Detailed Design Document

Bluetooth Android Fantasy Game

Prepared by: Alex Ciaramella, Matt Ng,

Dakota Pollitt, & Scott Ritchie

Computer Science Senior Project – Spring 2015

Instructor: Dr. Ganesh Baliga

**Table of Contents**

1. Introduction                                                                                                                               3

2. Module Design                                                                                                                          3  
 2.1 Connector 3

2.2 Story 4

2.2.1 Story Controller 4

2.3 Event 4

2.3.1 Event Controller 5

2.4 Character 6

2.5 Game End 6

3. Abstract Data Types                                                                                                                 7

4. COMPRISES diagram                                                                               8

5. USES diagram                                                                                                                          9

6. Integration Test Plan                                                                                                              10

6.1 Test Overview 10

**Introduction:**  
 This is the official design document for our Spring 2015 senior project. The project is a mobile gaming application that utilizes Bluetooth technology by connecting two to four players together in a unique fantasy adventure. Our application will be developed using the Java programming language and Android Studio. Our goal is to provide players with a fun, randomized adventure each time they play through the game. The system will include the following capabilities:

* Allow users to connect their device to nearby devices that have bluetooth enabled.
* Allow users to select a character to play through the game.
* Allow users to choose between options which will ultimately affect the story.
* Allow users to work together to battle randomized bosses.

**2. Module Design**

**2.1 Connector**

o **The purpose of the module:**

The purpose of this module is to give the user a way to connect to nearby devices. The user will be able to host a lobby, or join existing lobbies.

o **The rationale of the module:**

Without this module the user will have no way to connect to a host or lobby.

o **The provided interface:**

This module does not have any provided interfaces.

o **The required interface.**

This module will not require any interfaces.

**2.2 Story**

o **The purpose of the module:**

The purpose of this module is to give the user a reason to beat the bosses and ultimately play the game. It will also give the user options and freedom to make decisions which affect the story or go to the shop to buy or sell items. This module will feature a large textbox with a background image. It will give the user two or three options to continue with the story line. One of these options will be to use the shop in order to purchase items.

o **The rationale of the module:**

Without this module the user will have no reason to play the game and won’t have any freedom to make decisions as to how to continue on with the game. The purpose of this module is to give the user a GUI to pick how to continue on with the story as well as an option to buy items at the shop.

o **The provided interface.**

None

o **The required interface.**

void beginEvent()

void endEvent()

**2.2.1 Story Controller**

o **The purpose of the module:**

The purpose of this module is to control our story. It keeps track of Events that players have already completed, and stores Character data.

o **The rationale of the module:**

Separates responsibilities.

o **The provided interface.**

None

o **The required interface.**

None

**2.3 Event**

o **The purpose of the module:**

The purpose of this module is to give the users fun gameplay. Players will need to work together to defeat the bosses. Each battle will be turn based where each character gets a chance to go. Each player will be unable to see each other health/energy bar or status effects so players will need to communicate with each other to keep each other alive and defeat the boss. This module will contain a list of moves a character can use including attack, ability and item as well as a picture of the boss and a text box at the top. Players will need to use items and abilities they acquired from the shop module and character module to beat each event.

o **The rationale of the module:**

Players need opponents to face on their adventure. Events are the obstacles that players must overcome in order to complete their adventure. This module was created to give a user a user friendly GUI to use items and abilities they have acquired throughout their adventure.

o **The provided interface.**

**beginEvent()**

Description: The init method will initiate all the variables including the  
 BluetoothAdapter, the join and host buttons and the intent filters.

Parameters: None

Returns: None

**endEvent()**

Description: This method will try to find a device which has discovery turned  
 on. Specifically, it will look for a device who is ‘hosting a lobby’.

Parameters: None

Returns: None

o **The required interface.**

int getHealth()

int getEnergy()

MoveResult useMove(Character, Move, Character)

boolean useItem(Character, Item, Character)

List<Character> getPlayers()

List<Character> getEnemies()

**2.3.1 Event Controller**

o **The purpose of the module:**

The purpose of this module is to control our events. It serves as a link between the UI (Event module), and the Character module.

o **The rationale of the module:**

Separates responsibilities.

o **The provided interface.**

**void useMove(Character, Move, Character)**

Description: This method will use the move the user selected.

Parameters: Character using the move, the move, character the move is being used on

Returns: None

**void useItem(Character, Item, Character)**

Description: This method will use the item selected by the user.

