

Introduction

In this project you put in practice the previous knowledge of JS to develop a game using HTML, CSS & JS.

What are the main objectives in this project?

- Improve your knowledge of **JavaScript**
- Learn and improve your knowledge in logic processes
- Improve your knowledge working with **HTML, CSS** and **JS** together.

1. General analysis

In this pill you will develop several games with JavaScript to improve your skills and learn how to develop games with JavaScript and the browser.

2. Develop a game

This project must be developed in 4 teams, each one must make one of the following games that are explained below:

Game 1: Hangman

In this game you will play Hangman in which you will have to discover the hidden word by choosing letters.

Phase 1

You will have to guess a word by clicking the letter boxes on the screen before the hangman is drawn on the screen.

- You will have to display on the screen the letters of the alphabet in div boxes so that each letter is displayed in an element on the screen, and then when the div is displayed, you will have to compare the letter in the div to see if the word contains that letter.
- Once a letter is clicked you will have to remove the letter from the screen.
- If the letter is in the word, you will have to add the letter to the results.
- If the letter is present more than once in the word you will have to add it in each position.
- If the letter is not in the word, you will have to draw a part of the hangman.
- You will have to count the number of letters that are clicked to stop the game once the hangman has been completed.
- Once the hangman has been completed, if the user didn't guess the word, it means that the user lost.
- If the user guesses the word before the hangman has been completed, the user has won and the game ends.
- You can store the words that are used in an array of words.
- The words should have different lengths so that each round has a word that is longer than the previous word.

Chose a user name

User name

Start

User Scores

Mark

Currently playing...

Alex

25 seconds

Ana

13 seconds

Andrew

25 seconds

Guess the word

— — — — —

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| Ñ | O | P | Q | R | S | T | U | V | W | X | Y | Z | |

User Scores

Mark

Currently playing...

Alex

25 seconds

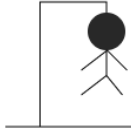
Ana

13 seconds

Andrew

25 seconds

You lost



Guess the word

R E S _ _



User Scores

Mark

Currently playing...

Alex

25 seconds

Ana

13 seconds

Andrew

25 seconds

Your won! 🎉

You won in 15 seconds.

Play again

User Scores

Mark

15 seconds

Alex

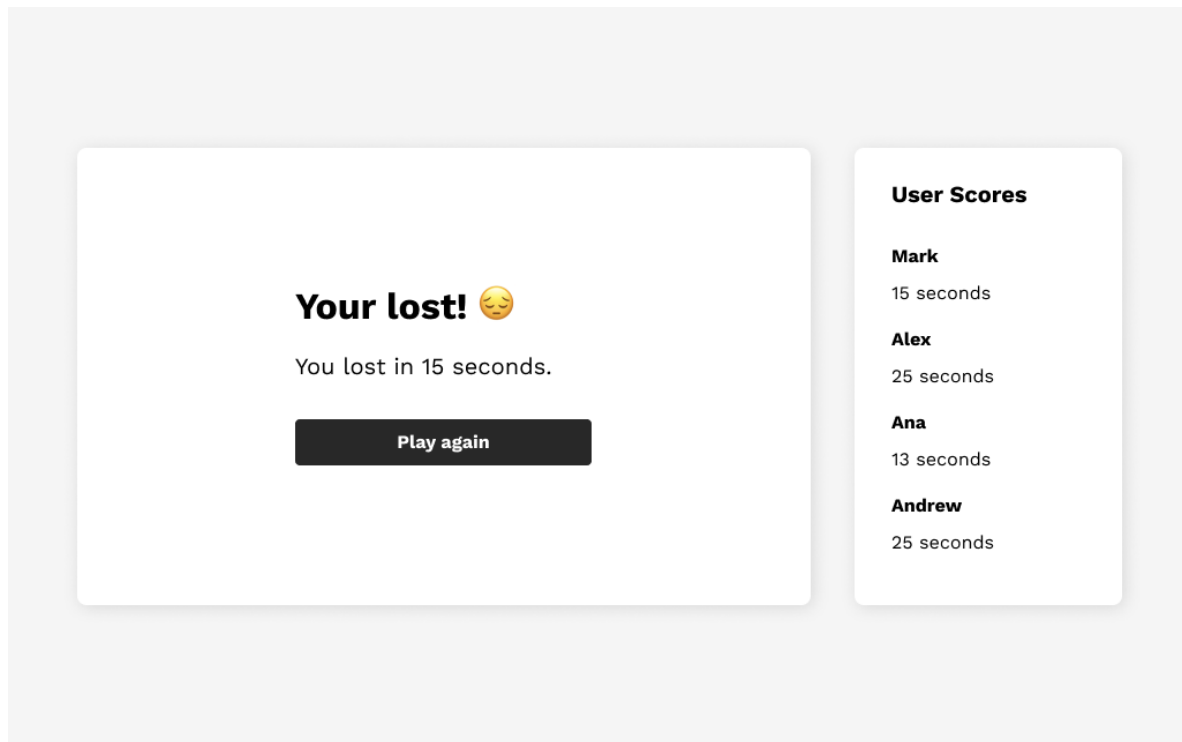
25 seconds

Ana

13 seconds

Andrew

25 seconds



Phase 2

After each round of the game, you can choose a word that is longer than the previous one so that after each round, the words are longer than before.

Extra Steps

1. All the game interaction should be usable with only the keyboard
2. Improve the design such that it is designed using a mobile-first strategy and responsive

Game 2: Clicks in 10 seconds

This game consists of measuring the number of clicks you can make in a 10 second period.

Phase 1

The game will have a start button that you will have to press before starting the game. Then, once the button is pressed and the game begins, you will have to create a timer to stop the game after 10 seconds have passed since the “Start game” button was pressed.

Once the game has started, you will also have to count the number of clicks that you make on a box on the screen.

Once the 10 seconds have passed, the game ends and stops counting the number of clicks on the box and displays the result to the user.

Chose a user name

User name

User Scores

Mark
Currently playing

Alex
4 clicks

Ana
20 clicks

Andrew
44 clicks

Start game

User Scores

Mark

Currently playing

Alex

4 clicks

Ana

20 clicks

Andrew

44 clicks

Click here

User Scores

Mark

Currently playing

Alex

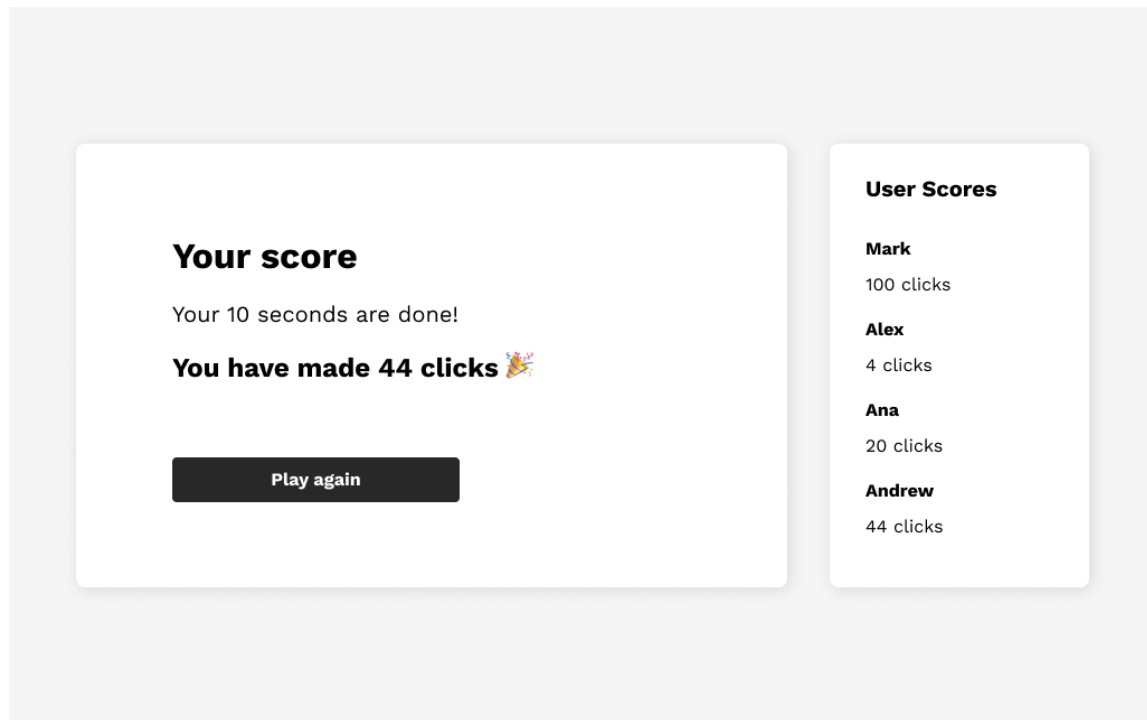
4 clicks

Ana

20 clicks

Andrew

44 clicks



Phase 2

In this phase the “Click here” box should move around the screen to a random position after each click and should change the size of the box.

