

|  |  |
| --- | --- |
|  | |
| 07.09.2017 | **Projet développement mobile - Roleplay**  **INF3dlm-a** |
|  |  |
|  | * Gabriel Griesser * Axel Bento da Silva |
|  |  |
|  | A destination de :   * Rizzotti Aïcha |

Abstract

Ce rapport décrit la réalisation du projet du cours Développement Mobile. Ce dernier consiste à créer un programme de notre choix, en Java, tournant sur un smartphone Android.

Une fois le groupe créé, nous avons réfléchi sur le type d’application que l’on pourrait faire. Étant les deux passionnés de jeu de rôle, nous nous sommes donc penchés sur un jeu de ce style.

Ce jeu est une application narrative dans laquelle l’utilisateur choisi sa propre destination en répondant aux questions.

Ce rapport décrira entre autres les recherches et les explications nécessaires à la bonne compréhension, la conception du programme, l’implémentation, le développement du jeu ainsi que l’utilisation de ce dernier.

Table des matières

[1. Introduction 4](#_Toc504568257)

[2. Analyse //A COMPLETER 4](#_Toc504568258)

[3. Conception 4](#_Toc504568259)

[3.1. Cahier des charges 5](#_Toc504568260)

[3.2. Diagramme de Gantt 5](#_Toc504568261)

[3.3. Scénario 5](#_Toc504568262)

[3.4. Modélisation 7](#_Toc504568263)

[3.4.1. Diagramme UML 7](#_Toc504568264)

[3.4.2. Convention de codage 7](#_Toc504568265)

[3.4.3. Use Case 8](#_Toc504568266)

[4. Implémentation //A COMPLETER 9](#_Toc504568267)

[4.1. Logiciel et bibliothèque 9](#_Toc504568268)

[4.1.1. Eclipse Java 9](#_Toc504568269)

[4.1.2. Swing 9](#_Toc504568270)

[4.2. Projets et classes 9](#_Toc504568271)

[5. Utilisation du jeu //A COMPLETER 9](#_Toc504568272)

[6. Problèmes rencontrés //A COMPLETER ? 10](#_Toc504568273)

[6.1. Incompréhension du stockage de données 10](#_Toc504568274)

[6.2. Accès bloqué aux données 10](#_Toc504568275)

[6.3. Problème lié aux émulateurs 10](#_Toc504568276)

[7. Améliorations futures 10](#_Toc504568277)

[8. Conclusion 11](#_Toc504568278)

[9. Sources 11](#_Toc504568279)

[10. Annexes 11](#_Toc504568280)

# Introduction

Pendant le 1er semestre de la dernière année, les étudiants de la HE-ARC ont du faire un projet, en Technologie d’intéraction. Si la création d’un nouveau programme ainsi que le langage sont choisis, le projet, lui, n’est pas imposé.

Le principe de base de l’application était de raconter une histoire à l’utilisateur, en le mettant à la place du héros, il doit faire des choix qui modifieront son chemin tout au long de l’histoire.

Ces différents choix permettent d’aborder l’histoire du héro sous un autre angle et permet donc d’augmenter la durée de vie de l’application.

Ce type de jeu narratif est un classique des jeux pc, mais n’étant plus à la mode lors de l’explosion des smartphones, il n’en existe que très peu sur le marché du mobile.

L’application est écrite en Java spécialisé développement Android. Parmi des choix de spécificités imposés, nous avons choisi les sauvegardes persistantes ainsi que l’utilisation de l’accéléromètre.

# Analyse

Pour ce projet Android, nous nous sommes penchés sur une utilisation simple mais efficace de ce que nous propose Android Studio et Java.

Avant de se lancer dans le projet, nous avons réfléchi à l’implémentation. Définir les objectifs était la priorité afin de pouvoir avancer sans se perdre en chemin. Nous avons donc fait le cahier des charges pour avoir une idée optimale du déroulement de notre jeu. Ce document décrit la façon dont notre projet verra le jour, les objectifs principaux et secondaires accompagnés de quelques détails. Nous nous sommes mis d’accord, avant d’attaquer le projet, sur une répartition des tâches, une convention de codage ainsi que certaines méthodes et manières d’aborder notre projet.

Ensuite, nous avons dessiné les différents schémas et diagramme nécessaires pour une meilleure réussite du projet. Ces schémas sont le diagramme de déroulement de l’application, le *use case* et le diagramme de Gantt.

# Conception

Cette partie est consacrée à la conception du projet. Premièrement, nous avons rédigé le cahier des charges et avons discuté des spécificités entre nous. Ensuite, après s’être réparti les différentes tâches du projet, nous nous sommes intéressés aux différents points chauds, puis avons réalisé un diagramme de Gantt pour avoir une meilleure visibilité du projet.

