

# Projet libre

mbourlet, gfernand, fbuoro, mchoong #e1r6

# Chapitre 1

## Présentation du projet

Le but du projet est de réaliser un jeu vidéo natif sur Android. Ce jeu sera un RPG au combat tour par tour mais avec un profond aspect stratégique, le joueur ne pourra pas choisir ses actions à chaque tour, il devra à la place programmer une IA (voir plus loin). Le projet se décomposera en deux parties : l'engine et le jeu

### 1.1 Engine

L'engine sera un moteur basique pour le jeu, il devra être réutilisable sur différents projets. Son but sera d'accélérer le développement futur et actuel du jeu. Il se divise en plusieurs Managers.

#### 1.1.1 Game

La classe Game est la classe centrale de l'engine. C'est elle qui gère la logique du programme avec son système de state. Au lancement la classe récupérera la sauvegarde la plus récente avant de lancer le jeu.

#### 1.1.2 Le DataManager

Le DataManager est celui qui gère les sauvegardes de l'utilisateur. Il offre deux possibilités :

- Sauvegarder en local : Le fichier est stocké sur le téléphone de l'utilisateur dans les fichiers de l'application.
- Sauvegarder sur le cloud : Si l'utilisateur se connecte avec son compte Google+ et nous en donne l'autorisation, le DataManager sauvegarde sur le cloud.

### **1.1.3 State**

Les states sont les différents états du jeu, ce sont elles qui permettront de gérer la navigation dans l'application.

## **1.2 Projet**

Le projet en lui même utilisera l'engine. Il devrait utiliser un maximum d'API offertes par Google, tel que le stockage sur le Cloud, le multi-joueur, les hauts-faits etc.

## Chapitre 2

# Systeme d'IA

Le joueur pourra contrôler une équipe de 3 personnages.  
Chaque personnage pourra avoir de nombreuses classes (Guerrier, Mage, Soigneur)

Chaque classe de chaque personnage aura son propre niveau.

Quand un personnage augmente de niveau dans une classe, il gagne des outils pour programmer son IA liée à cette classe, ces outils peuvent être décomposés en plusieurs types :

- Condition : Permet de conditionner le comportement de l'IA
- Action : Cela peut être des sorts, des attaques, etc...
- Slot : Permet de poser une nouvelle paire de conditions/actions

### 2.1 Explication du système de classe

Chaque personnage possède plusieurs classes, le joueur peut décider de changer la classe de ses personnages pendant un combat, permettant de faire varier les stratégies en fonction de l'adversaire (phase de rage, stratégie spéciale liée à un adversaire, etc...)

En fonction de la fréquence d'utilisation d'une classe dans un combat, celle-ci gagne plus ou moins de pourcentage de la somme d'expérience donnée lors de ce combat.

## Chapitre 3

# Planning de réalisation

Le planning du projet sera géré de manière agile (Sprint chaque semaine + réunion tous les jours de 15min pour faire un bilan rapide).

Cependant, nous allons réaliser plusieurs itérations du projet. Chaque itération aura pour but de rendre un produit fonctionnel, consistant en une amélioration du précédent :

- Itération 1 : Réaliser un combat entre deux IA basique (Alliée programmable / Ennemie)
- Itération 2 : Réaliser un système basique de classe
- Itération 3 : Réaliser un système de niveau + inventaire
- Itération 4 : Réaliser un début de campagne / quête basique
- Itération 5 : Ajout d'IA ennemie
- Itération 6 : Amélioration du système d'histoire / quête (branche)

A la fin de ces itérations, nous pourrions considérer que la partie obligatoire est finie. Nous prévoyons de réaliser quelques bonus, classés par ordre de difficultés :

- Ajout des achievements
- Beaucoup d'items
- Plus de classes
- Ladder solo
- Multijoueur
- Ladder multijoueur
- Animation entière du personnage

## Chapitre 4

# Design

Le jeu sera réalisé en pixel art. Voici un exemple de style illustré dans un premier exemple :



## Chapitre 5

# Communication

La communication se fera via un Website/devblog et sera appuyée par un compte twitter qui détaillera l'avancement du jeu en postant des photos, vidéos de gameplay etc. Le but étant d'attirer le plus de personnes à s'intéresser au projet avant même la sortie du jeu. C'est notamment pour cela qu'une phase de beta test sera disponible aux utilisateurs les plus intéressés.

La beta test sera gérée à l'aide de la google developer console. Le site proposera également l'inscription à une newsletter.

## Chapitre 6

# Monétisation

La monétisation du jeu se fera via les achats in-app et les pubs.  
Les achats in-app ne devront pas bousculer l'équilibre du jeu, notamment dans l'optique d'un multijoueur le plus stratégique possible.  
Nous prévoyons pour le moment l'achat de packs d'xp qui permettraient de level up plus vite.  
L'intégration des pubs se voudront non-intrusive pour ne pas gâcher l'expérience utilisateur. Elles pourront être par exemple intégrées sous forme de bannière à l'écran de score d'un combat. Elles ne devront pas gêner la navigation. Nous utiliserons des API tel qu'admob (utilisé par Rovio, Backflip Studio, Fingersoft... )pour intégrer des pubs ciblées et donc augmenter nos revenus.



## Chapitre 7

# Elements à vérifier en fin de projet

Il faudra vérifier que les différentes itérations fonctionnent correctement :

- Itération 1 : Réaliser un combat entre deux IA basique (Allie (programmable) / Ennemie)
- Itération 2 : Réaliser un système basique de classe
- Itération 3 : Réaliser un système de niveau + inventaire
- Itération 4 : Réaliser un système d'histoire / quête basique
- Itération 5 : Ajout d'IA ennemie
- Itération 6 : Amélioration du système d'histoire / quête (branche)

## Chapitre 8

# Bonus

- Ajout des achievements
- Beaucoup d'items
- Plus de classes
- Ladder solo
- Multijoueur
- Ladder multijoueur
- Animation entière du personnage