**Computer programming**

**Final Project**

Report

I.E.S. San Vicente

San Vicente del Raspeig (Alicante)

2017/2018

Teachers:

José Ignacio Cabanes / Ignacio Iborra

Student:

Brandon Blasco del Cid

1. Introduction

**Project name**

* Bomberman

**Made by**

* Brandon Blasco del Cid

**Short description of the project**

* Bomberman is a game for 4 players at the most, which consists in killing your opponent’s taking advantage of the characteristics of the map and your bombs. It is a graphical application that uses the SDL graphics library.

## 2. Functionality of the project

## After entering the program, a welcome screen will be displayed, where the user can choose between:

## • To play. (set game options)

## • Map creation.

## • Credits.

## • Controls

## • Exit game.

## If the user chooses the option to play, they will be asked how many players will play up to a maximum of 4 people, they will also be asked in which map they wish to play and the time of the game, having a minimum of 2 min and a maximum 8min. The game will consist in breaking blocks to get advantages to be more powerful than the opponents and thus be able to trap and eliminate them. Finally, the surviving player was the winner.

## If you choose the option of creating a map, you will access a screen which, you will be asked the name of the map and you can also design a map, finally you can save it and return to the main menu.

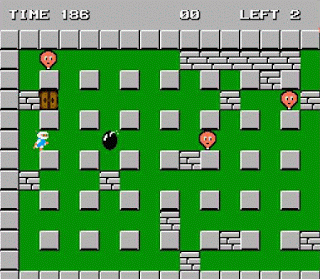
## If the option of Credits is chosen, a screen will be displayed in which the name of the creator of the game and records of previous games with the name of the player will appear.

## If the option of Controls is chosen, a screen with the controls of the game is shown to learn how to play.

## Finally, we will have an option to exit the game.

## 3. Screen prototype

The game screen will look like this:



## 4.Analysis

### 4a. Requisites

|  |  |
| --- | --- |
| Requisite | Date achieved |
| The main menu will be displayed, with animations and you can choose each of the options, but the options will not work yet. | 14-05-2018 |
| The option to start a game of the main menu will already have functionality, you can customize the games | 15-05-2018 |
| The settings will be displayed on the game screen | 16-05-2018 |
| the options of credits and controls will also take on playability, and you will be able to navigate through these options without problems | 17-05-2018 |
| The map and the details of the game will be loaded by a marker | 18-05-2018 |
| When entering a game the characters will be visualized and they can move | 21-05-2018 |
| The characters can not go through the blocks and can only go through the green areas | 22-05-2018 |
| Pumps can be placed and the explosion of the bombs will be displayed after a few seconds | 23-05-2018 |
| The destruction of blocks and the deaths of the characters through the fire of the bombs | 24-05-2018 |
| Emergence of advantages and disadvantages that players can take | 25-05-2018 |
| Creation of sudden death to avoid ties | 28-05-2018 |
| two people will be able to play | 29-05-2018 |
| 30-05-2018 |
| you can create custom maps | 31-05-2018 |
| 1-06-2018 |

### 4b. Basic pseudocode

Program body:

Display welcome screen

Wait for user to press a key

If key = ESC, quit program

If key = 1, start one player game

…

One player game:

