Code ::TimeRewind

Par Mathieu RABOT

Table des matières

1 Analyse préliminaire

1.1 Introduction

Le projet est une idée que j'ai depuis quelques années maintenant, c'est de faire un J'ai récolté plusieurs d'idées au fur et à mesure du temps que j'ai mis dans un fichier et j'ai décidé de réaliser ce jeu durant mon **Travail Pratique Individuel**.

Ça me permettait d'enfin réaliser ce projet et me donne **de la motivation** à travailler dessus, j'ai aussi décidé d'apprendre **un nouveau langage** pour moi en même temps, ce qui était un peu une sorte d'obligation. Je m'oblige à réaliser mon jeu en risquant de rater mon TPI.

Mais ça ne m'empêche pas de prendre énormément de plaisir à travailler dessus, je trouve vraiment ça très intéressant, ce que j'apprends et ce que je peux réaliser.

La plus grande chose que m'apporte la réalisation de mon projet c'est de la **satisfaction**, au fur et à mesure que je code différente fonction et que je vois que ça a **un impact**, que ça fonctionne comme je le vois ou alors je fais en sorte que ça corresponde à ce que je veux.

Je passe énormément de temps à **débugger mon code** mais c'est des parties très intéressantes qui peuvent m'apporter une nouvelle façon de réfléchir, d'avoir une vue d'ensemble de mon projet, je pense que c'est en débuggant que j'ai appris le plus de chose lors de la réalisation de mon projet.

C'est mon premier projet personnel que je réalise pleinement d'A à Z, ça m'apporte une première vue d'avoir **un projet** a dirigé et de la charge de travail conséquente que ça implique.

Au début du projet, j'étais justement très **paniqué** par la surcharge de travail que ça allait apporter et surtout avec le combo de commencer un nouveau langage en même temps.

1.2 Objectifs

Le but du projet est de réaliser un jeu vidéo d'aventure en **Java**.

Les objectifs suivant sont tirés de mon premier **cahier des charges**, se seront des actions que mon code devra implémenter.

Le héros du jeu devra pouvoir :
- Se battre
- Gérer son inventaire contenant objets et équipements
- S'équiper d'armes et autre équipements
- Evoluer au fil des combats

Le plus grand **but principal** est l'évolution du héros actuelle, ce dernier devra pouvoir **progresser dans le jeu** avec un système de combat **équilibré** et des **compétences** et ainsi gagner de **l'expérience** pour devenir de plus en plus fort.

1.2.1 Différentes difficultés à venir

Il y a plusieurs difficultés que je vais rencontrer ou que j'ai déjà rencontré dans la conception de mon projet.

En passant par l'apprentissage du Java jusqu'à la lecture de la documentation java pour régler un bug infime.

Les premières difficultés rencontrées étaient le fait de reprendre la programmation après un an de stage sans toucher du code, j'ai dû totalement réapprendre la POO (programmation orienté objet) et rien que l'utilisation de variable et la construction de classe.

Ensuite c'était l'apprentissage du Java, les conventions de nommage, la manière de l'écrire et de l'utiliser était différent du C# ce qui m'a pris une semaine ou deux pour pleinement me mettre à codé de manière plus détendue.

J'ai aussi eu quelque difficulté à m'adapter à mon environnement de travail, Eclipse offre plein de possibilités via son Marketplace intégré mais lors de la configuration de son projet avec l'ajout des différentes librairies, ça a dû me prendre un bon temps d'adaptation afin de tout comprendre.

1.2.2 Planification initiale

J'ai diviser ma planification en semaine, en ajoutant les différents jours ou semaine de vacance durant la durée du projet.

Et j'ai tout rediviser en catégorie des aspects que je vais devoir traiter dans mon projet avec les toutes les tâches ajouter par section.

Projet Code::TimeRewind	**44 jours**	**Lun 01.02.21**	**Jeu 01.04.21**	
Semaine 1	**5 jours**	**Lun 01.02.21**	**Ven 05.02.21**	
Documentation	**2 jours**	**Jeu 04.02.21**	**Ven 05.02.21**	
Création des maquettes utiles durant le long de mon projet				
Implementation	**1 jour**	**Jeu 04.02.21**	**Jeu 04.02.21**	
Commencer a constuire la view du menu en code				
Analyse	**1 jour**	**Lun 01.02.21**	**Lun 01.02.21**	
Chercher des informations de comment coder la vue en Java				
Autres				
Semaine 2	**5 jours**	**Lun 08.02.21**	**Ven 12.02.21**	**2**
Documentation	**1 jour**	**Lun 08.02.21**	**Lun 08.02.21**	
Créer le MCD et le MLD du projet	**1 jour?**	**Lun 08.02.21**	**Lun 08.02.21**	
Implementation	**4 jours**	**Mar 09.02.21**	**Ven 12.02.21**	
Créer le système de Sign In/Sign Up/Logout	4 jours	Mar 09.02.21	Ven 12.02.21	12
Analyse				
Autres				
Semaine 3	**5 jours**	**Lun 15.02.21**	**Ven 19.02.21**	**10**
Documentation				
Implementation	**5 jours**	**Lun 15.02.21**	**Ven 19.02.21**	
Créer le système de Sign In/Sign Up/Logout	4 jours	Lun 15.02.21	Jeu 18.02.21	
Créer la vue du lobby après s'être connecté	1 jour	Ven 19.02.21	Ven 19.02.21	20
Analyse				
Autres				
Semaine 4 Vacance Relache	5 jours	Lun 22.02.21	Ven 26.02.21	17
Semaine 5	**5 jours**	**Lun 01.03.21**	**Ven 05.03.21**	**24**
Documentation				
Implementation	**5 jours**	**Lun 01.03.21**	**Ven 05.03.21**	
Créer les interactions entre les différentes vues				
Analyse				
Autres				
Semaine 6	**5 jours**	**Lun 08.03.21**	**Ven 12.03.21**	**25**
Documentation				
Implementation	**5 jours**	**Lun 08.03.21**	**Ven 12.03.21**	
Créer les différentes vues avec les informations				
Analyse				
Autres				
Semaine 7	**5 jours**	**Lun 15.03.21**	**Ven 19.03.21**	**31**
Documentation				
Implementation	**5 jours**	**Lun 15.03.21**	**Ven 19.03.21**	
Créer le système de combat entre les alliés et ennemies				
Analyse				
Autres				
Semaine 8	**5 jours**	**Lun 22.03.21**	**Ven 26.03.21**	**37**
Documentation				
Implementation	**5 jours**	**Lun 22.03.21**	**Ven 26.03.21**	
Créer le système de combat entre les alliés et ennemies				
Analyse				
Autres				

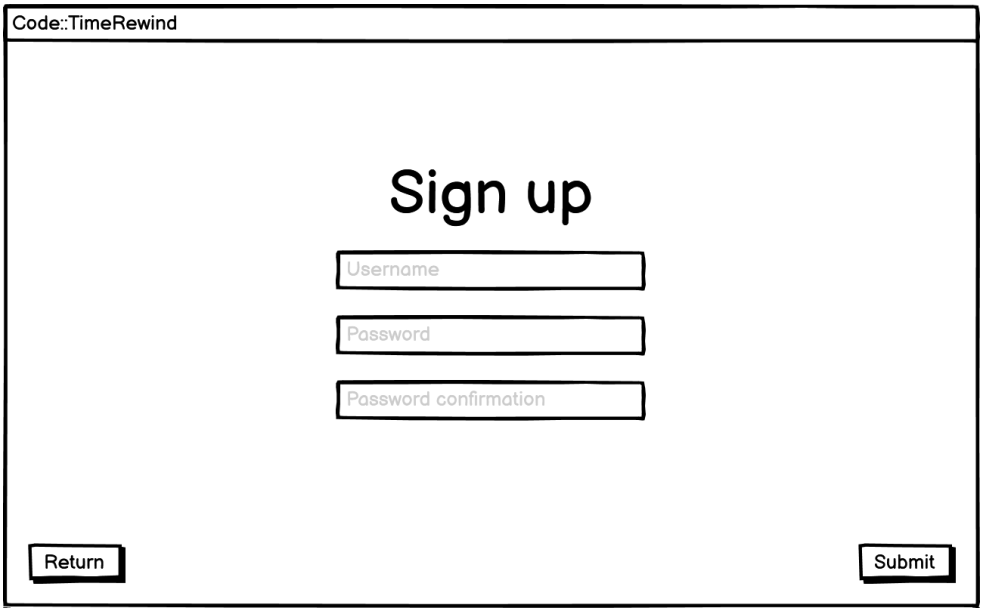
2 Analyse / Conception

2.1 Concept

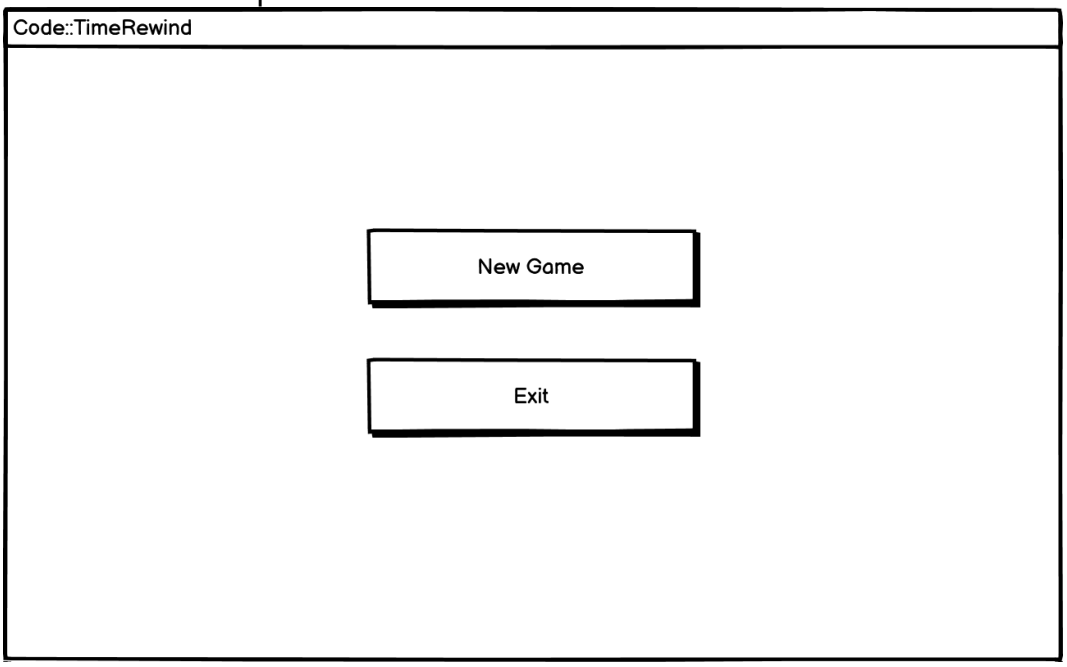
2.1.1 Gestion des maquettes

Au niveau de la conception, j'ai beaucoup réfléchis à la manière dont mon jeu va ressembler, c'est pour ça que j'ai dessiné plein de vue, il y en a certaine qui ne seront peut-être pas utilisé par manque de temps ou par changement d'avis durant le projet :

