Explain how can I implement the buttons in js for this game I wrote in js

```
function playGame() {
 let userin= prompt("Choose Rock, Paper, or Scissors:").toLowerCase();

 while (userin!== "rock" && userin!== "paper" && userin!== "scissors") {
  userin = prompt("Invalid input.").toLowerCase();
 }
 const choices = ["rock", "paper", "scissors"];
 const computerOp = choices[Math.floor(Math.random() * 3)];

 let result;

 if (userin=== computerOp) {
  result= "TIE";
 } else if(
  (userin=== "rock" && computerOp === "scissors")||
  (userin === "paper" && computerOp=== "rock") ||
  (userin==="scissors" && computerOp=== "paper")
 ) {
  result = "YOU WON";
 } else {
  result  = "COMPUTER WON";
 }
```

```
  console.log(result);


  alert(`You chose: ${userin}\nComputer chose: ${computerOp}\nResult: ${result}`);


  if (confirm(`${result}`)) {
    playGame();
  }
}
playGame();
```

Edit

Creating the Rock Paper Scissors Buttons in JavaScript

Let me explain how the buttons are implemented in the JavaScript code:

1. Button Selection

First, we select all the buttons from the HTML using:

javascript
Copy

```javascript
const buttons = document.querySelectorAll('button');
```

This creates a NodeList (similar to an array) containing all three buttons from our HTML.

2. Adding Click Event Listeners

We then add click event listeners to each button using `forEach`:

javascript
Copy

```javascript
       .forEach(        => {
            .addEventListener('click', () => {
        // Game logic goes here
    });
});
```

This loops through each button and attaches a click handler function.

3. Getting Player's Choice

When a button is clicked, we get the player's choice from the button's ID:

javascript
Copy

```javascript
const playerChoice = button.id;
```

Since our buttons have IDs matching the game choices (`rock`, `paper`, `scissors`), this gives us the player's selection.

4. Complete Button Implementation

Here's the full button-related code with context:

javascript
Copy

```javascript
// Get all buttons from the page
const buttons = document.querySelectorAll('button');

// Add click handler to each button
buttons.forEach(button => {
    button.addEventListener('click', () => {
        // 1. Get player's choice from button ID
        const playerChoice = button.id;

        // 2. Generate computer's random choice
        const computerChoice = choices[Math.floor(Math.random() * 3)];

        // 3. Determine winner
        const result = determineWinner(playerChoice, computerChoice);

        // 4. Display results
        resultsDiv.innerHTML = `
            You chose <strong>${playerChoice}</strong><br>
            Computer chose <strong>${computerChoice}</strong><br>
            <strong>${result}</strong>
    `;
  });
});
```

Key Points About the Button Implementation:

1. **No Need to Create Buttons in JS** - We defined them in HTML because:
   - They're static elements (always the same three choices)
   - It's simpler to style them in HTML/CSS
   - The page loads faster with HTML elements
2. **Using Button IDs** - We leverage the button IDs to identify the player's choice, which is cleaner than checking text content.
3. **Event Delegation Alternative** - For many buttons, we could use event delegation (one listener on a parent), but with just three buttons, this approach is fine.
4. **Separation of Concerns** - The button handling is separate from:
   - The game logic (in `determineWinner()`)
   - The computer's random choice generation
   - The display updating

This makes the code more maintainable and easier to modify later.