Display level selection screen

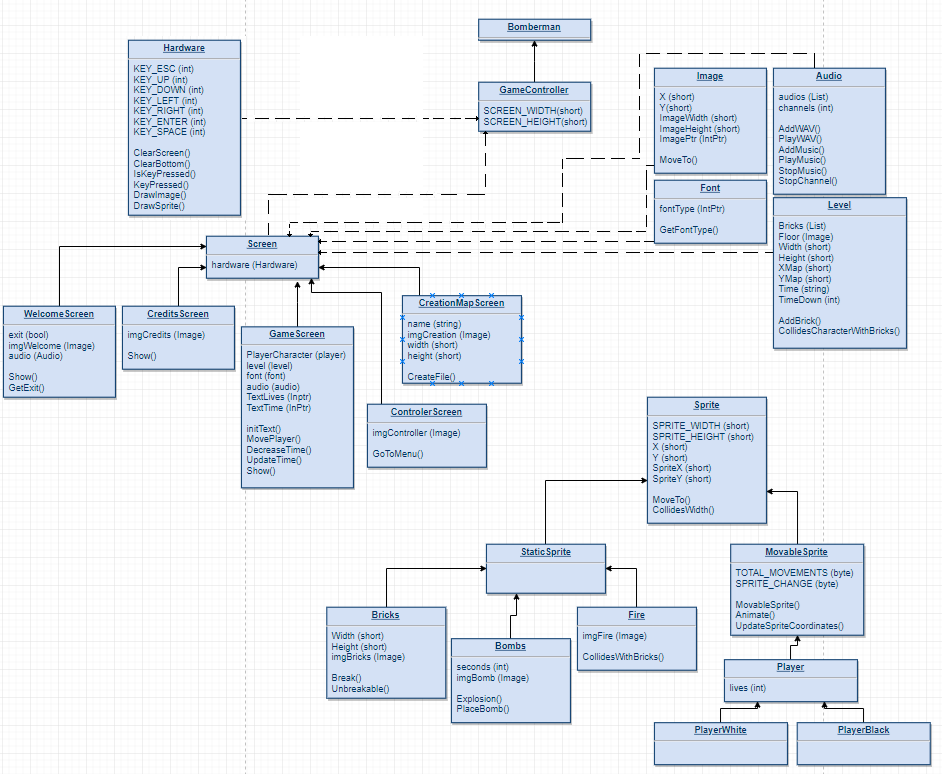
Wait for user to press a key

If key = ESC, return

…

…

### 4c. Classes diagram



## 5. Initial planning and expected deliveries

### 5a. Expected deliveries

• **Version 1**: I will focus on the creation of the class structure and the development of the main menu, in which I will access the different parts of the menu with a selection animation, this menu will consist of a bomb that will explode when the player selects an option with the enter key.

• **Version 2**: creation of the screen where the game options will be configured and the character selection to start the game (time, lives per player, map)

• **Version 3**: creation of the map (Part 1) in this part we will only represent the previous configurations that we have added in version 2 (lives, time, map).

The elements to be represented will be represented by large numbers to make it more visual for the user (especially time).

• **Version 4:** creation of the screen of the video game controls, where you will see the keys with which the game will be played and also the credit screen, which will show the players' registers, the name of the creator of the game and the information on the game (Version, name of the game, status of the next version, release date of the next version)

**• Version 5**: Creation of a game timer and if at the end we run out of time, the game will be tied and also the creation of the map (Part 2), which we will load from a file.

• **Version 6:** Mechanics of the players, here we will develop the movement of the players and the speed of movement.

• **Version 7:** collisions of players against blocks of the map

• **Version 8:** Operation of the pumps, with operation I mean the animations since this will have an expansion force (it depends on the force it will draw more fire or less) or it will explode in x seconds.

• **Version 9:** In this instalment I will continue with the bombs since I will have to check the collisions, if the fire of the bomb touches an indestructible block it does not break, or if it touches a character it will kill it and take a life from it. A detail that is also important is that if the fire destroys a block, the fire will end up in that block.

• **Version 10:** Creation of the advantages and disadvantages, this will apply to the characters to increase or decrease the firepower, the speed of the character, number of bombs

• **Version 11:** Death sudden, this will consist in that if the players have not been killed and the chronometer has reached zero the map will be filled with blocks until only one of the players is alive.

• **Version 12:** multiplayer game, at least 2 players

**• Version 1**3: creative interface mode (the user can develop their own maps and then reproduce them)

• **Version 14**: Created from the ReadLine method in sdl for character typing by the user.

• **Version 15:** Interface mode, error checking, within these checks is that the user does not make maps larger than the established limits, that the characters entered are correct.

### 5b. Real deliveries

* **Version 1 (May 14):** I have created the class structure. I also created the initial loop to start the main menu. I have also created the image hardware class
* **Version 2 (May 15):** today I almost fulfill my expectations, I have not complied because I had to adelnatar the work that I missed in the first delivery. In this installment I have finished the main menu and can now configure the options of the game
* **Version 3 (May 16):** I have had problems when passing the variables lives and time to the game screen, but to avoid wasting time I have started to draw the loaded map from a file.
* **Version 4 (May 17):** Raul Gogna, I create class ControllerScreen and CreditsCreen

I also had to create the Brick and BrickDestroyable classes to be able to represent them, something that was not planned at the beginning. Within the bad things went well

* **Version 5 (May 18):** I have changed the settings screen now I can only choose the map option, but I had problems when processing the data to print the 3-minute timer on the screen

## 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

…

### 6b. Entity-Relationship Diagram (If needed)

(…)

## 7. Problems found and solutions

I preferred not to use the SDL library directly, which I found difficult to learn, so I have used the classes "Hardware", "GraphicsElement" and "Font" that we had used previously in class games (...)

Although in the level 3 (hard) I check the next three movements, sometimes the computer makes foolish moves, but always in the first column. I have not found where the bug is, but I think the problem may lie in this portion of code, in which I check each column versus the previous one: (...)

I noticed that if I take the game to another computer which has not the DotNet Framework installed in it, the program closes without any error or warning message (...)

**(…)**

## 8. Improvements or restrictions to the starting design

I have had time enough to include the effect of animation when the pieces fall (…)

I have not been able to implement (…) because of (…)

## 9. Screenshots of the final project

(…)

## 10. Source code of the final project

(…)