**Computer programming**

**Final Project**

Report

I.E.S. San Vicente

San Vicente del Raspeig (Alicante)

2017/2018

Teachers:

José Ignacio Cabanes / Ignacio Iborra

Student:

Almudena López Sánchez

* 1. **1. Introduction**

**Project name**

Monopoly

**Made by**

Almudena López Sánchez

**Short description of the project**

This is the “Monopoly” game from two to four players, it will be by turns. It is a graphical application that uses the SDL graphics library.

* 1. **2. Functionality of the project**

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

• New game

• Load

• Credits

• Quit

If the user chooses a new game, then two modes will appear: vs AI or local mode from two to four players. If you choose local mode will be asked the amount of players. In any case, the game will begin with the player 1, in each turn the player can: throw dices, buy the street where the player is or pay the owner and buy houses if the player has all the necessary streets. The game ends when everyone goes bankrupt except one.

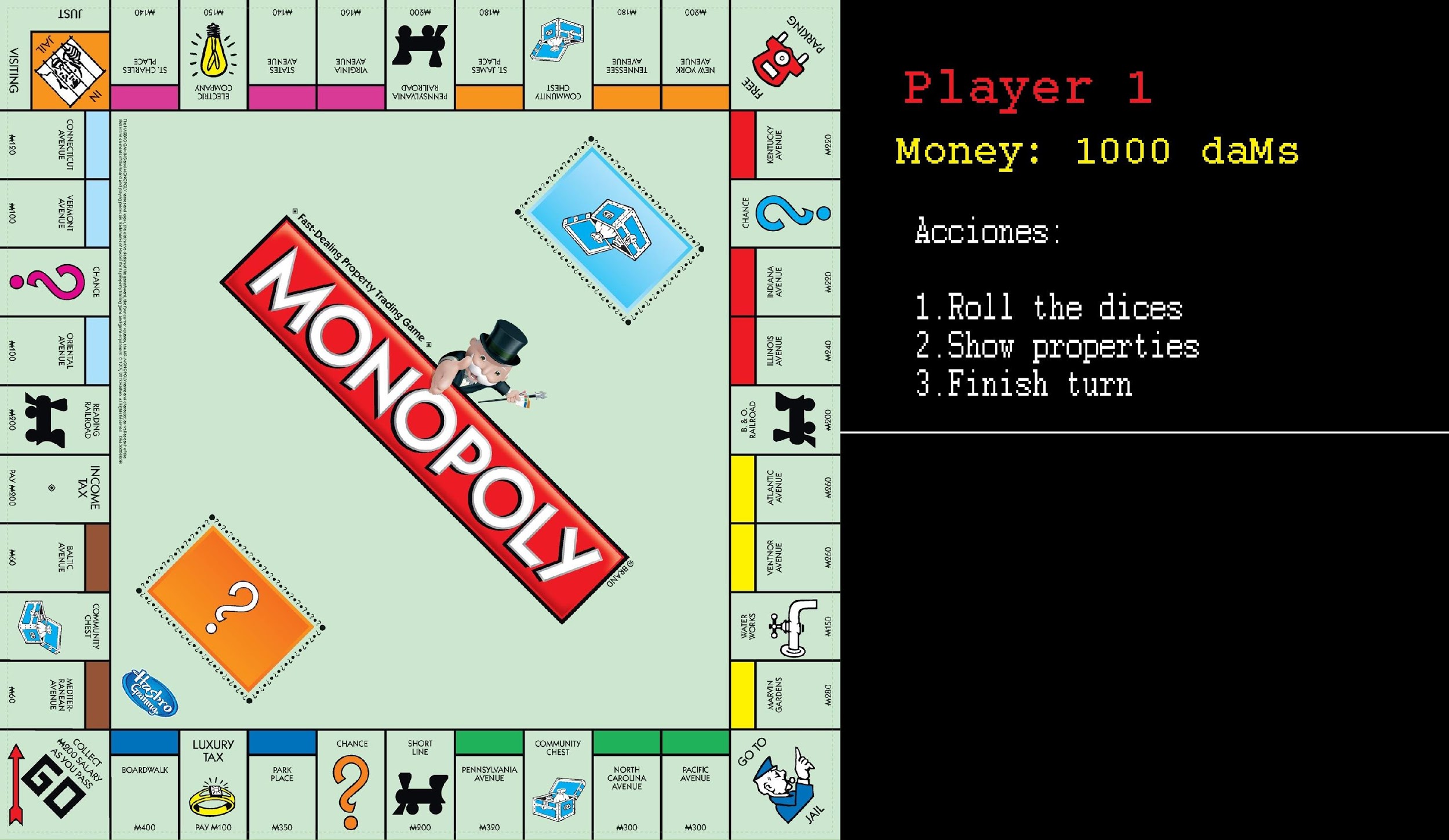
If the user chooses “Credits” will appear the credit screen.

