



07.09.2017

Projet développement mobile - Roleplay  
INF3dIm-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

<b>1. INTRODUCTION.....</b>	<b>4</b>
<b>2. ANALYSE .....</b>	<b>4</b>
<b>3. CONCEPTION.....</b>	<b>4</b>
3.1. CAHIER DES CHARGES.....	5
3.2. DIAGRAMME DE GANTT .....	5
3.3. SCÉNARIO.....	6
3.4. MODÉLISATION .....	7
3.4.1. Déroulement de l'application.....	7
3.4.2. Convention de codage.....	8
3.4.3. Use Case.....	8
<b>4. IMPLÉMENTATION .....</b>	<b>9</b>
4.1. LOGICIEL ET BIBLIOTHÈQUE .....	9
4.2. PROJETS ET CLASSES .....	9
4.2.1. Menu Activity.....	9
4.2.2. Introduction Activity.....	9
4.2.3. Game Activity.....	9
4.2.4. Load Activity.....	9
4.2.5. FileHandler .....	10
<b>5. UTILISATION DU JEU .....</b>	<b>10</b>
5.1. NOUVELLE PARTIE .....	11
5.2. CHARGER UNE PARTIE .....	11
<b>6. PROBLÈMES RENCONTRÉS .....</b>	<b>12</b>
6.1. INCOMPRÉHENSION DU STOCKAGE DE DONNÉES .....	12
6.2. ACCÈS BLOQUÉ AUX DONNÉES .....	12
6.3. PROBLÈME LIÉ AUX ÉMULATEURS .....	12
<b>7. AMÉLIORATIONS FUTURES.....</b>	<b>12</b>
7.1. AJOUT DE CHAPITRE.....	13
7.2. MISE EN PLACE D'ANIMATIONS .....	13
7.3. AMÉLIORATIONS SPÉCIFIQUES.....	13
<b>8. CONCLUSION.....</b>	<b>13</b>
<b>9. SOURCES.....</b>	<b>14</b>
<b>10. ANNEXES.....</b>	<b>14</b>

## 1. Introduction

---

Pendant le 1<sup>er</sup> semestre de la dernière année, les étudiants de la HE-ARC. 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 héros 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.

## 2. 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.

## 3. 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.

### 3.1. Cahier des charges

Le cahier des charges définit 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.

### 3.2. Diagramme de Gantt

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

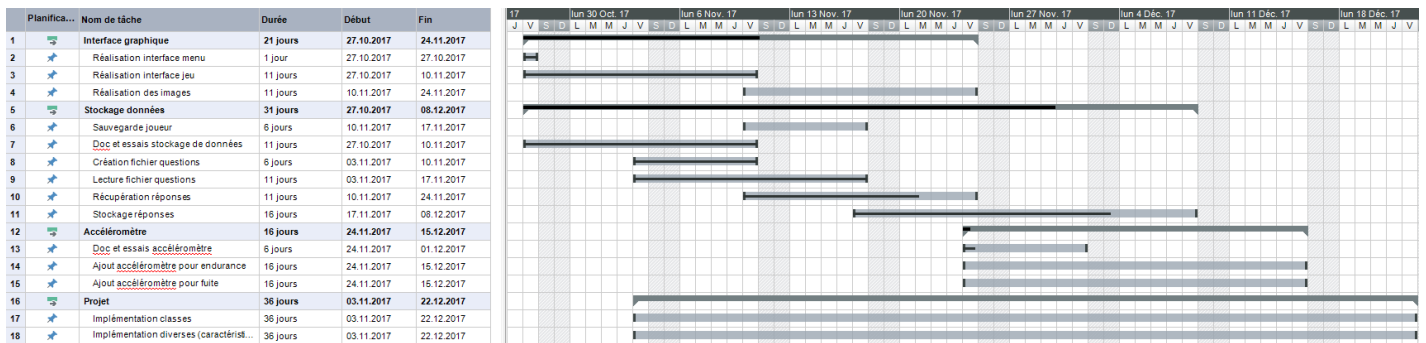
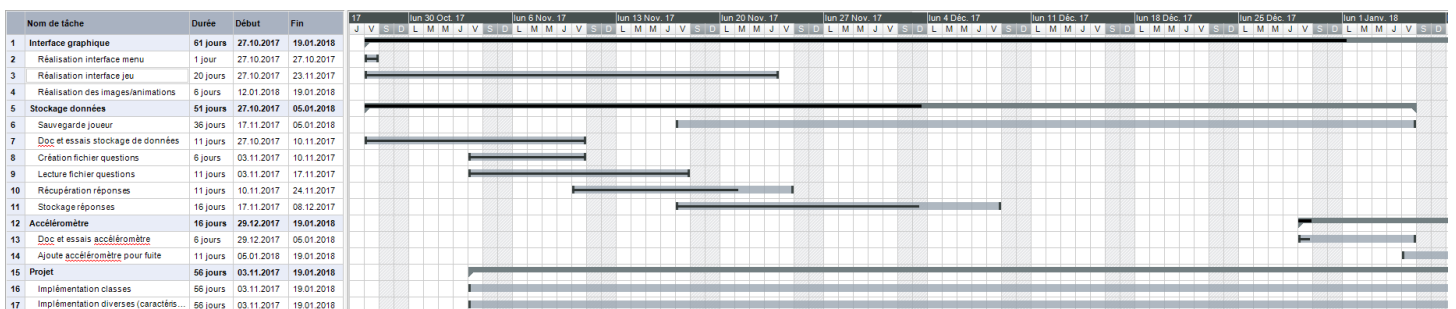


