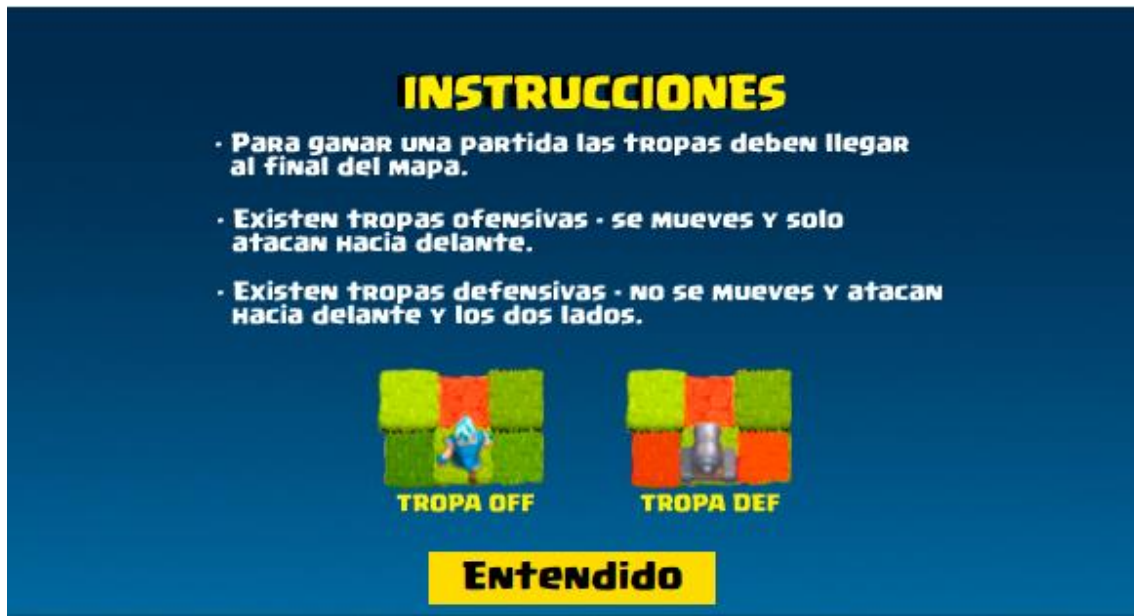
2.7 Interfície del Dialog Configuració – SettingsDialog



En aquest cas ens trobem amb un `JDialog` format per un `JPanel` que conté els dos `JButtons` i els ordena per mitjà d'un `BoxLayout`.

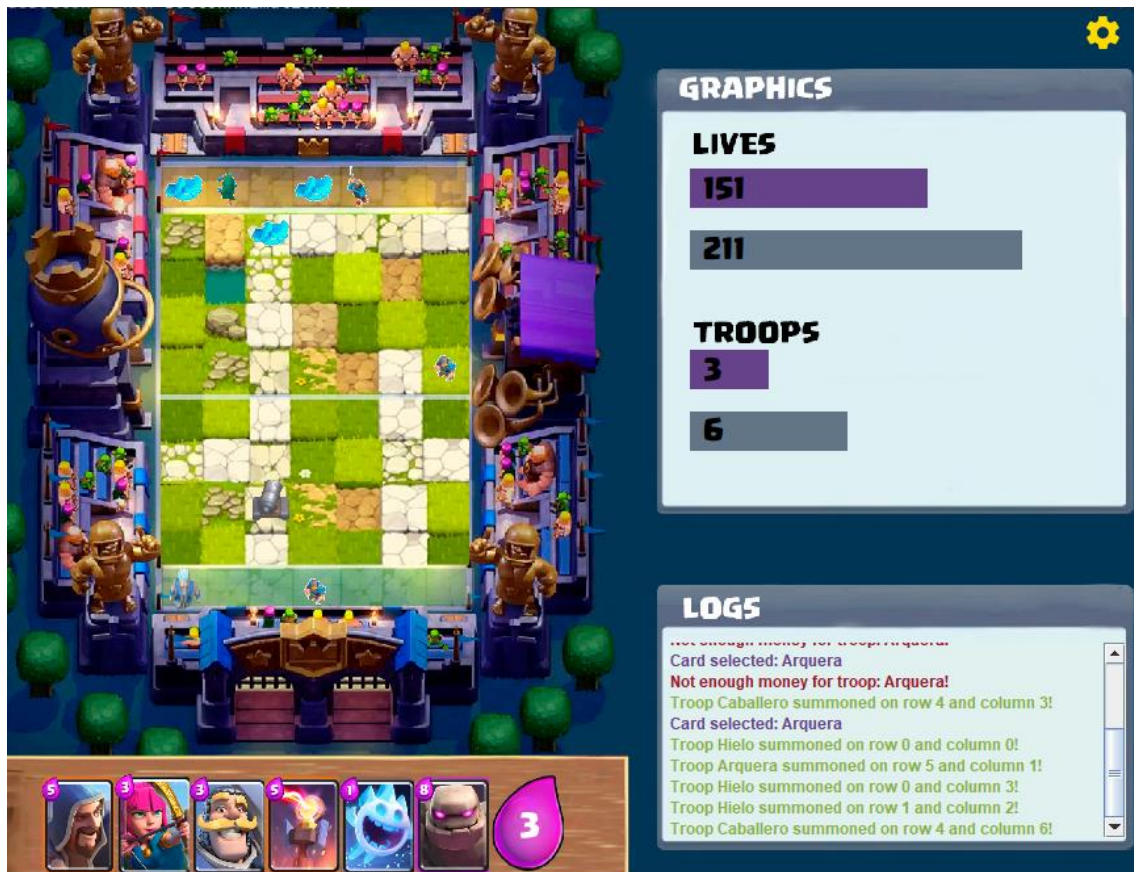
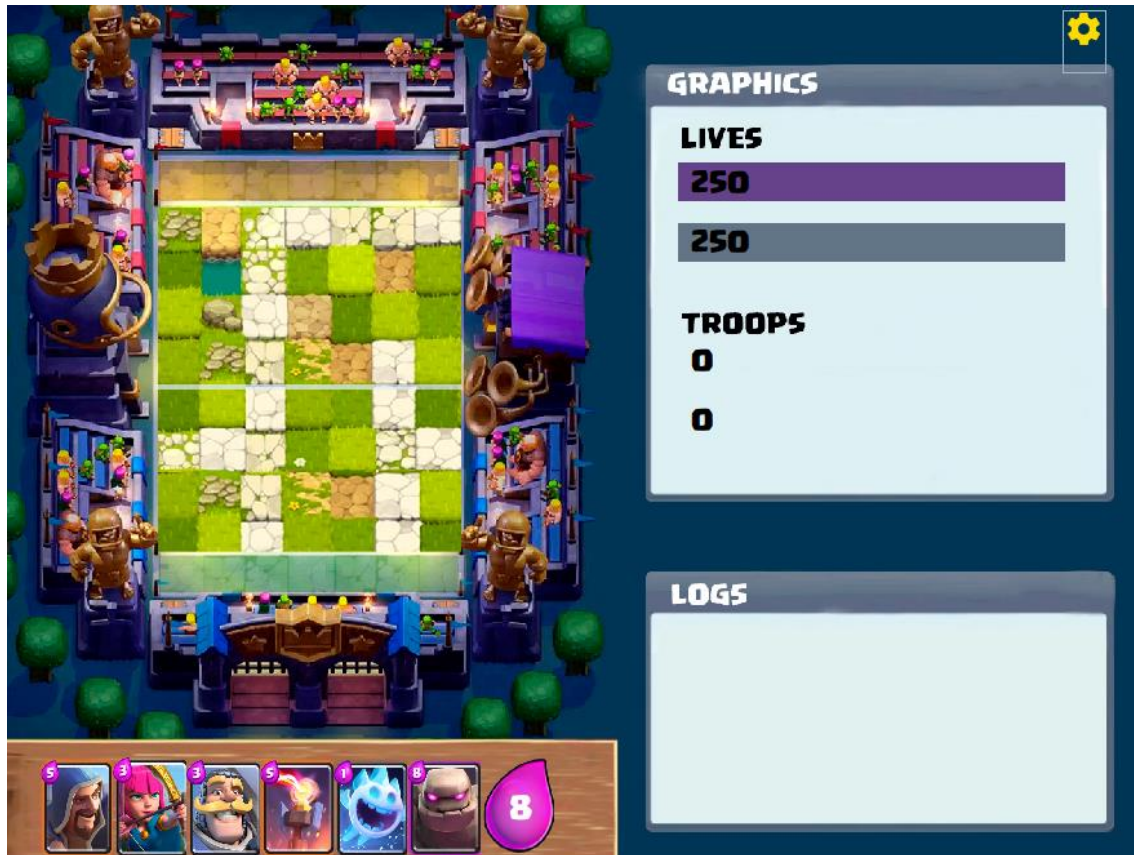
Els dos botons també estan modificats per a que canviïn d'opacitat al passar per sobre. En aquest cas estan formats per un `RoundJPanel` que canvia la forma del `JPanel` i arrodoneix les cantonades, sobre aquest es troba el `JButton` corresponent per tal de fer la crida al listener, aquest `JButton` també conté un `JDialog` amb el text corresponent del botó. En el cas dels botons es podia fer directament un `setText`, però aquesta funció posa el text d'una manera determinada que no permet modificar-los igual que un `JDialog`, per aquesta raó fem servir `JDialogs` per posar els texts als botons, a més que hem afegit una tipografia pròpia que tenim guardada i utilitzem a tots els títols de la pràctica.

