Requirements and Analysis Document for the Beat-to-the-Beat project (RAD)

**Contents**

1. **Introduction**1.1 Purpose of Application  
   1.2 General characteristics of application  
   1.3 Scope of Application  
   1.4 Objectives and success criteria of the project  
   1.5 Definitions, acronyms and abbreviations
2. **Requirements**  
   2.1 Functional requirements   
   2.2 Non-functional requirements  
    2.2.1 Usability  
    2.2.2 Reliability  
    2.2.3 Performance  
    2.2.4 Supportability  
    2.2.5 Implementation  
    2.2.6 Packaging and installation  
    2.2.7 Legal

2.3 Application models  
 2.3.1 Use case model   
 2.3.2 Use cases priority  
 2.3.3 Domain model  
 2.3.4 User interface   
 2.4 References  
  
  
**Version: 1.0  
  
Date: 2014-03-18  
  
Author:** malinth, dhampus, hebjorn, ponerik

This version overrides all other versions.

https://github.com/HampusDahlin/Beat-to-the-Beat.git

**1. Introduction**

This section gives a brief overview of the project.

**1.1 Purpose of application**

This project aims to create a computer game where we will utilize music to dynamically change how the world works. Our game is going to be a “Beat ’em up” game, so depending on the music’s beat you will have to time your punches in order to beat up the enemies. For definitions, terms and rules of the game see references.   
  
**1.2 General characteristics of application**

The application will be a desktop, standalone (non-networked), single player application with a graphical user interface for the Windows/Mac/Linux platforms.

The application will feature different levels, one for each song. The playable character will not be able to move, instead the ground will move under him with enemies coming with the beats of the current song. When the character is getting close to an enemy a button prompt will come up and the player has to hit that button. To complete a level one has to defeat all the enemies without dying. In order to die one has to fail several times at hitting the prompted buttons in time.  
  
Throughout the levels we will have NPCs (see Definitions) who will serve as the main storytellers. After finishing a level we will have cutscenes explaining the story further.

**1.3 Scope of application**

The application does not support multiplayer. The application will save progress at the start of each level, but is unable to save in the middle of the level. Score will not be implemented, see Possible future directions.

**1.4 Objectives and success criteria of the project**

1. It should be possible to play through a level killing all of the enemies to the beat of the music.

2. There should be NPCs and cutscenes to convey a story to the player.

**1.5 Definition, acronyms and abbreviations**

* GUI: Graphical User Interface
* Beat ’em up game: A game where the main focus is beating up enemies.
* NPC: Non Playable Character. A character in the game controlled by the computer.
* Cutscene: A sequence where the player has little to no interaction and further plot is revealed, through video/animation and/or pictures.
* Level: An area in a game which the player needs to complete in order to progress in the game.
* Java: platform independent programming language.
* JRE: Java Runtime Environment. Software needed to run as Java application.
* Host: a computer where the game will run.

**2. Requirements**

**2.1 Functional requirements**

The player should be able to:  
1. Start a new game  
2. Load previous save

3. Play the game. During a playthrough the player should be able to:

* Kill enemies by pressing the prompted button.
* Progress to the next level (and save progress between levels)
* Take damage
* Regain lost health between levels
* Defeat bosses by pressing the prompted buttons.

4. Exit the application, this does not save your position in the current level.

**2.2 Non-functional requirements**

**2.2.1 Usability**

Usability is a high priority. Normal users should be able to play the game within a very short time period. The game is simple enough that most players should be able to grasp the idea quickly. The game will start with a slower tutorial level to familiarize players with the game.

There will be a help button available in the game menu to explain controls and the main concept of the game, since there are no hard rules we only need this small explanation.

**2.2.2 Reliability**

NA

**2.2.3 Performance**

After a player initiates an action the game must respond at worst in 0.60 sec. Anything else in a music-based game will lead to frustration or lead the game to be downright unplayable.

**2.2.4 Supportability**

**2.2.5 Implementation**

To achieve platform independence the application will use the Java environment. All hosts must have the JRE installed and configured. The application must be installed on all hosts it will run (possibly downloaded)

**2.2.6 Packaging and installation**

**2.2.7 Legal**  
The program will use songs we are not owners of. This is a legal issue since we will distribute the game with the songs in it.

**2.3 Application models**

**2.3.1 Use case model**

See APPENDIX

**2.3.2 Use case priority**

1. HitButton

2. Start next level

3. Finish Level

4. Start new Game

5. Load game

**2.3.3 Analysis model**

see APPENDIX

**2.3.4 GUI**

The application will use a resizable GUI following standard conventions. The application will take in to consideration a few predefined screensizes. See APPENDIX for screens and navigational paths.

## Appendix:

### Usecases:

overview