Before quitting, the game will be saved, and it can be retrieved in the next session with the option “Load”.

The images and the background screen are read from file at the beginning of the game.

* 1. **3. Screen prototype**

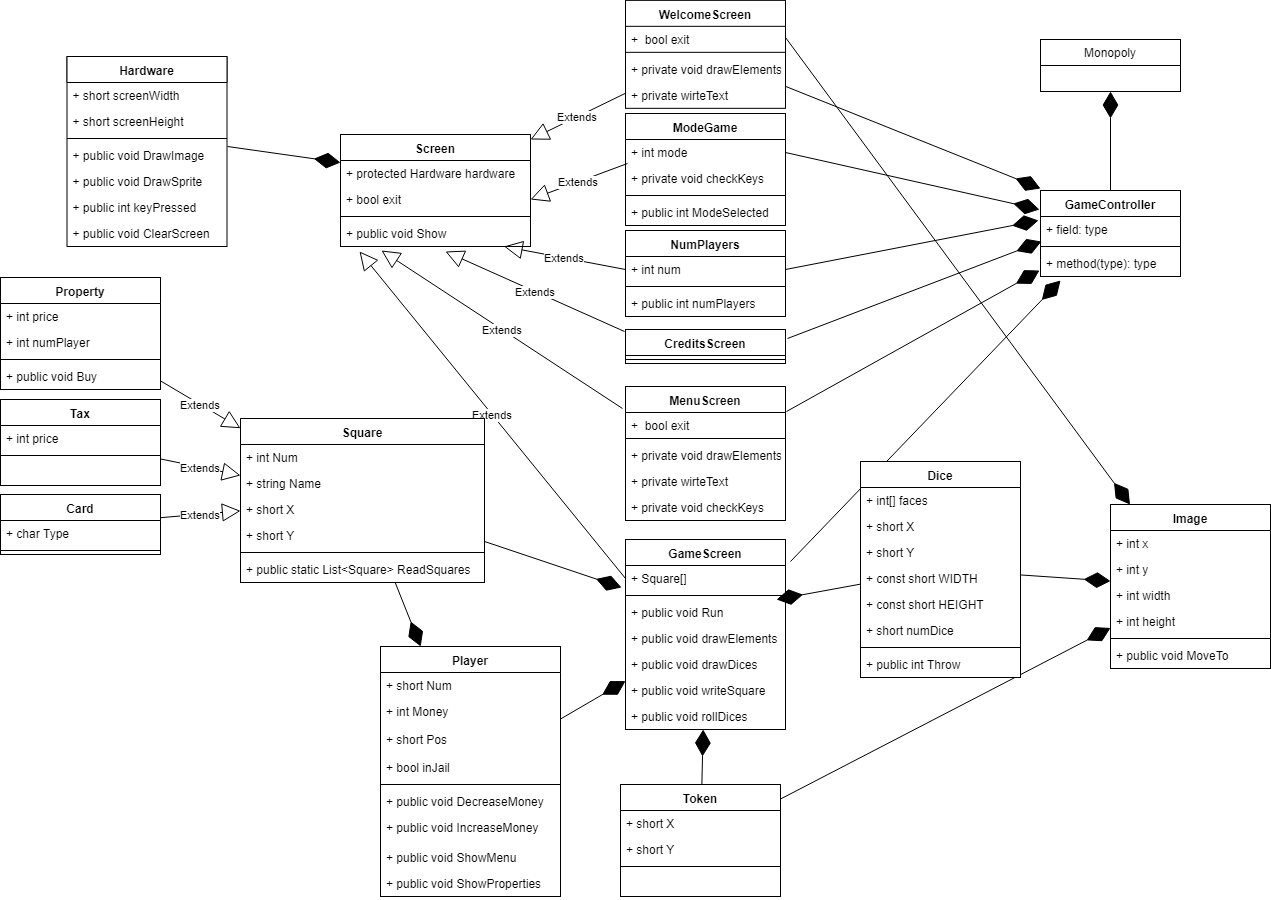
The game screen will look like this:



