

Functional Requirements

FR1: Display a matrix with user specified dimensions that contains “mirrors” that are initially hidden from the user. These cells will be shown as a pair of brackets with a space in between. Each cell must be identified by a letter and a number that are outside of the matrix, in a similar fashion to battleship.

FR2: Generate a user-specified number of mirrors on the matrix in random positions and random orientations. These will be represented with either a slash or a backslash, depending of their orientation. These mirrors will deflect the “laser”.

FR3: Allow the user to “fire” a laser, which will travel through the matrix and be deflected by any mirrors it collides with along the way. The user must specify the cell the laser will be fired from. The direction the laser will be deflected to depends on both the current direction of the laser and the orientation of the mirror. The deflection’s direction is determined by the following table.

Current direction	Slash mirror (/)	Backslash mirror (\)
Up	Right	Left
Down	Left	Right
Left	Down	Up
Right	Up	Down

After the shot, the matrix will display both the cell the laser was fired from and the cell it exited from, with an S and an E, respectively (only E will be displayed if these two cells are the same). The laser can only be fired from the edges of the matrix, and if the cell is a corner, the user must specify whether the shot will be fired vertically or horizontally.

FR4: Allow the user to take a guess at where a mirror is, followed by its orientation. To do this, the user must enter an L followed by (and not separated by a space) either L or R (left or right). If the user fails to guess any of the two, an X will be displayed in that position.

FR5: Display the user’s nickname along with the number of mirrors remaining to find every time the matrix is displayed.

FR6: Allow the user to return to the main menu at any time by typing “menu”.

FR7: Calculate and store the game’s score (regardless of whether the game was completed or not) and allow the user to see a list of previous scores ordered from greatest to smallest.

FR8: Display a main menu with options to play, see previous scores, see the instructions and exit.

Non-functional requirements

- The matrix can have only a maximum of 26 columns, as these are represented by English letters, which there are only 26 of. Also, the matrix must have a minimum of one row and one column.
- The number of mirrors cannot be greater than the number of cells in the matrix.
- The matrix consists of a multiply linked list. Each cell has a relation with every other adjacent cell.

- The scores must be stored in a binary tree and it has to be displayed in order by being traversed in inorder.
- The implementation must not include any kind of array, java collection or cycle (Except for the array returned by splitting a string).

Class Diagram