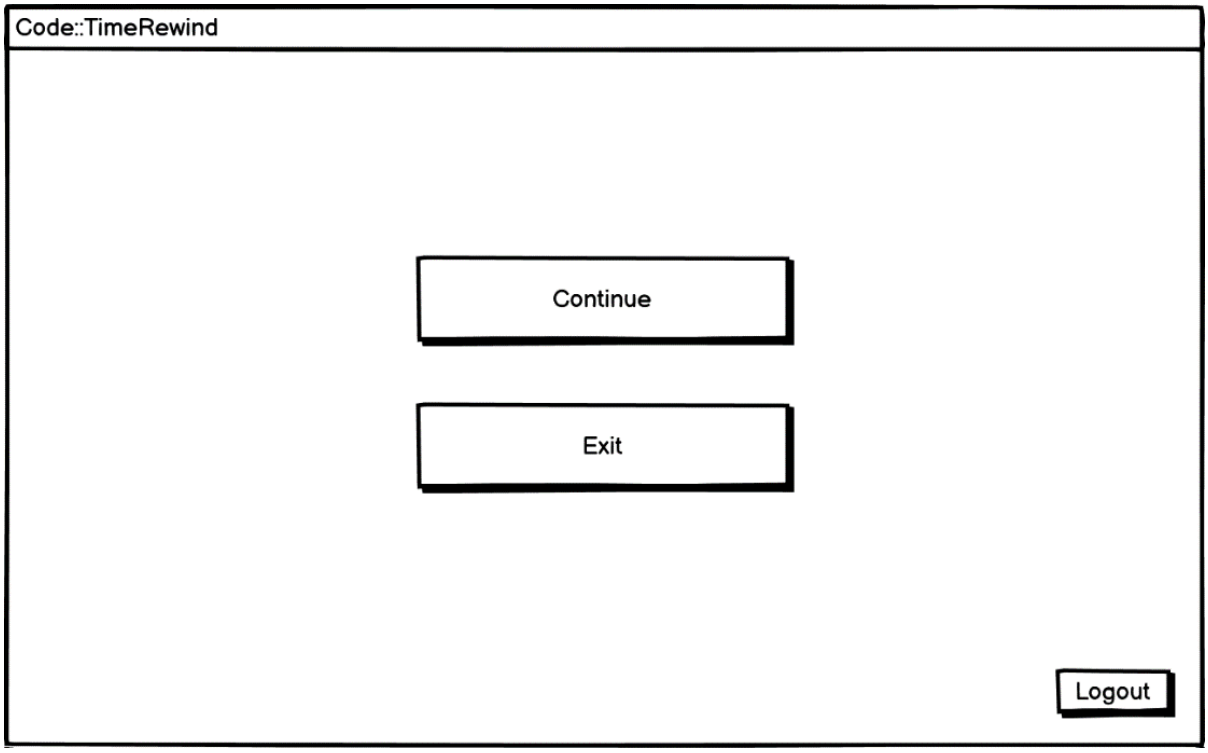
Création d'un compte dans le jeu :



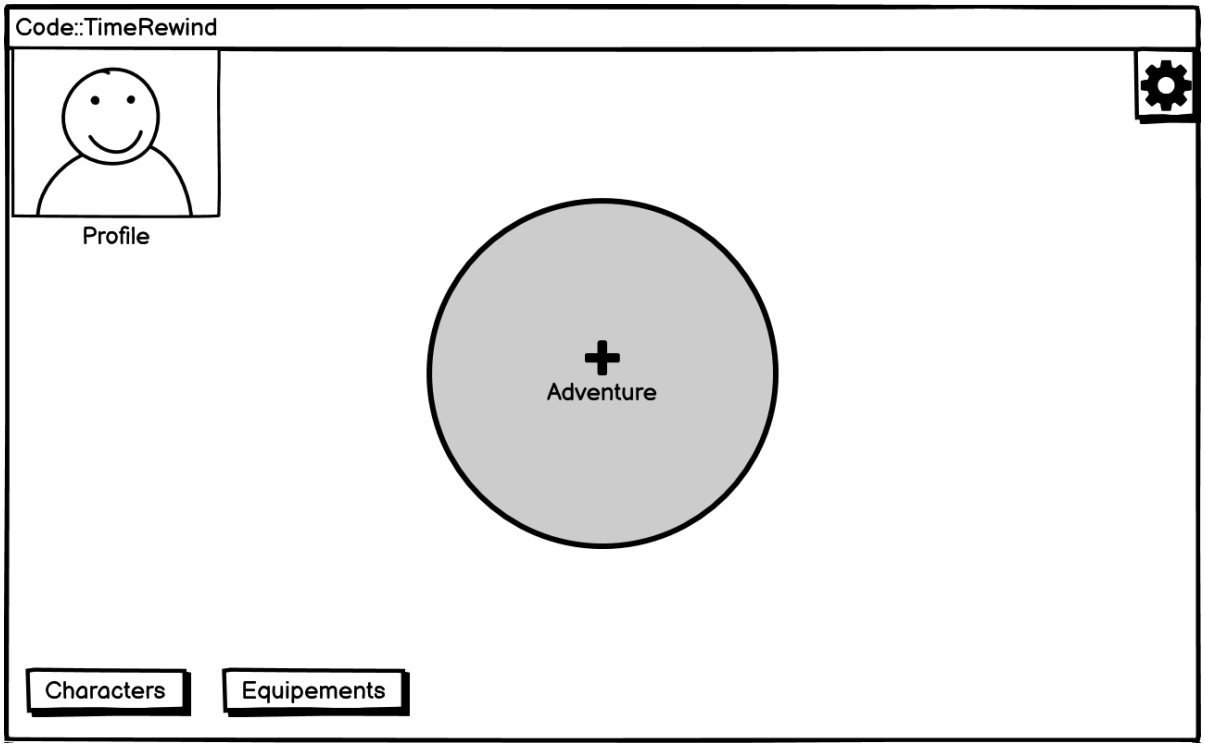
Vue si aucun compte n'est créé dans la base de donnée :



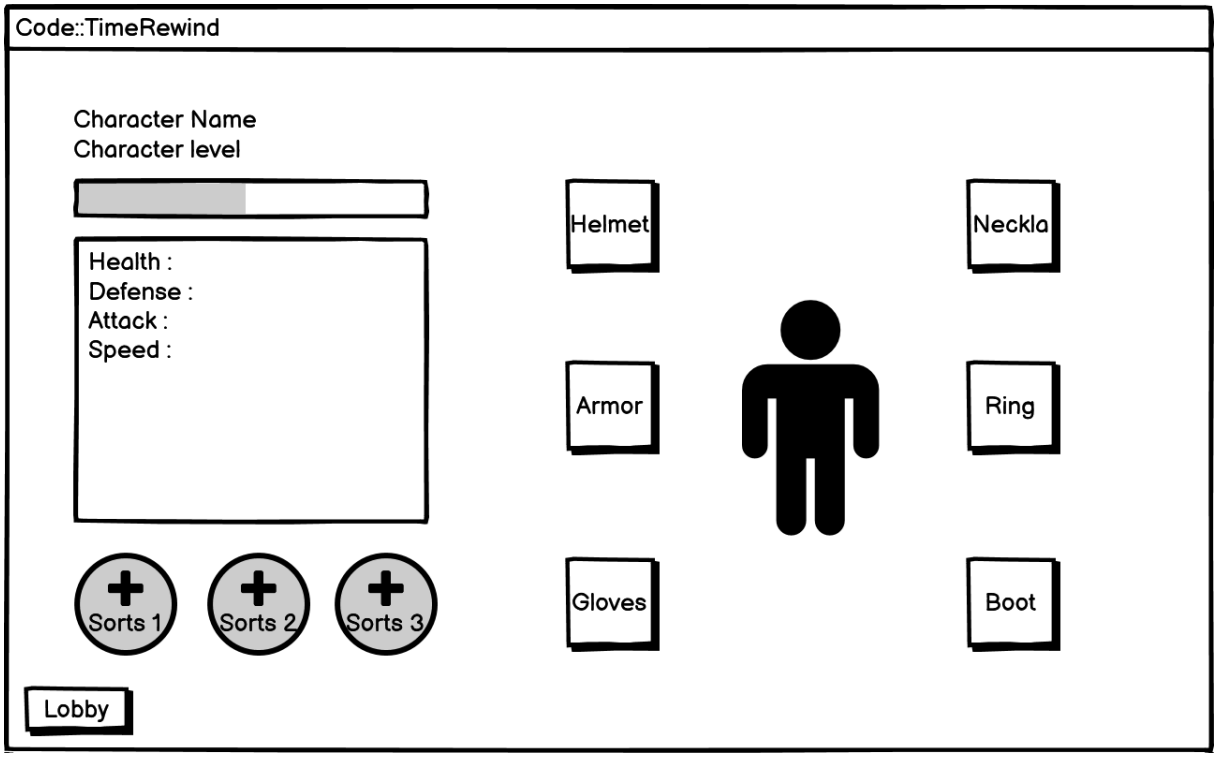
Vue si un compte existe déjà dans la base de donnée et propose à ne pas se connecter mais à directement jouer en cliquer sur « Continue ».



