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 seras 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 á la place programmer une IA plus ou moins basique (voir plus loin).
Le projet se décomposeras en deux parties : l'engine et le jeu

1.1 Engine

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

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 son système de state.
Au lancement la classe tente de se connecter automatiquement à l'api google pour que l'utilisateur se connecte et profite des achievements, de la sauvegarde sur le cloud etc.
L'utilisateur peut refuser la connection. Si il refuse la connection 3 fois de suite au lancement, le programme ne lancera plus la connection automatique. Cette donnée est stockée dans un fichier en cache.
Avant de lancer le jeu, la classe game va tenter de charger une sauvegarde via la DataManager.

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 telephone 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 également sur le cloud.

Le travail du DataManager est de faciliter les taches de sauvegarde et de chargement des données. Il permet lors du chargement de choisir automatiquement la bonne sauvegarde selon un timestamp ajouté au fichier et en gérant les conflits si nécessaire. Lors de la sauvegarde il s'occupe de sauvegarder soit en local soit en local + cloud selon l'état de connexion à l'API google.

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

Systéme d'IA

Le joueur pourras contrôler une équipe de 3 personnages.
Chaque personnage pourras avoir de nombreuses classes (Guerrier, Tank, Mage, Sorcier, Archer, Voleur...)
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 sous 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 peux 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 spciale 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 seras gérer de faon agile (Sprint chaque semaine + réunion tout les jours de 15min pour faire un bilan rapide).
Cependant, nous allons réaliser plusieurs itération du projet. Chaque itération auras pour but de rendre un produit fonctionnel.

- Itération 1 : Réaliser tout le systme de classe / level / item + ralisation d'un premier systme de combat contre une IA ennemie basique.
- Itération 2 : Réaliser la partie histoire du jeu, en développant les diffrentes IA enemies ainsi que le système d'histoire branchée.
- Itération 3 : Réaliser la partie multi-joueur du jeu, avec un système de ladder.

L'itération 1 auras une durée de 3 mois.
Les itération 2 et 3 auront chacune une durée de 2 mois sachant qu'elle seront réaliser en parallèle par deux équipe de deux.

Chapitre 4

Prototype design

Coming soon

Chapitre 5

Communication

La communication se feras via un Website/devblog et sera appuiée par un compte twitter qui détaillera l'avancement du jeu en postant des photos, vidd́os 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 interesséssés.
La beta test sera gérée a l'aide de la google dvelopper 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'éxperience utilisateur. Elles pourront être par exemple integrées sous forme de bannire a 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)
— Un mode solo fonctionnel
— Un multijoueur fonctionnel
— La présence d'achievements

Chapitre 8

Bonus

— Une monétisation bien intégrée