Figure 1 Diagramme de Gantt

Voici le diagramme de Gantt après une mise-à-jour.



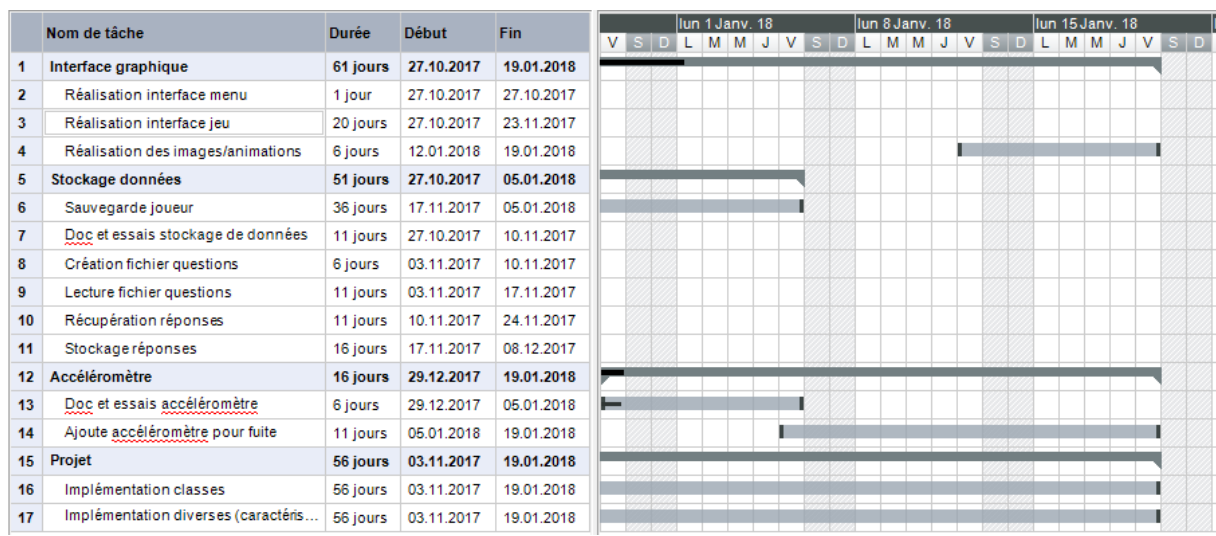


Figure 2, Planning maj

### 3.3. Scénario

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

```

1 Vous décidez de vous faire le tour de la pièce.
2 /A2/ S'approcher de la porte en fer.
3 /D1/ Regarder dans le fond de la salle.
```

Figure 3, 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 :

