

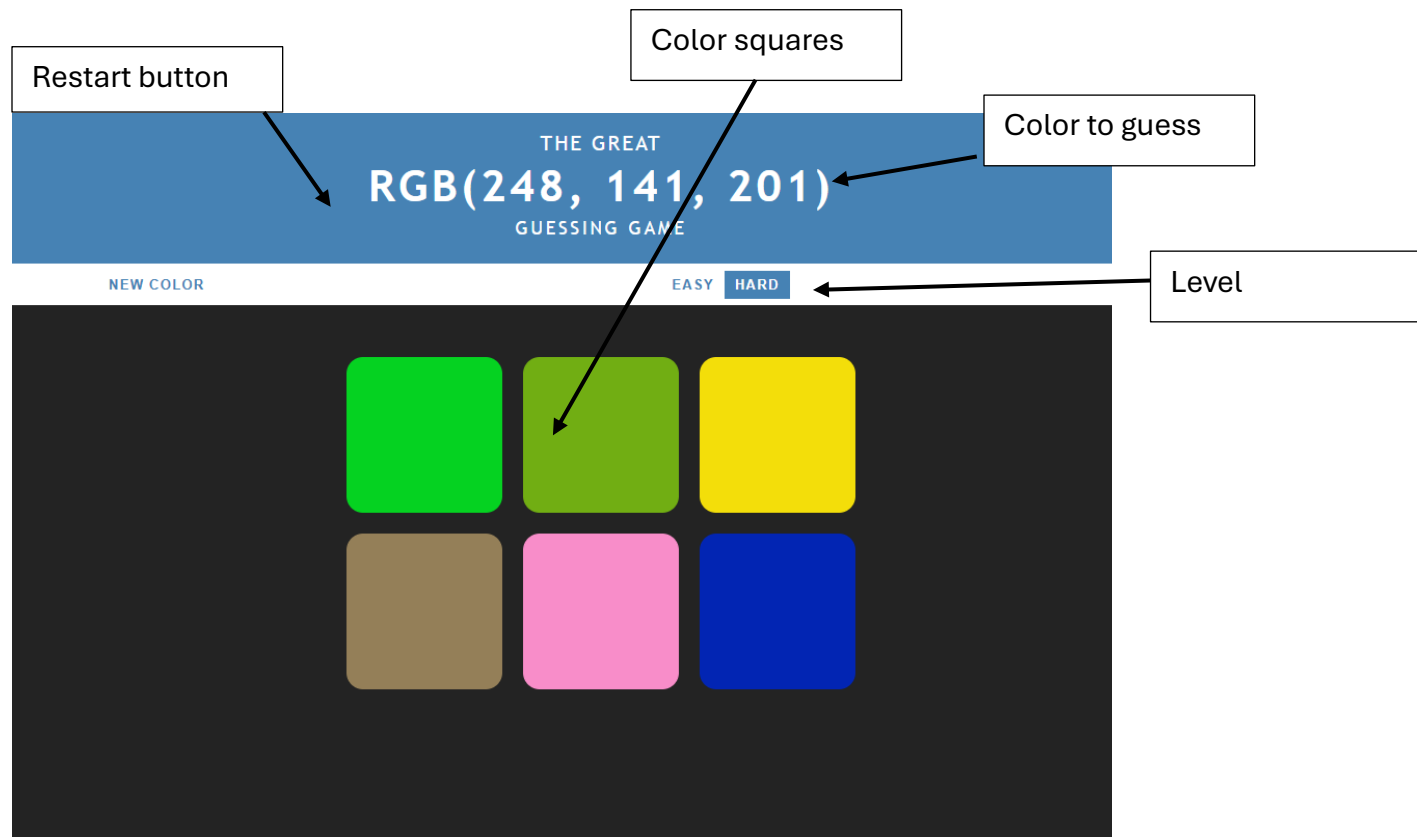
COSC 360 – Lab 5 (React)

RGB Color Guessing Game

In this lab, you will implement the RGB Color Guessing Game using **React**.

You will use React to modify the DOM, to create the RGB color guessing game. Every possible color can be represented with varying values of Red, Green and Blue (RGB) using numbers between 0 and 255. For example, Color white is RGB (0,0,0) where red is 0, green is 0 and blue is 0. Color Purple is RGB (255, 0, 255) where red is 255, green is 0 and blue is 255. Using different combinations of RGB values, we can generate any possible color.

In this game, you will generate 3(or 6) random colors using RGB values and represent them using color squares. You will then pick one of the colors for the user to guess. The user will click one of the squares to guess the color. If the guess is correct, you will display a success message, if it is wrong you will display an error message, allowing the user to try again till they get it correct. After each correct guess, the user can reset the game to generate a new color and play again.