## Cahier des charges

Le cahier des charges défini clairement les objectifs à atteindre. Ces derniers sont détaillés dans la spécification du projet. Ces deux documents nous ont permis d’obtenir une meilleure définition du but, des objectifs et des détails concernant le projet.

Dans le cahier des charges, nous avons défini nos objectifs principaux, secondaires, nos points chauds, le déroulement de la partie ainsi que quelques informations supplémentaires. Il servira à quiconque le lit, d’avoir une idée globale du projet et des objectifs à atteindre. Nous avons pu y intégrer les objectifs principaux accompagnés de quelques objectifs optionnels (améliorations). Les différents points chauds (que nous avons défini ensemble) y sont également notés afin de mieux se rendre compte de la difficulté du projet.

## Diagramme de Gantt

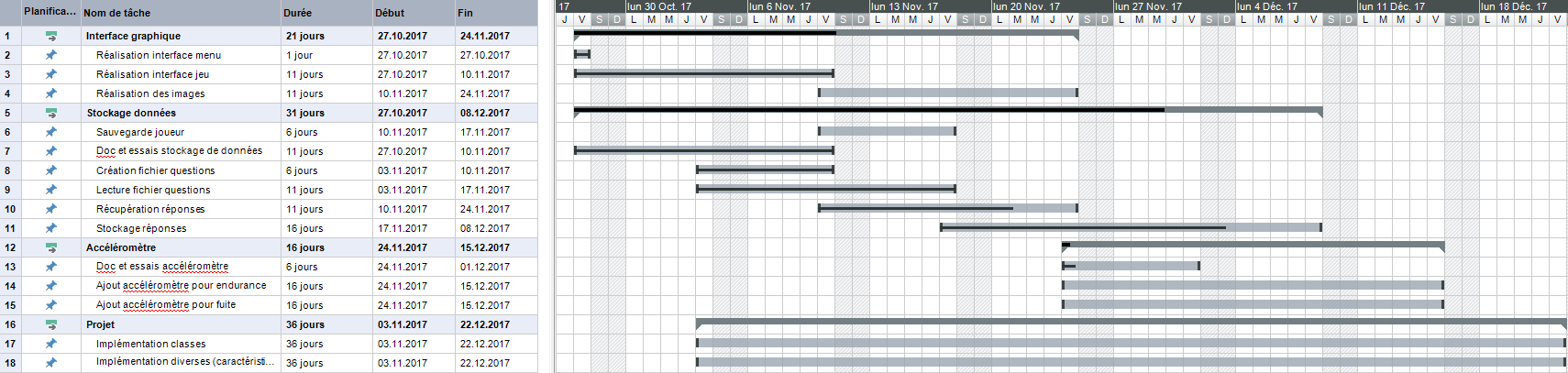
Le diagramme de Gantt est une manière de décrire, d’attribuer et de visualiser les différentes tâches nécessaires au bon déroulement au projet. Ce diagramme est à prendre avec des pincettes car les limites de temps données n’ont pas toujours été respectées.

Figure Diagramme de Gantt

## Scénario

Chaque partie de l’histoire est stockée dans un fichier séparé. Ce dernier est défini ainsi :



Figure , fichier de scénario

Au début du fichier se trouve le texte narratif, puis les lignes débutant par un « / » donne les nœuds vers les fichiers suivants. Entre les deux « / » se trouve le nom du fichier vers lequel la réponse pointe. La fin de la ligne est le choix qui sera présenté à l’utilisateur.

Il existe également des fichiers où le texte du scénario est remplacé par un mot-clé commençant par « \* », ces nœuds définissent des actions du héros comme se battre ou prendre la fuite.

Il existe un dernier type de fichier qui ne possède pas de nœuds vers lesquels se dirigé, puisqu’il s’agit des nœuds dans lesquels le héros meurt ou termine son histoire.

L’ensemble des fichiers de scénario représente le premier chapitre de l’histoire et est représenté par cet arbre :



Figure Arbre de scénario

Une histoire se suit par rapport aux lettres qui compose les fichiers de scénario, en sachant que A1 est le fichier de base de toutes les histoires. Par exemple

*A1, A2, A3, A4, A5, A6, A7, A*8 composent une histoire*. A1, A2, B1, B2, B3, B4* en font une autre.

Les fichiers commençant par D (*death*) indiquent que le personnage meurt s’il atteint un de ces nœuds.

## Modélisation

### Déroulement de l’application

Ce diagramme, bien que simple, nous permet d’avoir une vision globale du déroulement de l’application.