Extra Steps

1. Possibility to play with mouse click or spacebar. (Different user scores)
2. Improve the design such that it is designed using a mobile-first strategy and responsive

Game 3: Test your reflexes

In this game you will test your reflexes and reaction time.

The game consists of measuring the time it takes you to press a button once a timer has started.

The game will have a "Start Game" button that, once pressed, will display a message on the screen with the text of "Get ready..." for a random amount of seconds (from 0 to 10 seconds) and then it displays a "Stop game" button that you will have to press as quickly as you can to measure the time it takes you to press the button, this way you can measure your reaction time.

Once the "Get ready..." message disappears and the "Stop Game" button appears, you will have to start a timer using JavaScript dates to measure the time between the moment the "Stop Game" button has been displayed on the screen and the moment the "Stop Game" button is clicked to stop the game.

Once the button is clicked, you will have to stop the time counter and display the amount of seconds it took you to press the button to display the time on the screen and in the right hand panel with the username and the time it took the user to press the "Stop Game" button.

Chose a user name

User name

Start

User Scores

Mark

Currently playing

Alex

4 seconds

Ana

5 seconds

Andrew

2.4 seconds

Start game

User Scores

Mark

Currently playing

Alex

4 seconds

Ana

5 seconds

Andrew

2.4 seconds

Get ready...

User Scores

Mark

Currently playing

Alex

4 seconds

Ana

5 seconds

Andrew

2.4 seconds

Stop game

User Scores

Mark

Currently playing

Alex

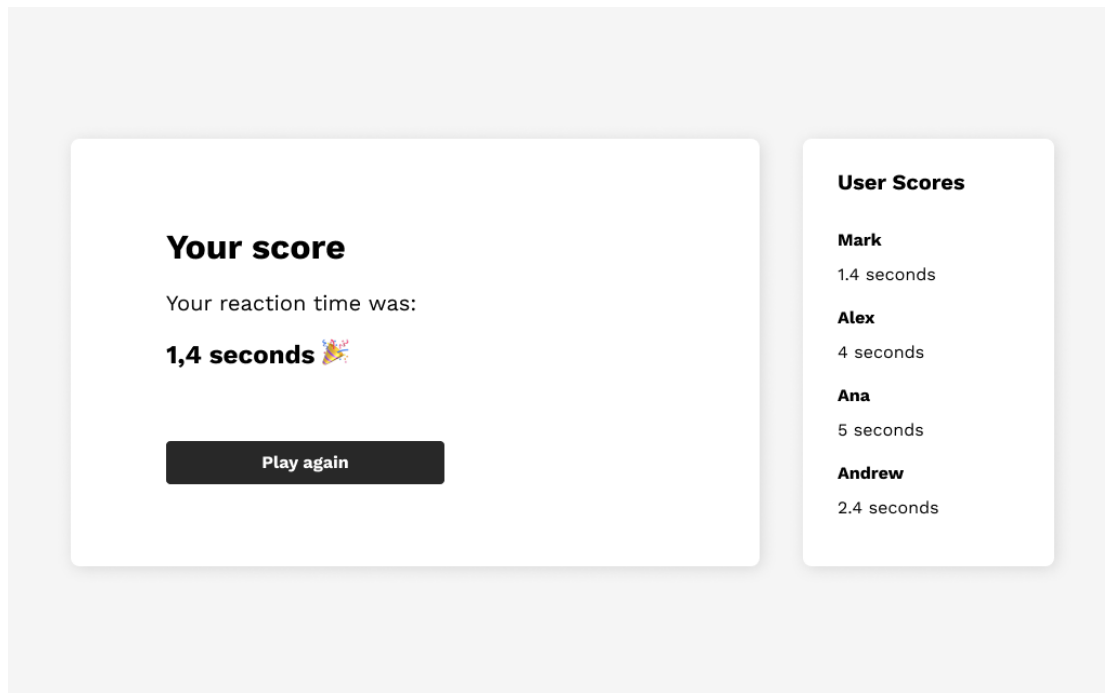
4 seconds

Ana

5 seconds

Andrew

2.4 seconds



Phase 2

In this phase the stop game button should appear in a random position on the screen for each round and not always in the same place.