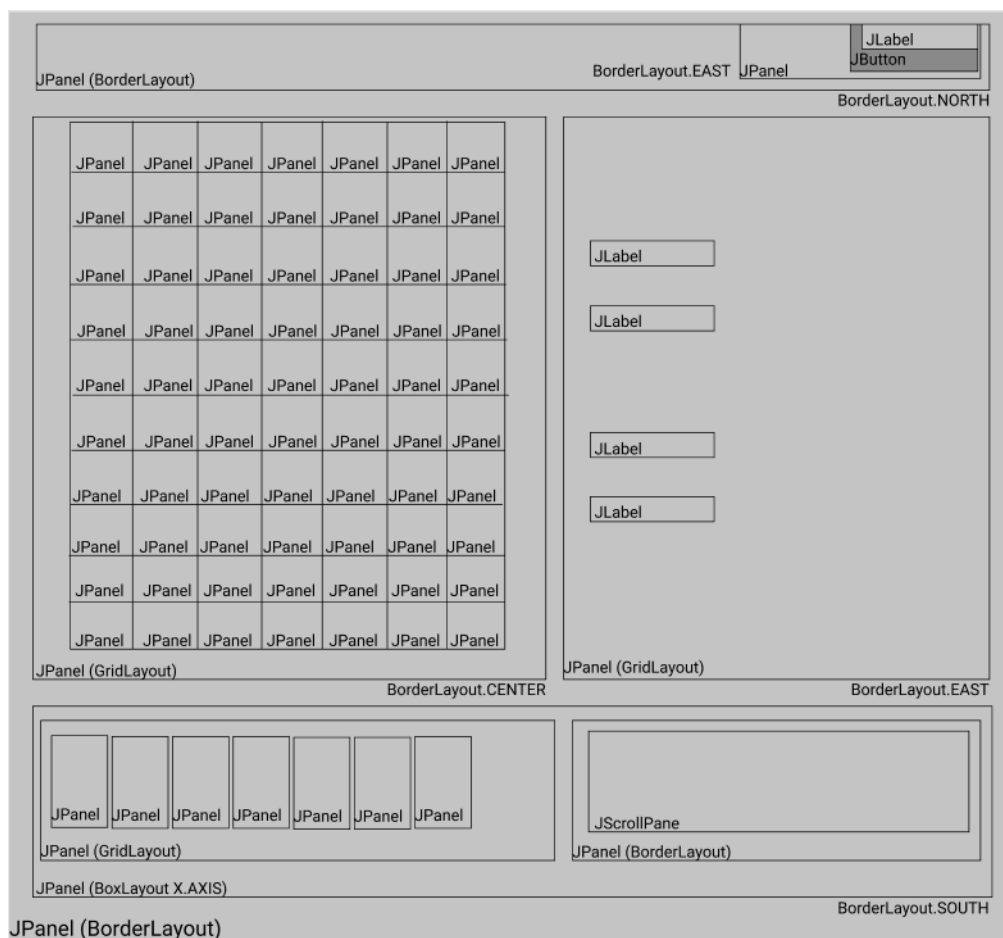
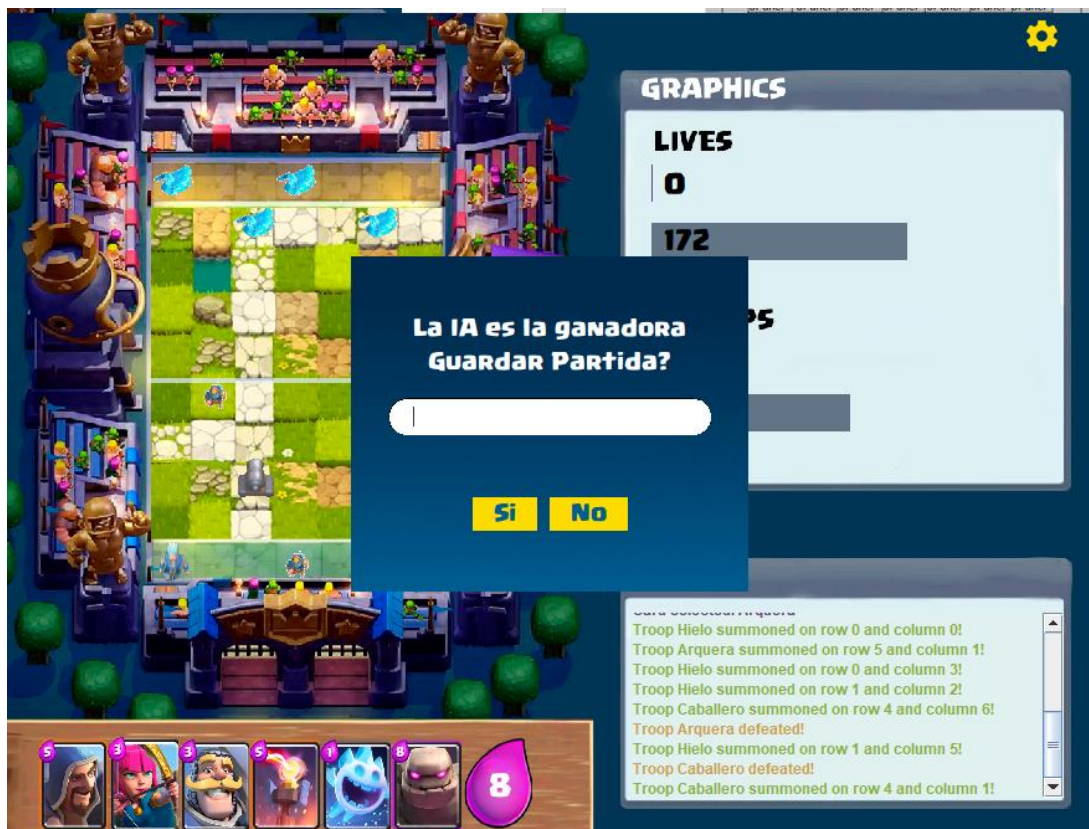
2.8 Interficie del Dialog d'instruccions del joc – InstructionsDialog



Aquest card conté la imatge de fons de les instruccions més un jButton per accedir al joc. El jButton conté un jDialog amb el text d'aquest.

2.9 Interficie del joc – GameView (JFrame)





Aquest JFrame està format per un JPanel bàsic que ordena segons BorderLayout els quatre JPanels que hem afegit i conté també la imatge de fons del joc.

El north panel conté el boto de Settings que està format per un JPanel bàsic que conté el botó corresponent.

Per la ma de la zona del westPanel tenim un JPanel que organitzat amb GridLayout per tal de crear una matriu per a cada JPanel que forma cada casella del joc. Cada JPanel té assignat un listener per poder fer un seguiment de les posicions de cada tropa sobre el taulell.

Tots els panels afegits sobre el panel principal tenen una imatge transparent de fons assignada per tal de que es vegi el taulell que hem posat.

Per la banda del eastPanel tenim el panell que conté els gràfics amb les vides de l'usuari i la IA en aquest cas es tracta d'un JPanel que també té declarat el layout en forma de GridLayout, per tal assignem cada element de manera matricial amb coordenades.

