

# Open Project

mbourlet, gfernand, fbuoro, mchoong #e1r6

# Chapter 1

## Project presentation

The goal of our project is to build a native video game on Android. This game will be a turn by turn battle role-playing game(RPG) but with a profound strategical aspect. Unlike other transitional turn by turn battle RPGs, the player cannot choose his actions at each turn, instead the actions of each turn are triggered by a list of AI that set by player. The project will be split into two parts: the engine and the game.

### 1.1 Engine

The engine will be the framework for the game, it will be reusable for different projects. Its purpose is to accelerate the current and future development of the game. It is divided into several parts.

#### 1.1.1 Game

Game class is the central class of the engine. It manages the program logic and the transition of game states.

At launch, the game tries to load the saved game automatically.

#### 1.1.2 DataManager

The DataManager manages the user backups. It offers two possibilities :

- Local Save : The game data 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 the game data to the cloud.

### **1.1.3 Others**

Other features of the engine maybe as below:

- TextureManager : Manage the graphical content
- AudioManager : Manage the audio content
- GameState: Manage the interface and logic
- Etc.

## **1.2 Projet**

The project is built mainly using the feature from our engine. For other features such as cloud storage, multiplayer, achievements and etc., it will use the technologies offered by Google.

## Chapter 2

# Gameplay

### 2.1 Roles

Player starts the game with 3 distinctive characters, with 3 initial roles. There are 7 roles available in the game, they will be introduced to user in the story mode, progressively. 7 roles as below:

- Fighter
- Healer
- Tank
- Black Magician
- Green Magician
- Thief
- Archer

Each character can play as different role throughout the game. Each role levels up with a character, depending how often it is used by that character in battles.

### 2.2 Behaviors

Each role has a set of behaviors. Behavior controls how a role act under certain condition, example behavior for a fighter as below:

- condition: nearest enemy, action: attack
- condition: enemy with lowest hp, action: attack

More behaviors can be define by leveling up a role. Each action cost certain amount of active time in battle.

Active time is the time in second that needed to start certain action in a battle, aka casting time.

## 2.3 Formations

With different roles, user can create different formations. Example formations as below:

- formation 1: Balanced
  - role1: fighter
  - role2: healer
  - role3: tank
- formation 2: Attackers
  - role1: fighter
  - role2: fighter
  - role3: fighter

Formations can be switched during battle. Each role has a specific action that can be initiate by user manually during battle. These actions cost double action time if initiated manually. The player will be able to control a team of 3 characters .

## Chapter 3

# Methodology

The project will use the Agile method (Sprint + meeting of 15 minutes every day to make a quick assessment ) .

The project will be divided into several steps. In each step, we aim to make a functional product with additional features, on top of the improvement of the previous step.

- step 1: Achieve a fight between two basic AI ( Allie (programmable) / Enemy )
- step 2: Achieve a basic class system
- step 3: Achieve level + inventory system
- step 4: Achieve the story mode / basic quests
- step 5: Add enemy AI
- step 6: Improve the story mode / quest (branch)

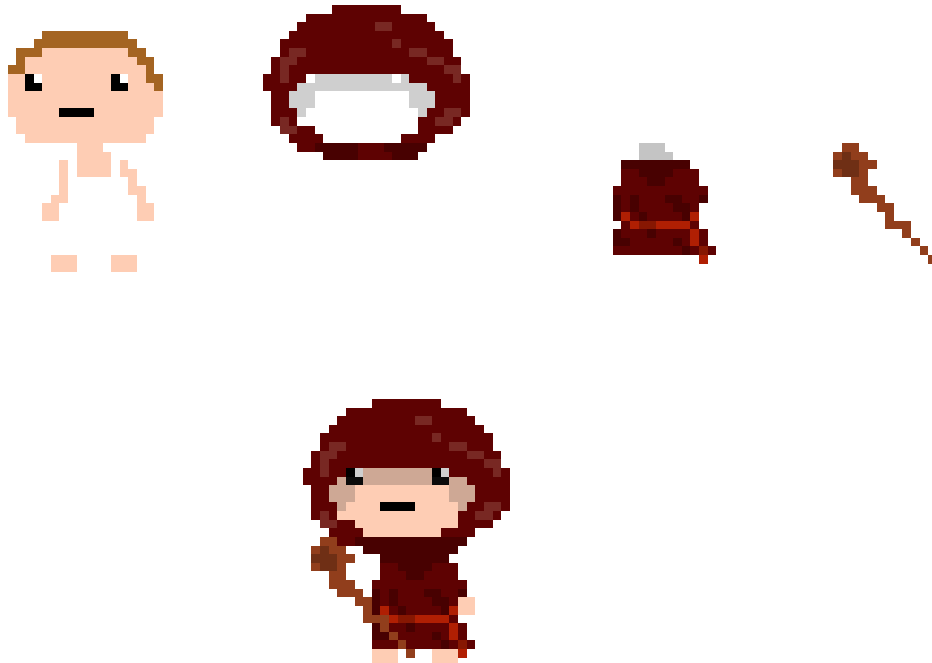
Upon finishing the final step, we can consider that the mandatory part is finished. Bonus features for the project maybe as below:

- Adding achievements
- More items
- More classes
- Solo ladder
- Multiplayer
- Multiplayer Ladder
- Entire character animation

## Chapter 4

# Design

The graphical assets of the game will be made using Pixel Art. Here are some of our implementation:



## Chapter 5

# Promotion

The promotion will be made via a Website / dev blog and will be helped by twitter account detailing the progress of the game by posting photos, gameplay videos etc. We want to attract people and make them interested in the project even before the release of the game. This is especially why a beta test phase will be available to the most interested users.

The beta test will be managed using the google developer console. The website will also sent newsletter to users.



## Chapter 6

# Monetization

The monetization of the game will be via in-app purchases and ads. The in-app purchases should not affect the fairness of the game, especially in the context of the multiplayer . We currently plan to offer paying experience packs that would help the player to level up faster.

The ads are non-intrusive and will not disturb the user experience. We will use admob API ( used by Rovio, Backflip Studio, Fingersoft ...) to integrate the targeted ads and increase our revenue.

## Chapter 7

# Checklist at the end of the project

- Iteration 1: Achieve a fight between two basic AI ( Allie (programmable) / Enemy )
- Iteration 2: Achieve a basic class system
- Iteration 3: Achieve level + inventory system
- Iteration 4: Achieve the story mode / basic quests
- Iteration 5: Adding enemy AI
- Iteration 6: Improving the system of history / quest (branch)

## Chapter 8

# Bonus

- Adding achievements
- More items
- More classes
- Solo ladder
- Multiplayer
- Multiplayer Ladder
- Entire character animation