Figure Diagramme déroulement application

### Convention de codage

* Les noms des classes sont en anglais, commencent par une majuscule et peuvent contenir une ou plusieurs majuscule 🡪 *FileHandler*
* Les noms des attributs et objets commencent par une minuscule et peuvent contenir une majuscule. Ils sont écrits en anglais 🡪 *attributeChanged*
* Les noms des méthodes commencent par une minuscule et peuvent contenir une ou plusieurs majuscules. Ils sont écrits en anglais 🡪 *savePlayer()*

### Use Case

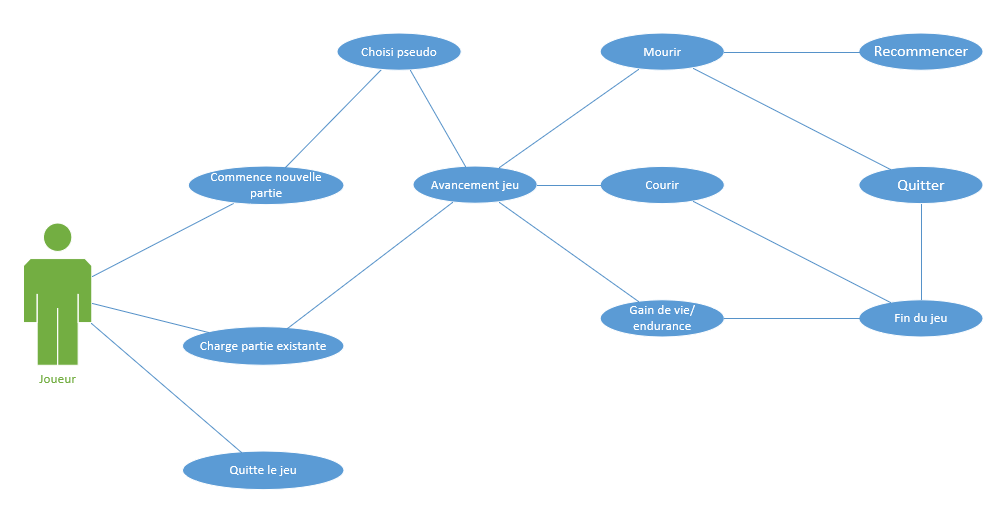
Le *use case*, ou diagramme de cas d’utilisation, défini la manière d’utiliser le programme et les différentes actions possibles.

Figure Use case

# Implémentation

Cette partie du rapport explique l’implémentation du projet. Nous détaillerons donc la façon d’atteindre le but principal de ce projet, les logiciels et bibliothèques utilisés, puis les différentes parties du programme, leurs spécificités et le rôle de chacune.

## Logiciel et bibliothèque

Comme durant les cours nous utilisons android studio pour le développement de l’application, qui nous permets d’utiliser les bibliothèques android ainsi qu’un émulateur sur lequel nous pouvons tester et débuguer notre application.

## Projets et classes

### Menu Activity

Menu activity est le menu de base dans lequel est disposé trois bouton, New Game, Load et Exit, qui permettent respectivement de lancer **IntroductionActivity**, de lancer **LoadActivity** et finalement de quitter l’application.

### Introduction Activity

Ce menu permet d’entrer un pseudo qui permettra de créer le fichier de sauvegarde à travers **FileHandler**, qui vérifiera également que le pseudo n’est pas déjà utilisé. Une fois un pseudo valide entrée, **GameActivity** est lancé.

### Game Activity

C’est dans cette activité que le jeu se déroule, elle fera appel au **FileHandler** pour charger les nœuds de l’histoire, puis les afficheras, elle gère également le choix des utilisateurs, la réception de bonus. Lorsqu’un évènement de fuite apparait dans un nœud elle utilisera l’objet **Accelerometor** pour calculer la distance parcouru par l’utilisateur, durant un certain temps.

Si cette activité est mise en pause, elle sauvegardera automatiquement l’avancement du joueur.

### Load Activity

Cette activité va utiliser **FileHandler**, pour récupérer la liste des sauvegardes et les affiche dans une listeView. Lorsque l’utilisateur clic sur l’un des choix, le fichier correspondant sera chargé et **GameActivity** sera lancé et démarrera directement au nœud correspondant au fichier de sauvegarde.

### FileHandler

Cette classe contient l’ensemble des fonctions permettant de gérer les fichiers de l’application, notamment :

* L’ouverture des fichiers de nœuds de l’histoire
* Vérifier qu’un pseudo n’est pas déjà utilisé
* Sauvegarder l’avancement d’une partie
* Charger une partie
* Récupérer la liste des sauvegarde

# Utilisation du jeu

Une fois le jeu lancé, l’utilisateur arrive dans le menu principal dans lequel plusieurs choix se présentent à lui. Nouvelle partie, charger partie, quitter.