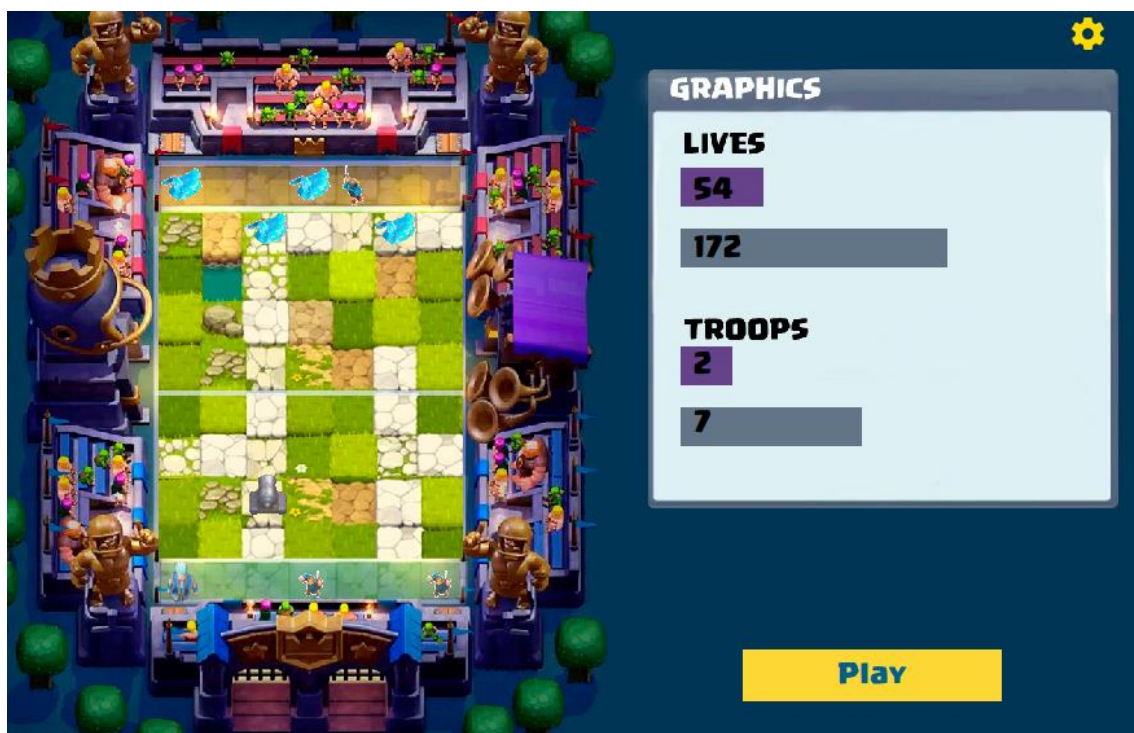
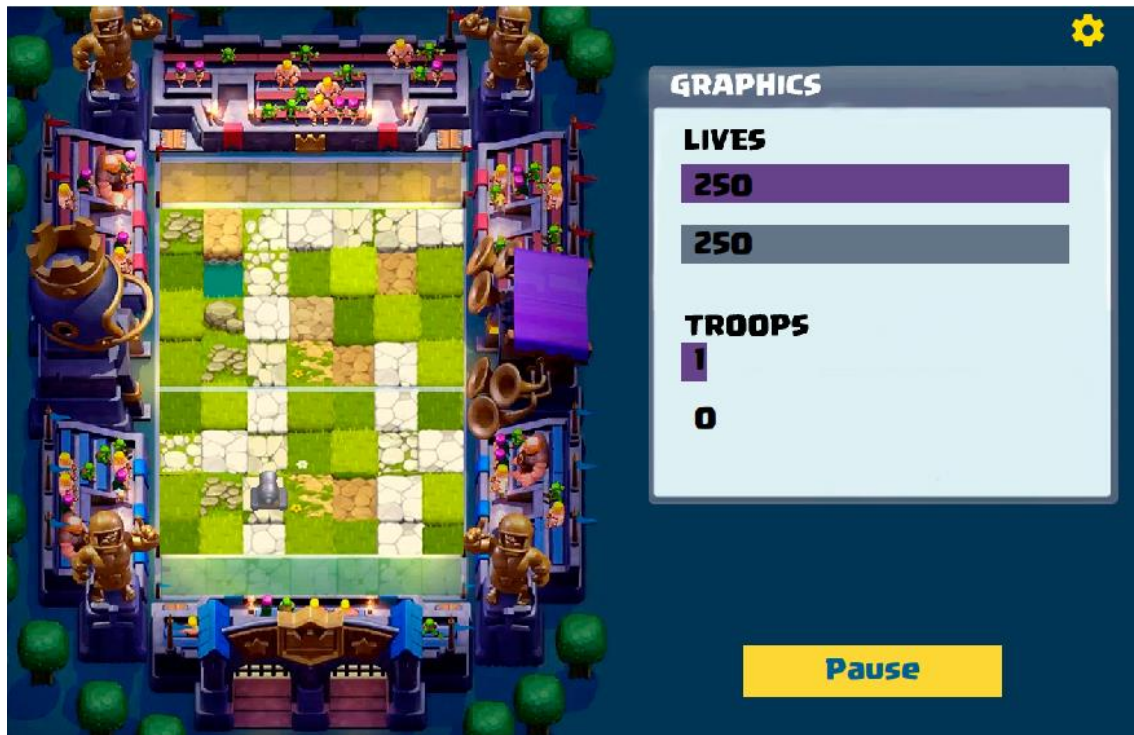
Finalment tenim el south panel que conté dos JPanels amb funcionalitats molt diferents. En aquest cas utilitzem BoxLayout ordenant cada element amb X_AXIS.

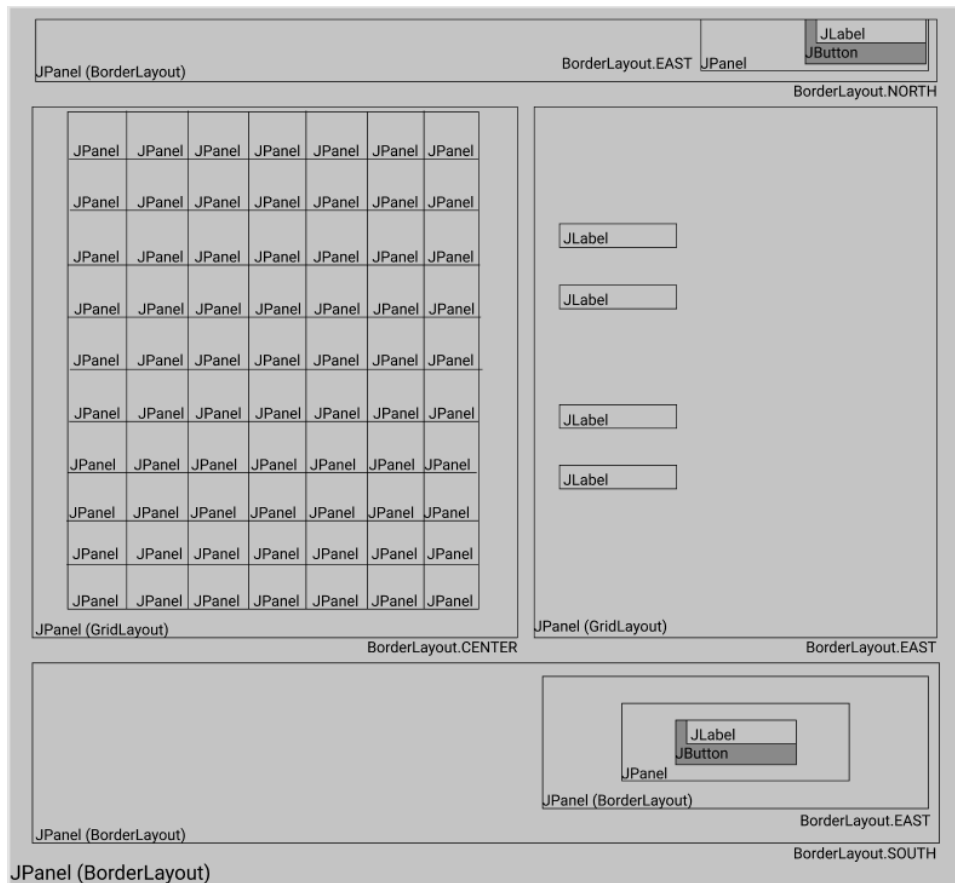
En el cas del JPanel afegit a l'esquerra conté les monedes i les cartes de les tropes, aquestes estan posades dins d'un JPanel ordenat també amb GridLayout. Cada carta conté el seu propi listener per així fer una acció quan es selecciona aquesta.

Per altra banda tenim el panell de la dreta organitzat amb BorderLayout que conté els missatges del Logs (que bàsicament son missatges de l'estatus de la partida, com la selecció d'una tropa o que se't han afegit més monedes, o que no tens monedes suficients per un personatge), tots aquests missatges es controlen per mitjà de listeners afegits a la resta d'elements del joc que ens retornen la informació de cada acció de la partida, igual passa en el cas de les monedes.

El JPanel dels Logs conté cinc JPanels a dins essent el JPanel de l'scrollable el central, ho hem fet d'aquesta manera per modificar la mida d'aquest degut a la posició de l'scroll al taulell poder-lo ajustar millor a la posició que volem.

2.10 Interfície del joc en repetició – ReplayView (JFrame)





Aquest JFrame està format per un JPanel bàsic que ordena segons BorderLayout els quatre jPanels que hem afegit i conté també la imatge de fons del joc. El north panel conté el boto de Settings que està format per un JPanel bàsic que conté el botó corresponent.

Per la ma de la zona del westPanel tenim un JPanel que organitzat amb GridLayout per tal de crear una matriu per a cada JPanel que forma cada casella del joc. Cada JPanel té assignat un listener per poder fer un seguiment de les posicions de cada tropa sobre el taulell.

Tots els panels afegits sobre el panel principal tenen una imatge transparent de fons assignada per tal de que es vegi el taulell que hem posat.

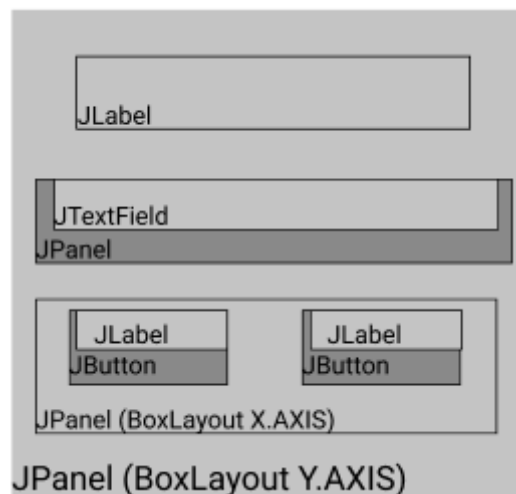
Per la banda del eastPanel tenim el panell que conté els gràfics amb les vides de l'usuari i la IA en aquest cas es tracta d'un JPanel que també té declarat el layout en forma de GridLayout, per tal assignem cada element de manera matricial amb coordenades.

Finalment tenim el south panel que conté dos jPanels amb funcionalitats molt diferents. En aquest cas utilitzem BoxLayout ordenant cada element amb X_AXIS.

En el cas del JPanel afegit a l'esquerra conté les monedes i les cartes de les tropes, aquestes estan posades dins d'un JPanel ordenat també amb GridLayout. Cada carta conté el seu propi listener per així fer una acció quan es selecciona aquesta.

Per altra banda tenim el panell de la dreta organitzat amb BorderLayout que conté els botons per pausar i reproduir la partida. Cadascun d'ells conté el seu JOptionPane corresponent.

