```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Rock Paper Scissors Game Modes</title>
  <style>
    body { font-family: Arial, sans-serif; text-align: center; background: #f0f0f0; }
    h1 { margin-top: 20px; }
    .score { font-size: 18px; margin: 10px; }
    button { margin: 10px; padding: 15px 25px; font-size: 16px; border-radius: 10px; border:
none; cursor: pointer; }
    .choice-btn { background: #4CAF50; color: white; }
    .reset { background: red; color: white; }
    select { font-size: 16px; margin: 10px; padding: 5px; }
    #result { font-size: 18px; margin: 20px; font-weight: bold; }
    .hand { font-size: 50px; display: inline-block; margin: 10px; transition: transform 0.3s; }
    #timer { font-size: 20px; margin: 10px; color: blue; }
  </style>
</head>
<body>
  <h1>Rock Paper Scissors Game</h1>

  <!-- Mode selection -->
  <div>
    Mode:
    <select id="mode">
      <option value="classic">Classic (RPS)</option>
      <option value="rpsls">RPSLS (Rock, Paper, Scissors, Lizard, Spock)</option>
      <option value="timed">Timed Mode (10s)</option>
    </select>
  </div>

  <div class="score" id="score">Player: 0 | AI: 0</div>
  <div id="round">First to 5 wins!</div>
  <div id="result">Make your move!</div>
  <div>
    <span id="playerHand" class="hand">❓</span>
    <span id="aiHand" class="hand">❓</span>
  </div>

  <div id="choices"></div>
  <div id="timer"></div>

  <button class="reset" onclick="resetGame()">Reset</button>

  <script>
    let playerScore = 0, aiScore = 0;
```

```
let playerHistory = [];
let lastOutcome = null;
let lastPlayer = null;
let lastAI = null;
let timerInterval;
let timeLeft = 10;

const hands = { Rock: "✊", Paper: "✋", Scissors: "✌", Lizard: "🦎", Spock: "🖖" };

// Weighted scoring
const weights = { Rock: 2, Paper: 1, Scissors: 1, Lizard: 3, Spock: 3 };

function getChoices() {
  let mode = document.getElementById("mode").value;
  if (mode === "classic" || mode === "timed") {
    return ["Rock", "Paper", "Scissors"];
  } else {
    return ["Rock", "Paper", "Scissors", "Lizard", "Spock"];
  }
}

function renderChoices() {
  let choices = getChoices();
  let div = document.getElementById("choices");
  div.innerHTML = "";
  choices.forEach(c => {
    let btn = document.createElement("button");
    btn.className = "choice-btn";
    btn.innerText = hands[c] + " " + c;
    btn.onclick = () => play(c);
    div.appendChild(btn);
  });
}

function counterMove(move) {
  const rules = {
    Rock: ["Scissors", "Lizard"],
    Paper: ["Rock", "Spock"],
    Scissors: ["Paper", "Lizard"],
    Lizard: ["Spock", "Paper"],
    Spock: ["Scissors", "Rock"]
  };
  // Return a random counter
  let counters = Object.keys(rules).filter(m => rules[m].includes(move));
  return counters[Math.floor(Math.random() * counters.length)];
}

// Win-Stay, Lose-Shift strategy
```

```
  function aiChoice() {
    let choices = getChoices();
    if (!lastOutcome || !lastPlayer || !lastAI) {
      return choices[Math.floor(Math.random() * choices.length)];
    }

    if (lastOutcome === "playerWin") {
      // Player won last round → assume they repeat → counter it
      return counterMove(lastPlayer);
    } else if (lastOutcome === "aiWin") {
      // Player lost → predict they switch to beat AI's last move
      return counterMove(lastAI);
    } else {
      // Draw → random
      return choices[Math.floor(Math.random() * choices.length)];
    }
  }

  function getWinner(player, ai) {
    const rules = {
      Rock: ["Scissors", "Lizard"],
      Paper: ["Rock", "Spock"],
      Scissors: ["Paper", "Lizard"],
      Lizard: ["Spock", "Paper"],
      Spock: ["Scissors", "Rock"]
    };
    if (player === ai) return "draw";
    if (rules[player].includes(ai)) return "playerWin";
    return "aiWin";
  }

  function play(playerChoice) {
    stopTimer();
    let comp = aiChoice();

    document.getElementById("playerHand").innerText = hands[playerChoice];
    document.getElementById("aiHand").innerText = hands[comp];

    let winner = getWinner(playerChoice, comp);
    if (winner === "draw") {
      document.getElementById("result").innerText = `Draw! Both chose ${playerChoice}`;
    } else if (winner === "playerWin") {
      playerScore += weights[playerChoice] || 1;
      document.getElementById("result").innerText = `You Win! ${playerChoice} beats
${comp}`;
    } else {
      aiScore += weights[comp] || 1;
```

```
      document.getElementById("result").innerText = `AI Wins! ${comp} beats
${playerChoice}`;
    }

    lastOutcome = winner;
    lastPlayer = playerChoice;
    lastAI = comp;

    updateScore();
  }

  function updateScore() {
    document.getElementById("score").innerText = `Player: ${playerScore} | AI: ${aiScore}`;
    if (playerScore >= 5 || aiScore >= 5) {
      let winner = playerScore >= 5 ? "You" : "AI";
      alert(`${winner} won the match!`);
      resetGame();
    }
  }

  function resetGame() {
    playerScore = 0;
    aiScore = 0;
    playerHistory = [];
    lastOutcome = null;
    document.getElementById("result").innerText = "Game Reset!";
    document.getElementById("playerHand").innerText = " ❓ ";
    document.getElementById("aiHand").innerText = " ❓ ";
    updateScore();
    if (document.getElementById("mode").value === "timed") startTimer();
  }

  // Timer functions
  function startTimer() {
    timeLeft = 10;
    document.getElementById("timer").innerText = `Time left: ${timeLeft}s`;
    timerInterval = setInterval(() => {
      timeLeft--;
      document.getElementById("timer").innerText = `Time left: ${timeLeft}s`;
      if (timeLeft <= 0) {
        clearInterval(timerInterval);
        aiScore++;
        document.getElementById("result").innerText = "Time's up! AI wins this round.";
        updateScore();
        if (document.getElementById("mode").value === "timed") startTimer();
      }
    }, 1000);
  }
```

```
    function stopTimer() {
      clearInterval(timerInterval);
      document.getElementById("timer").innerText = "";
    }

    document.getElementById("mode").addEventListener("change", () => {
      resetGame();
      renderChoices();
      if (document.getElementById("mode").value === "timed") startTimer();
      else stopTimer();
    });

    renderChoices();
  </script>
</body>
</html>
```

https://chatgpt.com/share/68ce86ea-9cec-8012-afa9-57385671f84d