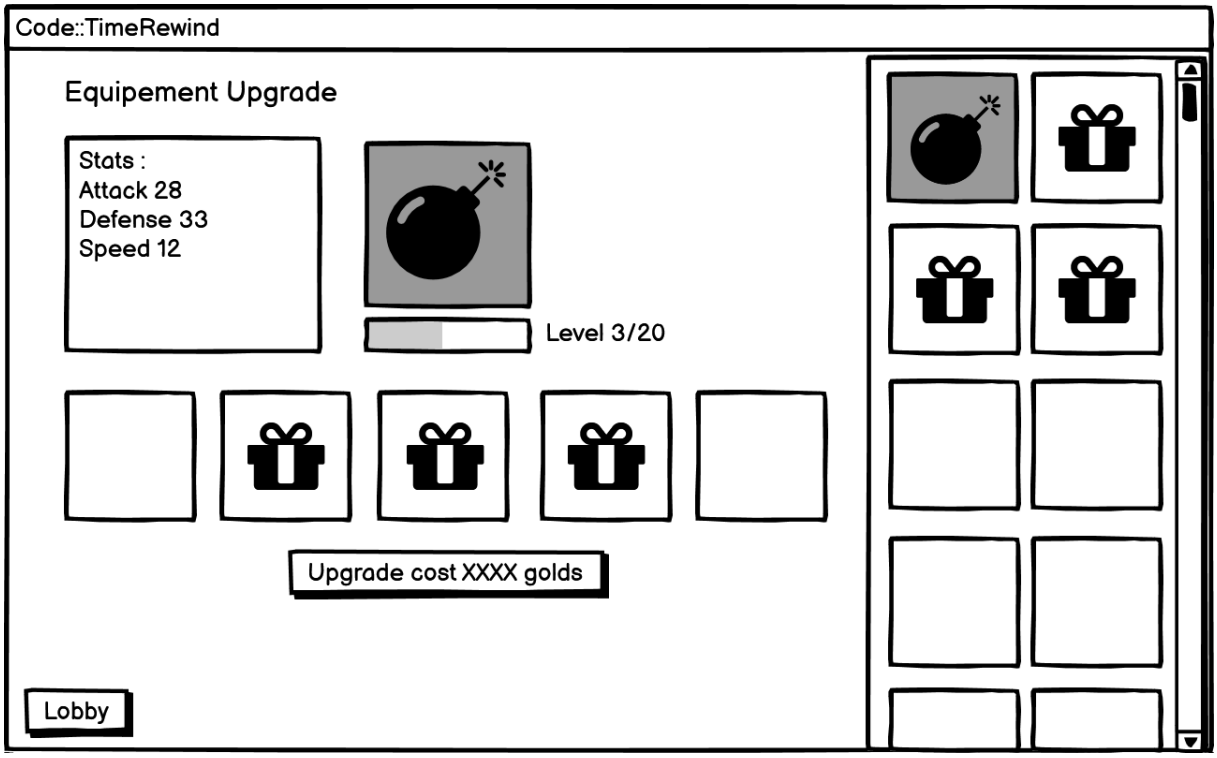
C’est la vue qu’on voit quand on a passé l’étape de la connexion.
On arrive directement sur le lobby où on peut choisir si on veut jouer ou si on veut se balader dans les différentes autres vues du jeu.



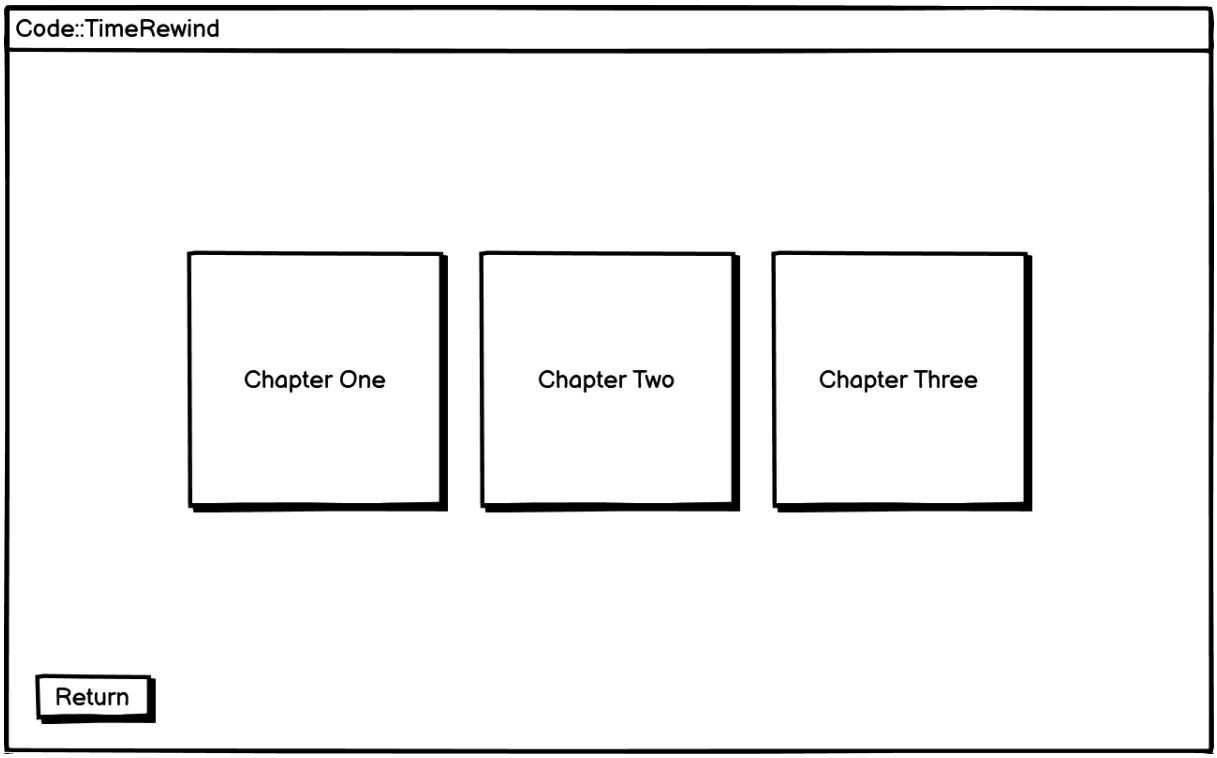
C’est la vue si on clique sur le bouton « Characters » dans la vue du lobby, elle affiche les stats actuelles du personnage ainsi que ses équipements et ses sorts.



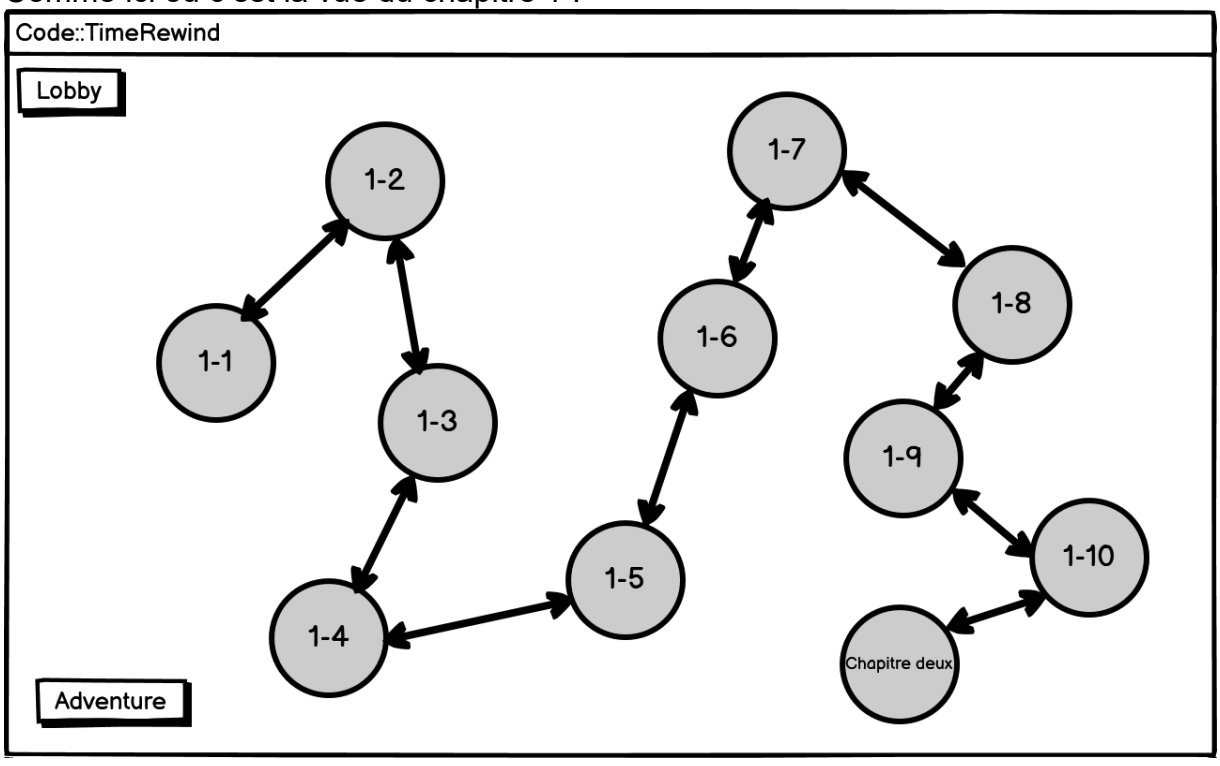
C’est la vue si on clique sur le bouton « Equipements » dans la vue du lobby, elle affiche tous les équipements que le compte a, ainsi que les stats de l’équipement sélectionné.



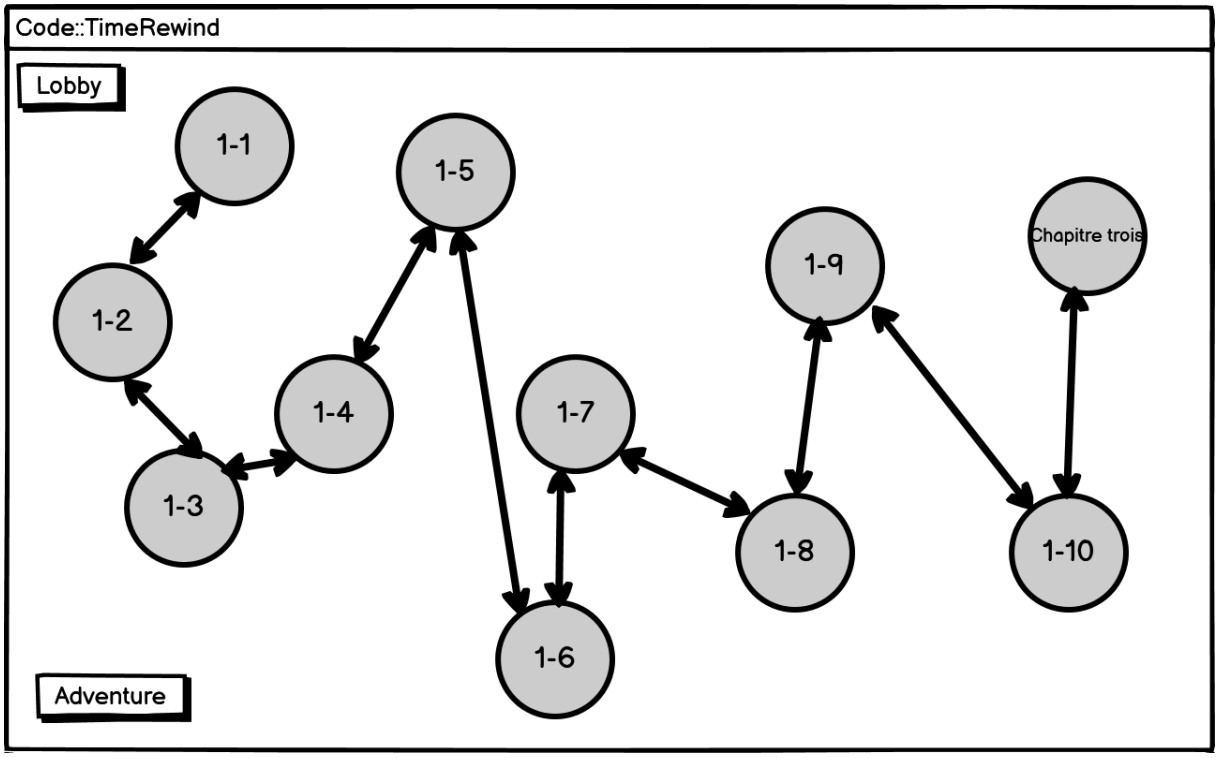
C’est la vue si on clique sur le bouton « Aventure » dans la vue du lobby, elle permet d’accéder aux différentes vues pour jouer au jeu. Elle accède à cette vue qui contient les différents chapitres du jeu. Pour l’instant il y en a que trois mais à l’avenir j’en rajouterais.



