

Open Project

mbourlet, gfernand, fbuoro, mchoong #e1r6

Chapitre 1

Project presentation

The goal of our project is to achieve a native video game on Android. This game will be a battle RPG turn by turn but with a profound strategical aspect, the player can not choose his actions at each turn, instead he will have to set a basic IA (see below).

The project will be split into two parts : the engine and the game.

1.1 Engine

The engine will be a basic engine for the game, it will have to be reusable on different projects. Its purpose will be to accelerate the future and current development of the game. It is divided into several Manager.

1.1.1 Game

Game class is the central class of the engine. He's the one who manage the program logic with his State System.

At launch the class tries to automatically connect to the google api for user connects and takes advantage of the achievements, the backup to the cloud etc. The user may refuse the connection. If he refuses the connection three times at launch, the program will no longer launch the automatic connection. This is in stock in a given cache file.

Before starting the game, the game class will attempt to load a backup via the DataManager.

1.1.2 Le DataManager

The DataManager is the one who manages the backups of the user. It offers two possibilities :

- Local Save : The file is stored on the user's phone in the application files.

- Cloud Save : If the user logs in with a Google+ account and gives us the authorization, the DataManager also backup to the cloud

The work of the DataManager is to facilitate the backup spots and loading of data . It choose the correct backup according to a timestamp adding to the file and by managing conflicts if necessary. When saving he save either locally or in locally + cloud connection according to the Google API connection state.

1.1.3 State

Les states sont les différents tats du jeu, ce sont elles qui permettront de gérer la navigation dans l'application. La game comprends une stack de State. Imaginons par exemple que l'utilisateur soit dans le menu principal, c'est donc la state liée a ce menu qui recevra toutes les actions de l'utilisateur. Si il lance une partie, la state de la partie se retrouvera en haut de la stack recevant ainsi toutes les instructions de la game, c'est elle qui pourra dessiner et recevoir les inputs de l'utilisateur.

Si l'utilisateur veut revenir a l'état precedent qui est donc le menu principal nous n'avons plus qu'à dépiler la state de la partie afin que la state du menu reprenne le contrôle sur le dessin et les inputs.

1.2 Projet

Le projet en lui même utiliseras l'engine. Il devras utilisé toutes les technologies offerte par Google, tel que le stockage sur le Cloud, le multi-joueur, les hauts-faits etc.

Chapitre 2

Systeme d'IA

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

Chaque classe de chaque personnage auras son propre niveau.

Quand un personnage augmente de niveau dans une classe, il gagne des outils pour programmer son IA lié a cette classe, ces outils peuvent être décomposer 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 Classe ?

Chaque personnage possède plusieurs classes, le joueur peut decider de changer la classes de ces personnages pendant un combat, permettant de faire varier les stratégies en fonction de l'adversaire (phase de rage, stratégie speciale liée a 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'experience donné par ce combat.

Chapitre 3

Planning de réalisation

Le planning du projet sera g r  de fa on agile (Sprint chaque semaine + r union tout les jours de 15min pour faire un bilan rapide).

Cependant, nous allons r aliser plusieurs it ration du projet. Chaque it ration aura pour but de rendre un produit fonctionnel, consistant en une am lioration du pr c dent.

- Iteration 1 : R aliser un combat entre deux IA basique (Allie (programmable) / Enemie)
- Iteration 2 : R aliser un systeme basique de classe
- Iteration 3 : R aliser un systeme de niveau + inventaire
- Iteration 4 : R aliser un systeme d'histoire / quete basique
- Iteration 5 : Ajout d'IA enemie
- Iteration 6 : Am lioration du systeme d'histoire / quete (branche)

A la fin de ces iterations, nous pourrons considerer que la partie obligatoire est finies. Nous prvoyons au cas ou de r aliser quelque bonus plus ou moins dur, pouvant tre r aliser en plus :

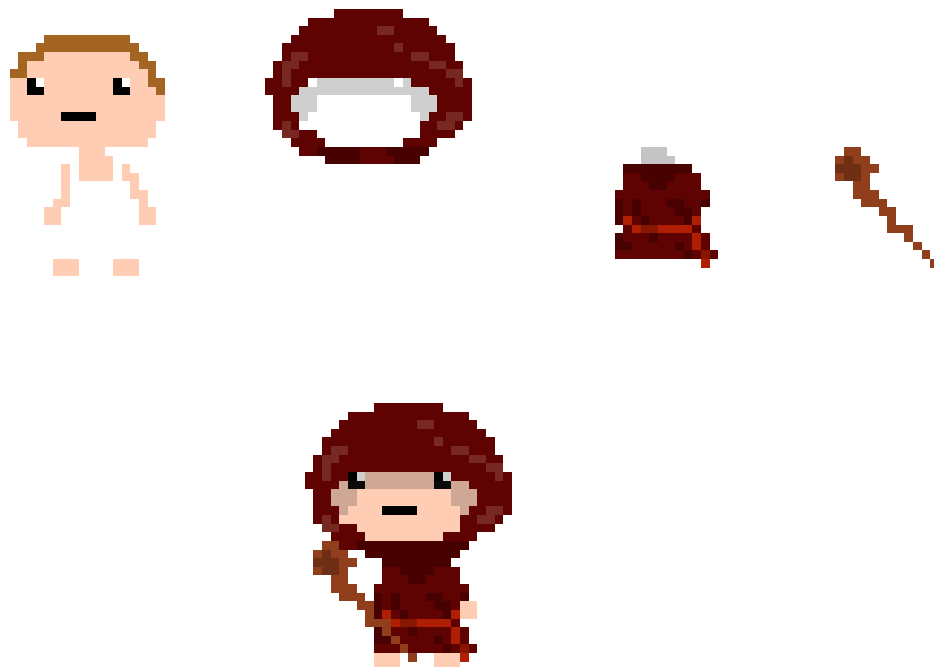
- Ajout des achievements
- Beaucoup d'item
- Plus de classes
- Ladder solo
- Multijoueur
- Ladder multijoueur
- Animation entiere du personnage

Chapitre 4

Design

Le jeu sera réalisé en pixel art. Chaque partie du corps des personnages et chaque item seront en fait différents calques que nous superposeront pour créer les images finales.

Cela nous permet de réaliser rapidement un grand nombre de sets de personnages et de rendre nos personnages principaux facilement personnalisables. Voici un exemple de nos premières réalisations :



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'intresser 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 permettrait de lvl up plus vite.

Les 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 comme 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

- Un système d'IA fonctionnel (voir chapitre Système d'IA)
- Une interface agréable
- Un mode solo fonctionnel
- Un multijoueur fonctionnel
- La présence d'achievements

Chapitre 8

Bonus

— Une monétisation bien intégrée