2.11 Interficie del Dialog per guardar el joc – SaveGameDialog



En aquest cas tenim un `JDialog` que es mostra un cop a finalitzat la partida. Aquest està format per una imatge de fons i ordenat per mitja de `BoxLayout` per tal de que el títol del dialog, el `JTextField` per guardar el nom de la partida i el `JPanel` que conté els botons estiguin ordenats un sobre l'altre, excepte en el cas dels botons que estan en un altre `JPanel` ordenat per `BoxLayout X_AXIS`.

3 Diagrama de classes

****Com que les captures de pantalla del disseny a UML pot arribar a ser il·legible en la memòria, s'ha adjuntat el fitxer UML i un fitxer PDF on es pot fer zoom al UML, per un millor visualitzat.**

Primerament vam dividir la pràctica en 3 grans apartats: `BusinessLayer`, `PersistenceLayer` i `PresentationLayer`. Aquest model es pot comparar amb el MVC tenint en compte que en aquest cas separem segons el que volem mostrar per pantalla, la lògica del joc i l'accés a la base de dades, a diferència del model MVC en el fet que tindríem el que mostrem per pantalla, la lògica del joc + l'accés a la base de dades i el controlador per passar la informació entre el model i la vista.

Tot i això hem creat carpetes de MVC dins de les de Business, Persistence i Presentation, de manera que hem creat una carpeta de Model dins de Business, i les carpetes de Controlador i Vista dins de Presentation.

A l'apartat de Presentation, dins del View, tenim tots els arxius relatius a Swing. En aquesta pràctica hem creat tres `JFrame`. El primer `JFrame` conté totes les pantalles de Benvinguda al programa, inici de sessió, registre de nou participant, menú, ranking i historial, conté totes aquestes pantalles gràcies al fet que ho hem organitzat per `CardLayout` i per tant cadascuna correspon a un `Card`. En el segon `JFrame` trobem el joc i en el tercer `JFrame` trobem la repetició d'una partida guardada.

Dins de l'apartat de Presentation tenim també el `Controller`, el qual ens permet controlar totes les accions dels listeners posats al Swing, així com transmetre la informació que es va generant al joc i enviar o rebre dades de la base de dades per mitjà d'interfícies ubicades a la capa de Persistence.

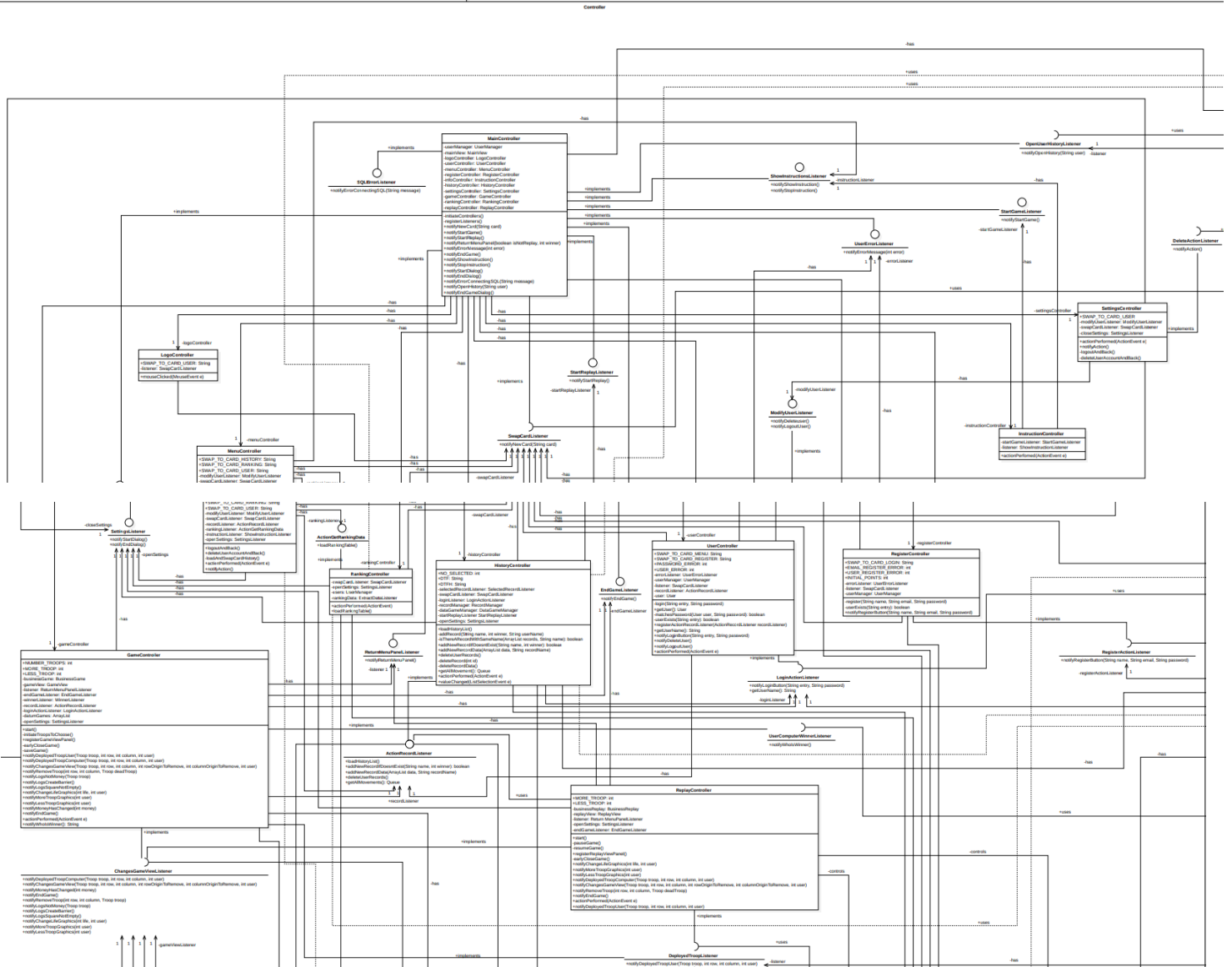
D'altra banda hem fet molt ús de les interfícies al llarg de tot el programa, en el cas de les interfícies dels controladors, algunes les fem servir per comunicar-nos entre controladors o amb algun dels panels del view.

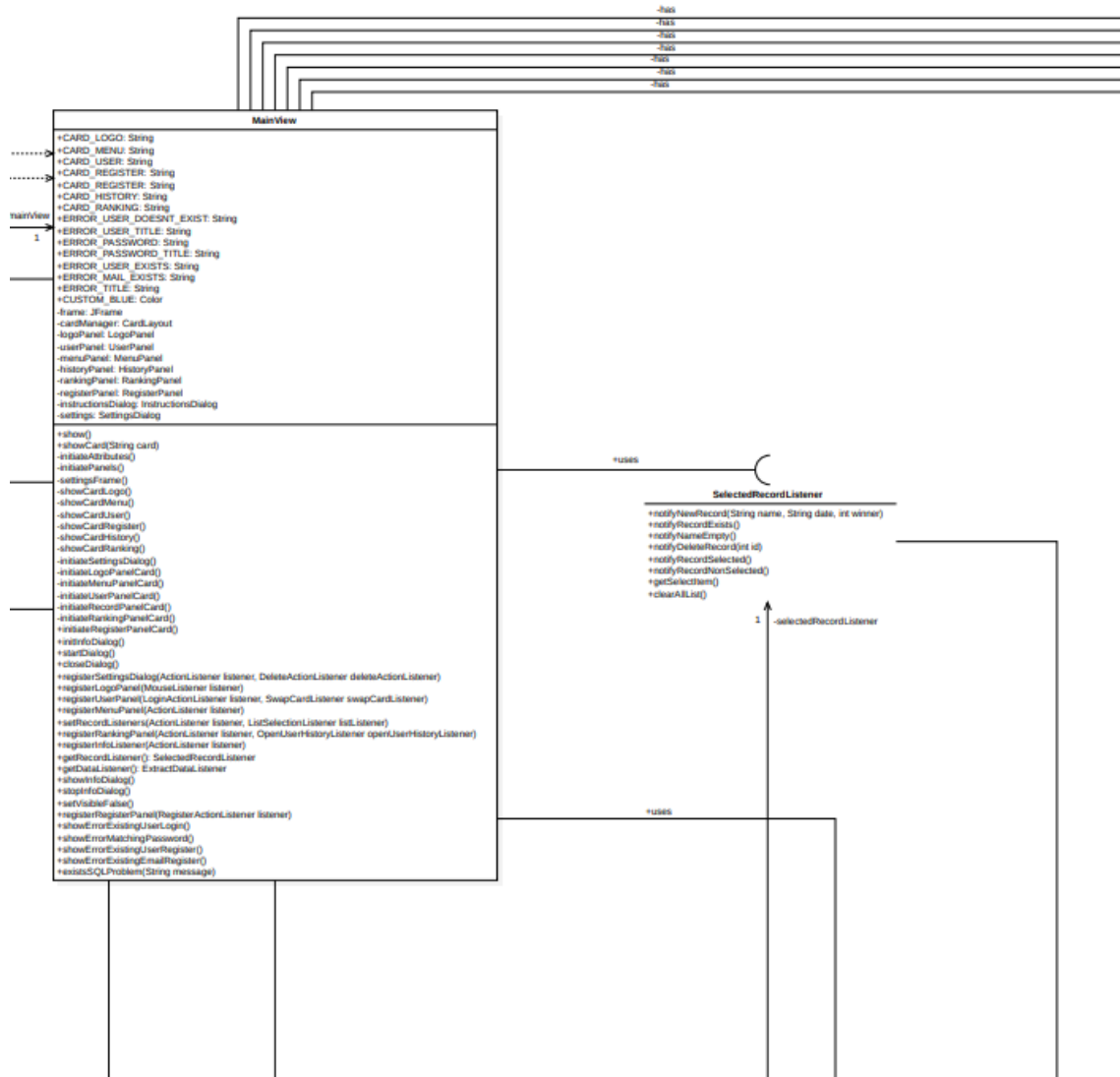
Per part del Persistence layer tenim diferents classes que utilitzem per connectar-nos a la base de dades, hem creat un DAO per cada tipus de tasca que necessitàvem (dades dels usuaris, connexió amb la BD, informació de les tropes, ...), cadascuna de les DAOs té una interfície corresponent que és la que els hi envia les ordres del que es demana des del controller o des d'on es necessita.