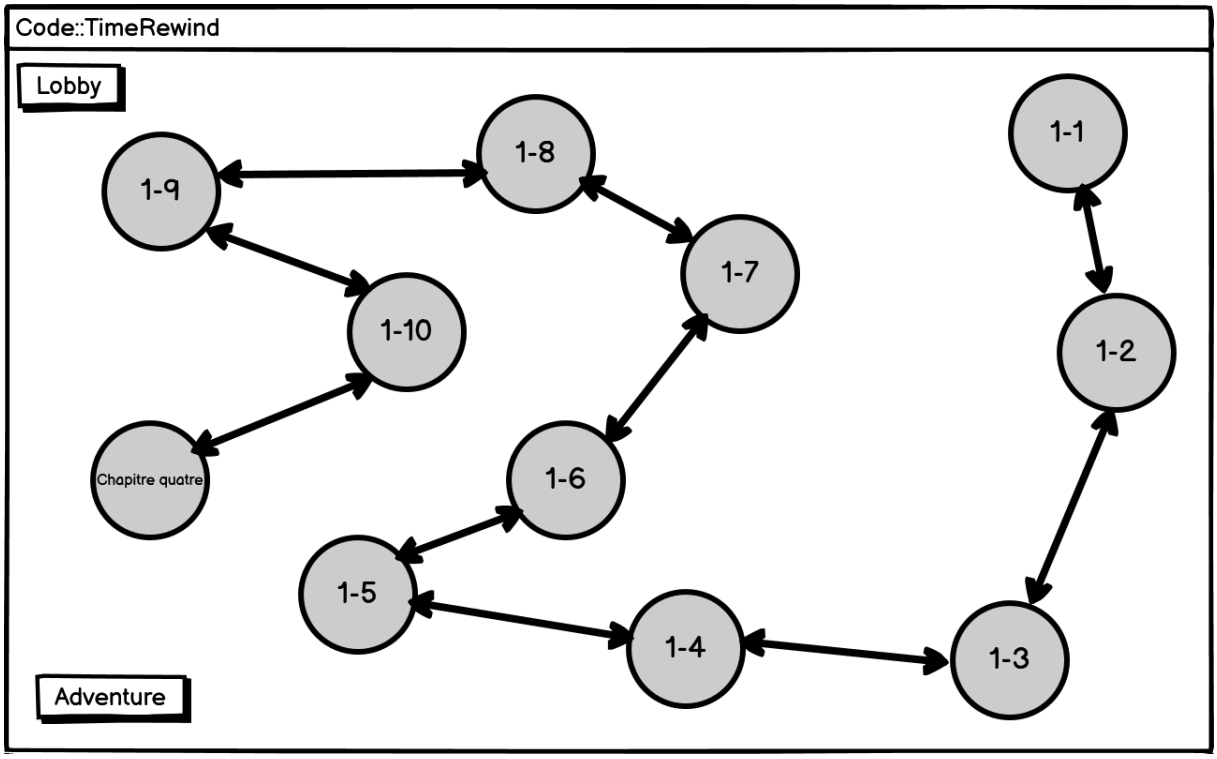
Après la vue des chapitres, nous avons la vue des niveaux par chapitres.
Chaque chapitre contiendra une dizaine de niveaux différents afficher comme ceci.
Comme ici ou c’est la vue du chapitre 1 :



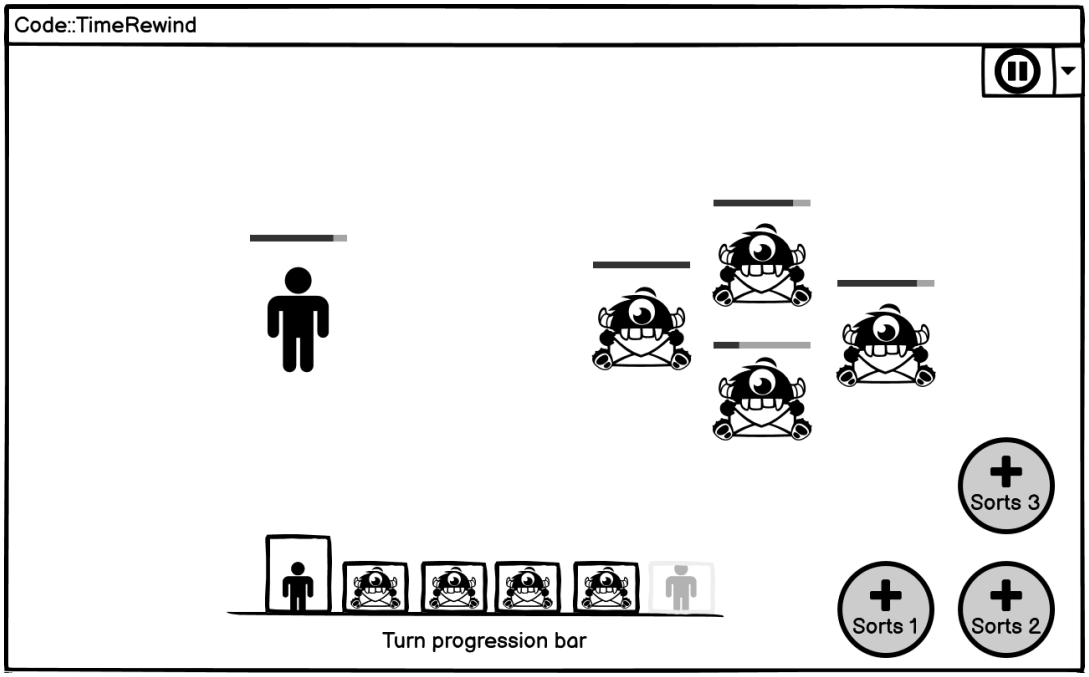
La vue du chapitre 2 :



La vue du chapitre 3 :



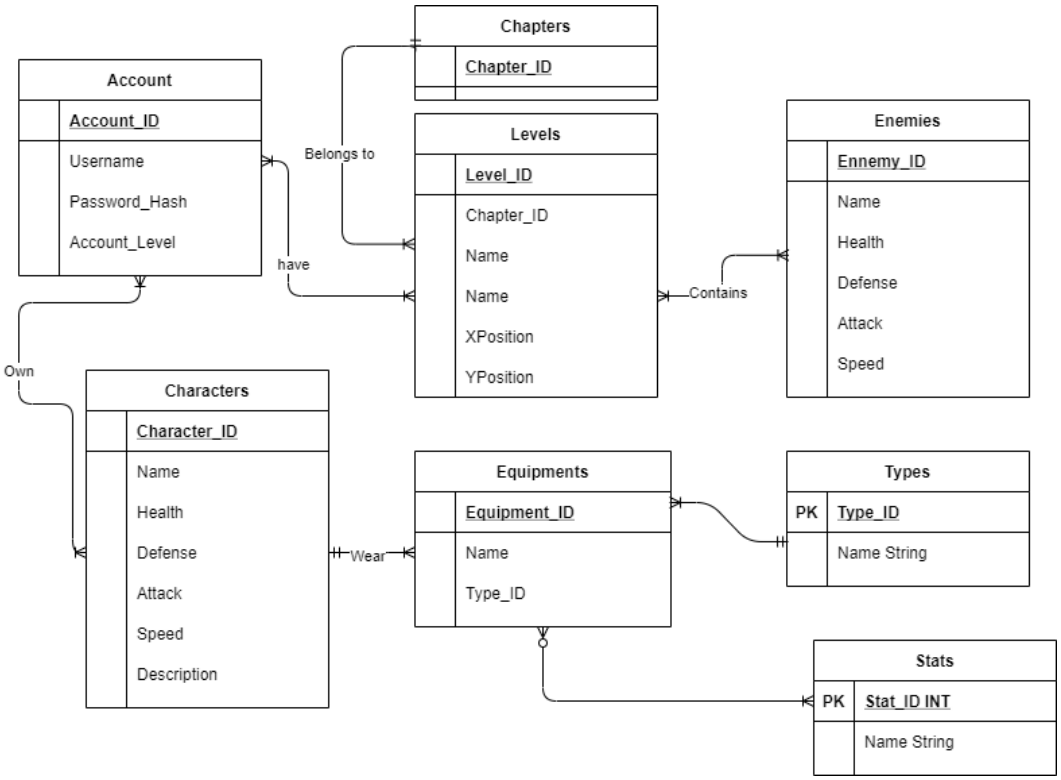
Cette vue c'est le moment où on clique sur un niveau et qu'on commence un combat, la vue affiche les différentes entités avec leur barre de vie et leurs sorts



