

Class Activity: Mini Game

Interactive Rock Paper Scissors – Play, Score & Win!

1. HTML Code:

Before:

```
<button class="btn" onclick="playerChoice('rock')">Rock</button>
<button class="btn" onclick="playerChoice('paper')">Paper</button>
<button class="btn" onclick="playerChoice('scissors')">Scissors</button>
```

After:

```
<button class="btn" onclick="playerChoice('rock')"><span
class="emoji">✊</span></button>
<button class="btn" onclick="playerChoice('paper')"><span
class="emoji">✋</span></button>
<button class="btn" onclick="playerChoice('scissors')"><span
class="emoji">✌</span></button>
```

Changes:

- Replaced text labels with emoji icons (✊, ✋, ✌) for a more engaging UI.
- Increased button size to make them more visually appealing.

2. CSS Code:

Before:

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
}
```

After:

```
@import url('https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap');

body {
    font-family: 'Press Start 2P', cursive;
    background: linear-gradient(135deg, #b35e3a, #d4a017, #2f9e55);
    text-align: center;
    color: white;
    overflow: hidden;
```

```
}
```

Changes:

- Changed font to "Press Start 2P" for a retro gaming feel.
- Added gradient background (#b35e3a, #d4a017, #2f9e55) for a funky color theme.
- Centered text and prevented scrolling with overflow: hidden;.

Before:

```
.btn {  
    background-color: #4CAF50;  
    color: white;  
    padding: 10px 20px;  
    font-size: 16px;  
    border: none;  
}
```

After:

```
.btn {  
    background: linear-gradient(135deg, #b32974, #2b82c9);  
    color: white;  
    border: none;  
    padding: 25px 35px;  
    margin: 20px;  
    font-size: 26px;  
    border-radius: 50%;  
    cursor: pointer;  
    transition: transform 0.3s, box-shadow 0.3s;  
    display: inline-flex;  
    align-items: center;  
    justify-content: center;  
}
```

Changes:

- Increased padding & font size for bigger buttons.
- Added border-radius: 50% to make buttons circular.
- Used a gradient background (#b32974, #2b82c9) instead of a solid color.
- Centered emoji icons inside buttons.

New Animations Added

```
.btn:hover {  
    transform: scale(1.3);  
    box-shadow: 0px 8px 25px rgba(179, 41, 116, 0.8);
```

```

}

.btn:active {
    animation: bounce 0.4s;
}

@keyframes bounce {
    0%, 100% { transform: translateY(0); }
    50% { transform: translateY(-15px); }
}

@keyframes fadeIn {
    from { opacity: 0; transform: scale(0.8); }
    to { opacity: 1; transform: scale(1); }
}

```

Changes:

- Buttons now grow in size (`scale(1.3)`) when hovered.
- Bounce effect added when buttons are clicked.
- Results now fade in smoothly.

3. JavaScript Code:

Before:

```
document.getElementById('result').innerText = `You chose ${choice}. Computer chose ${computerChoice}. ${resultMessage}`;
```

After:

```
document.getElementById('result').innerHTML = `You chose <b>${choice}</b>. <br> Computer chose <b>${computerChoice}</b>. <br><br> ${resultMessage}`;
resultDiv.style.animation = 'fadeIn 0.5s ease-in-out';
```

Changes:

- Used `` to bold the choices for better readability.
- Added line breaks `
` to space out result messages.
- Applied `fadeIn` animation for smoother result display.

New Scoreboard Feature

```
<div class="scoreboard">
    Player: <span id="playerScore">0</span> | Computer: <span id="computerScore">0</span>
</div>
```

Changes:

- Added a scoreboard to track wins & losses.
- Uses span to dynamically update player & computer scores.

New Score Update Logic in JavaScript

```
let playerScore = 0;
let computerScore = 0;

function playerChoice(choice) {
    const choices = ['rock', 'paper', 'scissors'];
    const computerChoice = choices[Math.floor(Math.random() * 3)];
    let resultMessage = '';

    if (choice === computerChoice) {
        resultMessage = "It's a tie!";
    } else if (
        (choice === 'rock' && computerChoice === 'scissors') ||
        (choice === 'paper' && computerChoice === 'rock') ||
        (choice === 'scissors' && computerChoice === 'paper')
    ) {
        resultMessage = 'You win! 🎉';
        playerScore++;
    } else {
        resultMessage = 'You lose! 😞';
        computerScore++;
    }

    document.getElementById('playerScore').innerText = playerScore;
    document.getElementById('computerScore').innerText = computerScore;
}
```

Changes:

- Added playerScore and computerScore variables.
- Updated score dynamically whenever a round is played.
- Added fun emojis ("🎉", "😞") in result messages.

CODE

```
game.html X | [navigation icons]
```

C: > Users > sonu0 > Downloads > game.html > html > head > style > .btn

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Rock Paper Scissors</title>
7      <link rel="stylesheet" href="styles.css">
8      <style>
9          @import url('https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap');
```

```
10     body {
11         font-family: 'Press Start 2P', cursive;
12         background: linear-gradient(135deg, #33daff, #10e8d6, #33ff77);
13         text-align: center;
14         color: white;
15         overflow: hidden;
16     }
17
18     .container {
19         margin-top: 50px;
20     }
21
22     h1 {
23         font-size: 40px;
24         text-shadow: 4px 4px 8px rgba(0, 0, 0, 0.3);
25     }
26
27     .choices {
28         margin-top: 30px;
29     }
30
31     .btn {
32         background: linear-gradient(135deg, #013fc4, #33ccff);
33         color: white;
34         border: none;
35         padding: 25px 35px;
36         margin: 20px;
```

```
28     .choices {
29         margin-top: 30px;
30     }
31
32     .btn {
33         background: linear-gradient(135deg, #013fc4, #33ccff);
34         color: white;
35         border: none;
36         padding: 25px 35px;
37         margin: 20px;
38         font-size: 26px;
39         border-radius: 50%;
40         cursor: pointer;
41         transition: transform 0.3s, box-shadow 0.3s;
42         display: inline-flex;
43         align-items: center;
44         justify-content: center;
45     }
46
47     .btn:hover {
48         transform: scale(1.3);
49         box-shadow: 0px 8px 25px rgba(255, 118, 136, 0.8);
50     }
51
52     .emoji {
53         font-size: 80px;
54     }
55
56     #result {
57         margin-top: 40px;
58         font-size: 30px;
59         font-weight: bold;
60         animation: fadeIn 0.5s ease-in-out;
61         padding: 10px;
```