Files Provided

You are given:

- App.jsx (starter JSX structure)
- App.css (styling – **do not modify**)
- index.html (React entry point – **do not modify**)

You must **not**:

- modify CSS
- query or manipulate the DOM directly
- use jQuery or any other framework except React.

Requirements

1. State

Use React state to store:

- number of squares (3 or 6)
- array of colors
- picked color
- status message
- header background color
- reset button text

2. Color generation

- Create a function that generates random RGB colors
- RGB values must be between 0–255
- Colors must be stored as strings in the form: `rgb(123, 45, 67)`

3. Game reset

When the game resets:

- new colors are generated
- a random color is selected as the target
- the header color resets to `#4682b4`
- status text is cleared
- button text is set to **NEW COLOR**

4. Difficulty

- Easy → 3 squares
- Hard → 6 squares
- The Hard level is selected by default

5. User interaction

- Clicking a square checks the guess
- Correct guess:
 - display **CORRECT**
 - change all squares and header to the correct color
 - change button text to **PLAY AGAIN?**

- Incorrect guess:
 - display **TRY AGAIN**
 - fade the selected square to #b3b3b3

6. React-only solution

- No document.querySelector
- No innerHTML
- UI updates must occur through React state

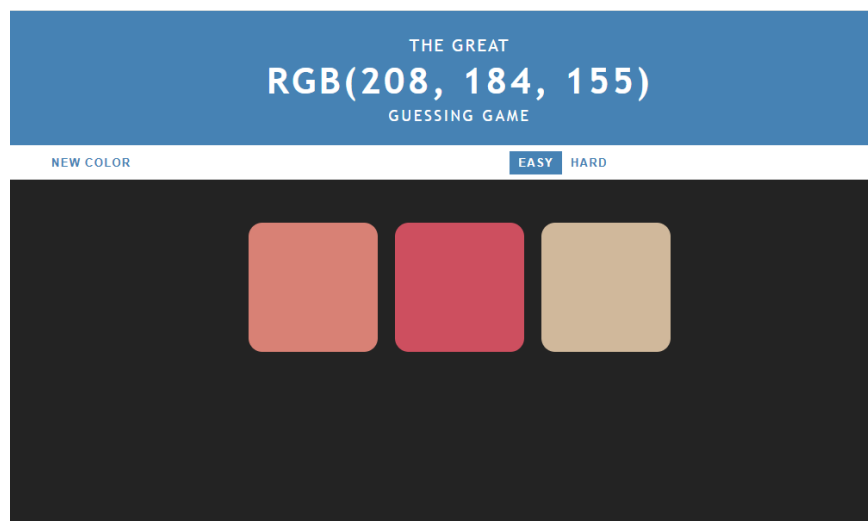
Submission

Submit your completed App.jsx file on Canvas.

