Requirements and Analysis Document for The Grupp

Version: 1.0

Date: 27/03/2018

Author: Sebastian Lind El-Bahrawy, Eric Mossberg and Alexander Nordgren

Contents

1 Introduction

The group looks to produce a simple, yet entertaining game. The game will contain five unlockable levels and one initial starting level. The levels will be accessible through an overwold map where the player might move and choose a level. A level will be a two dimensional side-scroller moving with the player character. Each level will be more challenging than the last, eventually leading up to a final boss fight. The player will be challenged by maneuvering increasingly difficult enemies and moving platforms.

Some general functionality include:

- The game features save slots.
- The game initially contain six levels.
- Each level, excluding the first, is unlocked by finishing the previous level.
- The players progress is automatically saved at each finished level and only at these points.
- Levels have checkpoints storing in-level session progress, allowing the player to start from that point whenever they die, up to a total of 3 times. After 3 deaths, the player is returned to the overworld.
- The game has sound in the form of music and .

1.1 Definitions, acronyms and abbreviations

- **Overworld Map** - A simple top view over the character and the map showing the progress of the player, simply a level menu.
- **Side-Scroller** - The "camera" follows the player character as it moves left or right.
- **HP** - Hitpoints. Represents an actors life total. If it reaches 0, the actor is killed.

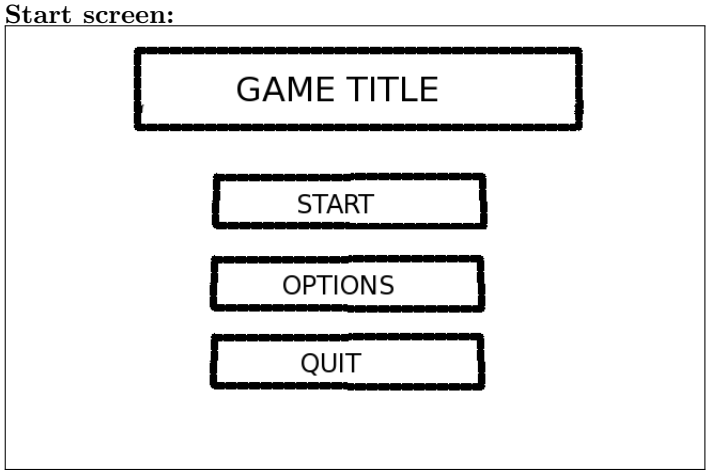
2 Requirements

2.1 User Interface

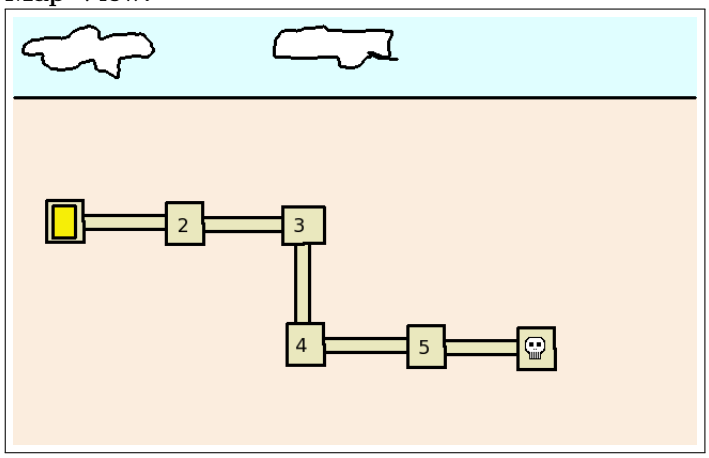
The game will have three menus and two playing views.

- The start menu
- The save menu
- The options menu
- the the overworld map view
- the level view.

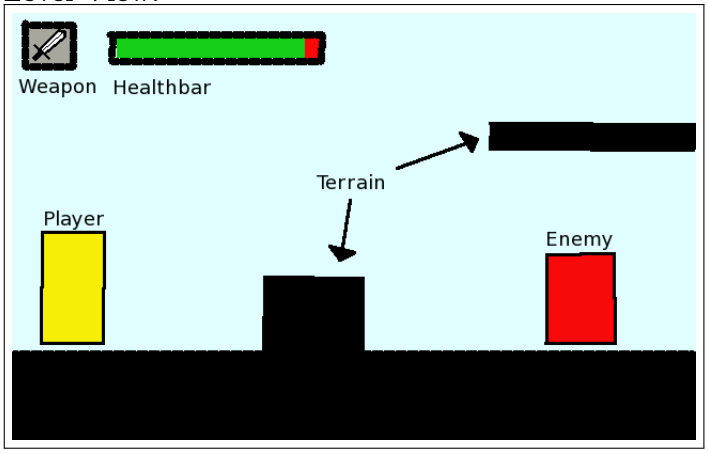
Start screen:



