Simon Game Using HTML, CSS & JavaScript

Project overview :

➢ Introduction:

The Simon Game is a classic memory game where players need to replicate a sequence of lights and sounds generated by the game. It tests the player's memory and concentration skills.

➢ Purpose:

The purpose of this project is to recreate the classic Simon Game using modern web technologies such as HTML, CSS, and JavaScript. It aims to provide an enjoyable and challenging gaming experience for users.

➢ Scope:

The project includes the development of the Simon Game interface, game logic, and user interaction. It covers the generation of random sequences, implementation of sound effects, difficulty levels, responsive design, and different game modes.

➢ Goals/Objectives:

- Create a fully functional Simon Game interface.
- Implement random sequence generation for gameplay.
- Incorporate sound effects for user feedback.
- Design and implement various difficulty levels.
- Ensure responsive design for compatibility across different devices.
- Provide different game modes such as Start, Restart, and Strict.
- Implement a scoring system to track high scores.

➢ Usage:

- **Starting the Game:**

To start the game, simply press any key on your keyboard . This will initiate the game sequence and prompt the player to begin.

- **Gameplay Instructions:**
  - Watch and listen to the sequence of lights and sounds played by Simon.
  - Repeat the sequence by clicking the buttons in the same order.
  - If you repeat the sequence correctly, Simon will add a new step to the sequence. Repeat the sequence again.
  - If you make a mistake, the game ends.

➢ Features:

- **Random Sequence Generation:**

The game generates random sequences of colors for each level, increasing in complexity as the game progresses.

- **Sound Effects:**

Each button press produces a corresponding sound effect to provide feedback to the player.

- **Difficulty Levels:**

The game offers multiple difficulty levels, adjusting the speed and complexity of the sequences to suit the player's skill level.

➢ Implementation:

- **HTML Structure:**

The HTML file contains the structure of the game interface, including buttons, score display, and game controls.

- **CSS Styling:**

The CSS file provides the styling and layout for the game interface, including colors, fonts, and animations.

- **JavaScript Logic:**

The JavaScript file contains the game logic, including sequence generation, user input validation, scoring, and event handling.

- **Libraries/Frameworks Used:**

No external libraries or frameworks are used in this project. The game is implemented using native HTML, CSS, and JavaScript.

➢ Code Snippets:

- **Sequence Generation:**

```
function nextSequence() {
  userClickedPattern = [];
  level++;
  $("#level-title").text("Level " + level);
  var randomNumber = Math.floor(Math.random() * 4);
  var randomChosenColour = buttonColours[randomNumber];
  gamePattern.push(randomChosenColour);

  $("#" + randomChosenColour).fadeIn(100).fadeOut(100).fadeIn(100);
  playSound(randomChosenColour);
}
```

- **Game Logic & Button Click Handlers:**

```
function checkAnswer(currentLevel) {

    if (gamePattern[currentLevel] === us                    evel]) {
      if (userClickedPattern.length === gamePattern.length){
        setTimeout(function () {
          nextSequence();
        }, 1000);
      }
```

var gamePattern: any[]

```
    } else {
      playSound("wrong");
      $("body").addClass("game-over");
      $("#level-title").text("Game Over, Press Any Key to Restart");

      setTimeout(function () {
        $("body").removeClass("game-over");
      }, 200);

      startOver();
    }
}
```

Styling Techniques:

```
body {
    text-align: center;
    background-color: #011F3F;
}

#level-title {
font-family: 'Press Start 2P', cursive;
font-size: 3rem;
margin:  5%;
color: #FEF2BF;
}

.container {
display: block;
width: 50%;
margin: auto;

}

.btn {
margin: 25px;
display: inline-block;
height: 200px;
width: 200px;
border: 10px solid black;
border-radius: 20%;
}

.game-over {
background-color: red;
opacity: 0.8;
}

.red {
background-color: red;
}
```

```
.green {
background-color: green;
}

.blue {
background-color: blue;
}

.yellow {
background-color: yellow;
}

.pressed {
box-shadow: 0 0 20px white;
background-color: grey;
}
```

Event Handling:

```
$(document).keypress(function() {
  if (!started) {
    $("#level-title").text("Level " + level);
    nextSequence();
    started = true;
  }
});
```

➢ Testing:

- **Test Cases:**
  - ✓ **Test case 1:** Verify that the game starts correctly when the Start button is clicked.
  - ✓ **Test case 2:** Test the game sequence generation to ensure randomness.
  - ✓ **Test case 3:** Check user input validation to handle incorrect sequences.

- **Test Results:**
  - ✓ **Test case 1:** Passed
  - ✓ **Test case 2:** Passed
  - ✓ **Test case 3:** Passed

- **Bugs/Issues:**
  - ✓ No major bugs or issues were identified during testing.

➢ Game Interface: