Functional Requirements.

- **FR1. Deploy a menu with three options.**

The program must start with a 3-option menu, one option to play, other one to show a list of players and scores, and the last one to exit the program.

- **FR2. Read Name of the player, size of matrix, and number of mirrors.**

The program must allow the user to enter the name of the player when starting a new game, followed by the size of the matrix and the number of mirrors present in the matrix. This information must be read by following the given format: name n m k. where n is the number of rows, m the number of columns, an k the number of mirrors.

- **FR3. Create a (n*m) size table with k mirrors.**

When a new game it´s started the program must create a board made by n*m cells, and randomly assign k mirrors in a random direction, right "/" or left "\".

- **FR4. fire a laser horizontally or vertically from any cell on the edge of the grid.**

The program must allow the user to select a cell located on the edge of the grid to shoot the laser, if the selected cell is located on a corner, the user must specify whether the laser is going vertically or horizontally, following the given format: rowColumn, ex.1B, if it's not shot from a corner it should be: rowColumnDirection, ex. 1AH or 1AV.

- **FR5. Locate a cell with a mirror.**

The program must allow the user to locate a cell with a mirror by entering the indicator of a search, the location to search, and the direction of the mirror in that cell following the given format: LRowColumnR or LRowColumnR. Ex. L1AR or L1AL.

- **FR6. Display board.**

When a new game is created, and every time an action is taken while playing, the program should display the complete board with each cell represented by "[ ]". After a shot is made the board should display the cell where the laser was shot filled whit the letter 'S' ex. [S], and where the laser came out with the letter 'E' ex. [E], in the case the cell is the same for start and finish, should fill the cell with the letter ‘M’ ex. [M]. if previously an attempt to locate a cell with a mirror failed, it should fill the searched cell with the letter ‘X’ ex. [X], and if the search was successful, the menu should continuously show the mirror of that cell. Ex. [/] or [\].

- **FR7 show nickname and mirrors remaining.**

While playing, when the board is shown, it should display the name of the player and the number of mirrors left to find.

- **FR8. Finish Current game.**

The user must be able to enter the word “menu”, to finish the current game at any time and return to the main menu. If the user finds all the mirrors in the board, it wins the game and returns the user to the main menu.

- **FR10.Display finished game configuration.**

After a game is finished the program should display the players name, the score, failed attempts to find mirrors, the size of the board, and the number of mirrors.

- **FR10. Save players information.**

After a game is played the program must save the name of the player, the score and the game configuration (n,m,k). the players are saved in a tree sorted by the scores.

- **FR11. List players and scores.**

The program must display all the players and scores by moving in in Order trough the tree that contains the players.