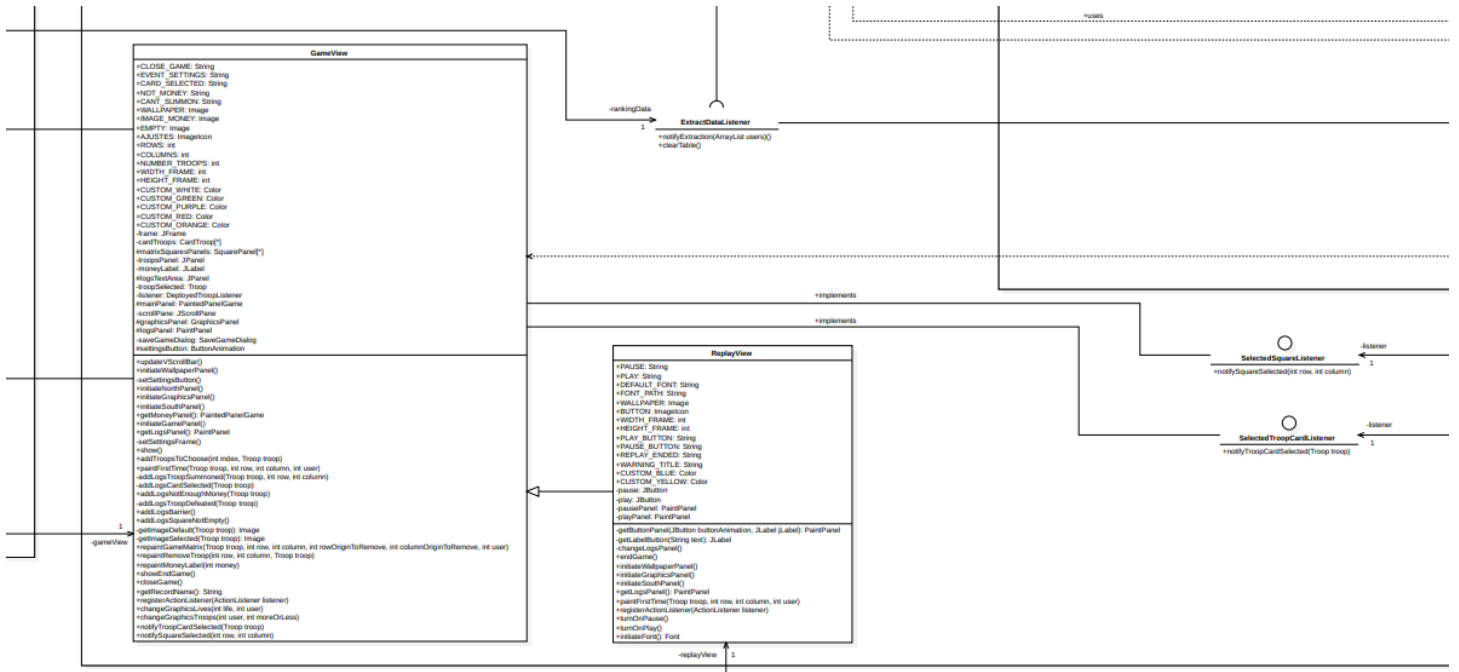
Per últim tenim el Business layer que conté tot el pes de la lògica del joc i la seva reproducció, aquesta conté també unes interfícies per transmetre els canvis que podem necessitar en altres apartats del programa.

3.1 Capa PresentationLayer

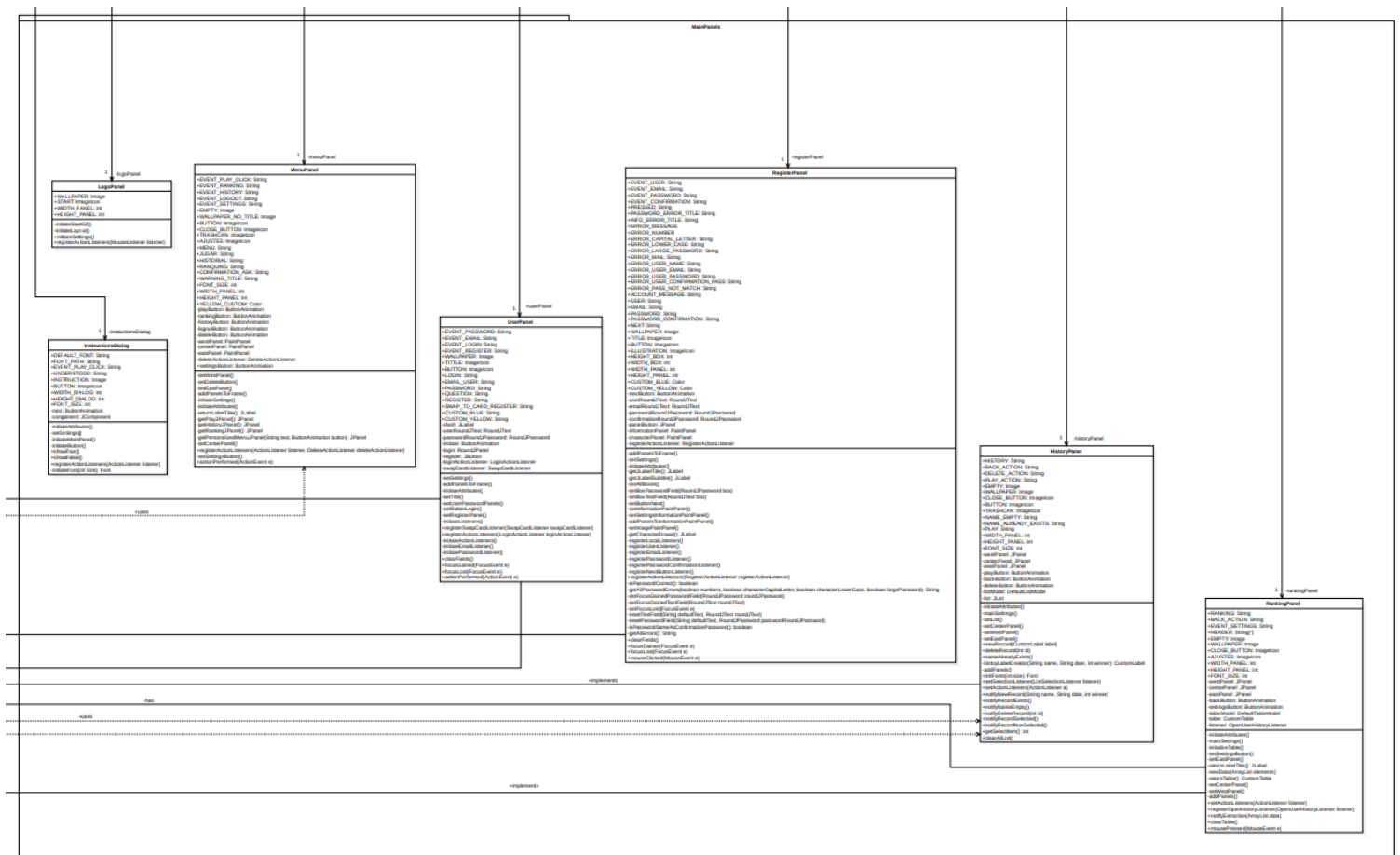
Controllers:

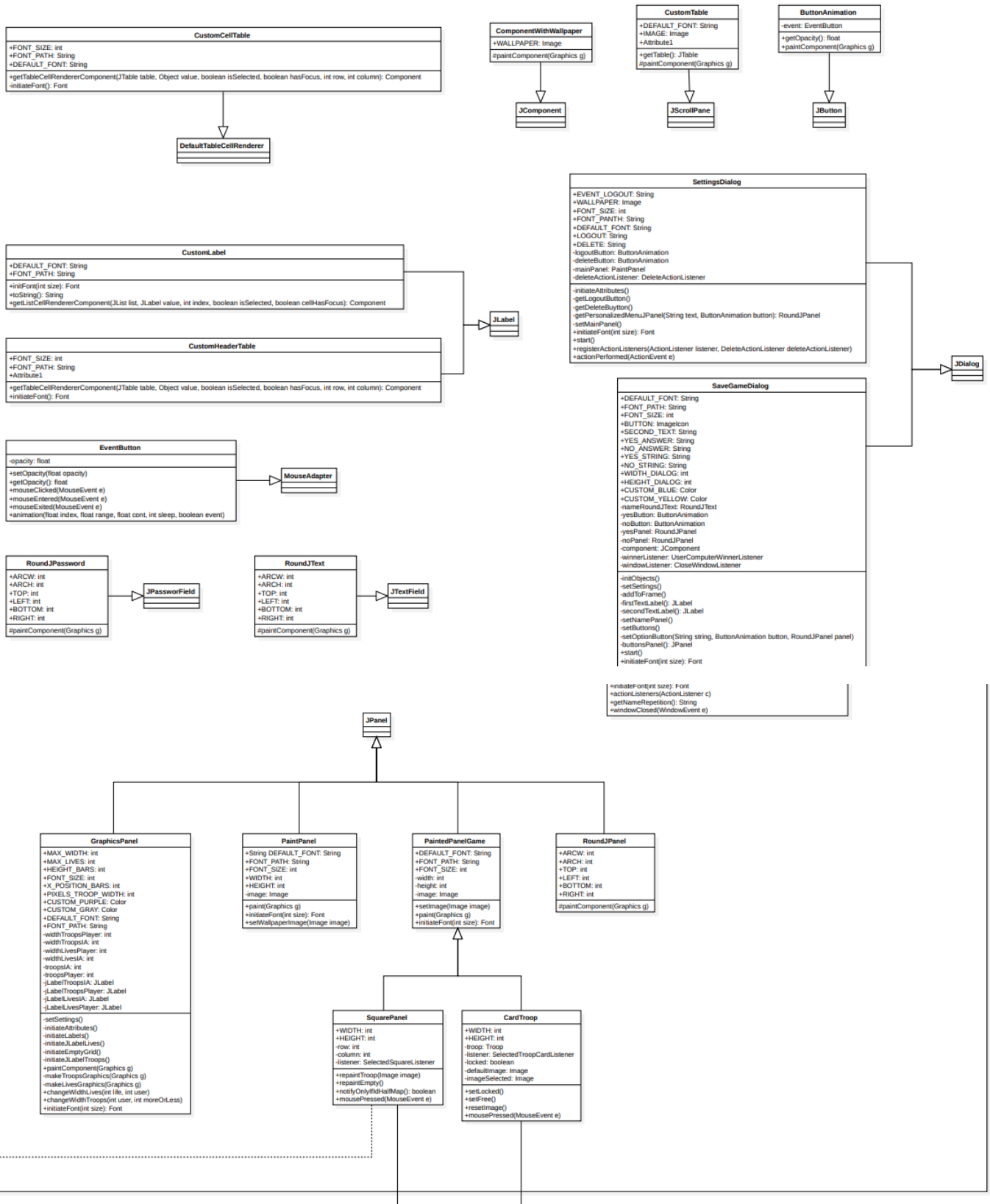


Main Views:



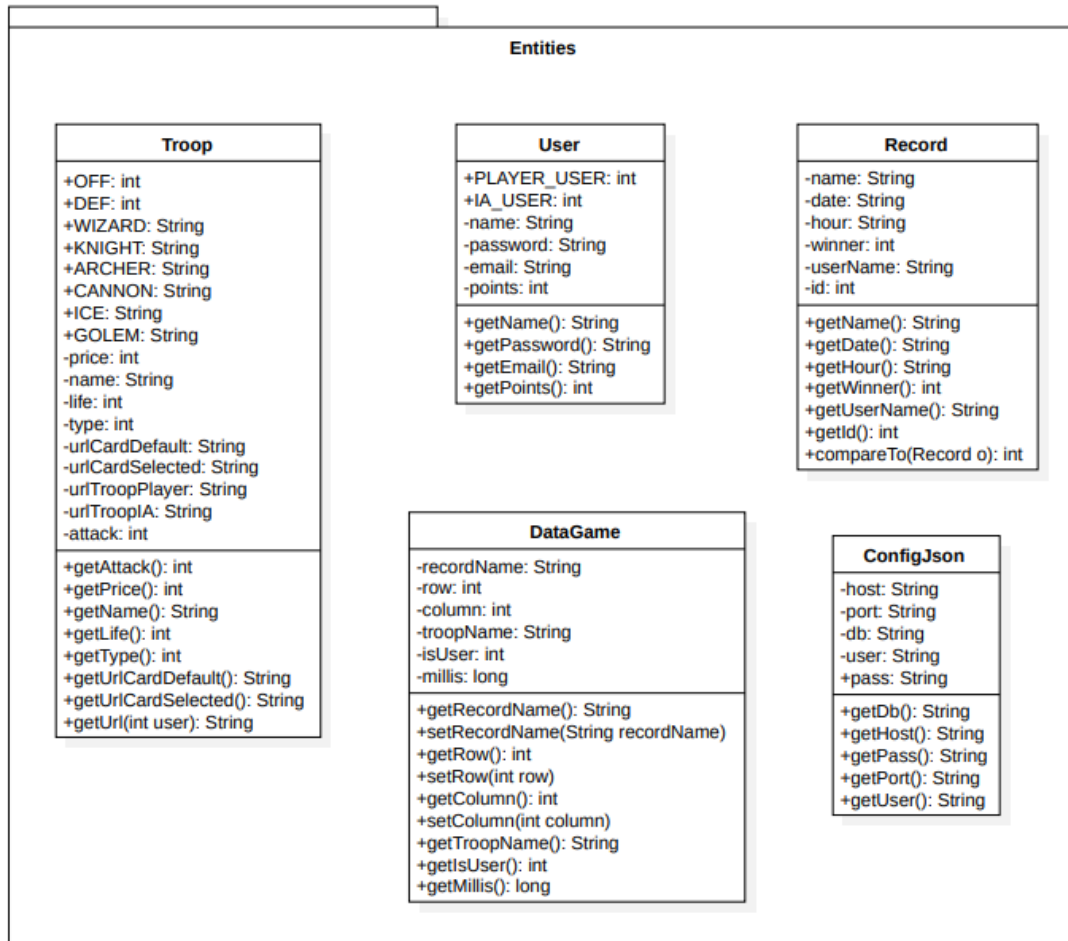
Main Panels:



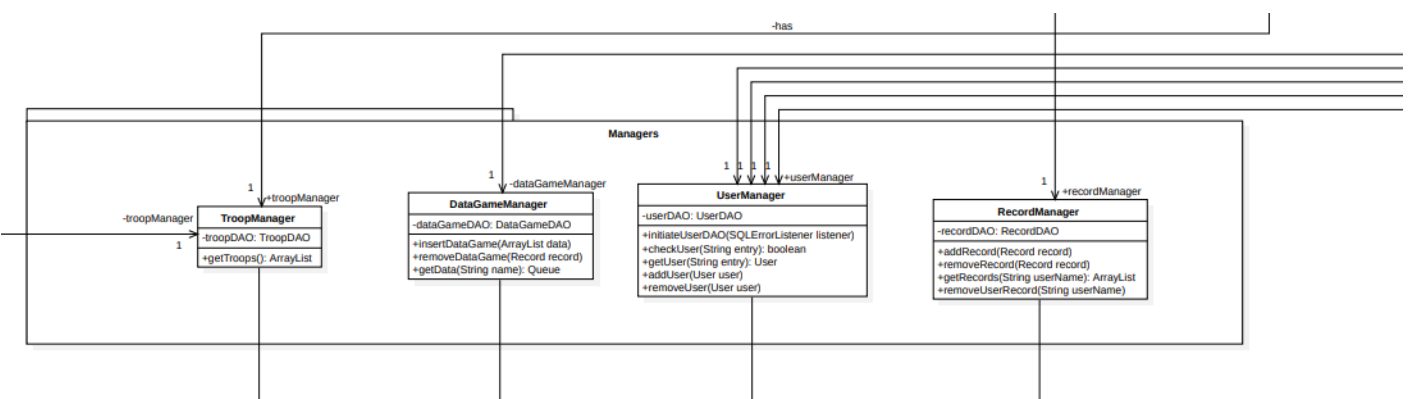
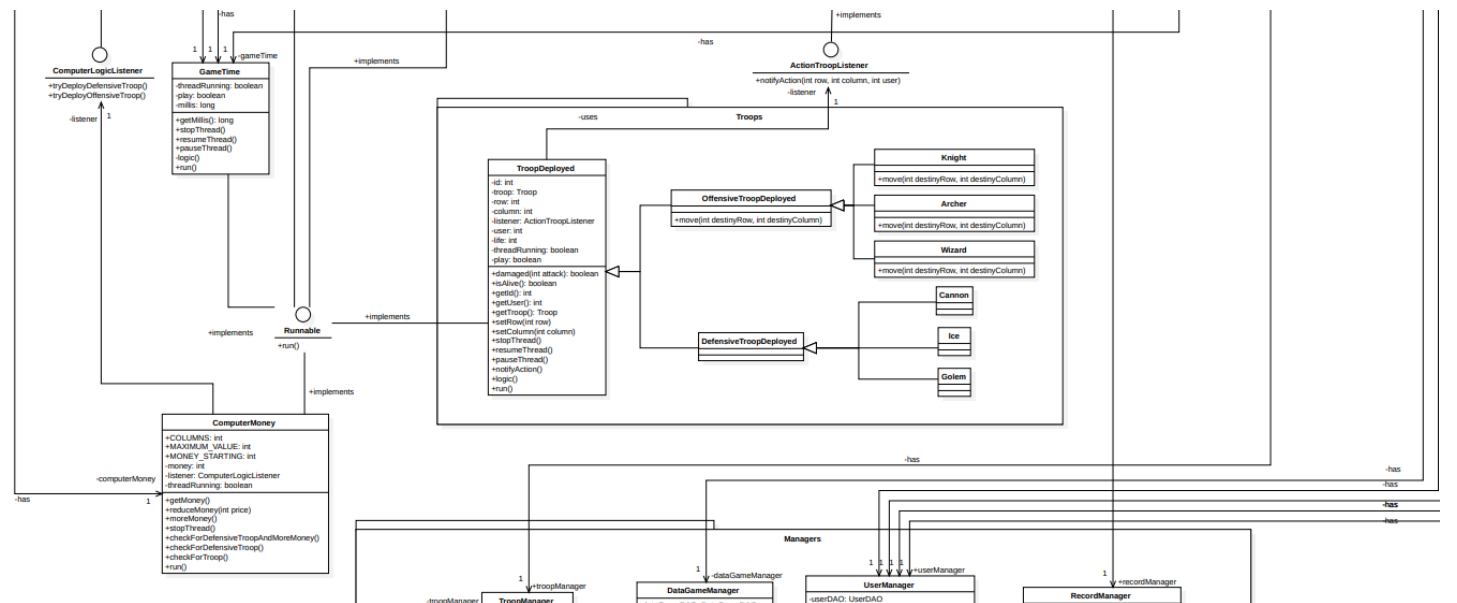
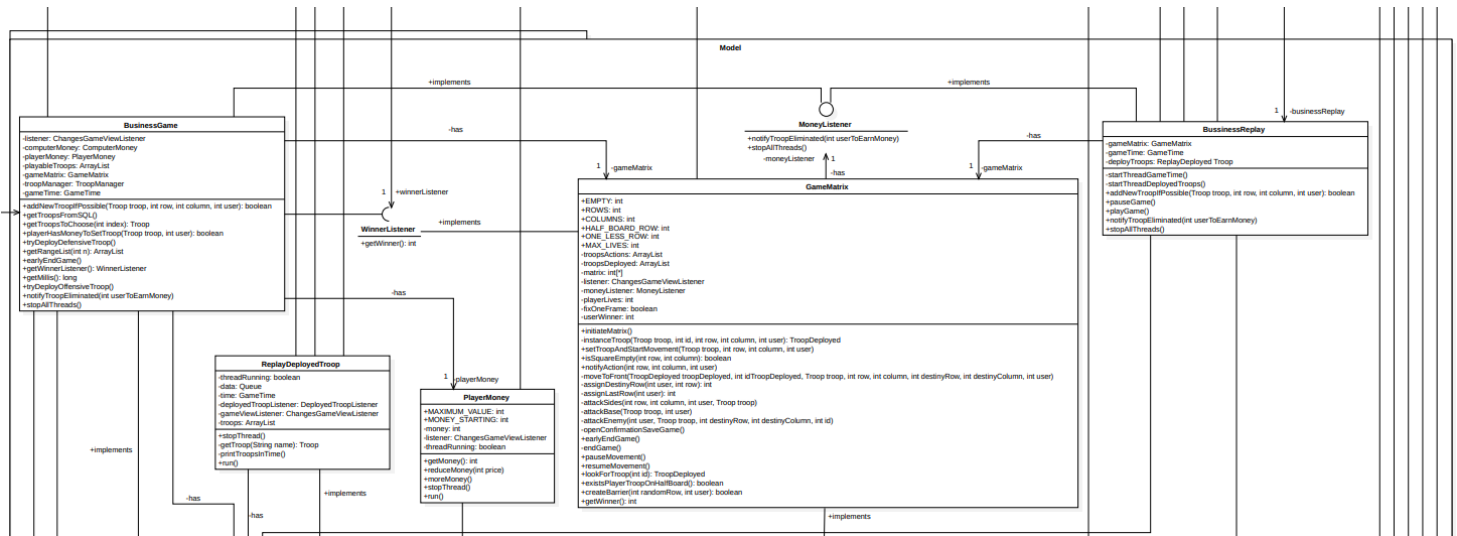
Helpers:**Helpers**