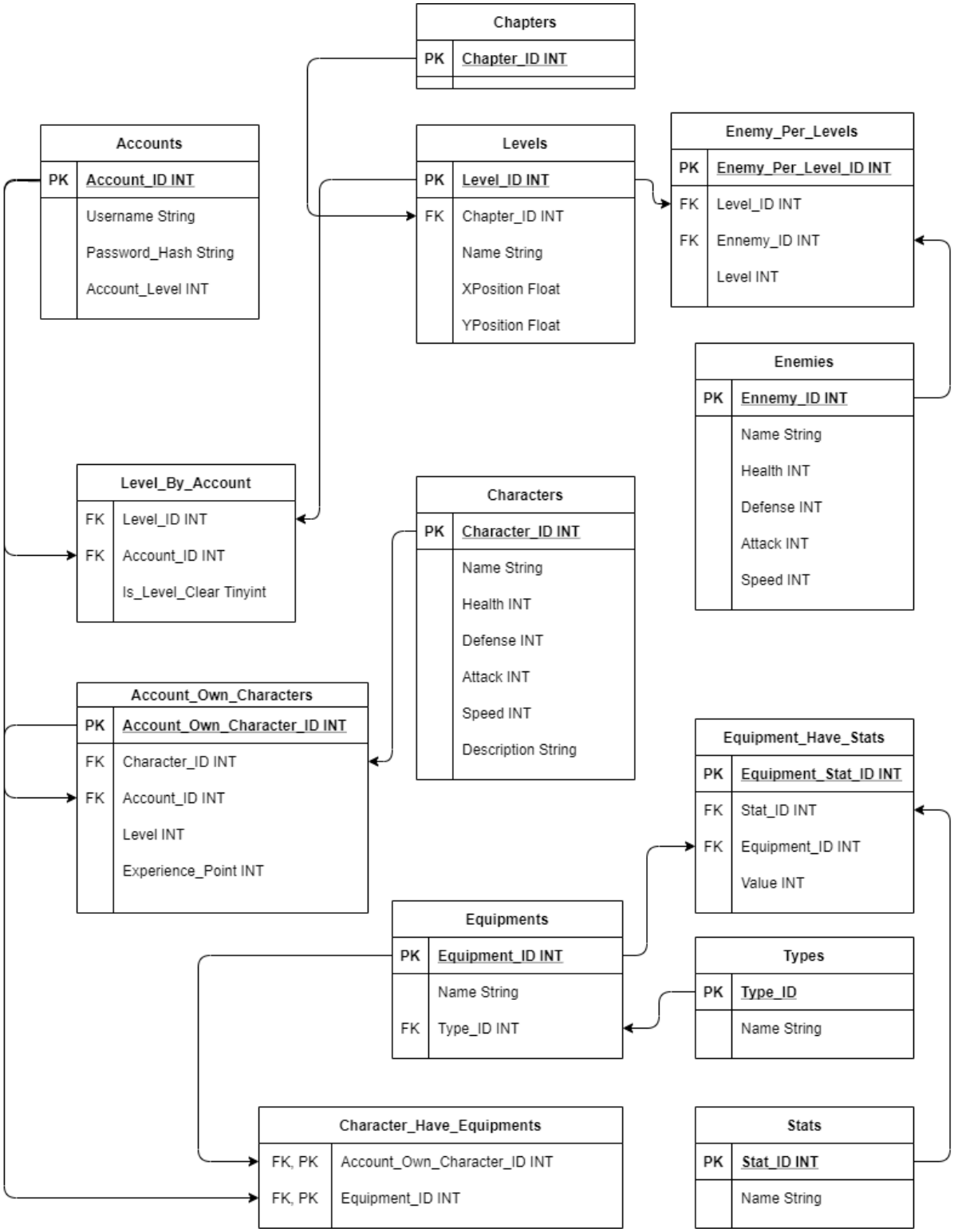
2.1.2 Gestion des données

Pour la gestion des données, j'ai créé un MCD et un MLD qui décrivent les relations entre les tables et la manière dont je l'imagine.

2.1.2.1 MCD



2.1.2.2 MLD


Chapters
PK Chapter_ID INT
Accounts
PK Account_ID INT
Username String
Password_Hash String
Account_Level INT
Levels
PK Level_ID INT
FK Chapter_ID INT
Name String
XPosition Float
YPosition Float
Enemy_Per_Levels
PK Enemy_Per_Level_ID INT
FK Level_ID INT
FK Ennemy_ID INT
Level INT
Enemies
PK Ennemy_ID INT
Name String
Health INT
Defense INT
Attack INT
Speed INT
Level_By_Account
FK Level_ID INT
FK Account_ID INT
Is_Level_Clear Tinyint
Characters
PK Character_ID INT
Name String
Health INT
Defense INT
Attack INT
Speed INT
Description String
Account_Own_Characters
PK Account_Own_Character_ID INT
FK Character_ID INT
FK Account_ID INT
Level INT
Experience_Point INT
Equipment_Have_Stats
PK Equipment_Stat_ID INT
FK Stat_ID INT
FK Equipment_ID INT
Value INT
Equipments
PK Equipment_ID INT
Name String
FK Type_ID INT
Types
PK Type_ID
Name String
Character_Have_Equipments
FK, PK Account_Own_Character_ID INT
FK, PK Equipment_ID INT
Stats
PK Stat_ID INT
Name String

2.2 Stratégie de test

Ma stratégie de test se repose sur la connexion à la base de donnée embarquée. Ça va effectuer des requêtes pour voir si mon projet a toujours les connexions nécessaires au bon fonctionnement de mon jeu.

Les tests pourront être lancé n'importe quand durant le projet, le but est que la connexion est toujours disponible.

Ça se représentera sous la forme de test unitaire qui me permettra de vérifier mes différentes méthodes qui se connecteront à la base de donnée.

2.3 *Risques techniques et solutions appliquées*

2.3.1 Eclipse

J'ai eu plein de problème avec l'utilisation de Eclipse, j'ai dû apprendre l'importation des libraires dans le logiciel et comme le logiciel fonctionnait pour bien compiler mon code.

J'ai regardé plein de tuto de comment paramétré mon environnement de travail et j'apprends toujours aujourd'hui.

2.3.2 Apprendre le Java

L'apprentissage du Java ma pris beaucoup de temps, j'avais déjà un peu avant le pré-TPI en suivant un tutoriel se trouvant dans OpenClassroom (tutoriel présent dans la bibliothèque des sources plus bas).

Le plus dur était l'héritage, j'ai énormément besoin de comprendre, de chercher un sens dans ce que je code, c'est pour ça que mon code de base ne comprenait aucun héritage que j'ai au fur et à mesure implémenté dans la suite de mon projet.

2.3.3 Documentation compliquée

La documentation en générale était un peu compliquée (si elle était existante !) parce que la Javadoc de Slick2D explique pas vraiment les fonctions avec des exemples ou des utilisations, ça décrit uniquement les paramètres demandés et comment appeler les différentes fonctions.

Les tutoriels sur le net sont aussi très peu fréquents, c'est vraiment difficile de trouver les bonnes informations à propos de bug ou encore de solution à nos problèmes.

Les bugs ont pris énormément de mon temps dû au manque de documentation et tutoriel bien que j'ai à chaque fois réussis à trouver une alternative.

2.4 Planification

La planification définitive de mon projet est la même que mon initial, étant donné que je n'avais pas vraiment compris le système de planification alors j'ai déjà fait suivre ma planification durant le long du projet au lieu d'en faire une complète au début.

Projet Code::TimeRewind	**44 jours**	**Lun 01.02.21**	**Jeu 01.04.21**	
Semaine 1	**5 jours**	**Lun 01.02.21**	**Ven 05.02.21**	
Documentation	**2 jours**	**Jeu 04.02.21**	**Ven 05.02.21**	
Création des maquettes utiles durant le long de mon projet				
Implementation	**1 jour**	**Jeu 04.02.21**	**Jeu 04.02.21**	
Commencer a constuire la view du menu en code				
Analyse	**1 jour**	**Lun 01.02.21**	**Lun 01.02.21**	
Chercher des informations de comment coder la vue en Java				
Autres				
Semaine 2	**5 jours**	**Lun 08.02.21**	**Ven 12.02.21**	**2**
Documentation	**1 jour**	**Lun 08.02.21**	**Lun 08.02.21**	
Créer le MCD et le MLD du projet	**1 jour?**	**Lun 08.02.21**	**Lun 08.02.21**	
Implementation	**4 jours**	**Mar 09.02.21**	**Ven 12.02.21**	
Créer le système de Sign In/Sign Up/Logout	4 jours	Mar 09.02.21	Ven 12.02.21	12
Analyse				
Autres				
Semaine 3	**5 jours**	**Lun 15.02.21**	**Ven 19.02.21**	**10**
Documentation				
Implementation	**5 jours**	**Lun 15.02.21**	**Ven 19.02.21**	
Créer le système de Sign In/Sign Up/Logout	4 jours	Lun 15.02.21	Jeu 18.02.21	
Créer la vue du lobby après s'être connecté	1 jour	Ven 19.02.21	Ven 19.02.21	20
Analyse				
Autres				
Semaine 4 Vacance Relache	5 jours	Lun 22.02.21	Ven 26.02.21	17
Semaine 5	**5 jours**	**Lun 01.03.21**	**Ven 05.03.21**	**24**
Documentation				
Implementation	**5 jours**	**Lun 01.03.21**	**Ven 05.03.21**	
Créer les interactions entre les différentes vues				
Analyse				
Autres				
Semaine 6	**5 jours**	**Lun 08.03.21**	**Ven 12.03.21**	**25**
Documentation				
Implementation	**5 jours**	**Lun 08.03.21**	**Ven 12.03.21**	
Créer les différentes vues avec les informations				
Analyse				
Autres				
Semaine 7	**5 jours**	**Lun 15.03.21**	**Ven 19.03.21**	**31**
Documentation				
Implementation	**5 jours**	**Lun 15.03.21**	**Ven 19.03.21**	
Créer le système de combat entre les alliés et ennemies				
Analyse				
Autres				
Semaine 8	**5 jours**	**Lun 22.03.21**	**Ven 26.03.21**	**37**
Documentation				
Implementation	**5 jours**	**Lun 22.03.21**	**Ven 26.03.21**	
Créer le système de combat entre les alliés et ennemies				
Analyse				
Autres				

Ça c'est en général très bien déroulé, cependant le temps ma rattraper et je n'ai pas planifié assez de temps pour faire ma documentation durant le long du projet.

2.5 Dossier de conception

Dans ce chapitre, je vais détailler les différents logiciels que j'utilise, pourquoi je les utilise et qu'est-ce qu'ils apportent à mon projet, c'est pourquoi j'ai découpé ce chapitre en plusieurs sous-chapitres qui me permettent de rentrer plus dans les détails.

2.5.1 Choix du matériel :

Au niveau du matériel choisis, je dois obligatoirement faire mon travail en classe, donc utilisé la machine mise à disposition par le CPNV.

La machine contient cette configuration :

Processeur :	Intel i7-6400 3.40GHz 8 cœurs
Carte Graphique :	Intel HD Graphics 530
Mémoire :	16 Go
Stockage :	160 Go
Système d'exploitation :	Windows 10

2.5.2 Environnement de travail :

Mon environnement de travail en général se fera sur Eclipse qui est un IDE spécialement fait pour Java.