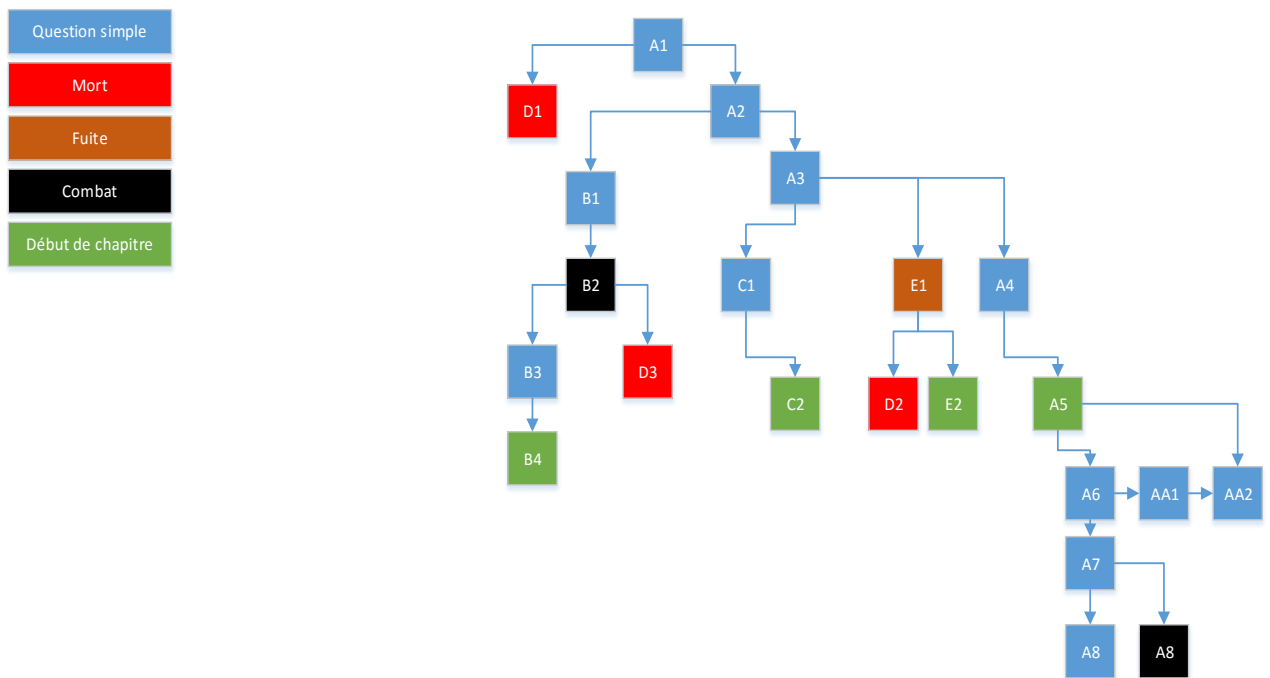


Figure 4 Arbres de scénario

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, A8 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.