Extra Steps

1. Every second that passes without clicking the position changes randomly
2. Improve the design such that it is designed using a mobile-first strategy and responsive

Game 4: Memory

In this game you must test your memory. In order to develop the game you will need 16 images, of which half will be repeated, so that each image will have its identical partner.

Phase 1

Chose a user name

User name

User Scores

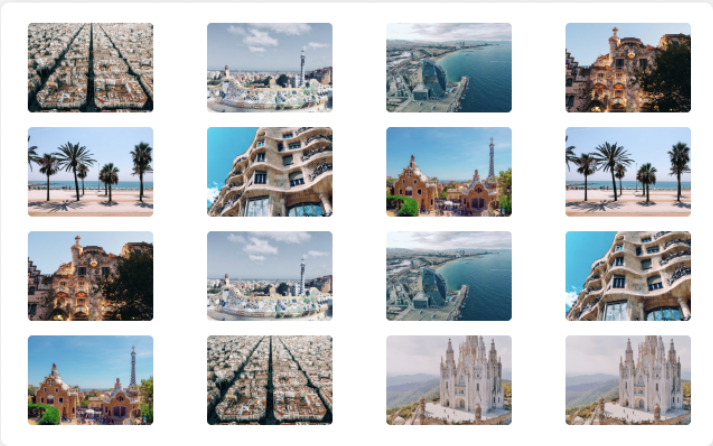
Mark
Currently playing...

Alex
25 seconds

Ana
13 seconds

Andrew
25 seconds

Guess the right card



User Scores

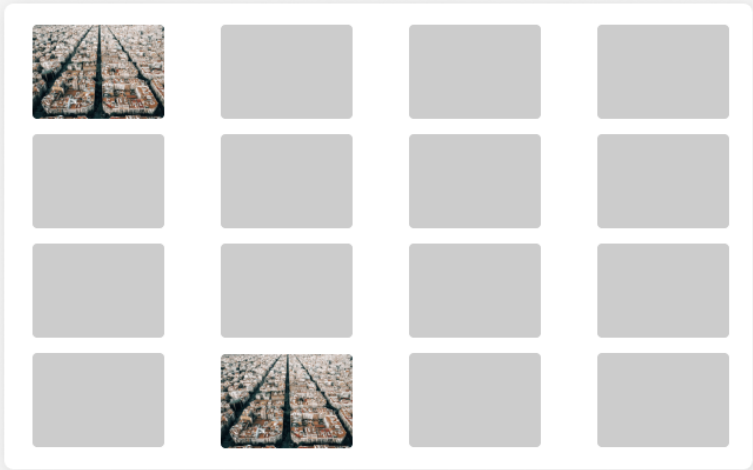
Mark
Currently playing...