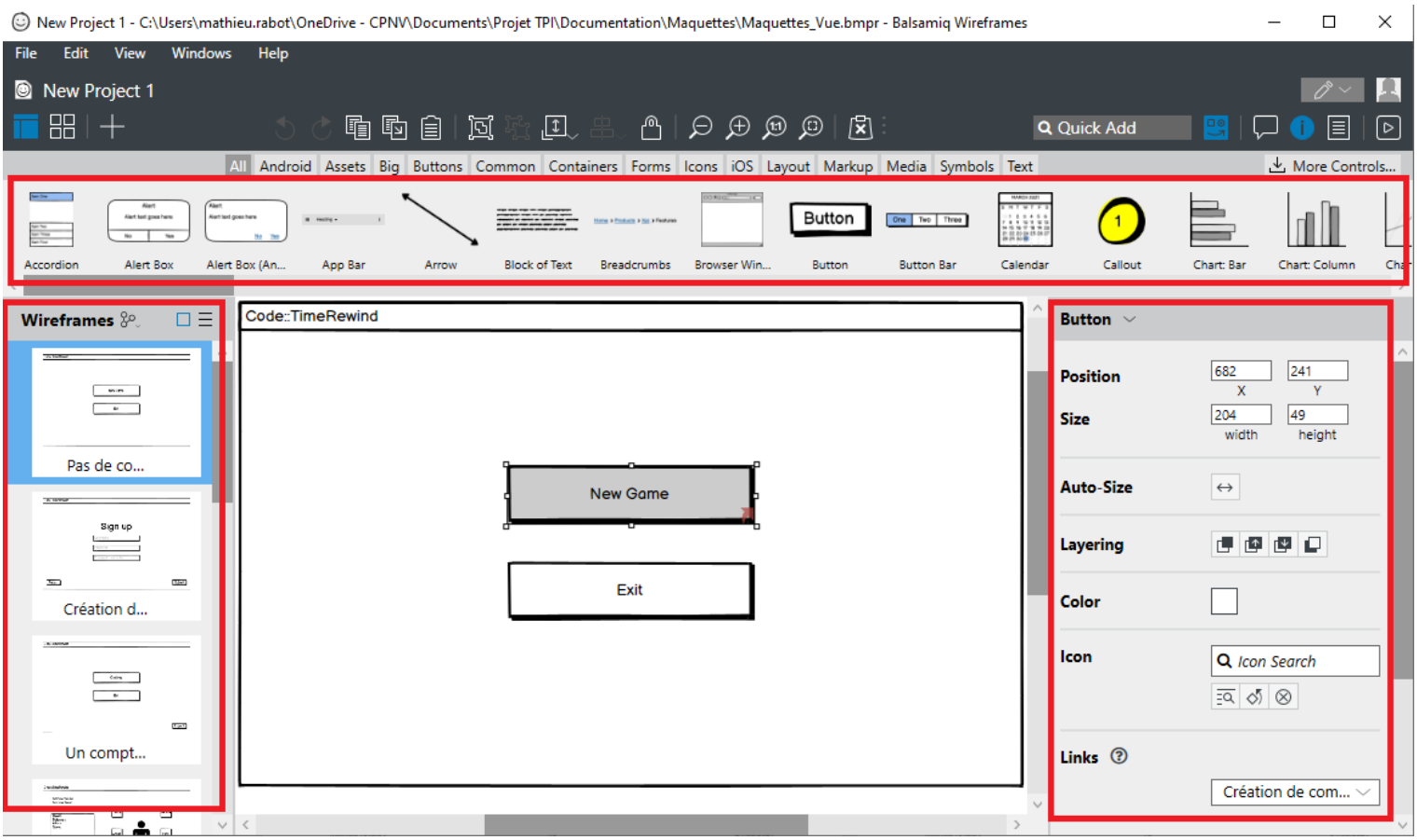
J'ai choisis cette IDE parce que Eclipse est gratuit et très complet dû au Marketplace intégré qui offre beaucoup de possibilité d'optimisation de la plateforme.

On m'a souvent conseillé de prendre IntelliJ comme plateforme de base pour coder en Java, mais la raison principale pour laquelle je veux utiliser Eclipse est que je vais continuer de programmer mon jeu après mon TPI, ce qui veut dire que je n'aurais plus accès a IntelliJ car je ne serais plus étudiant.

J'ai pris la dernière version stable de Eclipse (actuelle) qui est la 2020-12 (4.18)

2.5.3 Réalisation des maquettes :

Toutes la gestion de mes maquettes, que j'ai présentées dans le point 2.1, a été faite sur le logiciel Balsamiq Wireframes comme présenté ci-dessous :



- Le premier rectangle rouge contient les différents modules qui composent les vues, se sont plein d'outils qu'on peut déplacer dans notre vue et qu'on peut ensuite paramétrer.
- Le rectangle de gauche contient toute les vues qui composent notre projet, c'est l'endroit où on peut naviguer entre les différentes vues rapidement.
- Le rectangle de droite contient les différents paramètres qu'on peut modifier pour chaque module qu'on a déplacé dans notre vue, dans l'exemple ici, je peux configurer la position, la taille, la position du calque, la couleur, l'icône et le lien avec quel vue le bouton redirige.

2.5.4 Gestion des bases de données :

Pour la conception de ma base de données, j'ai décidé d'utiliser Draw.io qui est un logiciel très pratique pour faire de la modélisation en générale

Et pour l'implémentation de ma base de données, j'ai opté pour une base de données embarquée.

C'est un type de base de données très intéressant parce qu'elle permet de ne pas avoir besoin d'installer de service sur la machine du client ni d'avoir de système de base de données de base.

J'utilise Apache Derby, qui est la base de données embarquée d'Apache, toutes les requêtes créées sont écrites comme du SQL, ce qui permet de mettre à profit mes connaissances en requête et de pas avoir à en créer de nouvelle.

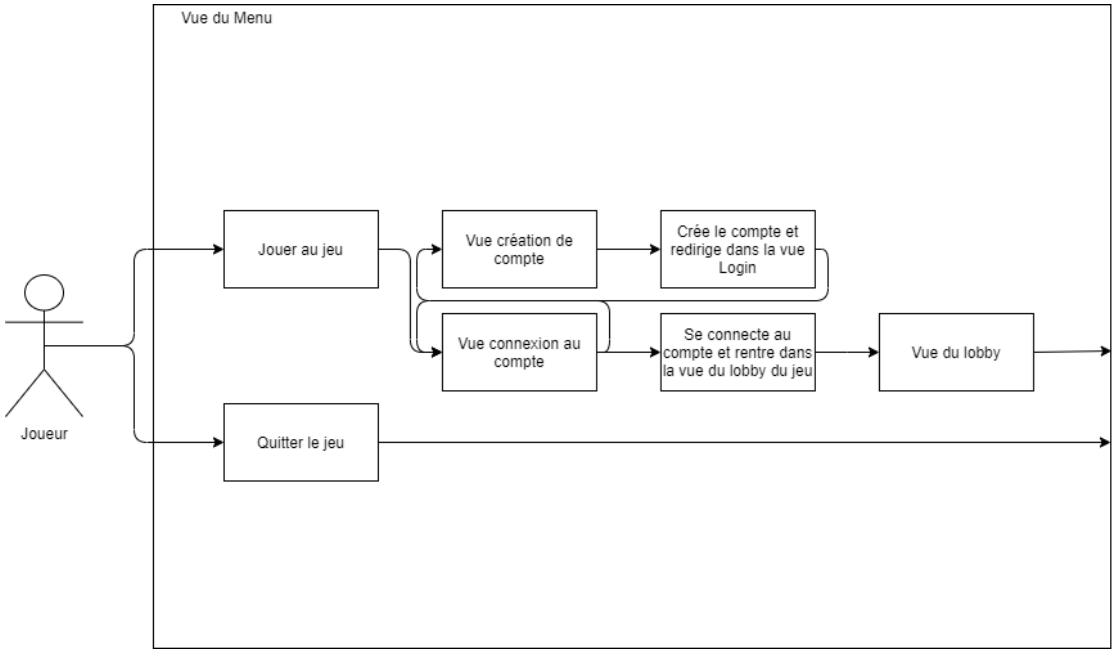
Et comme gestionnaire de base de données, donc pour interagir directement avec Derby, j'utilise DataGrip qui support les bases de données embarquées. Mais je peux aussi y accéder avec l'invite de commande de base de Windows via un utilitaire intégré à Derby qui s'appelle « ij ».

2.5.5 Diagramme projet :

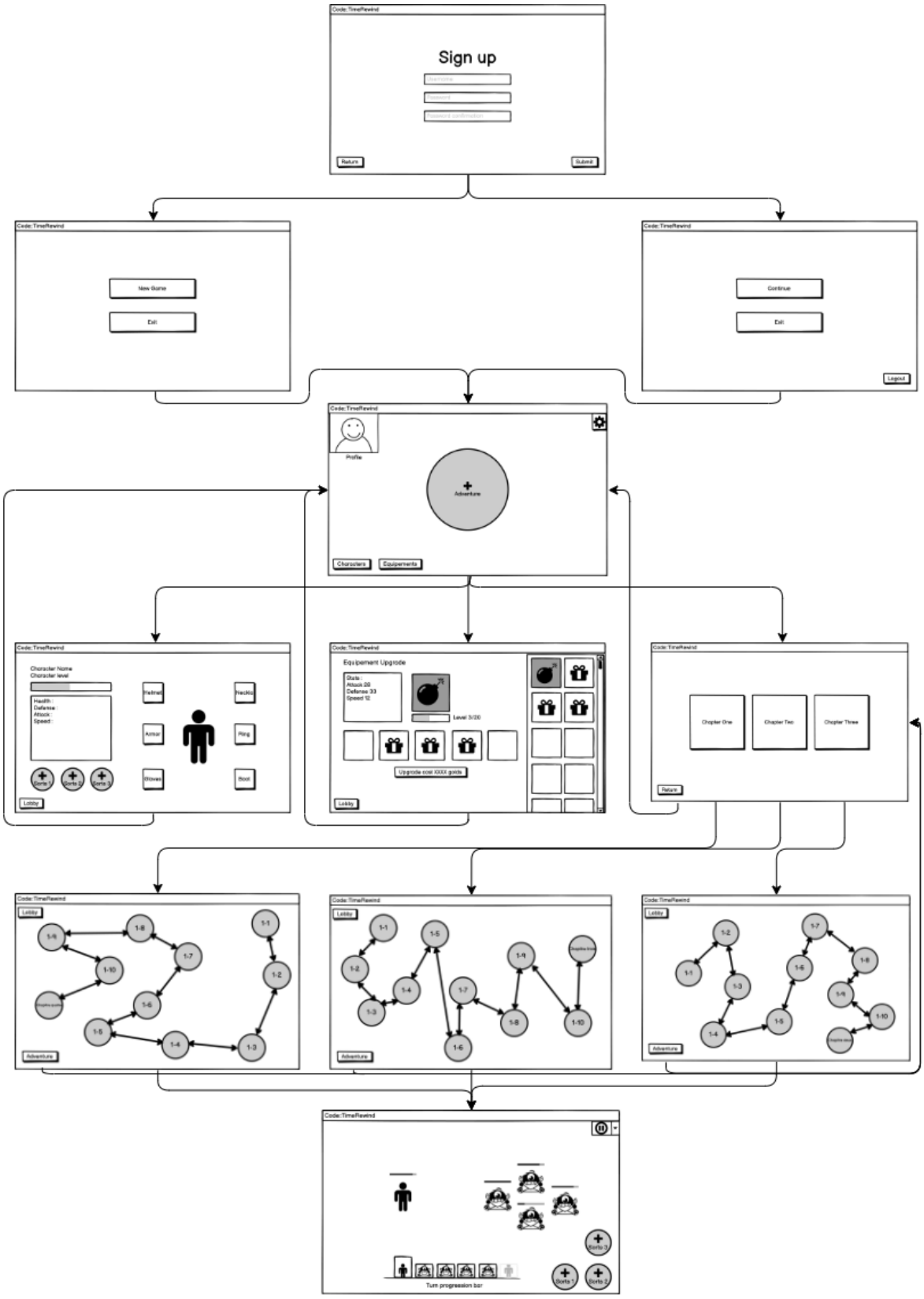
Ce diagramme représente les différents choix que feront le joueur quand il entrera dans le jeu, de base il se trouvera dans la vue du menu et il pourra décider de vouloir quitter le jeu ou de jouer, s'il décide de jouer, ça le redirige dans la vue de connexion ou il pourra choisir de se connecter ou de s'inscrire.