* 1. **4.Analysis**
     1. **4a. Requisites**

|  |  |
| --- | --- |
| Requisite | Date achieved |
| The program will allow a player to play against the computer, the ai will buy all the streets in which it falls if it has money. |  |
| The program will allow two to four players to play with turns. | X |
| The game will start showing an intro screen, when user press “space” will show menu screen. | X |
| The game will can be saved and user will can load it in the menu screen. |  |
| The menu screen will allow he user to enter a Credits screen, which will display data about the programmer. | X |
| The game will use SDL graphics library. | X |
| The game will allow each player to buy streets, houses and hotels. |  |
| Each player will have a menu with three options: Roll the dices, show properties and finish turn. | X |
| Each player will can choose the name and 1 token. |  |
| Each player will pay when he will be in a square with owner. |  |
| Each player will can sell in his turn. |  |

* + 1. **4c. Classes diagram**



**5. Initial planning and expected deliveries**

**5a. Expected deliveries**

* **Version 1:** Show board, 1 token and the dices in the screen show two random numbers.
* **Version 2:** Create a class player and show the menu’s player in the screen. Menu’s player will have 3 options: 1.- Roll the dices, 2.- Show properties, 3.- Finish turn. Make functional “Roll the dices” moving token over squares with an array of classes.
* **Version 3:** Read file of squares and fill the array of them. Make different classes depending of the type of the squares (do inheritance). Display the data on the screen of the square which player falls.
* **Version 4:** Make to buy streets, stations and utilities when the player is in it and it will appear as the property of the player. (In the option: “2.- Show properties”)
* **Version 5:** Welcome screen and menu screen with 4 options: New game, Load, Credits and Quit. Make functional “New game” and “Quit”.
* **Version 6:** Player increase or decrease his money when he is in a determinate square.
* **Version 7:** Add screen to select the number of players and change players in game (Make turns).
* **Version 8:** Pay when a player falls in in a street with owner.
* **Version 9:** Credit screen. Add more tokens.
* **Version 10:** Save and load game.
* **Version 11:** Read file of cards (chance and community cards) and add to array of structs. When the player gets one of them do actions that it says.
* **Version 12:** Add one token for each player and make it able to choose one of them and set the name of the player.
* **Version 13:** Make player vs AI option, and screen to choose it. If the AI have enough money when it falls in a property it will buy always. Do logical of jail.
* **Version 14:** When the player has all the streets of a colour make to buy houses and hotels (another option in the player’s menu: “Buy house”).
* **Version 15:** Sell houses, hotels and streets. The player will can to sell houses and hotels to the bank. The streets to another player. (Another option in the player’s menu: “Sell”)
* **Version 16:** Make players lose when they can not pay. End the game when only one player will be in game.
* **Version 17:** Make a screen between number players screen and the game. In this screen you will can choose the player’s name and your token.

**5b. Real deliveries**

* **Version 1 (May 14):** As expected: Show board, 1 token and the dices in the screen show two random numbers when user press “1”.
* **Version 2 (May 15):** As expected: Create a class player and show the menu’s player in the screen. Menu’s player will have 3 options: 1.- Roll the dices, 2.- Show properties, 3.- Finish turn. Make functional “Roll the dices” moving token over squares. (I prefer use a list of squares).
* **Version 3(May 16):** Read file of squares and fill the array of them. Display the data on the screen of the square which player falls.

Classify depending on type of square is left. (Create different types of classes)

* **Version 4(May 17):** As expected (5th deliver)
* **Version 5(May 18):** Created different types of classes and classified them
* **Version 6(May 21):** Player increase or decrease his money when he is in a determinate square.
* **Version 7(May 22):** Change players in game (Make turns), made to buy and show properties.

To Do: screen of select number of players.

* **Version 8(May 23):** Corrections in the menuToBuy and gameScreen
* **Version 9(May 24):** Pay when a player falls in a street with owner. Added class numPlayersScreen.
* **Version 10(May 25):** Added LoadScreen, change menu and display all images and texts.

## 6. File formats

### 6a. Plain files format

File Squares:

Separated by ‘-’. Its order like:

Street: num-name-price-colour

Station: num-name-price-type

Utilities: num-name-price-type

Cards: num-name-type

Tax boxes: num-name-price

start, jail, parking, go to jail: num-name

These squares are classificated like:

Property: street, station, utilities

Tax : tax boxes

Card: cards (change/comunity chest)

Example:

1-start

2-MEDITERRANEAN AVENUE-60-brown

3-community chest-CC

4-BALTIC ANENUE-60-brown

5-income tax-200

...