3.2 Capa BusinessLayer

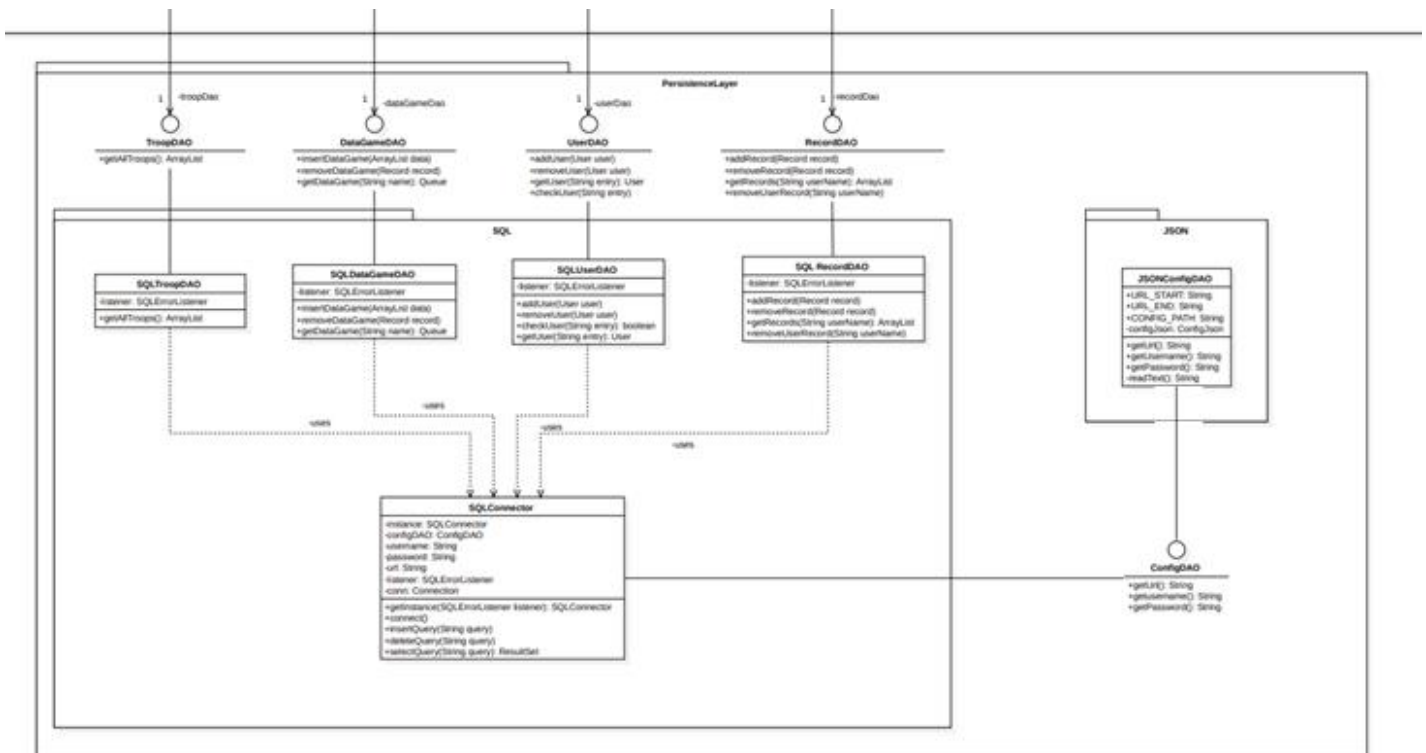
Entities:



Model:



3.3 Capa PersistenceLayer



4 Metodologia de desenvolupament

4.1 Age Royale Sprint 1

Per al primer sprint vam decidir prendre les següents històries:

- Comprensió de les especificacions del projecte
- Anàlisi i disseny del projecte
- Documentació inicial
- MockUps
- UML inicial
- Configuració de la base de dades

Les quals ens les vam dividir de la següent manera. La número 1, 2 i 3 les vam fer tots ja que és necessari que tots estiguem al dia de sobre de què tracta el projecte i que és el que s'ha de fer per complir amb tots els objectius, també entre tots donar idees i començar a encarrilar el projecte.

La història 4 es va realitzar entre la Inés i l'Oriol ja que són els que podien aportar idees més artístiques per a les pantalles. Per realitzar els dissenys van utilitzar Photoshop com a eina.

El UML es va realitzar entre la Marta i el Pol, que van començar amb la primera idea d'implementació del programa. Per realitzar el diagrama van utilitzar StarUML.

I l'última història la van realitzar l'Inés i el Giovanni que de tots els de el grup són els que més coneixement en base de dades tenen. Per realitzar el primer disseny de base de dades van utilitzar MySQL.

4.2 Age Royale Sprint 2

Les històries escollides per al segon esprint van ser les següents:

- Lògica dels canvis de pantalles
- Pantalles inicials (Logo, Inici, Registre i Menú)
- Lògica del joc UML inicial
- Lògica de l'inici de sessió i registre del usuari

Per a aquest sprint ens vam organitzar de la següent manera. La lògica per de canvis de pantalla es va encarregar la Inés que va escollir la forma en com fer el salt entre aquestes.

El disseny en swing de les pantalles inicials va ser encarregat a la Inés i Oriol ja que van ser els que van realitzar els MockUps.

La lògica de el joc la va realitzar el Pol ja que, era qui més temps va poder dedicar entre tots els de l'equip.

I la lògica de l'inici de sessió i registre van ser fets per el Giovanni i la Marta ja que, saben gestionar bases de dades i saben com manejar-la des de Java.

4.3 Age Royale Sprint 3

Per aquest tercer sprint les històries escollides van ser les següents:

- Finalització de la lògica del joc
- Historial
- Repetició d'un joc
- Revisar especificacions del projecte

La finalització de la lògica de joc va ser portada a terme pel Pol que ja tenia feta gran part de el joc.

El historial i la reproducció d'una partida va ser fet per Giovanni: en general tota la pantalla dels jocs guardats per l'usuari i la lògica de reproducció de cada joc.