Après s'être inscrit, la vue change pour celle de la vue ou le joueur devra se connecter ce qui le fera rentrer dans la vue du lobby et pourra commencer à jouer.

Et s'il avait déjà un compte, il pourra directement se connecter pour accéder au jeu et continuer avec sa progression actuelle.



Ce diagramme représente l'architecteur / la disposition des maquettes et leur ordre dans le projet, on peut voir comment accéder à une certaine maquette et en sortir.



3 Réalisation

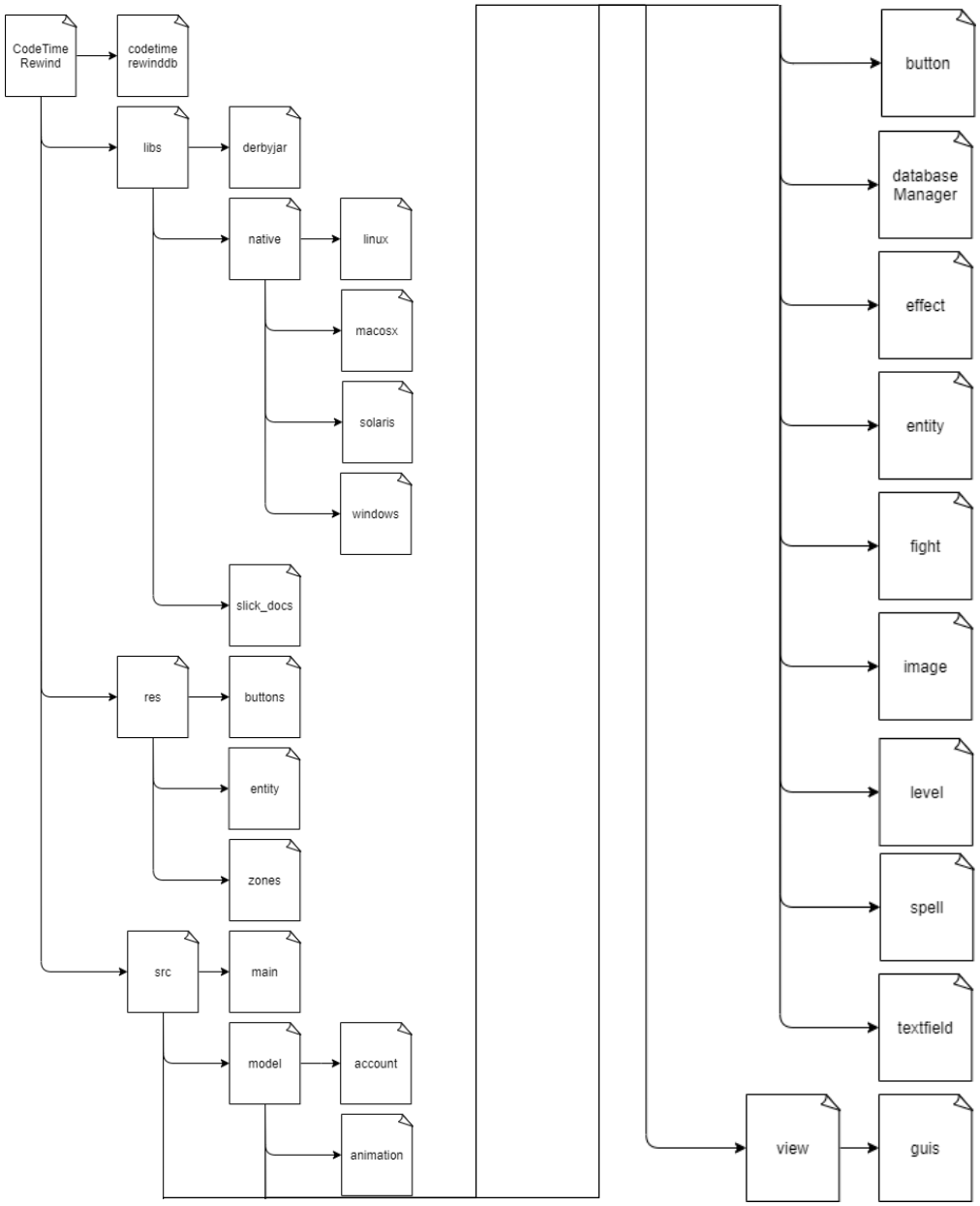
3.1 Dossier de réalisation

3.1.1 Répertoire du logiciel :

Mon Eclipse se trouve dans le dossier :
C:\Users\mathieu.rabot\eclipse\java-2020-122\eclipse

L'entièreté du projet est créée dynamiquement, ce qui veut dire que ça importe peu l'endroit où l'application se trouve, tant que les différents autres dossiers sont avec dans le même dossier, l'application fonctionnera.

3.1.2 Liste des fichiers de mon projet :



3.1.2.1 Code-TimeRewind
Ce dossier est le point principal de mon projet, qui contient toutes les libraires ainsi que mes classes.

3.1.2.2 Codetimerewinddb
Ce dossier est le dossier crée automatiquement au lancement de mon application, il contient ma base de donnée embarquée.

3.1.2.3 Libs
Ce dossier contient toutes les librairies que j'utilise dans mon projet, l'explication en détail de chaque librairie se trouvera quelque chapitre en dessous.

3.1.2.4 DerbyJar
Ce dossier contient toutes mes librairies concernant ma base de donnée.

3.1.2.5 Native
Ce dossier contient les natives de mon projet, je définis sous quel OS je veux exécuter mon jeu et les .dll dans le dossier native fera la transcription.

3.1.2.6 Slick-docs
Ce dossier contient les différentes documentations pour la librairie Slick qui est ma librairie principale.

3.1.2.7 Res
Ce dossier contient toutes mes ressources nécessaires au fonctionnement de mon projet, il contient surtout les images de mes personnages, et des décors.

3.1.2.8 Res/Buttons
Ce dossier contient toutes les images pour mes différents boutons dans le jeu, il existe deux types de boutons principaux, le normal et celui qui est pressé qui est plus assombrie que l'autre pour simuler le bouton appuyer.

3.1.2.9 Res/Entity
Ce dossier contient toutes les images de mes différents personnages dans le jeu.

3.1.2.10 Zones
Ce dossier est plus générique parce qu'il contient des conteneurs que j'ai appelé des « zones » qui sont aussi des images mais qui servent beaucoup pour l'affichage de l'expérience ou encore pour mettre un fond aux sorts et aux statistiques.

3.1.2.11 Src
Ce dossier contient toutes mes classes et mon projet en générale.

3.1.2.12 Main
Ce dossier contient les classes principales de mon jeu, il contient le Main qui est le point d'entrée, ainsi que le Game qui est le gestionnaire qui fait la liaison entre toutes les différentes vues.

3.1.2.13 Model

Ce dossier contient les classes qui gère la liaison entre les vues et les données dans la base de donnée ou les données stockées en générale. (Il sera amélioré pour faire la liaison entre le futur Controller et les données, il y aura plus de lien direct avec les vues après.

3.1.2.14 Model/Account

Ce dossier contient la classe account, qui est la classe qui va stocker les informations du compte actuel, tel que le mot de passe hashé ou le nom de compte, ainsi que la liste des personnages que le compte possède.

3.1.2.15 Model/Animation

Ce dossier contient les différentes animations en combat, elle permet de faire des combats plus dynamique et plus intéressant que si les persos ne pouvaient pas bouger.

3.1.2.16 Model/Button

Ce dossier contient des boutons pour chacune de mes vues, ça hérite des boutons qui se trouve dans la librairie de Slick2D, la différence c'est que dans ces boutons je définis leur position, leur taille et leur image par défaut.

3.1.2.17 Model/DatabaseManager

Ce dossier contient mon database manager, qui est ma classe qui fait le lien entre la base de données et mon jeu, ça permet d'exécuter différentes requêtes SQL et de mettre à jour la BDD.

3.1.2.18 Model/Effect

Ce dossier contient les différents effets que peut faire les sorts des personnages, tel que des effets actif ou passif.

3.1.2.19 Model/Entity

Ce dossier contient les différentes Entity dans le jeu, tel que les personnages du joueur et les ennemies, et permettent leur gestion.

3.1.2.20 Model/Fight

Ce dossier contient les informations stockées pour faire un combat ainsi que son contrôleur.

3.1.2.21 Model/Image

Ce dossier contient les classes des images qui sont des héritages de la classe Image de Slick2D, elle me permet de donner une taille, une position et de dessiner les images sur la vue.

3.1.2.22 Model/Level

Ce dossier contient les différentes informations par rapport aux levels, tel que le nombre d'ennemis dans le level, sa position ou encore dans quel chapitre il se trouve.

3.1.2.23 Model/Spell

Ce dossier contient les différents sorts des personnages, que se soit des monstres ou des personnages, ils contiennent des sorts avec des effets et des temps de chargement.

3.1.2.24 Model/TextField

Ce dossier contient des text fields, qui sont des champs text que j'utilise pour faire mes bars dans le login et le register qui demande a l'utilisateur de rentrer son nom et mot de passe. Ça hérite de la classe du même nom de Slick2D.

3.1.2.25 View

Ce dossier contient mes différentes vues que l'utilisateur verra lorsqu'il lancera le programme et jouera.

3.1.3 Version de mon produit :

La version actuelle de mon jeu est la 1.0 qui est une version stable mais qui comporte encore énormément de bug, lorsque je vais en résoudre plusieurs d'un coup, je vais faire passer le jeu à la version suivante.

3.1.4 Description des librairies utilisées :

3.1.4.1 Junit
Cette libraire me permet de faire des tests unitaires dans mon projet, ça me permet de tester des fonctions et de vérifier si les tests fonctionnent, ça voudrais dire que mon code est fonctionnel.

3.1.4.2 Lombok
Cette librairie est très pratique, elle me permet de faire en sorte que mes accesseurs et mes setters dans chacune de mes classes soient déjà utilisable sans avoir besoin de les coder ex :

```
@NoArgsConstructor
@Getter
@Setter
public class GuiAdventure extends BasicGameState {
    private int stateId;
    private Image background;
    private Button lobbyButton;
    private Button adventureButton;
```

Pour mes attributs LobbyButton et AdventureButton, je n'aurais pas besoin de déclarer les getters et les setters grâce a Lombok.

3.1.4.3 Lwjgl et lwjgl_util
Ces librairies sont obligatoires pour pouvoir utiliser Slick2D, ça permet de générer certains graphismes et certaines options.

3.1.4.4 Native-linux, Native-mac et native-Windows
Ces libraires me permettent de définir sous quel Os je veux que mon programme s'exécute.

3.1.4.5 Slick
C'est la librairie principale de mon programme, c'est une librairie spécialement faite pour les jeux en 2D, elle gère la boucle du jeu, les graphismes, les sons, les vues etc…

3.1.4.6 Suite Spring
La suite Spring c'est des librairies qui me permettent de gérer l'encryptions de mon mot de passe et sa désencryptions.

