

Guess the number game web app

Development
process



TABLE OF CONTENTS

01

Implementing the web app
basic/initial structure

02

Adding JavaScript logics to
manipulate the DOM dynamically

03

Making the final adjustments for the
visual

04

Finalizing code refactoring and
running final tests

05

Deploying the web app online

06

Completing the README document



Implementing the web app
basic/initial structure

Skills used

Code writing

WHY WAS THIS STEP IMPORTANT

This step was important to ensure that the initial structure of the web app was properly established (structure on which JavaScript would then be able to work with).

WHAT WAS THE GOAL

Ensure that all static content for the web app is present in the HTML and CSS documents, with appropriate class names and ids to allow easy reference to it later using JavaScript.

A yellow square containing the letters 'JS' in a bold, dark grey, sans-serif font, representing JavaScript.

JS

Adding JavaScript logics to
manipulate the DOM
dynamically

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

Implementing all the code to transform the previously static web app into an interactive and dynamic one was important to ensure it responds correctly to users actions.

WHAT WAS THE GOAL

- Create event listeners
- Create the function associated with each event listener
- Create conditional logics and supporting codes to handle every possible case users may face while playing:
 - No input
 - Wrong answer
 - Correct answer
 - Reset game
 - Game lost

LANDING VIEW

Again!

(Between 1 and 100)


Guess the number!



Check!

Start guessing...

~~100~~ Score: 100

 Highscore: 0

NO INPUTS

(DYNAMICALLY HANDLE VIA THE DOM)

Again!

(Between 1 and 100)


Guess the number!

?




Check!



No number 

 Score: 100

 Highscore: 0

WRONG ANSWER WITH NEW HINT

(DYNAMICALLY HANDLE VIA THE DOM)

Again!

(Between 1 and 100)

Guess the number!

?

63

Check!

Number too low ↓

~~100~~ Score: 98

🏆 Highscore: 0

*Score
decreases by
2 for every
wrong answer

WRONG ANSWER WITH NEW HINT

(DYNAMICALLY HANDLE VIA THE DOM)

Again!

(Between 1 and 100)

Guess the number!

?

89

Check!

Number too high 🔼

¹⁰⁰ Score: 96

🏆 Highscore: 90

***Score
decreases by
2 for every
wrong answer**

CORRECT ANSWER

(DYNAMICALLY HANDLE VIA THE DOM)

Again!

(Between 1 and 100)

Guess the number!

81

81

Check!

Correct number! 🎉

🏆 Score: 96

🥇 Highscore: 96

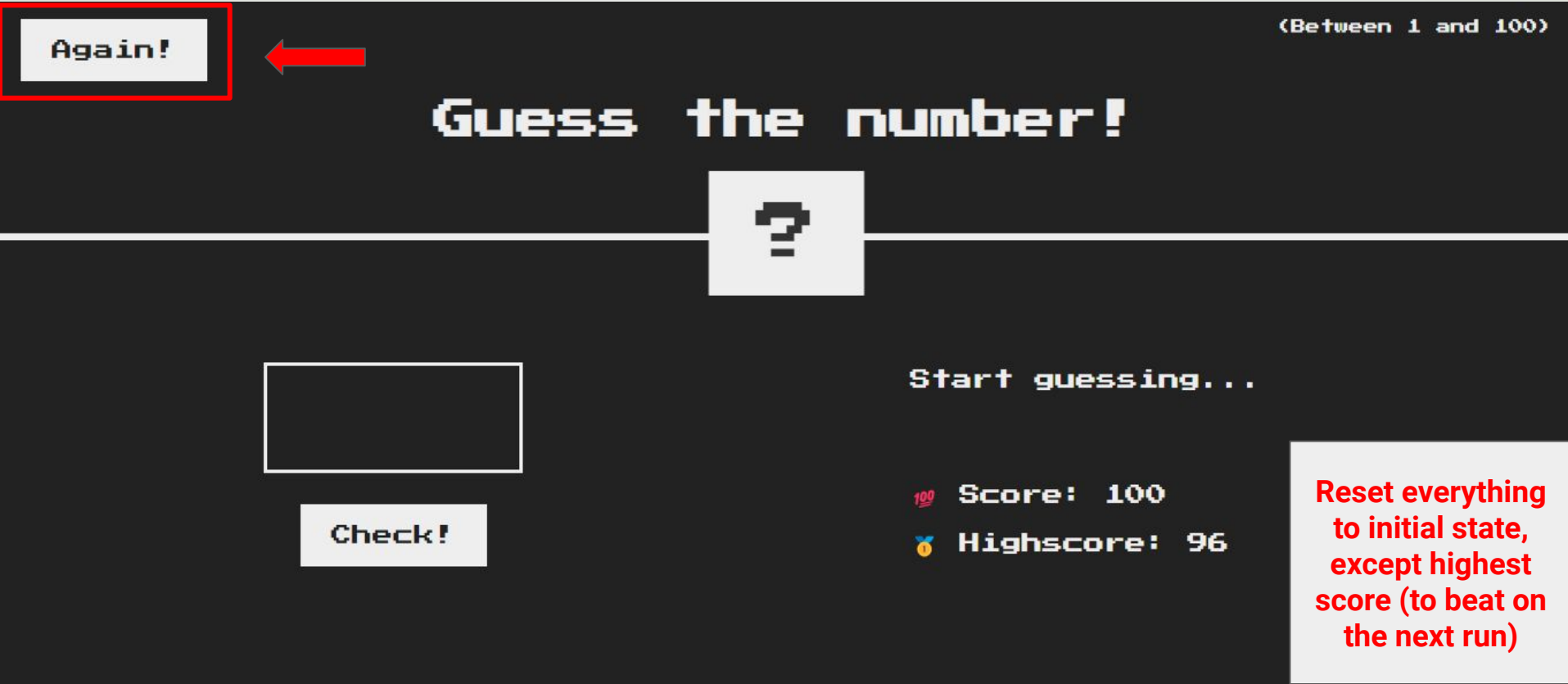
-Secret number revealed

-Congrats message

-High score set (if better than last run)

USER CLICKS ON AGAIN BUTTON

(DYNAMICALLY HANDLE VIA THE DOM)



GAME LOST

(DYNAMICALLY HANDLE VIA THE DOM)

Again!

