

Disseny i programació orientats a objectes

Practica del segon semestre



Grup C4

Biel Carpi

Marc Cano

Rafael Morera

David Bassols

Pol Saula

Índex

1.	<i>Especificacions del projecte</i>	<i>3</i>
2.	<i>Disseny de la interfície del projecte</i>	<i>4</i>
3.	<i>Diagrama de classes</i>	<i>17</i>
4.	<i>Metodologia de desenvolupament.....</i>	<i>18</i>
5.	<i>Cost Temporal</i>	<i>20</i>
6.	<i>Conclusions</i>	<i>21</i>
7.	<i>Bibliografia</i>	<i>22</i>

1. Especificacions del projecte

Per al primer semestre de l'assignatura de Disseny i Programació Orientat a Objectes, se'ns va demanar, en grups de dues persones, implementar un simulador d'una competició de "Batalla de Gallos". Aquesta va ser vital per tal d'obtenir una sòlida base per al projecte que hem hagut de dissenyar per a aquest segon semestre.

Aquest cop, l'equip de becaris de l'assignatura ens ha demanat implementar, seguint un criteri i unes eines concretes, una còpia d'un joc d'estratègia molt conegut i famós arreu del món, *Clash Royale*.

Aquest, es basa en un combat entre dos jugadors, cadascun dels quals posseeix una base o castell i un numero indeterminat de tropes. L'objectiu principal és fàcil: intentar destruir la base enemiga abans de que la teva sigui destruïda.

L'objectiu del projecte, anomenat *War Royale*, es basa en implementar una còpia de *Clash Royale*. La única diferència és que, al jugar, en comptes de fer un Player vs. Player, se'ns demana un Player vs. Bot. La implementació ha de ser amb Java, utilitzant Swing.

Per tant, haurem de fer una primera part interactuant amb una base de dades (en concret, MySQL) per tal de proporcionar un sistema de LogIn i poder-te també registrar.

Seguidament, hem d'implementar una "pantalla principal" on l'usuari podrà prendre diferents accions, tals com començar una nova partida, fer un LogOut, esborrar el seu compte o bé mirar el ranking de jugadors i l'historial de partides de cada jugador.

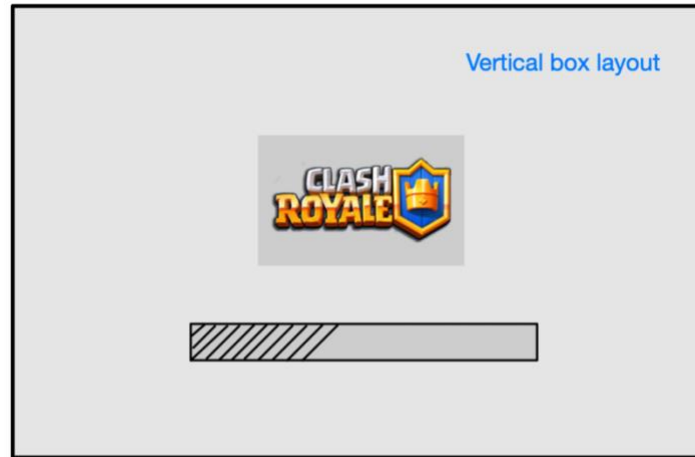
Un cop implementada totes les vistes del joc, cal implementar el joc en sí. Un mapa on puguis tirar cartes/tropes (tal i com en el joc real) i les teves cartes batallin amb les que tira el bot. També cal mostrar algunes estadístiques en temps real, tals com el numero de tropes en el camp com la vida de cada tropa, i poder guardar aquesta partida (donant-li un nom quan s'acabi la batalla) per veure-la posteriorment.

Per a la implementació d'aquest projecte, s'ha decidit dissenyar el codi seguint amb una arquitectura per capes juntament amb el patró de disseny *Model-View-Controller* (MVC). S'ha decidit seguir aquest patró per tenir definida una estructura sòlida i organitzada del codi, d'aquesta manera, aconseguim mantenir les vistes desacoblades del model, de forma que aquest pot treballar independentment de com s'han decidit implementar les vistes.

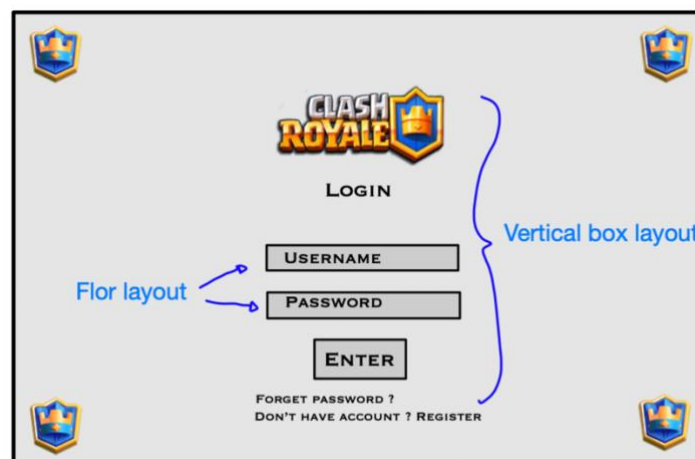


2. Disseny de la interfície del projecte

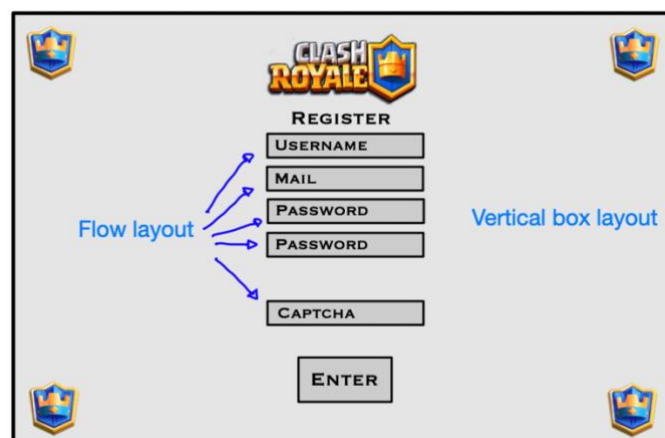
Abans de començar a programar, vam dissenyar els *mockups* de tot el que seria el joc. Inicialment vam tenir les següents idees de vistes, cosa que hem respectat durant tot el projecte i el resultat final ha sigut l'esperat.



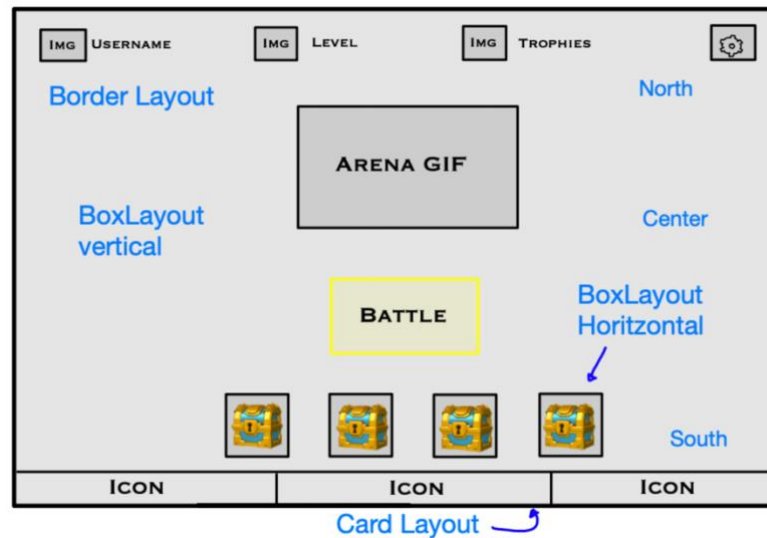
Pantalla d'inici (Splash Screen)