3.1.4.7 Derby
Les librairies Derby c'est celle qui permettent de faire fonctionner ma base de donnée embarquée en gérant les requêtes SQL et les liens avec les différents drivers la faisant fonctionner.

3.2 Description des tests effectués

3.2.1 Différents tests :

3.2.1.1 Connexion à la base de donnée

Ce test permet de vérifier si la connexion à la base de donnée est possible.

Résultat :
Si la connexion est possible, ça return l'état de la base de donnée et met fin au test.
Si la connexion est impossible, ça lance une « SlickException » et met fin au test.

3.2.1.2 Insertion des différents niveaux

Ce test permet de vérifier si on peut insérer les différents niveaux dans la base de donnée (les niveaux faisant énormément de jonction entre plusieurs tables).

Résultat :
Si l'insertion est réussie, ça return un booléen TRUE et ça met fin au test.
Si l'insertion a échouée, ça return un booléen FALSE et ça met fin au test.

3.2.1.3 Vérification d'un test de connexion à la base de donnée coté utilisateur

Ce test permet de vérifier si un utilisateur arrive bien, à se connecter avec son compte.

Résultat :
Si la connexion est réussie, ça return un booléen TRUE et ça met fin au test.
Si la connexion a échoué, ça return un booléen FALSE et ça met fin au test.

3.2.1.4 Insertion des différents personnages existants

Ce test permet de vérifier si on peut insérer les différents personnages dans la base de donnée (Les personnage ayant beaucoup de jonction entre plusieurs tables).

Résultat :
Si l'insertion est réussie, ça return un booléen TRUE et ça met fin au test.
Si l'insertion a échouée, ça return un booléen FALSE et ça met fin au test.

3.3 Erreurs restantes

Ce chapitre définis les différentes erreurs qui reste dans mon projet qui fait que mon jeu n'est pas encore totalement jouable, ça contient aussi les améliorations futures que je vais devoir faire pour résoudre le problème.

3.3.1 Finir un combat

Pour l'instant, c'est possible de finir un combat mais ça a aucun effet sur le jeu.

Nous pouvons affronter des ennemies si on va dans le mode aventure, utilisé nos différents sorts et les battre et même que quand il y a plus d'ennemie devant nous, la page de fin d'un combat s'affiche.

Mais la gestion pour dire si le combat est fini dans la base de donnée n'est pas implémenter.

Pour les actions à suivre, je dois encore mettre à jour la base de donnée à la fin d'un combat, c'est ce qu'il me reste à faire pour cette erreur.

3.3.2 Enregistrer l'expérience gagné lors d'un combat

Lorsque le combat est fini, des points d'expériences sont attribués aux différents alliés pour les récompenser d'avoir finis le niveau et surtout pour les faire progresser dans le jeu en les faisant augmenter de niveau et de statistique.

La gestion de l'expérience n'est pas encore finie, ce qui inclue que la base de donnée est pas à jour selon l'expérience gagnée du personnage.
Le jeu actuellement a pas vraiment de progression direct. Nous pouvons juste combattre sans rien gagner.

Pour la suite, je vais juste continuer à créer le système de gain d'expérience et faire en sorte que notre allié puisse s'améliorer dans le jeu en augmentant son niveau.

3.3.3 Ajouter des niveaux

La gestion des niveaux est totalement finie, nous pouvons rentrer dans un niveau, et participer à un combat, la gestion des niveaux est en chapitre, nous avons trois chapitres avec une dizaine de niveau normalement ce qui n'est pas le cas actuellement.

Le chapitre est pas complètement rempli de niveau, le chapitre deux en contient qu'un seul et le chapitre trois n'est à aucun, ce qui veut dire que lorsqu'on va cliquer pour accéder au chapitre trois, le jeu va se fermer dû à une erreur dans le code.

Il faudrait juste que j'ajoute plus de niveau dans le jeu pour pallier à ce problème.

3.3.4 Ajouter la gestion des équipements

Dans mon cahier des charges, la gestion des équipements est un point qui revient et que je n'ai pas réussis à implémenter, durant le long de mon projet, j'ai focaliser mon temps et les implémentations que j'apportais à mon esprit ou à ma vision des choses. J'ai implémenté des choses logiques pour moi tel qu'un login et un register ou encore mis en place différentes vues qui n'était pas forcément demandé et je suis passé à côté de la gestion des équipements.

Je vais totalement devoir commencer la gestion des équipements et la faire fonctionner pour la suite du projet.

3.4 Liste des documents fournis

Comme demandé dans mon dans mon cahier des charges, les documents fournis et autres livrable sont les suivants :

- Une planification initiale (disponible plus haut dans le rapport)
- Un rapport de projet
- Un journal de travail (disponible plus bas dans le rapport)
- Le code source du jeu (fournis via un dépôt GitHub)
- L'exécutable/l'installateur du produit fini

4 Conclusions

4.1 Objectifs atteints ?

La plupart des objectifs que je m'étais fixé sont pas atteints, je pensais pouvoir gérer les fonctionnalités de mon jeu et les implémentés plutôt rapidement mais je n'ai pas vu le temps passé et j'ai dû rapidement rattraper ma documentation et laisser mon code de côté, ce qui m'a valu que mon jeu a pas atteints ce qu'il devait atteindre.

4.2 Points positifs / négatifs

Négatif :
Le jeu est très mal codé, ma vision du Java lors du début du projet et ma vision actuelle a beaucoup changé, je vois toutes les erreurs et les mauvaises implémentations que je faisais à l'époque et j'ai envie de les corriger.
Même si je suis encore très loin de voir ce que c'est qu'un code très bien codé en Java.

Il y a encore énormément de bug qui fait que le jeu est injouable et ça me déplait fortement, j'aimerais beaucoup pouvoir continuer et tous les corriger.

Positif :

Au début du projet j'avais un peu peur de pas savoir ce que je veux faire au niveau des graphismes, n'étant pas un grand dessinateur, je devais trouver des sprites et des fonds que je peux utiliser et je suis plutôt content de ce que j'ai pu trouver.

Ma compréhension du POO est en amélioration, j'avais aussi peur de mes lacunes au niveau de la POO et je suis content de voir que ça s'améliore.

Même si mon jeu ne fonctionne pas tout à fait, je trouve le système très amusant à coder et je suis assez content du résultat que ça pourra prendre une fois débugger.

4.3 Difficultés particulières

L'exportation du .exe, c'est un cauchemar, ça fonctionne une fois sur deux et il y a encore plein de chose que je n'arrive pas à comprendre avec ça. J'espère vraiment que ça va mieux aller dans la suite du projet.

4.4 Suite pour le projet

Il y a plein de fonctionnalité et de bug que je vais devoir corriger lors de la suite.

Je vais continuer ce projet lors du TPI qui commence le 3 mai 2021, et j'espère que ça se passera mieux sans trop de stress et que tout fonctionnera.

5 Annexes

5.1 Sources – Bibliographie – Acquisition des connaissances

Ce chapitre va contenir tous les liens et différentes sources qui m'ont permis d'apprendre à coder en Java et qui en renforcé les connaissances que j'avais en POO ainsi que les librairies que j'ai utilisées pour coder mon jeu.

- Apprendre a utilisé Slick2D : lien.
- Apprendre a utilisé Slick2D : lien.
- Apprendre à coder en Java : lien.
- Javadoc de Slick2D : lien.
- Mauro Santos, aide lors de question concernant la POO notamment sur l'héritage.
- Alexis Haldy, aide lors de quelques bugs complexe.

5.2 Journal de travail

Jour	Semaine	Temps [h]	Type	Description
30.mars	14	3.00	Documentation	Suite de la documentation du projet
29 .mars	14	3.00	Documentation	Suite de la documentation du projet
26.mars	13	3.00	Documentation	Suite de la documentation du projet
25.mars	13	3.00	Documentation	Suite de la documentation du projet
23.mars	13	3.00	Documentation	Début de la documentation du projet (Que j'aurais dû commencer dès le début du projet) avec création de l'introduction
22.mars	13	4.00	Documentation	Suite de la création de la fin des combats
19.mars	12	4.00	Implémentation	Début de l'affichage de fin des combats avec l'expérience gagnée

18.mars	12	4.00	Implémentation	Ajout des animations au combat et début d'implémentation d'animation de mort des personnages
16.mars	12	4.00	Implémentation	Suite de création du système de combat en faisant perdre des points de vies aux ennemis et aux alliés
15.mars	12	3.00	Implémentation	Création des barres de vies et début de création du système de combats
12.mars	11	4.00	Implémentation	Début d'implémentation des personnages sur l'écran des combats
11.mars	11	4.00	Implémentation	Création des boutons pour les levels et les faire fonctionner
09.mars	11	3.00	Implémentation	Suite d'ajout des levels que le joueur pourra jouer
08.mars	11	4.00	Implémentation	Mise à jour du MLD avec les levels
08.mars	11	4.00	Implémentation	Début d'ajout des levels que le joueur pourra jouer
05.mars	10	3.00	Implémentation	Création de bouton pour la vue + changement du code pour adapter à la classe Button + mise à jour du MLD
04.mars	10	4.00	Implémentation	Création de bouton pour la vue + rajout de la classe Entity qui fait hériter Character et Enemy
02.mars	10	4.00	Implémentation	Création de la bar de progression dans Character
01.mars	10	4.00	Implémentation	Continuer la vue Character en rajoutant les stats du Character actuel et sa description, création du bouton aventure et création des boutons Lobby, Character et inventaire
18.févr	8	3.00	Implémentation	Finir de setup le register et login
16.févr	8	4.00	Implémentation	Créer les conditions de connections et d'enregistrement
15.févr	8	4.00	Implémentation	Créer les views Login et Register
12.févr	7	4.00	Implémentation	Créer les classes JsonManager, JsonManager test et Account pour gérer les Accounts de utilisateurs .
11.févr	7	4.00	Implémentation	Créer les boutons "Play" et "exit" dans la view et résoudre des bugs
09.févr	7	2.00	Documentation	Apprendre à utiliser Slick2D
09.févr	7	2.00	Documentation	Commencer à gérer le GUI et les différents Controller dans le projet
08.févr	7	4.00	Documentation	Créer le MCD et MLD pour mon projet
05.févr	6	4.00	Documentation	Créer les maquettes pour les views
04.févr	6	1.00	Analyse	Apprendre à utiliser les Jframe
04.févr	6	3.00	Documentation	Prendre contact avec Mr. Viret pour le cahier des charges
02.févr	6	4.00	Analyse	Début d'un petit projet pour apprendre plus facilement le java
01.févr	6	3.00	Documentation	Installation d'environnement de travail
01.févr	6	1.00	Analyse	Acquisition de compétence en Java

5.3 Manuel d'Installation

Le manuel d’exportation du projet se trouve dans le dossier « ExportationProcess » qui se trouve dans le projet sous : Code-TimeRewind\Exportation Process\.

5.4 Archives du projet

Pour accéder à mon projet directement prêt à être utilisé, donc à son installeur, il faut accéder à ce lien et télécharger la dernière release disponible.