### 3.4. Modélisation

#### 3.4.1. Déroulement de l'application

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

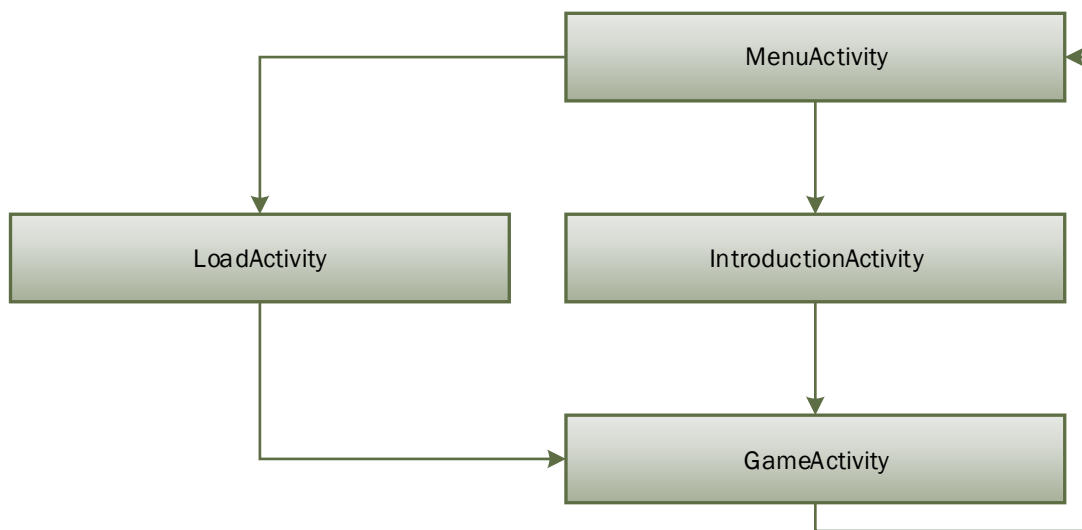


Figure 5 Diagramme déroulement application

### 3.4.2. 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()*

### 3.4.3. Use Case

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

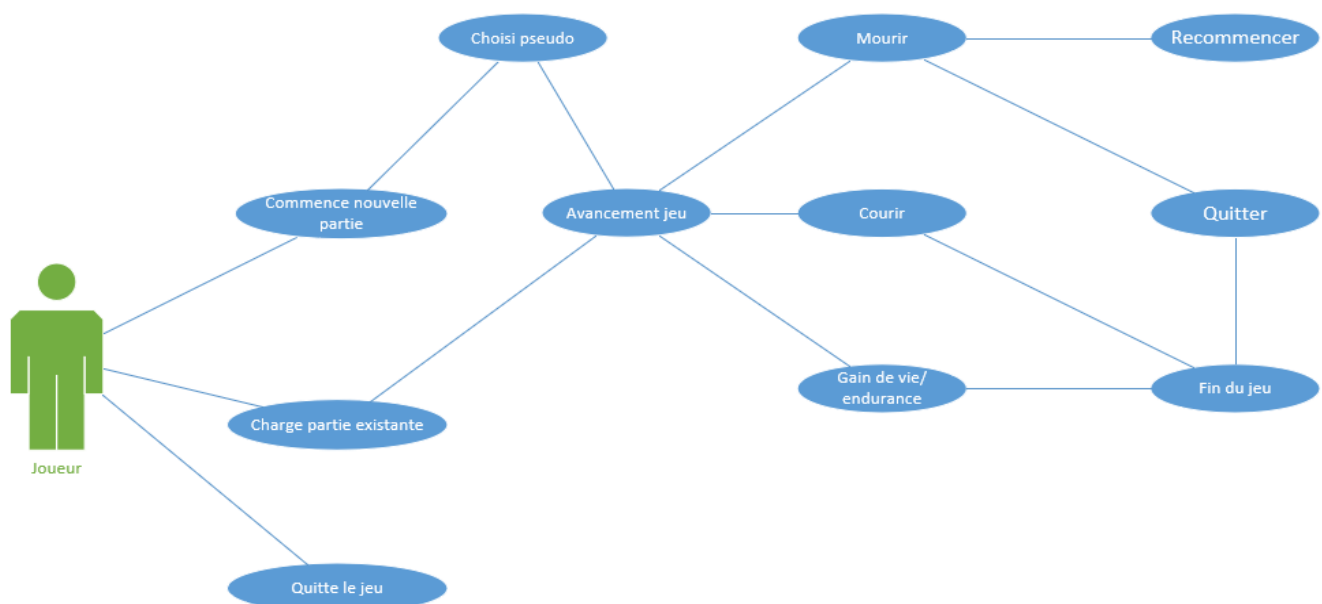


Figure 6 Use case



## 4. 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.

### 4.1. Logiciel et bibliothèque

Comme durant les cours nous utilisons Android Studio pour le développement de l'application, qui nous permet d'utiliser les bibliothèques Android ainsi qu'un émulateur sur lequel nous pouvons tester et déboguer notre application.

### 4.2. Projets et classes

#### 4.2.1. Menu Activity

*MenuActivity* est l'*activity* sur laquelle nous arrivons en lançant le jeu. Trois boutons sont alors disponibles : *New Game*, *Load* et *Exit*, qui permettent respectivement de lancer **IntroductionActivity**, **LoadActivity** ou finalement de quitter l'application.