(Between 1 and 100)

Guess the number!

?

7

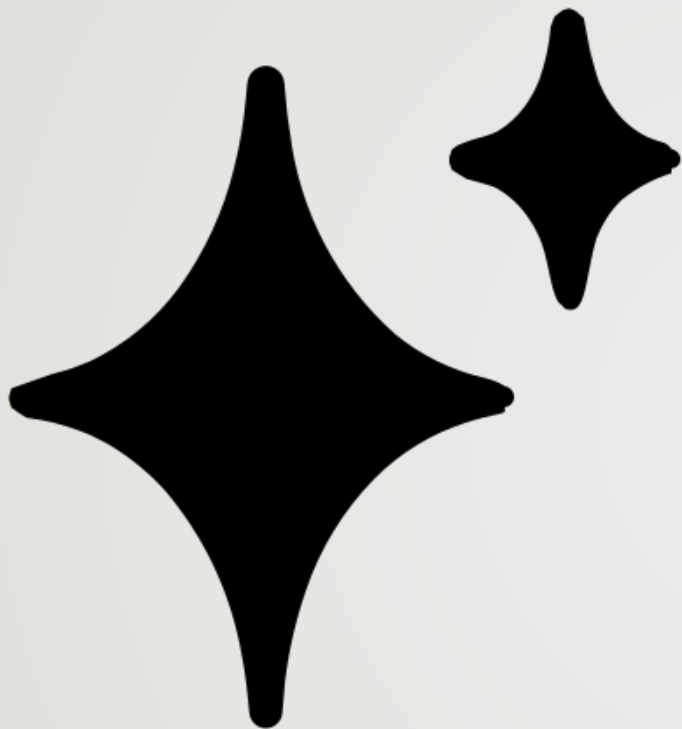
Check!

You lost the game 😞

¹⁰⁰ Score: 0

🏆 Highscore: 96

Game lost (if
users reach
score 0)



Making the final adjustments
for the visual

Skills used

Research
Code writing

WHY WAS THIS STEP IMPORTANT

This step was important to provide the web app a more engaging and dynamic background, and to create a more finished and visually attractive product.

WHAT WAS THE GOAL

Implementing an animated visual using an online CSS code generator, with special attention given to finding the right balance between a visually appealing background that is however not disturbing for users.

ANIMATED BACKGROUND

([see live version](#))

Guess the number!

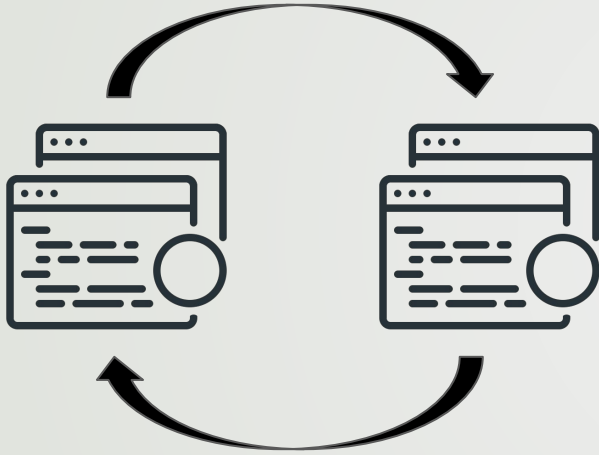
?

Check!

Start guessing...

¹⁰⁰ Score: 100

[🏆] Highscore: 0



Finalizing code refactoring
and **running** final tests

Skills used

Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

When possible, it is good practice to refactor codes. When there are a lot of duplicate codes, and some functionalities need to be changed for example, multiple identical updates may be necessary across various locations and files, leading to a potentially lengthy and error-prone process. Code refactoring helps prevent this by making the code cleaner, more logical, and more concise.

WHAT WAS THE GOAL

Replacing the previously duplicated lines of code to accomplish the same actions, but with fewer code lines, thus significantly reducing the length of the code. To do this, different functions were created.

Testing the game with all possible scenarios that users might encounter to ensure everything works as expected.

Fixing bugs as needed.

```
const displaySecretNumber = function (number) {  
  document.querySelector('.number').textContent = number  
}
```



github-pages

Deploying the web app online

Skills used

N.O.

WHY WAS THIS STEP IMPORTANT

This step was important to make the *Guess the number* web app publicly available by hosting it on GitHub Pages (gh-pages).



README .

md

Completing the README document

Skills used

Communication
Content writing

WHY WAS THIS STEP IMPORTANT

Ensure the project is well documented and easily accessible by anyone interested.

WHAT WAS THE GOAL

Updating and completing the README file located in the *Guess-number-game* Github repository. The goal was to ensure that all relevant information regarding this project is accessible under these three categories:

- Project description
- User interface
- Technical aspects

README SAMPLE - FULL VERSION ON GITHUB