I finalment, revisar les especificacions del projecte que ho vam fer entre tots per veure que del que portem acabat en aquell moment que s'estava complint del que es demana a l'enunciat.

4.4 Age Royale Sprint 4

Per a aquest últim sprint es van agafar les històries que quedaven per fer:

- Gràfics
- Rànquings
- Qualitat de codi
- Memòria
- UML final

Els gràfics van ser assignats a l'Oriol i el Pol, que després es van millorar a nivell de vista gràfica pel Pol i l'Inés.

El Rànquing es va realitzar per la Marta i la Inés: mostrar tots els usuaris registrats a la base de dades i tenir les seves puntuacions i ordenar-los per aquestes.

El UML final es va realitzar per la Marta, la Inés i el Pol, que van realitzar els canvis al UML inicial per finalment obtenir el UML final ja explicat..

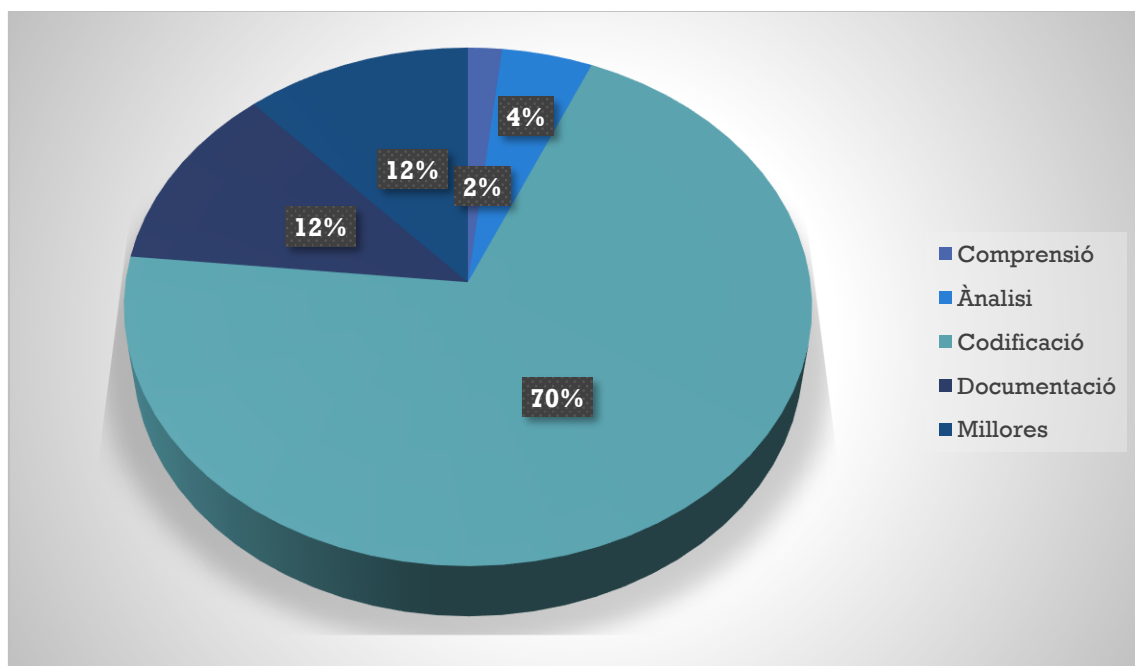
I finalment la qualitat de codi i la memòria va ser realitzat entre tots els membres del grup.

5 Dedicació

En aquesta secció explicarem i descriurem una estimació del temps emprat en total per la realització del projecte i de la memòria com a equip.

- Pel que fa al temps dedicat a la comprensió de les especificacions es podria estimar en unes 3 hores.
- Pel que fa al temps dedicat a l'anàlisi i disseny del projecte i del joc es podria estimar en unes 8 hores.
- Pel que fa al temps dedicat a la codificació del projecte (base de dades i Java) es podria estimar en unes 120 hores entre tots els membres del grup.
- Pel que fa al temps dedicat a la documentació i autoaprenentatge de la teoria necessària per realitzar completament el projecte es podria estimar en unes 20 hores entre tots els membres del grup.
- Finalment, s'han de destacar les hores extres aportades per l'equip per millorar visualment i gràficament totes les pantalles del joc. La majoria de dibuixos són originals realitzats per eines de treball com Photoshop i Illustrator. Aquest treball extra es podria estimar en unes 20 hores més.

Per tant, per completar el treball s'han emprat un total de 171 hores aproximadament entre tots els membres del grup.



6 Conclusions

D'aquest treball arribem a aprendre moltes coses molt importants tant en l'àmbit de treball com a escala personal. Pel que fa al treball i a l'assignatura, vam aprendre la importància de tenir bon estructura en l'ordre de les diferents classes i interfícies utilitzades, un programa ben definit definint per cada capa en el desenvolupament utilitzant la Layered Architecture, i la importància d'aprendre a utilitzar eines que faciliten la distribució de treball i el seu flux com Jira i BitBucket.

Gràcies a Jira, es va mantenir una gran organització de tasques per realitzar al llarg de la feina tenint així, ben definits els objectius a complir, i facilitant la distribució de treball per a cada persona del grup. També gràcies a aquesta eina es podia revisar i ajudar entre diferents membres del grup i a la vegada al grup sencer durant el procés de cada tasca sabent que tan avançat o endarrerit es portava el treball i així marcar en cada ocasió el ritme de treball oportú per cada moment i cada setmana.

El fet que utilitzem BitBucket per al desenvolupament d'aquest programa, vam entendre la utilitat i importància de treballar per branques amb les eines de Git. Això és degut al fet que en assignar una tasca diferent per a cada integrant tot era tan fàcil com obrir una nova branca per a aquesta tasca i així, anar desenvolupant-la de manera que no s'anava trepitjant el treball de l'altra persona. En el possible cas de fer mal, o generar errors a la feina que ja estava feta, no ens ha suposat un gran problema, ja que el treball important es trobava protegit en la branca des d'on es va crear aquesta nova, donant la facilitat de fer la feina nova sobre una còpia del que ja funciona i no danyar aquest treball ja fet.

En l'àmbit personal vam entendre la importància de tenir una bona comunicació en el que és treballar en un equip, ja que treballar en equip no és només repartir tasques i tornar-se a veure només quan s'hagi d'unir tot, és molt més que això. És mantenir-se informat de com va el procés de treball, els inconvenients trobats pel camí, ajudar-se entre tots, donar solucions i generar un bon ambient de treball mostrant interès pel projecte i l'assignatura i així, entre els companys de grup fent sentir la feina cada vegada menys com una obligació.

Gràcies a aquest treball hem après a tractar amb diferents personalitats i a saber portar-lo en el camí de la millor manera per aconseguir el millor resultat en l'objectiu final i en el marge de temps marcat.

7 Bibliografía

1. Clash Royale/Removed and Changed Sprites - The Cutting Room Floor. The Cutting Room Floor [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
https://tcrf.net/Clash_Royale/Removed_and_Changed_Sprites
2. Javadoc displaying value on an inner class constant using @value. Stack Overflow [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en: <https://stackoverflow.com/questions/11426959/javadoc-displaying-value-on-an-inner-class-constant-using-value>
3. Writing Javadoc examples | OpenWritings.net. Home | OpenWritings.net [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en: <https://openwritings.net/pg/java/writing-javadoc-examples>
4. Terminating a Java Program. Stack Overflow [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://stackoverflow.com/questions/22452930/terminating-a-java-program>
<https://www.codota.com/code/java/methods/javafx.swing.JDialog/setDefaultCloseOperation>
5. How to Delay Code Execution in Java | Baeldung. Baeldung [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://www.baeldung.com/java-delay-code-execution>
6. How to Use GridLayout (The Java™ Tutorials > Creating a GUI With JFC/Swing > Laying Out Components Within a Container). Moved [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://docs.oracle.com/javase/tutorial/uiswing/layout/grid.html>
7. How to Use BoxLayout (The Java™ Tutorials > Creating a GUI With JFC/Swing > Laying Out Components Within a Container). Moved [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://docs.oracle.com/javase/tutorial/uiswing/layout/box.html>
8. How to use JLabel as a String. Stack Overflow [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://stackoverflow.com/questions/25041861/how-to-use-jlabel-as-a-string/25044926>
9. Java cannot access class, class file not found. Stack Overflow [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://stackoverflow.com/questions/38549649/java-cannot-access-class-class-file-not-found>
10. Codota. AI Code Autocomplete for All Languages and IDEs | Codota [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://www.codota.com/code/java/methods/java.awt.Graphics/fillOval>
11. How to make a rectangle in Graphics in a transparent colour? Stack Overflow [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://stackoverflow.com/questions/8110975/how-to-make-a-rectangle-in-graphics-in-a-transparent-colour>
12. Sorting in Java | Baeldung. Baeldung [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en: <https://www.baeldung.com/java-sorting>
13. JDialog (Java Platform SE 7). Moved [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://docs.oracle.com/javase/7/docs/api/javafx.swing.JDialog.html>

14. How to Use Modality in Dialogs (The Java™ Tutorials > Creating a GUI With JFC/Swing > Using Other Swing Features). Moved [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://docs.oracle.com/javase/tutorial/uiswing/misc/modality.html>
15. How to draw an outline around text in AWT? Stack Overflow [en línea]. [sin fecha] [consultado el 17 de mayo de 2021]. Disponible en:
<https://stackoverflow.com/questions/10016001/how-to-draw-an-outline-around-text-in-awt>