#### 4.2.2. Introduction Activity

Ce menu sert de liaison entre le menu d'accueil et le jeu. L'utilisateur devra entrer un pseudo qui servira à la création du fichier de sauvegarde à travers **FileHandler**. Une fois un pseudo valide entré, l'*activity* **GameActivity** est lancée.

#### 4.2.3. Game Activity

C'est dans cette *activity* que le jeu se déroule, elle fera appel à **FileHandler** pour charger les nœuds de l'histoire, puis les affichera. Dans cette classe est également géré le choix des utilisateurs et la réception de bonus. Lorsqu'un événement de fuite apparaît dans un nœud elle utilisera l'objet **Accelerometor** pour calculer la distance parcourue par l'utilisateur, durant un certain temps.

Si cette *activity* est mise en pause, elle sauvegardera automatiquement l'avancement du joueur.

#### 4.2.4. Load Activity

Cette *activity*, couplée à **FileHandler**, permettra la récupération des sauvegardes et leurs affichages dans une *listView*. Lorsque l'utilisateur choisit un pseudo, le fichier correspondant sera alors chargé, **GameActivity** sera lancé et démarrera directement au nœud correspondant au fichier de sauvegarde.

#### 4.2.5. FileHandler

Cette classe contient l'ensemble des fonctions permettant la gestion des 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

### 5. 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.

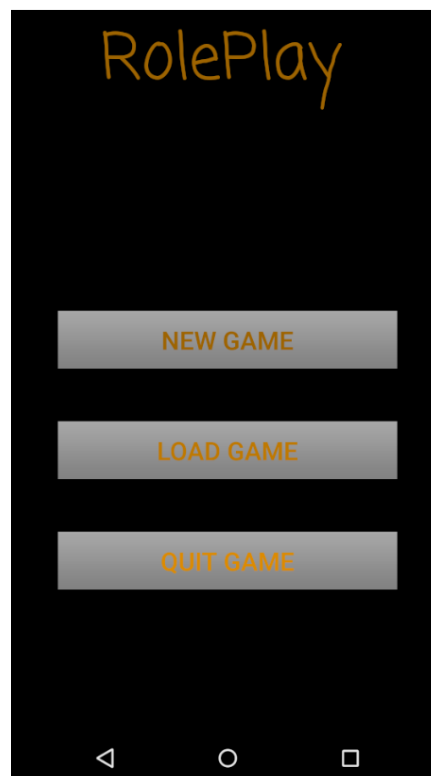


Figure 7, menu du jeu

### 5.1. 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.

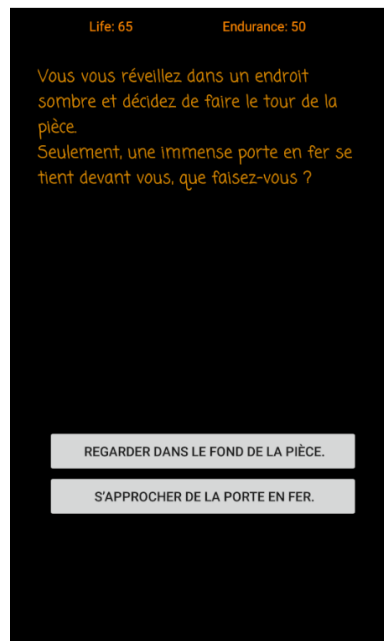


Figure 8, déroulement du jeu

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 fini (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.

### 5.2. 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.

## 6. Problèmes rencontrés

---

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

### 6.1. 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.

### 6.2. 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 bel et bien 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. Cette modification nécessita plusieurs opérations notamment le routage de l'appareil. Cela 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 mène au problème suivant.

### 6.3. 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.

## 7. Améliorations futures

---

Malgré notre cahier des charges pratiquement rempli, le jeu peut encore être amélioré de diverses manières.

Certaines améliorations sont « essentielles » pour une meilleure immersion, d'autres ne le sont pas forcément.