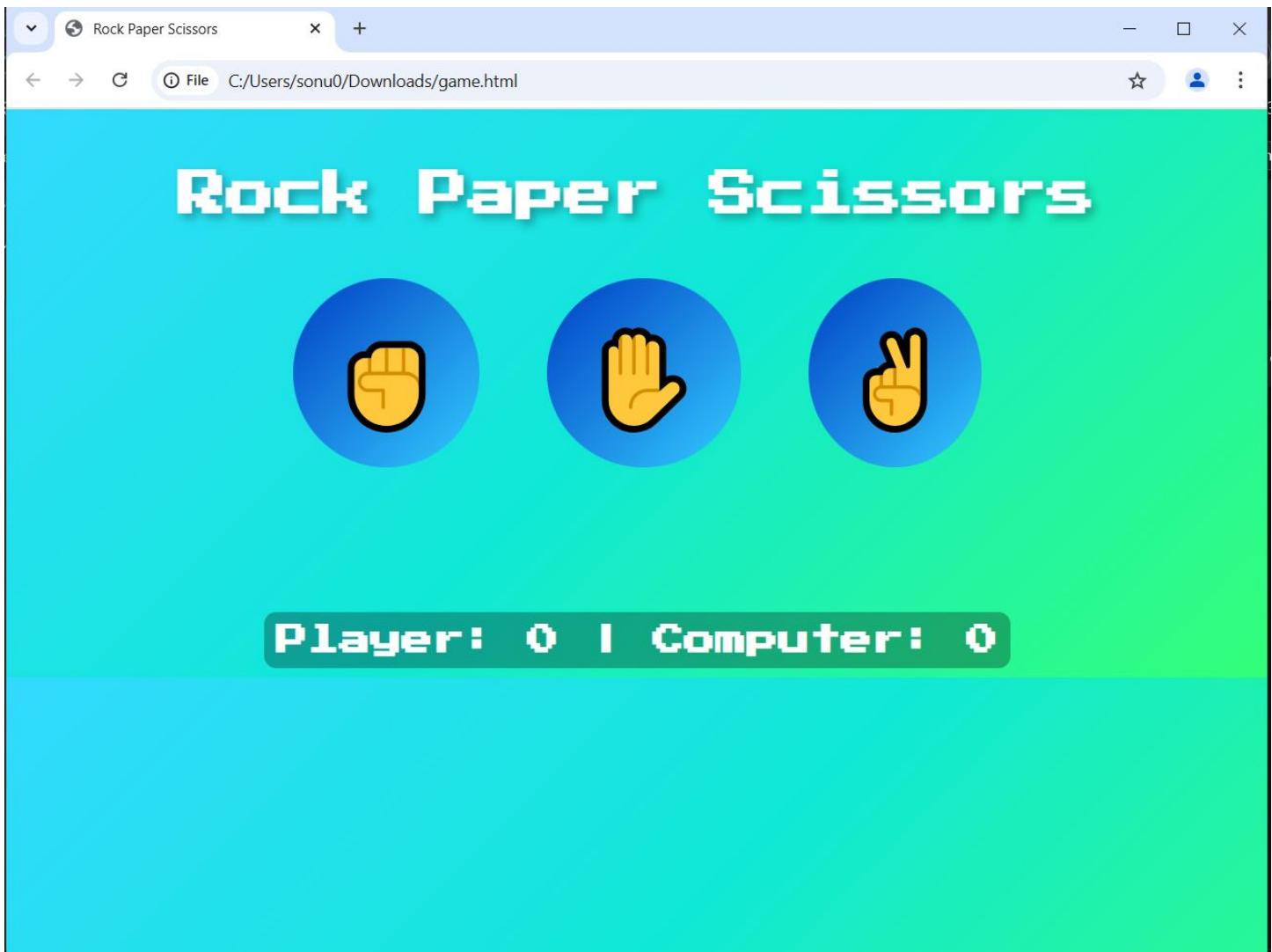
```
61    |     padding: 10px;
62    | }
63
64    .scoreboard {
65        margin-top: 40px;
66        font-size: 26px;
67        font-weight: bold;
68        padding: 10px;
69        background: □rgba(0, 0, 0, 0.3);
70        border-radius: 10px;
71        display: inline-block;
72    }
73
74    @keyframes fadeIn {
75        from { opacity: 0; transform: scale(0.8); }
76        to { opacity: 1; transform: scale(1); }
77    }
78
79    @keyframes bounce {
80        0%, 100% { transform: translateY(0); }
81        50% { transform: translateY(-15px); }
82    }
83
84    .btn:active {
85        animation: bounce 0.4s;
86    }
87    </style>
88    </head>
89    <body>
90        <div class="container">
91            <h1>Rock Paper Scissors</h1>
92            <div class="choices">
93                <button class="btn" onclick="playerChoice('rock')"><span class="emoji">✊</span></button>
```

```

92  <div class="choices">
93    <button class="btn" onclick="playerChoice('rock')"><span class="emoji">>✊</span></button>
94    <button class="btn" onclick="playerChoice('paper')"><span class="emoji">>✋</span></button>
95    <button class="btn" onclick="playerChoice('scissors')"><span class="emoji">>✌</span></button>
96  </div>
97  <div id="result"></div>
98  <div class="scoreboard">Player: <span id="playerScore">0</span> | Computer: <span id="computerScore">0</span></div>
99
100 </div>
101 <script>
102   let playerScore = 0;
103   let computerScore = 0;
104
105   function playerChoice(choice) {
106     const choices = ['rock', 'paper', 'scissors'];
107     const computerChoice = choices[Math.floor(Math.random() * 3)];
108     let resultMessage = '';
109
110     if (choice === computerChoice) {
111       resultMessage = "It's a tie!";
112     } else if (
113       (choice === 'rock' && computerChoice === 'scissors') ||
114       (choice === 'paper' && computerChoice === 'rock') ||
115       (choice === 'scissors' && computerChoice === 'paper')
116     ) {
117       resultMessage = 'You win! 🎉';
118       playerScore++;
119     } else {
120       resultMessage = 'You lose! 😞';
121       computerScore++;
122     }
123
124     let computerScore = 0;
125
126     function playerChoice(choice) {
127       const choices = ['rock', 'paper', 'scissors'];
128       const computerChoice = choices[Math.floor(Math.random() * 3)];
129       let resultMessage = '';
130
131       if (choice === computerChoice) {
132         resultMessage = "It's a tie!";
133       } else if (
134         (choice === 'rock' && computerChoice === 'scissors') ||
135         (choice === 'paper' && computerChoice === 'rock') ||
136         (choice === 'scissors' && computerChoice === 'paper')
137       ) {
138         resultMessage = 'You win! 🎉';
139         playerScore++;
140       } else {
141         resultMessage = 'You lose! 😞';
142         computerScore++;
143       }
144
145       document.getElementById('playerScore').innerText = playerScore;
146       document.getElementById('computerScore').innerText = computerScore;
147
148       const resultDiv = document.getElementById('result');
149       resultDiv.innerHTML = `You chose <b>${choice}</b>. <br> Computer chose <b>${computerChoice}</b>. <br><br> ${resultMessage}`;
150       resultDiv.style.animation = 'fadeIn 0.5s ease-in-out';
151     }
152   </script>
153 </body>
154 </html>
155

```

OUTPUT



Rock Paper Scissors



You chose rock.
Computer chose rock.

It's a tie!

Player: 0 | Computer: 0

Rock Paper Scissors



You chose paper.
Computer chose rock.

You win! 🎉

Player: 6 | Computer: 5