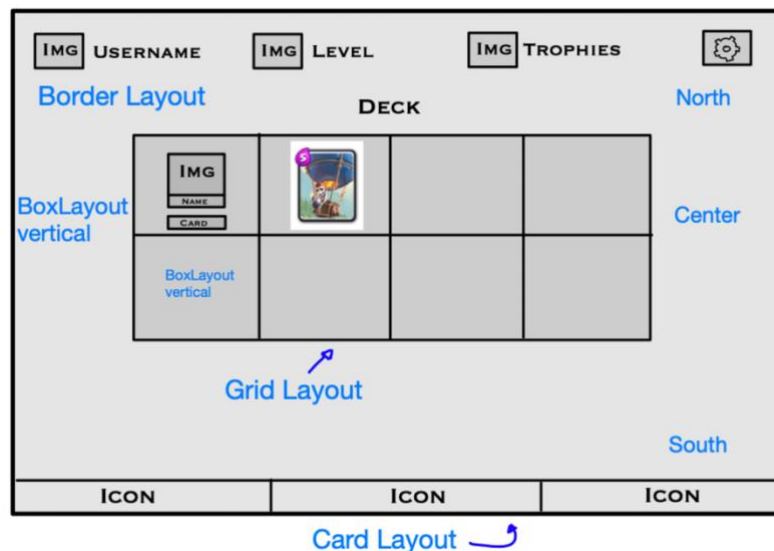
Pantalla de Login



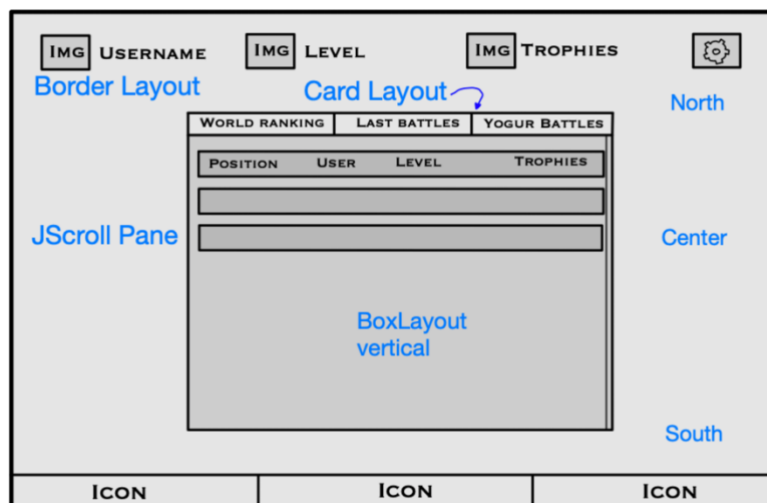
Pantalla de Register



Menú. Pantalla de la Batalla (centre)



Menú. Pantalla de les Cartes (esquerra)



Menú. Pantalla dels Rankings (dreta)

Tot i que algunes vistes han sigut més complicades de construir del que esperàvem, el resultat final, tal i com hem dit, ha sigut l'esperat.

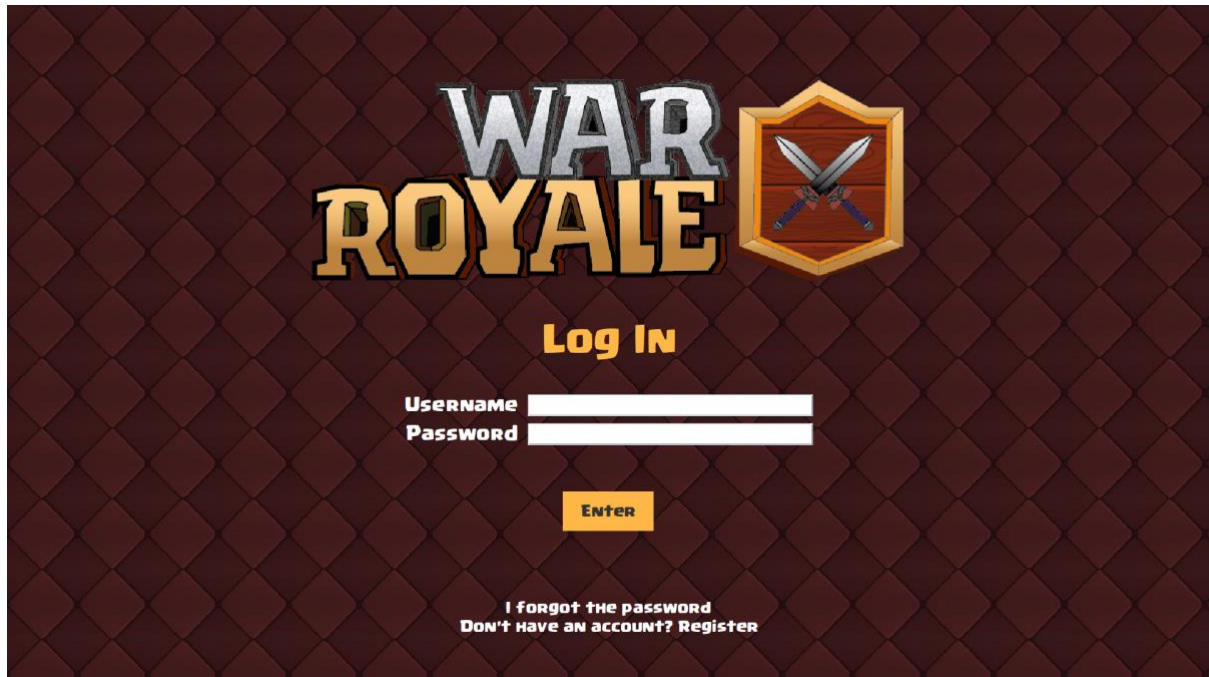
Cal dir que totes les vistes s'afegeixen al centre d'un panel que hi ha posat en el `contentPane` del `JFrame`. De tal manera, aquest `contentPane` ja defineix el “look” de la nostra aplicació. Com es pot veure en cada vista del joc, de *background* sempre hi ha aquest conjunt de *tiles* de color vermell (aquestes es pinten manualment una a una, en la classe `MenuBackgroundPanel`).

Splash Screen:



La Splash Screen (pantalla de càrrega del joc) és, sense dubte, la vista més senzilla. Consisteix simplement del logo del joc (un `RoyaleLabel`) i una `RoyaleProgressBar` (els nostres tipus de `JProgressBar` i de `JLabel`) que mostra el progrés de càrrega del joc (en segon pla s'estan carregant els recursos del joc, les musiques i sons, comprovant la connexió amb la base de dades, etc.).

Aquests dos components estan dintre un `JPanel` amb un `BoxLayout Vertical`, que ahora està dintre un altre `JPanel` amb un `GridBagLayout` sense constraints (per tal de centrar-ho tot tant verticalment com horitzontalment).

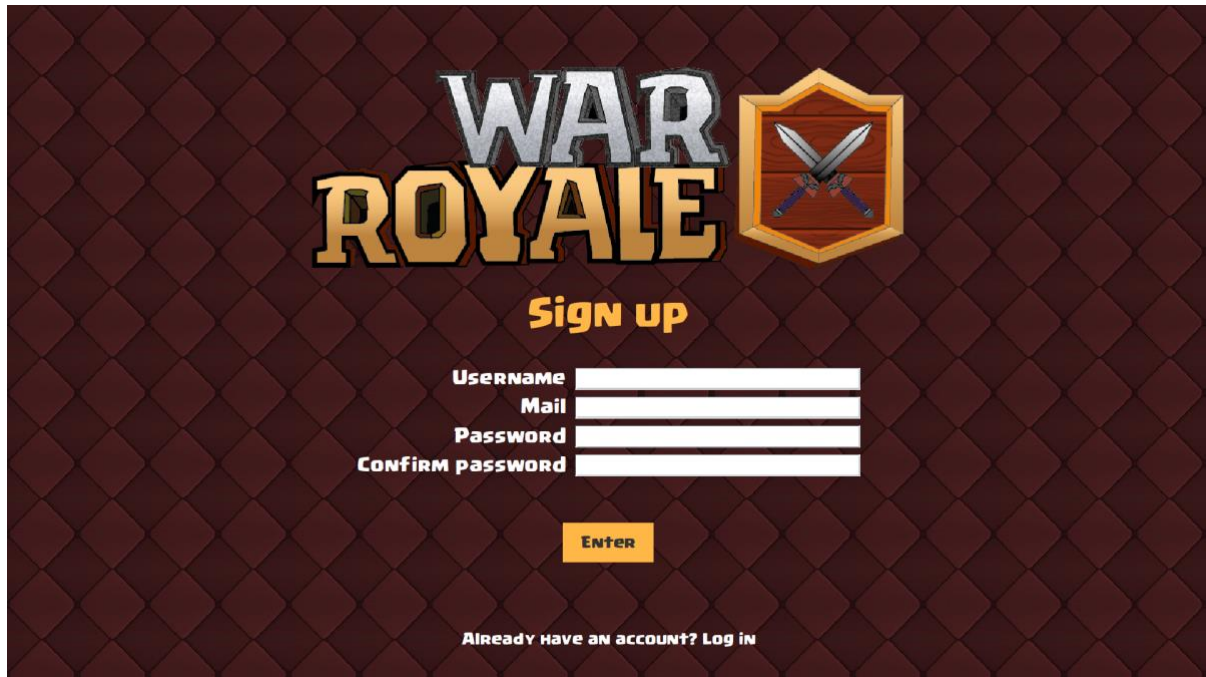
Log In Screen:

Aquesta pantalla segueix el disseny de la anterior, amb el JPanel amb BorderLayout Vertical contingut en un altre JPanel amb GridBagLayout sense constraints per centrar-ho tot.

També hi trobem un RoyaleLabel amb el text de Log In, un RoyaleButton (el nostre JButton) per tal de comprovar si el LogIn és correcte (i passar a la següent pantalla). En cas de que el botó es premi i hi hagi algun error amb les dades introduïdes, es mostrarà el missatge d'error corresponent.