Map View:



Level View:



2.2 Functional requirements

The functionality will be dependent on the current state of the game, as follows:

Main menu

The main menu intended as a gateway to get into the game. The player may do the following:

- Start the game, either from the beginning or from a save file
- Adjust settings via the options menu, like sound levels and refresh rate
- Exit the game

Overworld

After getting into the game, the player is greeted with an overworld map that contains several tiles on a line, each tile representing a level. It will have the following functionality:

- Move. Movement to a tile is only allowed if the player has beaten the previous tile's level
- Start a level from the current tile that the player is positioned at
- Return to main menu

Level

- Move. Movement to a tile is only allowed if the player has beaten the previous tile's level
- Start a level from the current tile that the player is positioned at
- Return to main menu

The user will be able to interact with the buttons by simple mouse clicks allowing the player to change the view.

The user will be able to control the player character using the keyboard it may instruct the character to jump, attack and walk.

The user may transition in between levels on the map view using the keyboard.

2.3 Non-functional requirements

Game

The game needs to be able to save player progression

The game should have options to turn off sound.

The game will be developed in modular way allowing for future implementations.

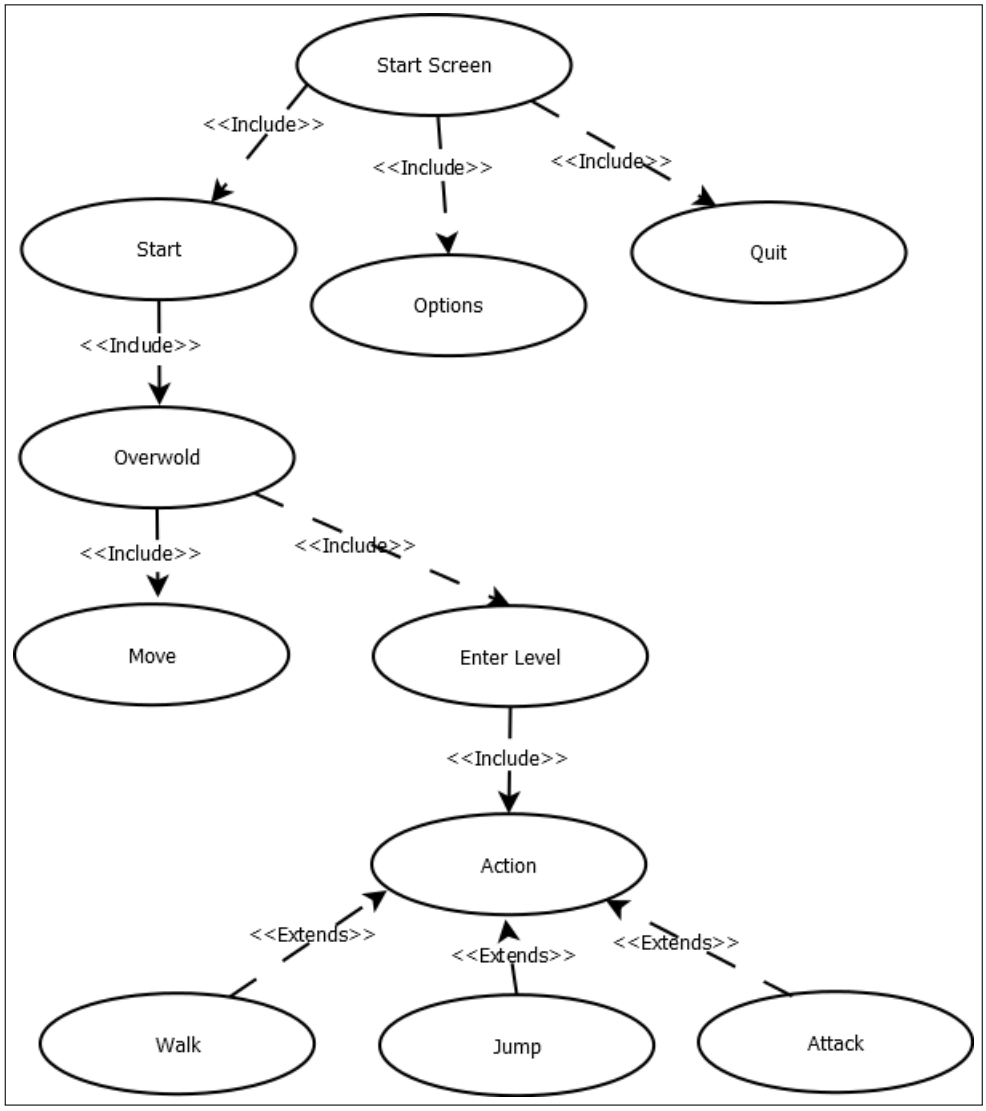
Technical

The game needs to be runnable on Mac, Linux and Windows.