Parameters: Character that uses item, Character that benefits, Item  
 used

Returns: None

**List<Character> getPlayers()**

Description: This method will get a list of all the current players.

Parameters: None

Returns: A list of characters.

**List<Character> getEnemies()**

Description: This method return a list of all characters which are  
 enemies in the current event.

Parameters: None

Returns: A list of characters.

o **The required interface.**

public void useMove(Move move)

public void useItem(Item item)

**2.4 Character**

o **The purpose of the module:**

The purpose of this module is to give the user a character to play during their adventure. Along the way, a character acquires new abilities and skills by leveling up. These skills and abilities will be specific to each character. This module will form the background end of the game.

o **The rationale of the module:**

This module is needed to add more interesting and fun features to the game, giving the player a certain style of play. This module will determine how strong each character is and how strong the bosses are.

o **The provided interface.**

**int getHealth()**

Description: Method used for returning a Character’s health

Parameters: None

Returns: An integer representing the character’s health

**int getEnergy()**

Description: Method used for returning a Character’s energy

Parameters: None

Returns: An integer representing the character’s energy

**Effect useMove(Move)**

Description:

Parameters: None

Returns: An effect which contains an integer for how much damage should be  
 done to the target and a list of Characters which represents the status effects  
 which should be applied to the target

**boolean useItem(Item)**

Description:

Parameters: The Item used.

Returns: True if using the item was successful, False if it was unsuccessful

o **The required interface.**

None

**2.5 Game End**

o **The purpose of the module:**

The purpose of this module is to end the game after certain conditions are met. These conditions include, death of all characters, victory of all events or a user quitting the game. This module will notify that the game has or is about to end and give the user options on how to proceed. These options include restart the event, quit the event or quit the game.

o **The rationale of the module:**

Without this module the user will never know when the game ends. It is important to give the option to the player to restart the game or part of the game when the game is over to encourage the player to play again.

o **The provided interface.**

The Game End module will not provide any provided interfaces to any other module.

o **The required interface.**

None

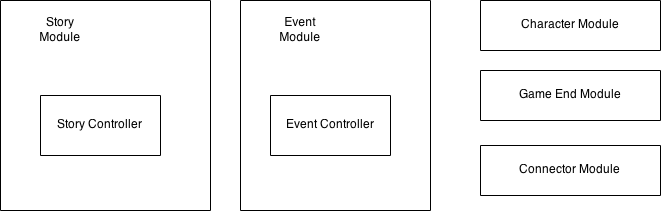
**3. Abstract Data Types**

Effect{  
 int base\_damage  
 List<Characters> status\_effects  
 }

Item{  
 String description  
 Image img  
 Stats stats  
 }

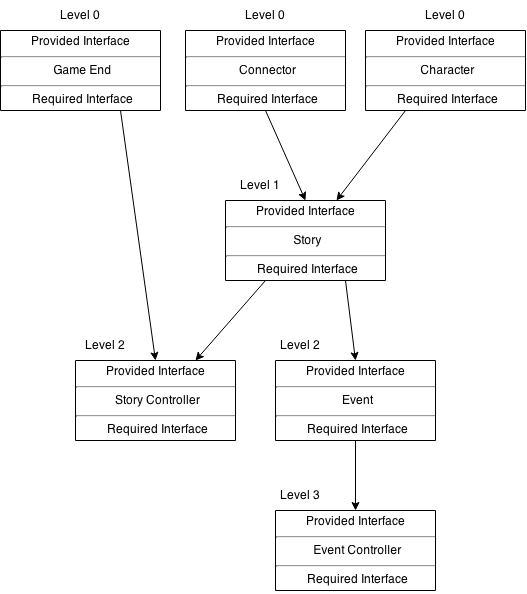
Move{  
 String name  
 String type  
 Image img  
 String description  
 Effect effect  
 }

**4. COMPRISES Diagram**



|  |
| --- |
| Comprises (1).png |

**5. USES Diagram**



**6. Integration Test Plan  
  
 6.1 Testing Overview**

Testing our game will consist of us running each feature (Characters, Moves, Items, Events, Stories, etc.) in as many possible scenarios as time will allow. This will hopefully expose any situations where the game crashes, bugs out, or makes the game too easy or difficult.

We’ll also be constantly testing the integrity of the Bluetooth connection between users, trying multiple different Android devices, and testing that a playthrough goes uninterrupted, and stays synced with the other players in your party.