### 7.1. Ajout de chapitre

Nous pourrions, par exemple, ajouter plusieurs chapitres. En effet, le principe même de création de fichiers de narration n'est pas compliqué. L'utilisateur peut, quand il le souhaite, créer d'autres fichiers et les lier aux précédents afin de modéliser son histoire comme il le souhaite. Bien sûr, l'ajout de fichiers de narration signifie également un ajout de propriétés (vie/endurance) ou d'actions (courir/combattre).

### 7.2. Mise en place d'animations

La mise en place des animations, bien qu'elles couperaient un peu l'immersion, permettrait d'ajouter une idée de contenu qui parlerait à un public plus large. Le seul réel « problème » est de trouver les animations, ou de les créer nous-même.

### 7.3. Améliorations spécifiques

Certaines améliorations concernant les actions peuvent-être réalisées, comme l'utilisation de la géolocalisation pendant une fuite afin d'empêcher toute triche de la part de l'utilisateur. Cette même géolocalisation, toujours couplée à l'accéléromètre, pourrait également servir d'atout si l'utilisateur se déplace tout en jouant. Cela pourrait, par exemple, augmenter son endurance de 1 tous les 5 mètres.

Nous avons également pensé à une amélioration assez technique concernant les combats. Si le joueur décide de combattre, nous pourrions, au lieu d'utiliser une formule calculant la probabilité de gain/perte, utiliser un capteur de mouvement pour obliger le joueur à manier son téléphone comme une épée afin de battre l'ennemi.

Toujours avec ce capteur de mouvement, il serait possible d'ajouter une action *freeze* qui obligerait le joueur à ne pas bouger un certain laps de temps (pour se cacher par exemple).

Un menu pause ne nous semblait pas urgent car ce genre de jeu est assez « lent ». L'utilisateur peut très bien ne pas faire de choix pendant plusieurs heures, cela n'affectera en rien son épopée.

## 8. Conclusion

---

Pour conclure ce rapport sur notre projet informatique, nous pouvons dire que les objectifs principaux ont été atteints.

Après avoir choisi ce projet, nous nous sommes focalisés sur un planning ainsi que diverses « aides » quant à sa réalisation (diagramme, use case, cahier des charges). Suite à cela, nous avons identifié les points chauds afin de pouvoir les anticiper correctement. Nous nous sommes beaucoup entretenus sur comment aborder le projet sans se perdre en cours de route.

Une fois fait, nous avons directement attaqué, en parallèle, l'interface graphique du jeu ainsi que les fichiers contenant le scénario. Il nous semblait important de réaliser ces deux tâches avant autre chose afin de commencer sur une bonne base.

Ensuite, nous avons pu continuer l'implémentation avec un maximum d'informations cohérentes ce qui a permis de réduire les risques.

L'implémentation s'est faite au plus proche de ce qui avait été convenu oralement. Le travail était régulièrement mis à jour par chaque membre du groupe en *pushant* nos modifications sur le répertoire Git du projet.

Nous nous sommes ensuite focalisés sur le chargement/sauvegarde des fichiers. Cette tâche s'est révélée plus difficile que nous le pensions et a énormément ralenti notre avancement du projet. Parallèlement à ça, nous avons commencé à utiliser l'accéléromètre et à implémenter quelques propriétés indispensables (icônes, combat, fuite, ajout endurance/vie, etc...).

L'interface graphique est, quant à elle, ergonomique et sobre. Nous avons préféré laisser part à l'imagination de l'utilisateur tout au long de l'histoire (qui est encore courte).

Le projet a été conçu de façon à encourager l'utilisateur à créer sa propre histoire. Il peut, sans difficulté, ajouter de nouveaux fichiers narratifs à ceux déjà existant. Si certaines améliorations sont faites sur ce projet, ce dernier ne pourra qu'être plus immersif et encore plus encourager l'utilisateur à explorer toutes les faces de l'histoire.

## 9. Sources

---

Documentation Android: <https://developer.android.com/index.html>

Android OpenClassroom: <https://openclassrooms.com/courses/creez-des-applications-pour-android>

Icônes: <https://opengameart.org/>

## 10. Annexes

---

Ci-joint :

- Cahier des charges
- Diagramme de déroulement
- Planning
- Cas d'utilisation
- Code source du jeu
- Présentation