Alex
25 seconds

Ana
13 seconds

Andrew
25 seconds

Guess the right card



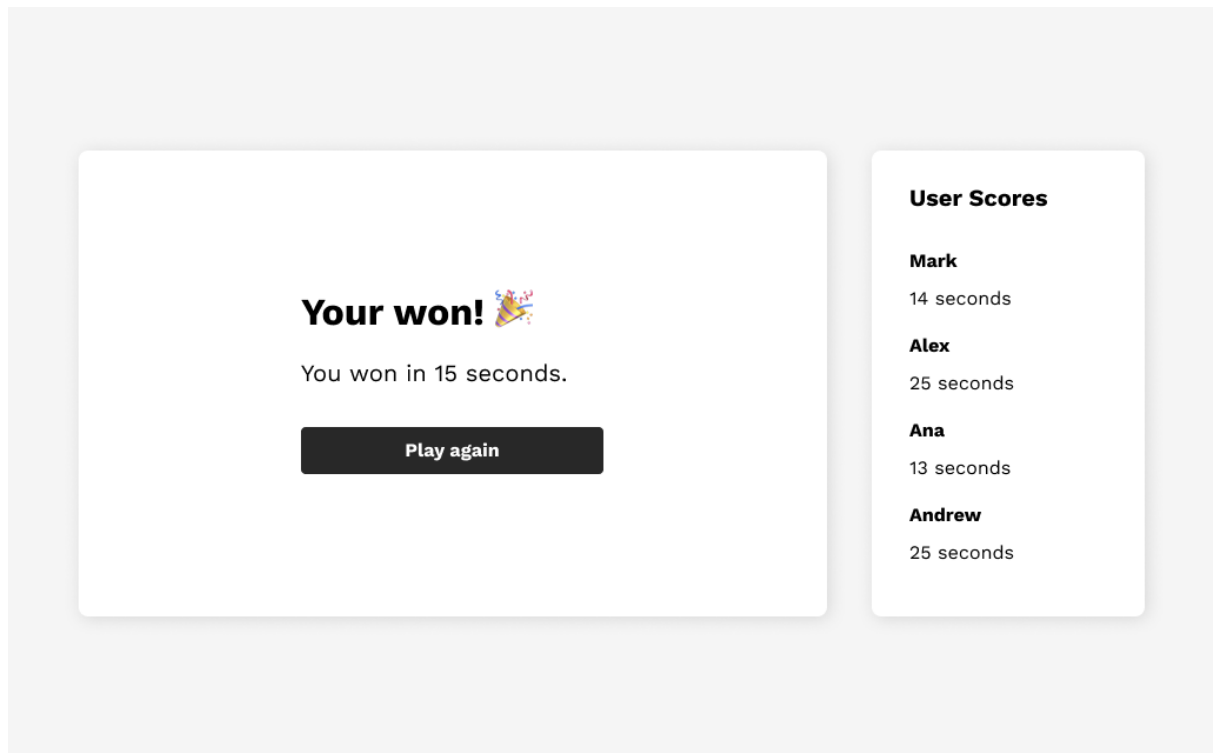
User Scores

Mark
Currently playing...

Alex
25 seconds

Ana
13 seconds

Andrew
25 seconds



When the game starts, all 16 images should be displayed randomly on a 4x4 grid for 3 seconds.

Once these 3 seconds have elapsed, the images should be hidden as if it were a board game and you had turned them over. In this way you will be able to see these 16 elements in the same order that they were but without showing what image is in each cell.

In phase one you will have to click images two by two to know if they are identical, in the negative case the current two selected images are hidden and you continue with the game. If the images that you clicked are identical they will be discovered during the game.

You win if you match all the images on the board. The score is calculated with the time that you spend to win the game.

Phase 2

In phase two you must add a second game mode, by entering your username to start the game you can choose easy or hard mode. The easy mode will be the previous mode (phase 1).

In this hard mode you will have to solve all the pairs without making a mistake, once you make a mistake showing two cards and that they are not the same, you will lose the game.

Extra Steps

1. All the game interaction should be usable with only the keyboard
2. Improve the design such that it is designed using a mobile-first strategy and responsive

3. Requirements

- You must use only one HTML file to develop this game, this means that you must hide and show dynamic elements without reloading the page.
- Before starting the game, you must enter your username on the screen to start the game.
- The ranking of the players who have played a game
 - Must always be visible on screen
 - Refreshing the page will lose the information stored in the ranking, since the information must be stored in JS variables
 - When a player enters his username and clicks on the start button, the player must be added to the ranking with the message "Currently playing ..."
 - When this player finishes the game, the ranking will be dynamically updated to show your score instead of the previous message

- When the player finishes the game, the Scoring screen will be displayed with the results of the game.
- All game processes must be carried out without reloading the page
- You will not be able to use cookies, localStorage or any similar technology
- You must store:
 - Current game
 - User name
 - Game duration
 - Scoring of the current game
 - Ranking
 - User name
 - Scoring
- The ranking will show above the last record that has been entered
- After each game the total score is stored (if you refresh the browser is lost)
- You cannot use third-party libraries
- You must use Git for the development of the project and branches for each feature, try not to separate branches by person

4. Deliverables

To evaluate the project you will need the following deliverables:

- Repository with the code that includes also the documentation bellow
- A **presentation** in **Google Slides** explaining:
 - Explain what lessons you've learned during this project
 - How have you decided to organize the three stages of the game that you have had to develop?

5. Resources

- License free images <https://unsplash.com/>