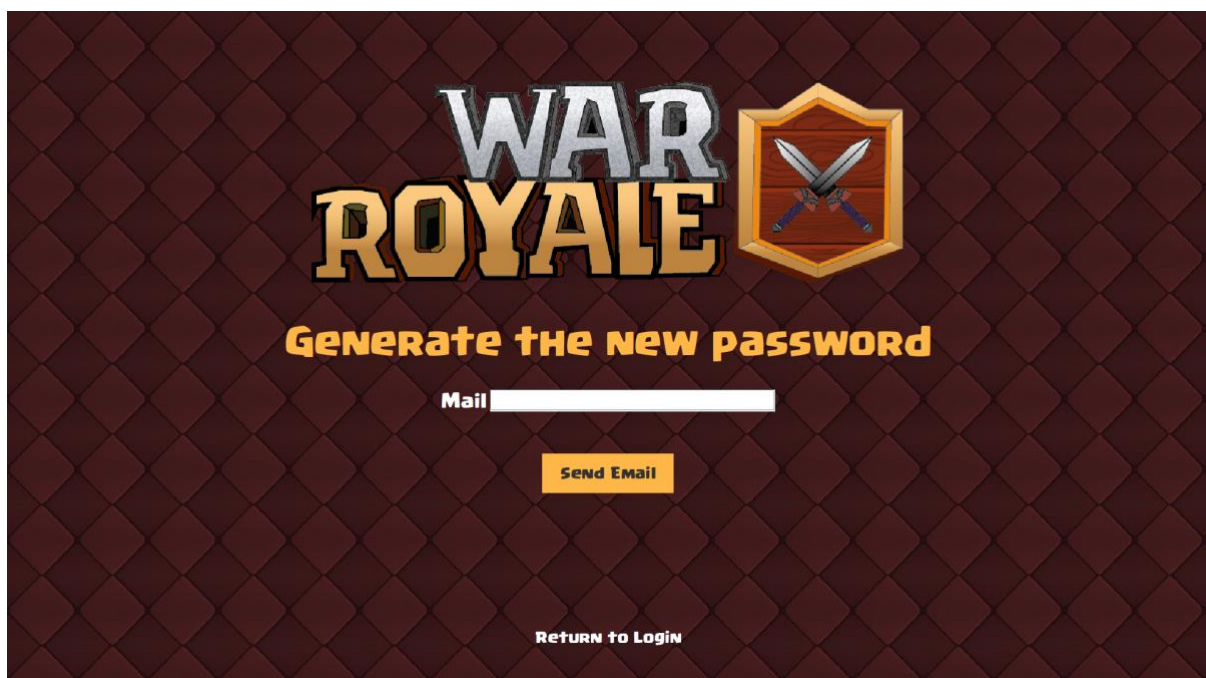
Just al mig, hi tenim quatre components (dos RoyaleLabel, un RoyaleTextField i un RoyalePasswordField). Aquests quatre elements els gestiona un JPanel mitjançant un GroupLayout, per tal que els TextFields sempre tinguin la mateixa mida.

Finalment, a sota (encara amb el BorderLayout Vertical) hi tenim dos RoyaleLabel. Aquests Labels tenen la habilitat de poder-se clicar, i ens porten tan a la vista de recuperació de contrasenya com a la vista per registrar-se.

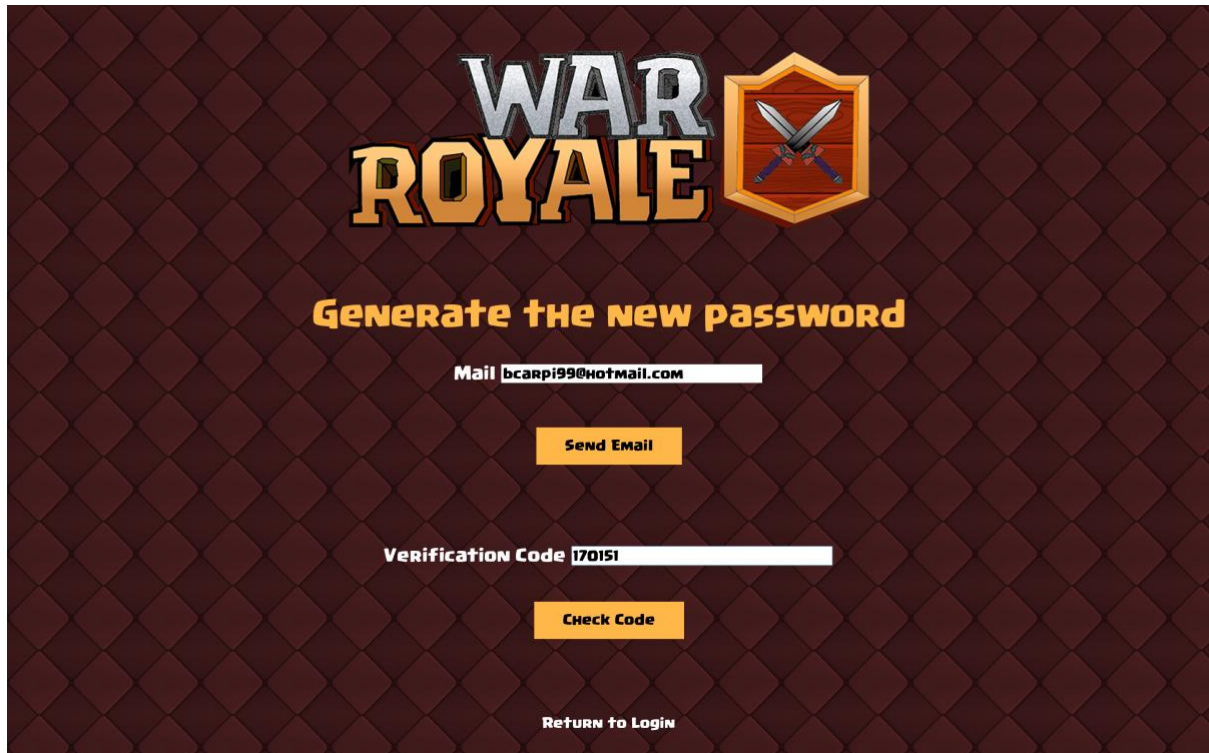
Register Screen:

The register screen for War Royale features a dark red background with a diamond pattern. At the top center is the game logo, with "WAR" in silver and "ROYALE" in gold, next to a shield icon containing two crossed swords. Below the logo, the text "Sign up" is displayed in orange. The registration form consists of four white text input fields, each preceded by a label: "Username", "Mail", "Password", and "CONFIRM password". Below these fields is an orange button labeled "Enter". At the bottom center, there is a link that reads "Already Have an account? Log in".

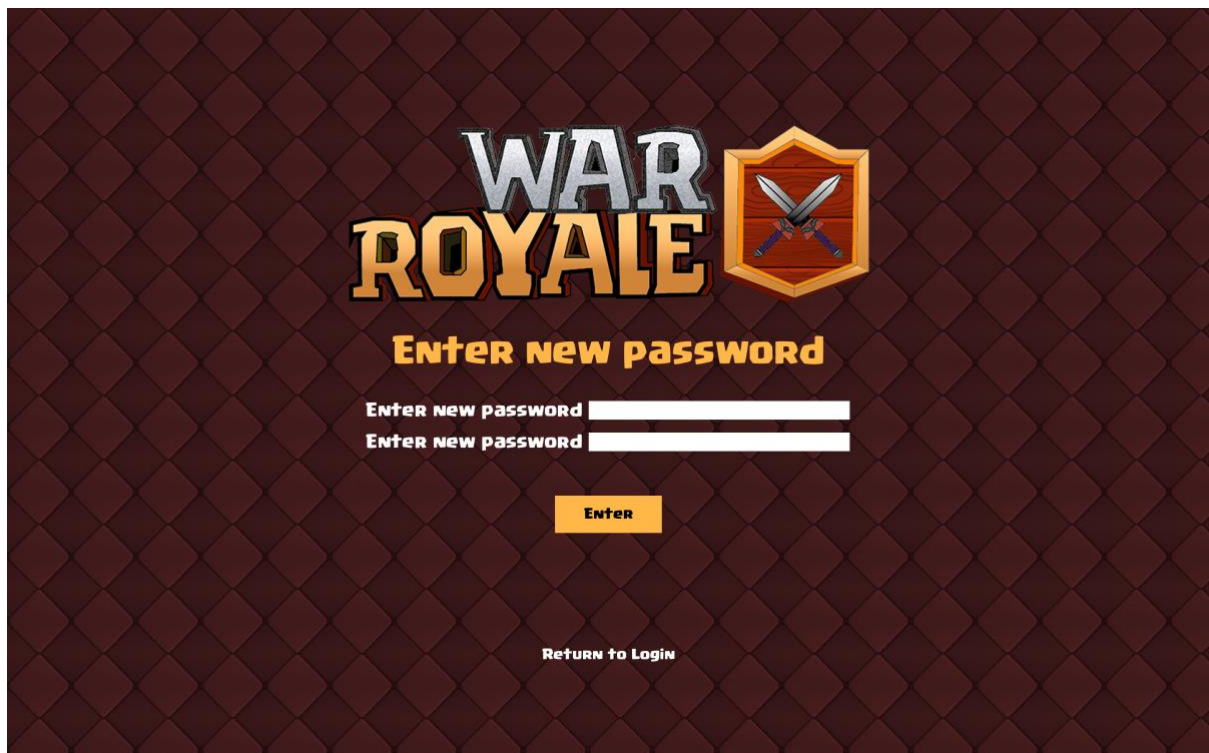
Exactament el mateix que la vista anterior de LogIn. Tots els TextFields i els seus respectius Labels estan agrupats utilitzant un GroupLayout per tal d'aconseguir, com hem dit abans, que tinguin la mateixa amplada.

PasswordForgottenScreen:

The password forgotten screen for War Royale has the same dark red diamond-patterned background and logo as the register screen. Below the logo, the text "Generate the new password" is shown in orange. The form includes a single white text input field labeled "Mail". Below this field is an orange button labeled "Send Email". At the bottom center, there is a link that reads "Return to Login".



The image shows a web form for resetting a password in a game called 'WAR ROYALE'. The background is dark red with a diamond pattern. At the top, the game logo 'WAR ROYALE' is displayed in a stylized font, with 'WAR' in white and 'ROYALE' in orange. To the right of the logo is a shield icon with two crossed swords. Below the logo, the text 'Generate the new password' is written in orange. There is a text input field for an email address, which contains 'bcarpi99@hotmail.com'. Below this field is an orange button labeled 'Send Email'. Further down, there is a text input field for a 'Verification Code', which contains '170151'. Below this field is an orange button labeled 'Check Code'. At the bottom of the form, there is a link that says 'Return to Login'.

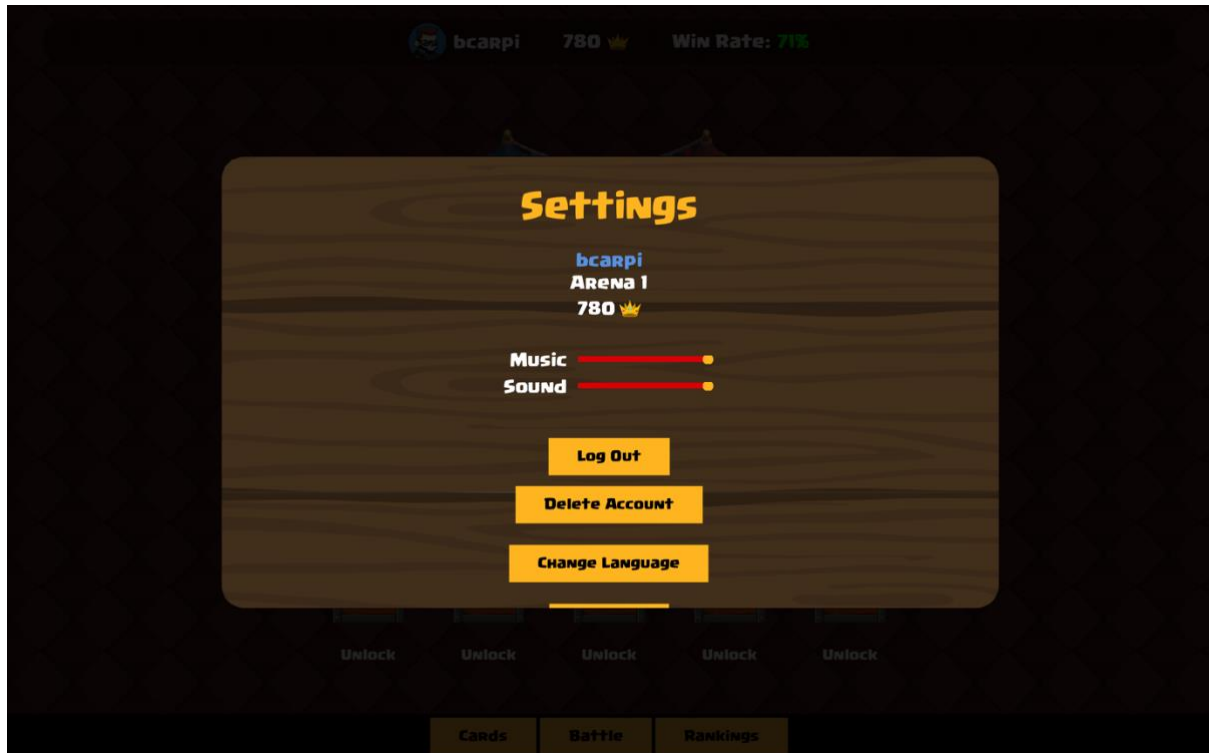


The image shows a web form for entering a new password in the same 'WAR ROYALE' game. The background is dark red with a diamond pattern. At the top, the game logo 'WAR ROYALE' is displayed in a stylized font, with 'WAR' in white and 'ROYALE' in orange. To the right of the logo is a shield icon with two crossed swords. Below the logo, the text 'Enter new password' is written in orange. There are two text input fields for the password, both containing the same text. Below these fields is an orange button labeled 'Enter'. At the bottom of the form, there is a link that says 'Return to Login'.

Els layouts i components utilitzats són exactament els mateixos que en les vistes anteriors.

Aquí podem veure que se'ns demana l'e-mail per recuperar la contrasenya. Un cop introduït correctament, l'aplicació ens envia un correu amb un codi. Si introduïm bé el codi, ja ens deixa introduir una nova contrasenya (la hem de confirmar).

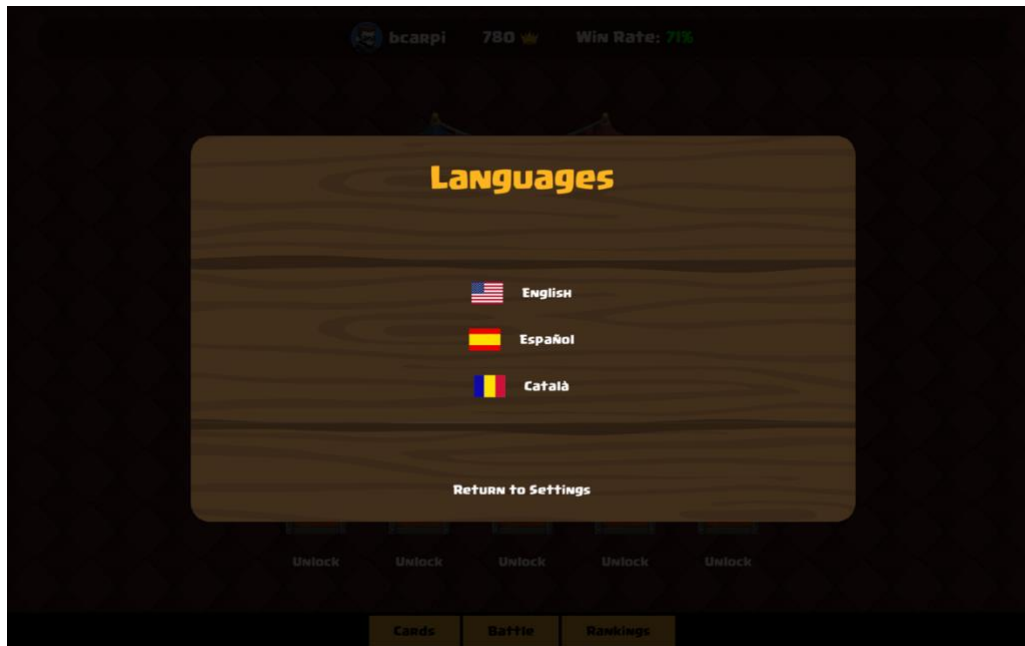
Per tal d'aconseguir aquestes transicions, simplement traiem i afegim JPanels.

Settings:

Aquesta vista és una de les més complexes del joc. Es pot obrir pressionant el botó ESC des de qualsevol punt del joc, i el seu contingut canviarà depenent de la pantalla on estiguem en aquell moment.

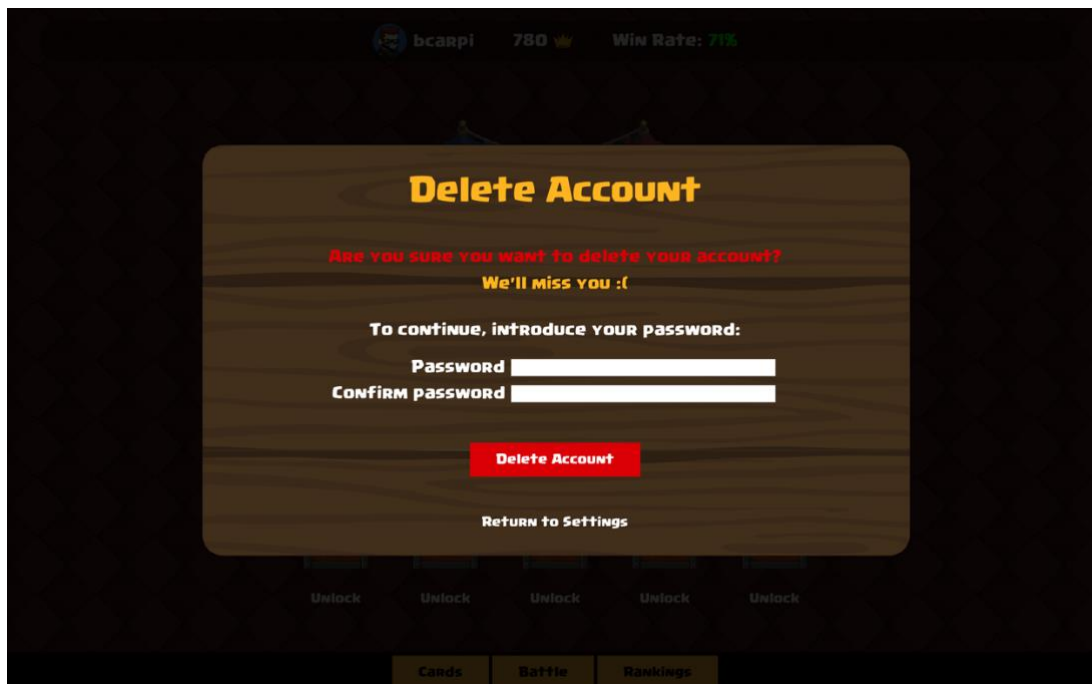
Com es pot veure, és un JPanel que es posa a sobre de tot (cosa difícil amb Swing, però es pot aconseguir manipulant el GlassPanel del JFrame). Aquest dibuixa una fusta que té la mida del JScrollPane que hi ha posat al mig de tot (li modifiquem la preferredSize perquè ocupi un 60% del height de la pantalla). El JScrollPane està centrat tant vertical com horitzontalment ja que està contingut en un altre JPanel amb un GridBagLayout sense constraints.

El que hi ha dintre el JScrollPane és un JPanel amb un BoxLayout Vertical, on hi podem observar diferents RoyaleLabels i RoyaleButtons. També, per tal de controlar el volum de la música, tenim disponible dos RoyaleSlider (els nostres JSlider).

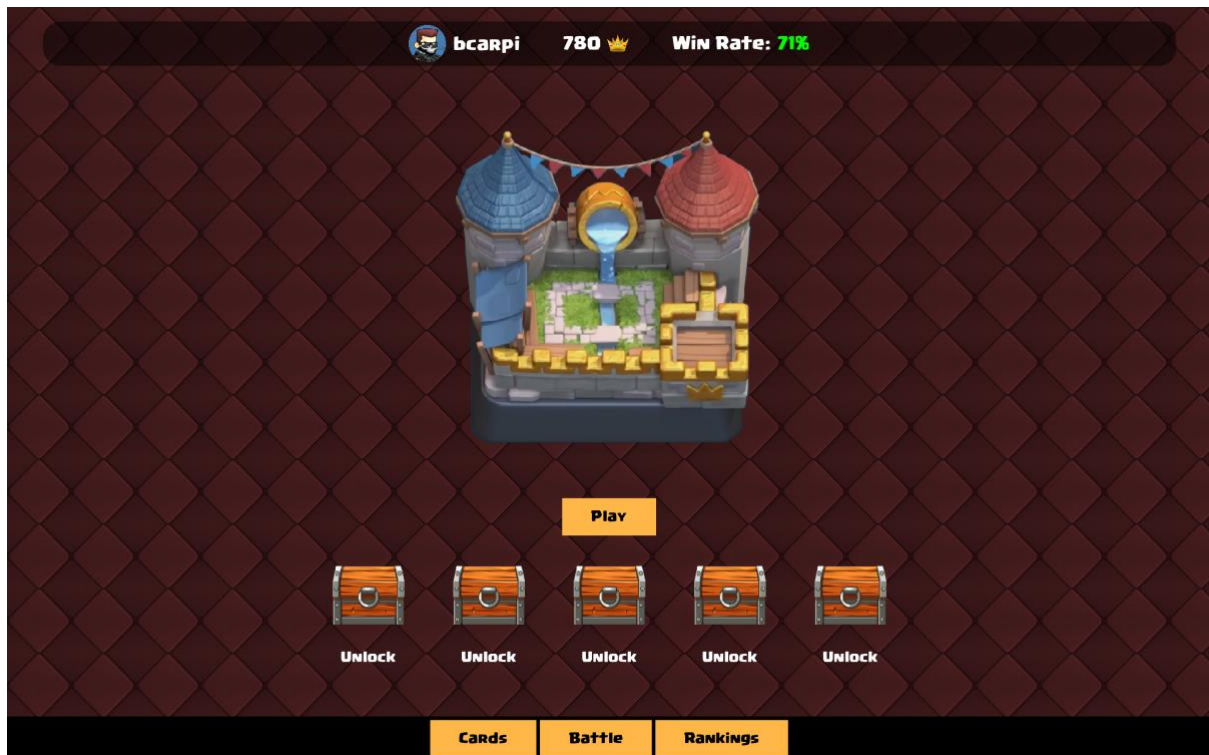
Change Language:

Dintre Settings, podem clicar el botó de Change Language (tenim tres idiomes disponibles per al nostre joc, i podem canviar entre ells). Aquesta acció canviarà el viewport del JScrollPane, i hi posarà aquest nou JPanel.

Seguim amb un BoxLayout Vertical, i BoxLayout Horitzontal per tal de que les banderes (que són Labels) estiguin al costat dels RoyaleLabel.

Delete Account:

També a dintre de Settings, és un altre simple JPanel (com els explicats anteriorment) que ens permet esborrar el nostre compte.

MainMenu Battle:

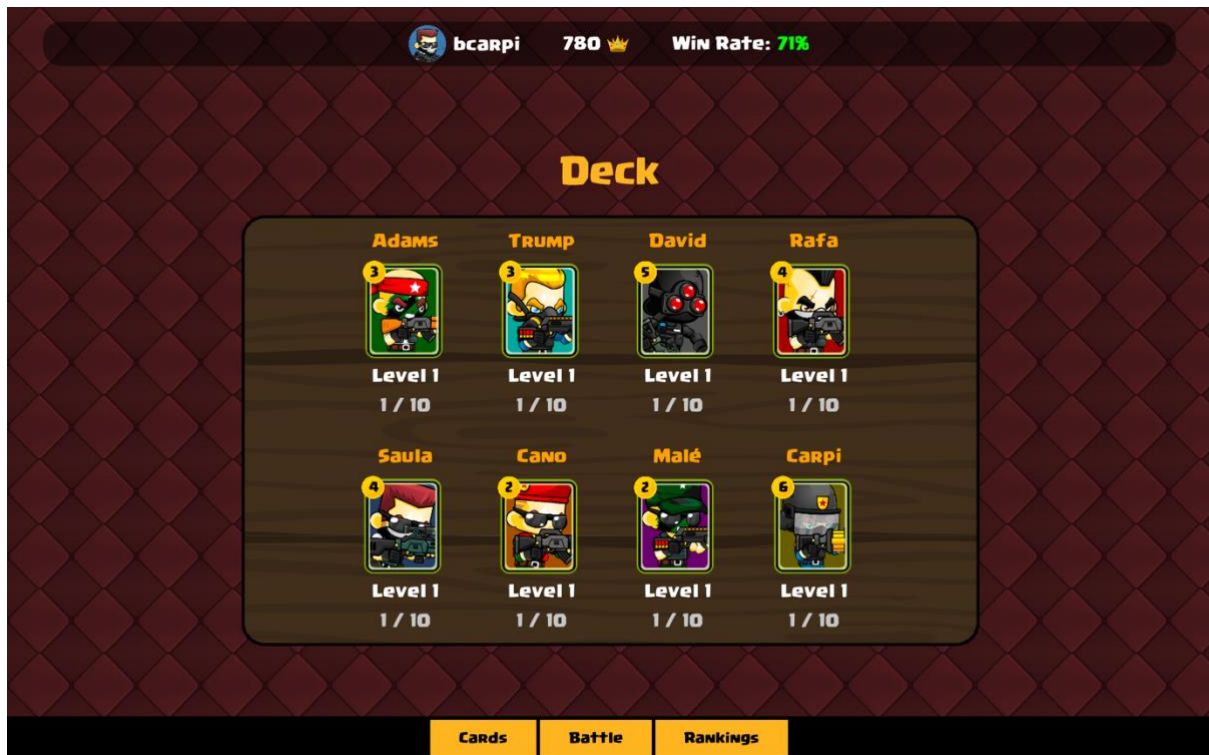
Aquí ja entrem en el MainMenuScreen. És una vista que està dividida en tres parts (battle, cards i rankings). Per tal de canviar entre elles, simplement tenim un JPanel principal amb un BorderLayout i anem canviant el JPanel del centre.

En el nord (comú a les 3 vistes), hi podem veure un petit menú transparent que t'informa sobre el teu usuari. És un JPanel amb un BoxLayout Horitzontal, on els dos primers elements són JPanels amb FlowLayout per ajuntar dos RoyaleLabels (imatge i text).

Per fer el background, simplement aquest JPanel del nord és un JPanel sense opacitat i hi pintem un *rounded rectangle* amb les eines que ofereix Swing.

En el sud (comú a les 3 vistes), hi veiem també un simple JPanel amb un BoxLayout Horitzontal que conté els 3 botons per canviar de vistes (canviar el JPanel del centre, com hem dit).

En el centre, ens trobem amb un BoxLayout Vertical amb una Imatge (RoyaleLabel), un RoyaleButton per començar la partida i un altre JPanel amb BoxLayout Horitzontal per posar-hi tots els cofres.

Cards (Deck):

Un cop es clica el botó de “cards”, es canvia el centre del JPanel principal i apareix aquest nou JPanel al mig.

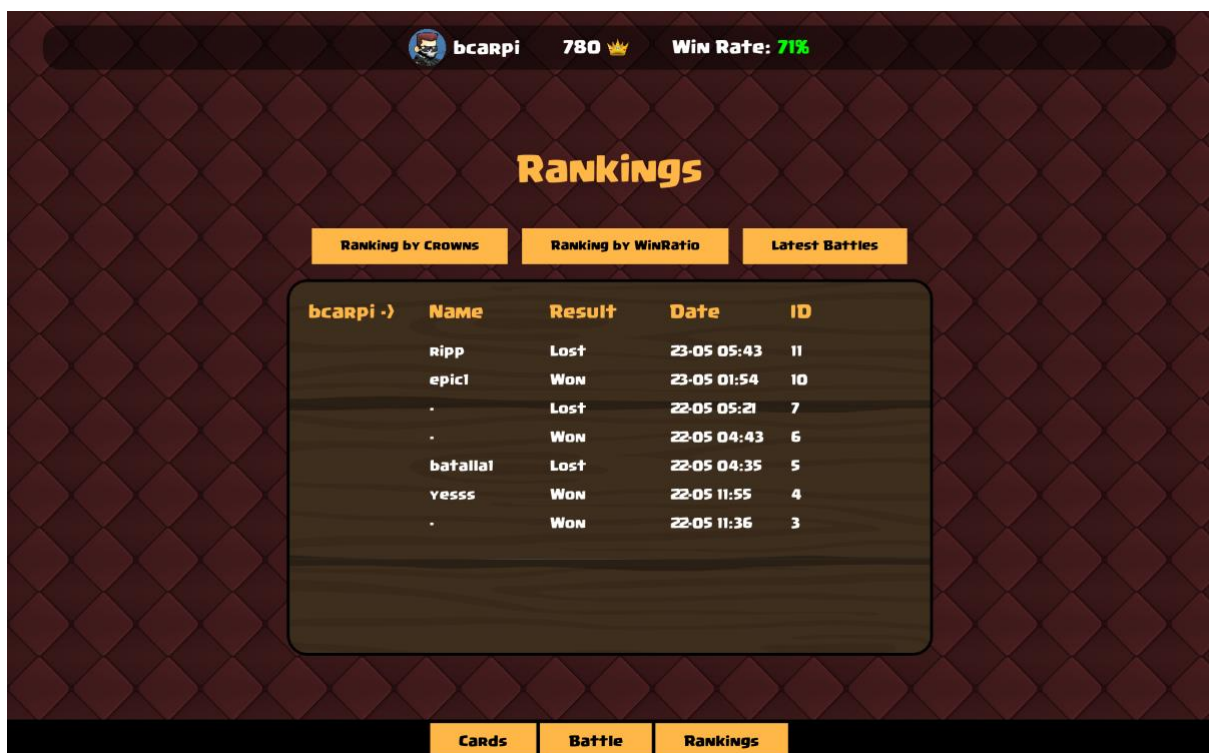
Conté un JPanel principal amb un GridBagLayout sense constraints (per tal de centrar-ho tot), i un altre JPanel amb un BoxLayout Vertical. A dintre aquest últim, ens hi trobem un RoyaleLabel (amb el text de Deck) i un JPanel amb un GridLayout de 2x4.

A cada posició del GridLayout, si ens hi fixem, tenim de nou un JPanel amb un BoxLayout Vertical.

La fusta de darrere està pintada depenent de la posició del JPanel que conté el GridLayout, amb les eines natives de Swing.

Rankings:

Player	Wins	Plays	Crowns
bcarpi	48	67	780
iTrust	4	7	61
kroketo	1	1	59
Marta	2	5	34
ALex	2	5	16
Maria	2	4	12
bielcarpi	0	0	0
M_cano19	0	0	0
Raial	0	0	0
Tsur	0	1	0



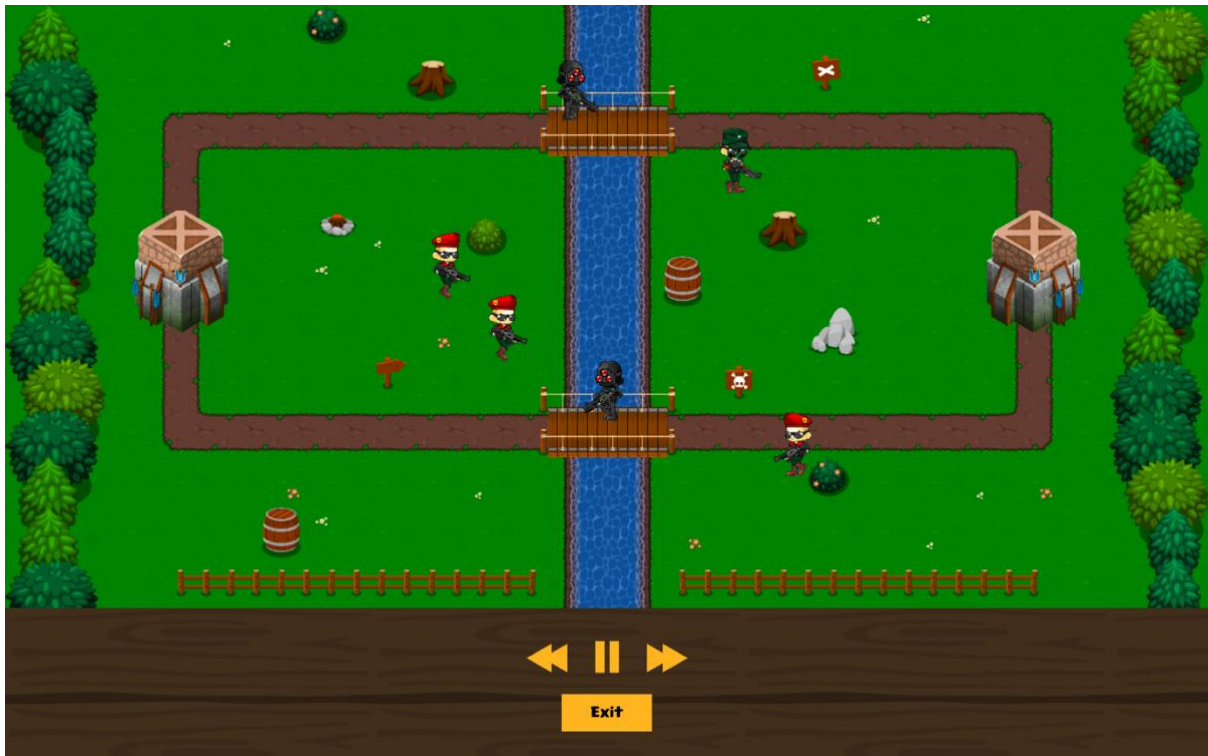
bcarpi ->	Name	Result	Date	ID
	ripp	Lost	23-05 05:43	11
	epic1	Won	23-05 01:54	10
	-	Lost	22-05 05:21	7
	-	Won	22-05 04:43	6
	batallal	Lost	22-05 04:35	5
	yesss	Won	22-05 11:55	4
	-	Won	22-05 11:36	3

Aquesta vista, que apareix al clicar el botó de “rankings” és exactament igual que la anterior (la de “cards”). La única cosa que canvia, es que en comptes de tenir un JPanel amb un GridLayout, tenim un JScrollPane on a dintre hi ha una JTable que mostra informació. Aquesta taula canvia depenent del RoyaleButton pressionat.

Battle:

La complexitat de la batalla es basa en pintar tots els sprites utilitzant `paintComponent()`. Pel que fa als `JPanels` i `layouts`, és bastant senzilla.

Tenim un `JPanel` principal. Al seu centre hi tenim un altre `JPanel` on s'hi pinta tota la batalla (no hi ha cap component `Swing`), i al sud hi tenim un `JPanel` amb les cartes (són imatges que s'agrupen dintre un altre `JPanel` amb un `BoxLayout` `Horitzontal`). Just a sota les cartes, hi pintem manualment dos *rounded rectangles* (un a sobre de l'altre) per representar l'or del personatge.

Reproduce Old Battle:

Al clicar una batalla antiga en el ranking, aquella es reproduirà. La vista és simplement la mateixa que la anterior, i l'únic que canvia es que ara al sud hi tenim un RoyaleButton per abandonar la reproducció i els controls agrupats en un JPanel amb BorderLayout Horitzontal.

Settings a Battle:

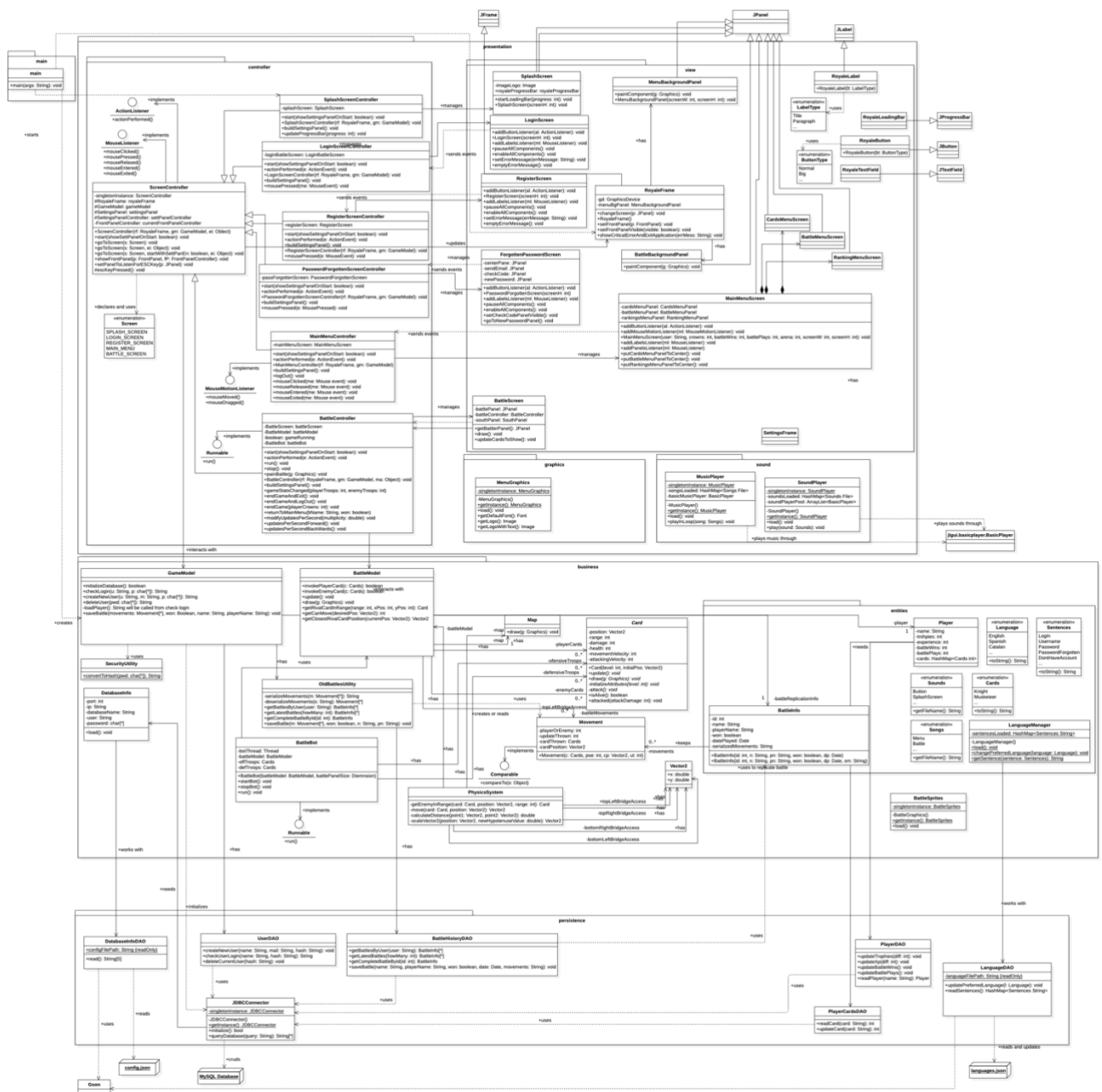
Tal i com hem dit, la pantalla de Settings canvia depenent de la vista on s'obra. En aquest cas, si la obrim des de la Batalla podem veure les estadístiques que se'ns demanaven (pintades també manualment amb paintComponent(), en temps real i amb els eixos escalats)

3. Diagrama de classes

La part més important d'aquest projecte ha sigut, sense cap dubte, el fer un bon disseny inicial del projecte abans de començar a picar una sola línia de codi.

Finalment el projecte ha ocupat **12.000** línies, però a mesura que anàvem afegint més i més mòduls ens anàvem donant compte de lo fàcil que era tot havent fet el previ disseny inicial. També cada setmana, ens anàvem parant per seguir pensant el disseny de les classes abans de començar a implementar-les, i així anar fent més i més gran el diagrama de classes.

Principalment, el disseny es basa en separar el programa seguint una *layered architecture*, i aplicant MVC. Tenim un controller abstracte principal del qual hereten tots els altres controllers, i una screen (vista) abstracte principal de la qual hereten totes les altres vistes.



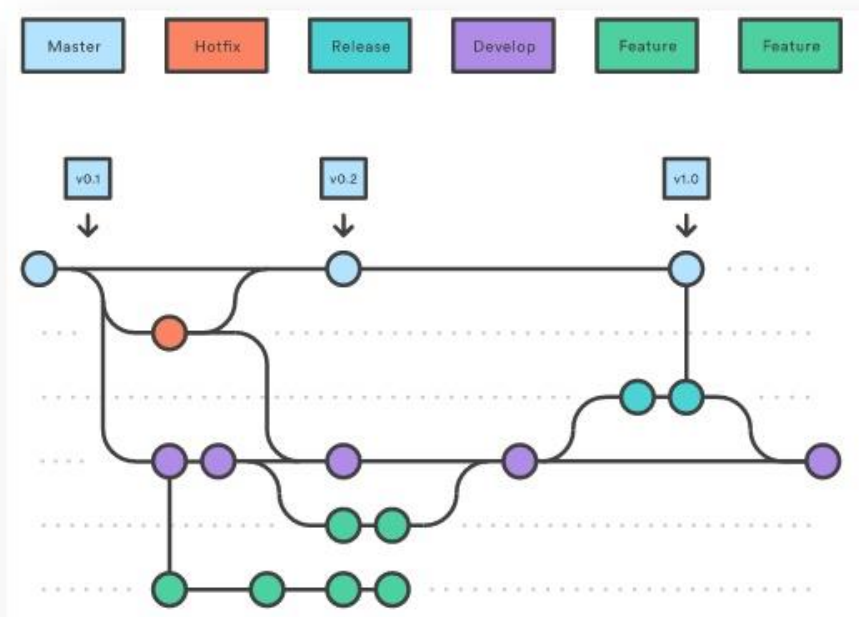
4. Metodologia de desenvolupament

Per poder realitzar el desenvolupament d'aquest treball de forma ordenada, estructurada i en equip, hem utilitzat diferents eines que serveixen per això mateix. En aquest cas, hem utilitzat git per poder realitzar els diferents controls de versions del nostre joc i el Jira per poder gestionar els *sprints*.



Implementar tantes funcionalitats diferents, i més treballant en equip, pot ser una feina gairebé impossible si no s'utilitza un gestor de versions de control. Per això mateix, l'eina de git ens ha facilitat moltíssim la vida i hem pogut anar distribuint i implementant diferents aspectes del joc a l'hora sense cap mena de problema.

Aquesta eina l'hem utilitzat creant una nova branca per cada implementació nova important que anàvem a realitzar (tal i com es veu en la següent imatge). Gairebé cada èpic que tenim al Jira ha estat una nova branca en la que s'ha realitzat la implementació d'aquest apartat. A més, sempre hem anat realitzant els *merge* i *pull requests* cap a la branca *dev* que es d'on sortien les diferents branques, lo que vol dir que en la nostra branca *main* només podem trobar la versió final del joc.



L'altre eina que hem utilitzat ha estat el Jira, la qual ens ha estat molt útil per tot moment tenir accés a tots els diferents *epics* a implementar (lo qual és molt eficient a l'hora de tenir present una idea de la feina que queda per realitzar i organitzar-te millor). A part d'això, ens ha permès poder distribuir la feina entre els diferents membres del grup d'una manera molt senzilla. Un altre aspecte a comentar es que aquesta eina et permet tenir en tot moment constància de quin apartat fa cadascú i en quin estat està aquesta tasca a realitzar.

Això ha estat gràcies a que totes les tasques estaven distribuïdes en estats diferents:

- **To do:** La tasca ha estat escollida per realitzar-se durant aquest *sprint* i encara no s'han començat a implementar.
- **In progress:** Es troben en aquest apartat totes aquelles tasques les quals han estat assignades i s'han començat a implementar. Com és evident, una tasca estarà col·locada gairebé tot el temps en aquest apartat, ja que es el que implica un cost temporal major.
- **In Review:** Secció molt important on es pot trobar una tasca. Es troben en aquest punt una tasca quan la persona a la qual li ha estat assignada ja consideri que ha acabat la seva implementació. En aquest punt, són la resta de companys de l'equip que revises/realitzen testing a aquesta funcionalitat per poder acabar de trobar diferents bugs o d'altre banda, confirmar que ja està implementada correctament i la tasca pot passar al següent i últim estat.
- **Done:** Estat en el que apareixen les diferents tasques que ja han estat revisades per diferents membres de l'equip i després d'un testing s'ha corroborat que ja funciona correctament.

Vista aquesta eina i tot el potencial que té, ja es bastant intuïtiu el mètode que hem utilitzat per poder organitzar-nos. Cada setmana decidíem en equip quines funcionalitats volíem implementar pel següent *sprint*, assignàvem aquestes als diferents membres del grup, i si veïem que alguna tasca estava molt temps en progrés, ens posàvem en contacte amb el company per mirar d'ajudar-lo i anar avançant adequadament treballant en equip.

5. Cost Temporal

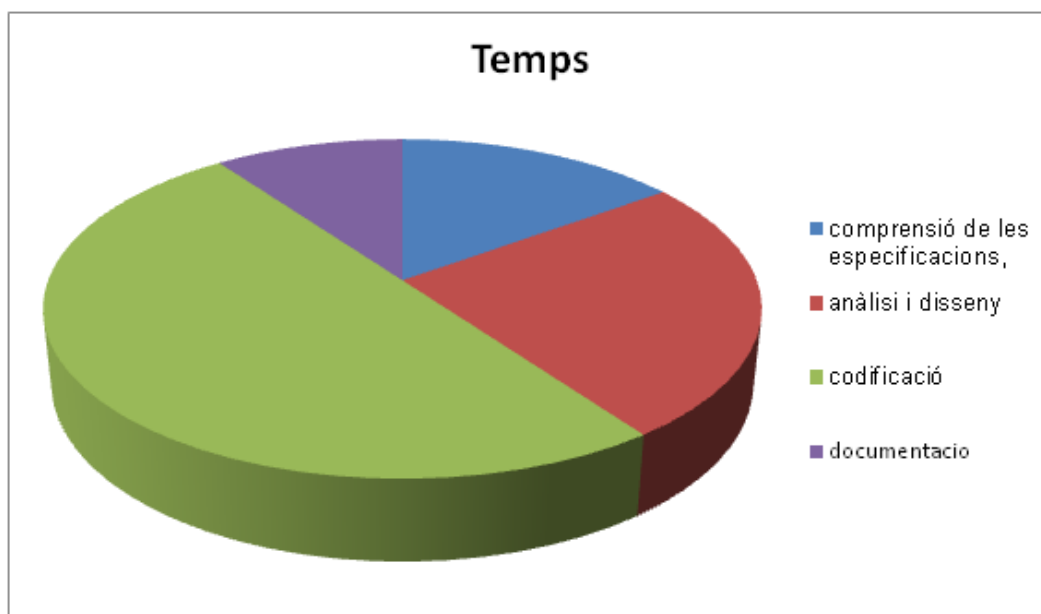
Durant la realització d'aquest projecte, pel que fa a la distribució del cost temporal i organització del projecte, ho hem repartit en 4 diferents *sprints*, comprensió de les especificacions, anàlisi i disseny, codificació i documentació. Aquesta metodologia de treball a nivell de *sprints*, ve donada per l'enunciat del projecte.

Primer de tot la comprensió de les especificacions va ser relativament senzill, ja que estan explicades a l'enunciat. Vam mirar d'entendre tot el que s'havia de fer a la perfecció i vam intentar pensar com estructuraríem tot el projecte en si.

L'anàlisi i disseny ens va portar molt més temps, ja que vam haver de pensar com estructurar l'UML perfectament a causa de la dimensió d'un projecte d'aquest nivell. Aquesta dedicació a pensar l'UML va ser un punt a favor a l'hora de codificar el codi, ja que tot ben fet i ben estructurat és un punt a favor a l'hora de resoldre un problema. El disseny de la base de dades i servidor també ens va portar una part de dedicació d'aquest *sprint*.

Per altra banda la codificació de tot el codi és amb molta diferència la més dedicada amb hores d'aquest projecte. Les vistes ens van portar bastantes hores ja que vam crear nosaltres els nostres propis Components per aconseguir el que realment volíem. Una part molt costosa a nivell d'hores va ser la lògica del joc, el sistema de físiques, la construcció del mapa i el *management* dels sprites, però realment ha valgut la pena pel resultat obtingut.

Per últim la documentació no ens ha suposat molta càrrega de treball, ja que ha estat redactar tot el desenvolupament d'aquest projecte, a més dels coneixements adquirits.



6. Conclusions

Per molts de nosaltres, aquest projecte ens ha servit per a millorar les nostres habilitats en quant a treball en equip es refereix, a part d'aspectes tècnics de la pròpia assignatura. Primer de tot, ens agradaria parlar sobre els coneixements adquirits per la pròpia pràctica i es que, hem dissenyat i implementat un joc, la qual cosa no ha estat gens fàcil. Una gran diferència entre aquesta pràctica i la del primer semestre, sense cap mena de dubte, es el contingut d'aquesta.

L'última pràctica que vam haver d'implementar, no requeria de molta recerca externa per a fer-la, ja que amb els coneixements adquirits a classe ja en teníem pràcticament suficient. Ara però, amb aquest segon semestre a punt de finalitzar, veiem que sense recerca externa, el projecte resultant no hauria quedat ni la meitat de bo i potent. Conceptes com arquitectura per capes i l'ús de la llibreria *SWING* són principalment els dos grans pilars que han portat aquesta pràctica. Entendre be com funciona el patró *MVC*, saber-lo implementar, i entendre perquè serveix, ens ha servit a tots i cadascun dels membres del grup a poder dissenyar un codi estructurat i coherent. A part d'això, la part gràfica del joc feta amb *SWING*, ha estat un clar exemple que la pràctica constant i cometre errors es la millor manera d'aprendre un nou concepte.

A part d'això, ens agradaria comentar tots els beneficis que ha tingut fer aquest projecte en grups de cinc. Traient de banda que sempre es fa més amè treballar acompanyat a treballar sol, creiem que tots hem après la importància que té en un àmbit de treball cooperatiu la comunicació, ja sigui aprendre a parlar als demés companys o aprendre a escoltar les idees i opinions dels altres. Òbviament, tant un projecte de la universitat o en un projecte en un àmbit laboral del món real, sempre es faran grups on hi haurà diferents nivells i personalitats. I una cosa que hem vist amb aquesta pràctica, es la importància d'escoltar i saber aprendre els uns dels altres, perquè de segur que cada membre d'un grup tindrà quelcom interessant a aportar, ja sigui tècnic o personal.

Llavors, que ens ha aportat aquest projecte? Ha estat una gran oportunitat per utilitzar eines molt potents com *GitHub* que perfectament ens podem trobar en un àmbit laboral real i que s'utilitza per molts professionals per tal de tenir un control de versions dels seus projectes. A part, hem consolidat no només els nostres coneixements relacionats amb *Java*, sinó que també a la programació orientada a objectes i tots els seus avantatges gràcies a *IntelliJ*, un software molt potent per a desenvolupar projectes en *Java*. A part d'això, hem hagut de dissenyar nosaltres mateixos la base de dades per tal d'emmagatzemar i posteriorment consultat, dades necessàries per la correcta execució del joc. Hem après tots els avantatges de programar seguint un patró com pot ser el *MVC* i, tot i que sigui mes tediós, de cara a totes les classes i paquets que s'han d'implementar, un codi estructurat i organitzat destaca molt més i li dona molt més potencial. I per últim, hem après a treballar en equip, a respectar les opinions dels demés, i a esforçar-nos perquè ningun membre del equip es quedés enrere.

7. Bibliografia

Stackoverflow. (s. f.). Stack Overflow - Where Developers Learn, Share, & Build Careers. <https://stackoverflow.com>

JavaTM Platform, Standard Edition 7 API Specification. Api Java. <https://docs.oracle.com/javase/7/docs/api/>