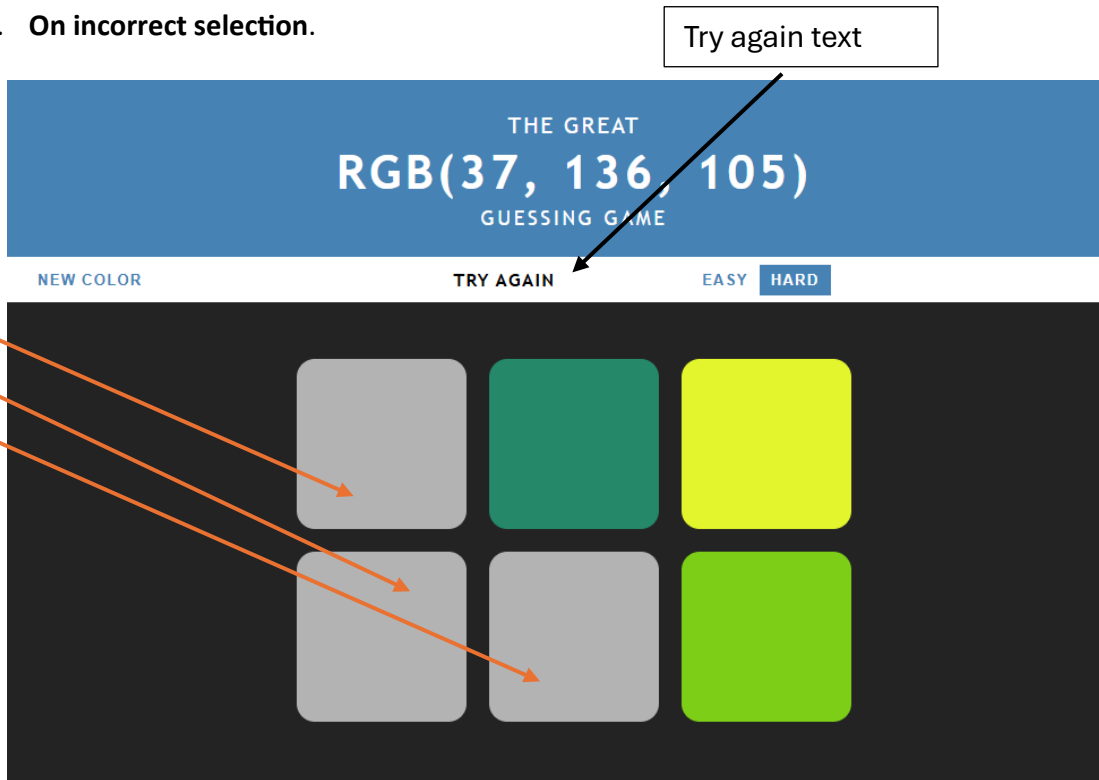
1. On game load or reset button click



2. On changing level – Easy level should have only 3 squares



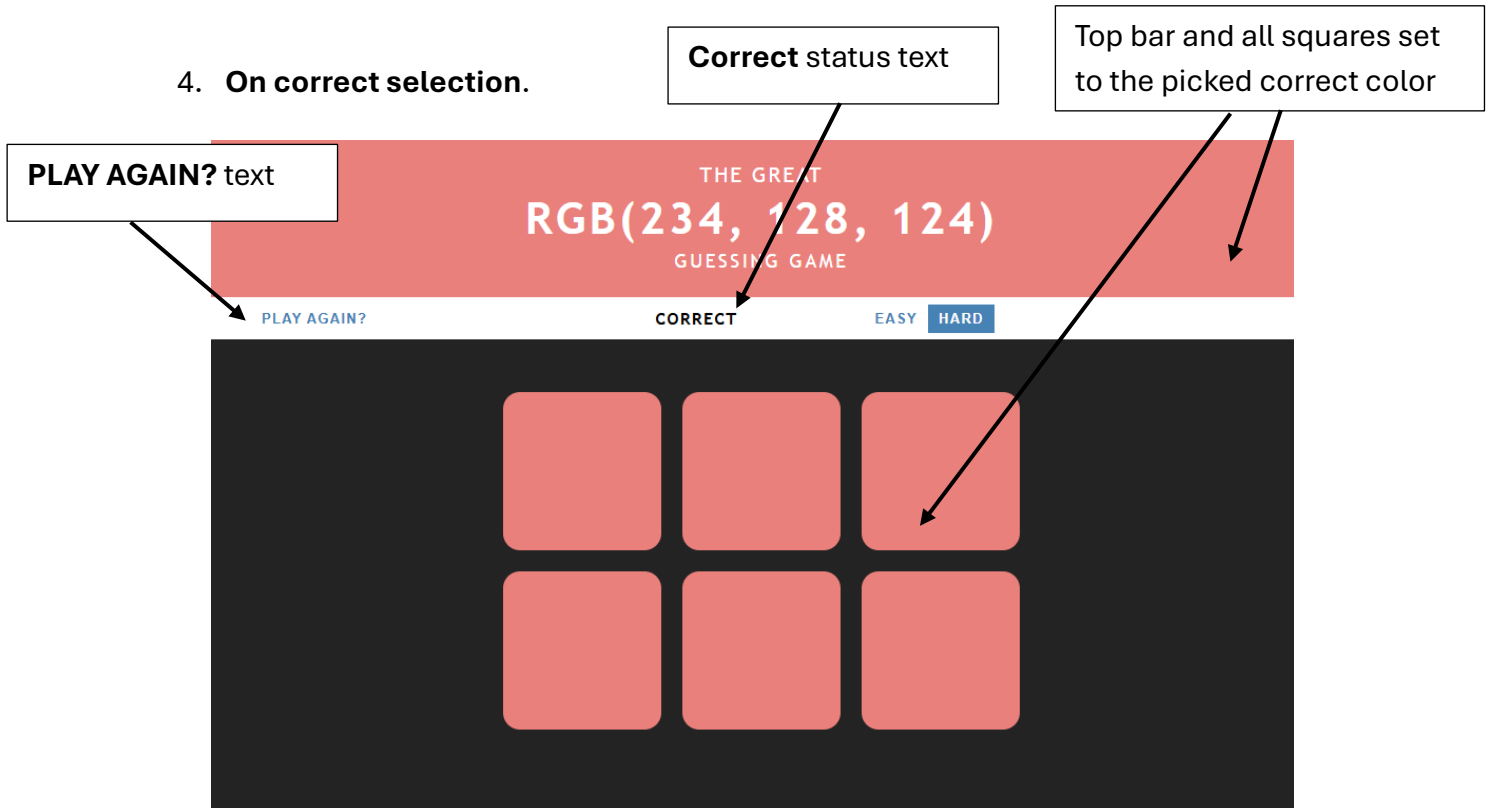
3. On incorrect selection.



Try again text

Incorrect selections faded to #b3b3b3 color

4. On correct selection.



Correct status text

Top bar and all squares set to the picked correct color

PLAY AGAIN? text