The finished product will be delivered with instructions regarding the installation and running of the program.

The application will be created in Test driven fashion, as new functionality is implemented, tests will be run to prevent issues with the finished product.

3 Use Cases


Start Screen
<<Include>>
<<Include>>
<<Include>>
Start
Options
Quit
<<Include>>
Overwold
<<Include>>
<<Include>>
Move
Enter Level
<<Include>>
Action
<<Extends>>
<<Extends>>
<<Extends>>
Walk
Jump
Attack

3.1 Use case listing

1. Menu

Summary: The player is in the main menu.
Priority: Low
Extends:
Includes: Overworld
Participators: Player

	Actor	**System**
1.1	Press start Button	
1.1.1		Save slots appear
1.1.2	Press save slot	
1.1.3		Game starts
1.2	Press options Button	
1.2.1		Menu switches to the options menu
1.3	Press the quit button	
1.3.1		The program exits

2. Overworld

Summary: The player is in the overworld view
Priority: Medium
Extends: Menu
Includes: Move & Enter
Participators: Player

	Actor	**System**
1.1	Pressed movement key	
1.1.1		See 3. Move
1.2	Pressed enter key	
1.2.1		See 4. Enter level

3. Move

Summary: The player moves in a direction on the overworld map
Priority: Medium
Extends: Overworld 1.1
Includes: Overworld
Participators: Player

	Actor	**System**
1		Receives directional input
2		Attempts to move the character
2.1 Legal move		Moves the player character in given direction.
2.2 Illegal Move		If the movement requested is illegal, eg. the player cant go that way or the level in that direction is not yet unlocked, nothing happens

4. Enter level

Summary: The player enters a level
Priority: Low
Extends: Overworld 2.1
Includes: Walk & Jump & Attack
Participators: Player

	Actor	**System**
1		Receives enter input
2		The level corresponding to the characters position is opened

5. Walk

Summary: The player has entered a level.
Priority: High
Extends: Enter Level
Includes:
Participators: Player

	Actor	**System**
1	Press walking button	
2		System checks if the characters hitbox would overlap with any other hit box, indicating a collision.
2.1 No collision		If no collision is detected, the character is moved on the x-axis
2.2 Collision		If a collision is detected, the character is moved next to the colliding object.

6. Attack

Summary: The player performs an attack
Priority: High
Extends: Enter Level
Includes:
Participators: Player

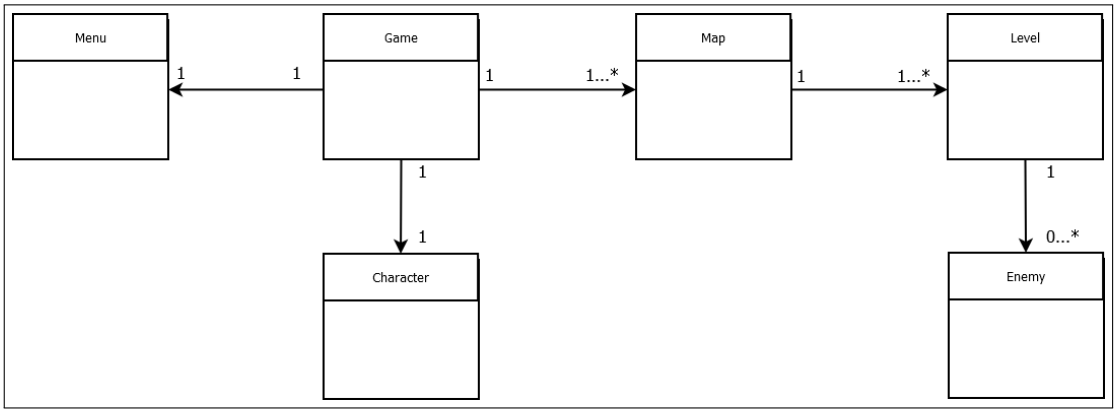
	Actor	**System**
1	Press attack button	
2		Fires an attack in the direction the player character is facing, and performs checks to see if the attack hit something.
2.1 Collision detected		If the attack hits an enemy, the enemy is dealt damage. If the enemy's HP reaches 0, it is killed.

7. Jump

Summary: The player performs a jump
Priority: High
Extends: Enter Level
Includes:
Participators: Player

	Actor	**System**
1	Press jump button	
2		The player character jumps upwards in an angle determined by the character's facing direction.
2.1 Collision detected		If collision is detected at any point during upwards momentum, all momentum is lost, and the character begins falling.

4 Domain model



4.1 Class responsibilities

Game: Represents the game as a whole.
Menu: Menu accepting user input.
Map: Overworld map showing available levels.
Level: The core of the game, a two-dimensional platform where a character is controlled.
Character: The unit controlled by the player.
Enemy: An obstacle for the player to overcome.

5 References