**Computer programming**

**Final Project**

Report

I.E.S. San Vicente

San Vicente del Raspeig (Alicante)

2017/2018

Teachers:

José Ignacio Cabanes / Ignacio Iborra

Student:

Ángel Rebollo Berná

-= Note: all the texts included here are just an example

of the way that the report should look when finished

Each student must replace these texts with

the ones corresponding to their projects =-

* **1. Introduction**

**Project name**

ZeldaSDL

**Made by**

Ángel Rebollo

**Short description of the project**

This is an adventure rpg game, like the original made in 1986, but with improved graphics and gameplay (modern spriteSheets and improved controls).This version will also have the addition of roguelike gameplay (unique map each play). It is a graphical application that uses the SDL graphics library.

* **2. Functionality of the project**

After entering the program, a welcome screen with the logo on it will we displayed, waiting for the player to press the start button.

After that, the main menu will appear with this options:

• New Run.

• Stats.

• Credits.

• Quit.

if the player chooses “New Run”, the game should process 10 levels at random. or at least take them from a librarie yet created and put together.

On each level the player must fight enemies spawned at random locations and he must find a chest hidden on each level that contains the key for the next level.

On the final level (the 10th), the player will enter on a special room where he will fight a stronger enemy (a final boss).

Once everything is over the game will have stats of like damage taken, time passed to complete, score….

if the “stats” option is selected there will be displayed all the “Runs” that the player had, showing how the player did.

The credits menu will show the developer name, and the libraries used to develop the game.

The game will NOT have a pause screen at all as a feature (like “Dark Souls” games) adding a step further on terms of difficulty.

The enemies will have a chance of dropping a heart (to increase health) but very rare (1/25)

* **3. Screen prototype**

The game screen will look like this:

* 
* **4.Analysis**
* **4a. Requisites**

|  |  |
| --- | --- |
| Requisite | Date achieved |
| The program will allow a player to play a “Run” |  |
| The player should have the attack moves (sword) |  |
| The enemies will move freely and attack the player |  |
| The program will generate new maps or spawn them from a library |  |
| The game will start showing an intro screen |  |
| The menu screen will allow the user to enter a Help screen, which will display hints on how to play |  |
| The menu screen will allow the user to enter a Credits screen, which will display data about the programmer |  |
| the player should have the special attack move(bombs that damage on range) |  |
| The game will use SDL graphics library, and the game can be controlled with an controller (ps3 or xbox360) |  |
| The game should have a roll mechanic that disables the hitbox of the player for a fraction of a second so he can dodge enemy attacks |  |

* **4b. Basic pseudocode**

Program body:

Display welcome screen

Wait for user to press start (on controller) or enter (on keyboard)

The menu screen will have scrolable buttons so you can access via controller

…

New Run game:

Load the first level on screen and the run will begin.

Once the door is unlocked and crossed the next level will load

…

…

* **4c. Classes diagram**
* 
* **5. Initial planning and expected deliveries**
* **5a. Expected deliveries**
* **Version 1:** Basic game structure and soft logic.
* **Version 2**: player class (movement with controller and keyboard + animations + all his variables) created.
* **version 3:** Enemies created with basic IA (random movement and shooting at the direction of the player) + attacks (throw small rocks that rests 1 hearth of the player each hit).
* **Version 4:** Player´s main attack (sword on range, normal enemies looses ½ of his hp).
* **Version 5:** Developed the drop mechanics (when killing an enemy).
* **Version 6:** Class level that reads from file and create a tile each time.
* **Version 7:** Scroll on and between levels (when the door for the next level crossed, the camera moves away from the previous level so it can be unloaded from memory).
* **Version 8:** Implemented the bomb mechanic and area damage (the player drops a bomb on its current location that explodes at 5 seconds of delay damaging nearby enemies and the player itself if on range).
* **Version 9:** Developed the lock mechanism, inventory system, and keys mechanic (the door to the next screen is locked by default. if the level does not contains a “k” char, it should be dropped with the last enemy. If the player has the key inside its inventory the door should open when touched).
* **Version 10:** Developed the final boss and unique move patterns (the enemy itself has another sprite, health, and does not move at random. He seeks the player, and when he is on range he attacks. When he is attacked he falls back and tries again).
* **Version 11:** Developed the time, score, and health mechanics + Optimization system to destroy the out of the camera sprites.
* **Version 12:** Developed the credits screen, Stats screen (that reads from a stats file), Game over screen, and WelcomeScreen.
* **Version 13:** Implement a save state that saves the enemies position and objects between levels.
* **Version 14:** Developed the procedural design (A map generated at random Each time, but most of the maps will be preGenerated for more visual quality. This will be used random between the preRendered levels and the random levels to give the game more rePlayability).
* **Version 15:** Implement the roll mechanic that disables the hitbox of the player for 0.2 seconds and changes its sprite while doing the animation.
* **5b. Real deliveries/**

14/05/2018 - Created Basic squeleton able to create screens, images, prites, and recognices controls.

15/05/2018 - Problems inheriting clases, and player yet dont have animations.

* **6. File formats**
* **6a. Plain files format**

The scores table is saved to a text file, which contains the name of a player in one line, and the corresponding score in the following line, and so on, to complete the 10 highest scores, as in this example:

John

125

Peter

116

…

* **7. Problems found and solutions**
* **8. Improvements or restrictions to the starting design**
* **9. Screenshots of the final project**
* **10. Source code of the final project**