## Nouvelle partie

Pour lancer une nouvelle partie vous devez entrer un pseudo, qui servira de nom au fichier de sauvegarde. Ce nom doit donc être unique et permettra au joueur de commencer une nouvelle partie.

Une fois la partie lancée, le principe est le suivant : vous devrez lire l’histoire qui vous est présentée et, à chaque « partie » d’histoire, faire un choix parmi les propositions données. Chaque choix sera déterminant quant à la suite de l’histoire. Vous devrez donc faire une suite de choix qui pourront vous amener à des combats ou des besoins de fuir et cela jusqu’à ce que votre personnage décède ou atteigne une fin d’histoire.

Une fois le jeu finie (ce dernier ne comporte qu’un seul chapitre), il est possible de relancer une partie pour essayer de découvrir les autres parties de l’histoire.

## Charger une partie

Lorsque vous décidez de charger une nouvelle partie, une liste comportant les différentes sauvegardes, avec le nom du pseudo, sera affichée. Il suffira de cliquer sur la sauvegarde voulue pour la charger.

# Problèmes rencontrés

Durant le déroulement du projet, nous avons rencontrés quelques problèmes, qui, pour la plupart, ont été résolus.

## Incompréhension du stockage de données

Une erreur que nous avons fait lors de la conception, a été de ne pas spécifié le type de stockage de données, un membre du groupe voulais utiliser *l’internal storage* et l’autre pensais que l’objectif demandé était de stocker les données sur un support externe (*sdCard*), après avoir rencontré le problème, nous nous sommes mis d’accord sur l’utilisation de fichier interne.

## Accès bloqué aux données

Un problème rencontré durant la création du *FileHandler* (gestionnaire de sauvegarde et de lecture des fichiers), est que nous n’avions pas accès au dossier data de l’émulateur à travers « *Android Device Monitor* », ce qui nous posait des problèmes de débogages, notamment pour savoir si les fichiers de sauvegardes étaient créés.

Après quelques recherches nous avons compris qu’il s’agissait d’un problème de droit de lecture/écriture, que nous avons tenté de modifier, ce qui nécessitait plusieurs opérations notamment le routage de l’appareil, cette direction nous a fait perdre beaucoup de temps, pour au final ne pas apporter de résultats concrets.

Nous avons heureusement trouvé une autre solution temporaire, qui était de ne plus utiliser l’appareil de test fournit par l’école, mais d’utiliser un émulateur avec un niveau d’API supérieur ou égal à 23, qui n’ont pas ces problèmes de droits.

Ce qui nous à mener au problème suivant.

## Problème lié aux émulateurs

Nous avons eu de nombreux problèmes liés aux émulateurs, crash au démarrage, freeze alors que l’application n’est même pas lancée, les *error* 0x502 que nous n’avons pas pu corriger. Tous ces problèmes ont passablement ralenti, la partie sauvegarde et chargement de fichier.

# Améliorations futures

## Une représentation de combat

Pour l’instant les combats sont faits instantanément pour l’utilisateur et il ne reçoit que les résultats (s’il a vaincu l’ennemi, s’il a perdu de la vie ou s’il est mort). Cette représentation pourrait se faire sous la forme d’une animation (voir 7.2.)

## Mise en place des animations

La mise en place des animations bien quelles couperait une partie du jeu qui consiste à imaginer le monde dans lequel l’histoire se déroule, permettrait d’ajouter une idée de contenu qui parlerait à un public plus large.

## Ajout de chapitre

L’histoire étant très complexe à écrire et que peu importante pour un projet scolaire, il n’y a que très peu de nœuds dans le scénario, mais en ajouter permettrait de prolongé la vie du jeu.

# Conclusion

Le cahier des charges a été quasiment rempli, à l’exception des animations qui ont été remplacé par de simple images, pour aider le joueur à la compréhension du jeu (exemple : réception d’un bonus).

# Sources

Documentation java : <https://docs.oracle.com/javase/7/docs/api/>

Documentation Swing : <https://www.jmdoudoux.fr/java/dej/chap-swing.htm>

Diverses aides : <https://stackoverflow.com/>

# Annexes

Ci-joint :

* Cahier des charges
* Spécification du projet
* Diagramme UML des classes
* Diagramme de séquence
* Planning
* Cas d’utilisation
* Code source du jeu
* Exécutables sous formes de .jar