

We've seen the use of the Cascading Style Sheet (CSS) feature of HTML in changing fonts. CSS is used to control all aspects of the page's display – size, colors, fonts, backgrounds, etc.

The 'sliding box' game works as follows:

A box slides from the upper left, to the lower right. If you click the sliding box, the score goes up, however, each time you click the box, there's a small chance the box will slide immediately down to the right. Once the box reaches the lower-right corner, the game is over. Your goal is to click as quickly as possible to generate the highest score.

Start a new file, and copy in the following code for the sliding box game:

```
<!DOCTYPE html>
<html>
<style>
  #myContainer {
    width: 400px;
    height: 400px;
    position: relative;
    background: yellow;
  }

  #myAnimation {
    width: 50px;
    height: 50px;
    position: absolute;
    background-color: red;
  }
</style>

<body>

  <p>
    <p>Score:<a id="myScore">0</a></p>
    <button onclick="myStopGo()" id="myButton">Click to
start</button>
  </p>

  <div id="myContainer">
    <div id="myAnimation" onclick="ScoreUpdate()"></div>
  </div>

  <script>
    let score = 0;
    let started = false;

    function ScoreUpdate() {
      //update the score if the box is moving
      if (started) {

        score += 1;
        document.getElementById("myScore").innerHTML = score;

      }
    }
  </script>
</body>
</html>
```

```
function myStopGo() {  
    if (started == false) {  
        //change the button string, start moving if not,  
        otherwise stop  
  
        score = 0;  
        document.getElementById("myButton").innerHTML = "Click  
to stop";  
  
        document.getElementById("myScore").innerHTML = score;  
        myMove(); //move the object  
        started = true;  
  
    } else {  
  
        myMove();  
        started = false;  
    }  
}  
var id;  
  
function myMove() {  
  
    var elem = document.getElementById("myAnimation");  
    if (started == false) {  
        id = setInterval(frame, 20); //start the object moving  
        document.getElementById("myButton").innerHTML = "Click  
to stop";  
  
    } else {  
  
        clearInterval(id);  
        document.getElementById("myButton").innerHTML = "Click  
to start";  
  
        score = 0;  
  
    }  
    var pos = 0;  
    var randomOffset = 0;  
    var chance = 0;  
  
    function frame() {  
        //reposition the box. Keep it inside the animation
```

```
        // change the button if the box reaches the lower
right and stop the score from going up
        if (pos >= 350) {
            //Reached the lower right hand corner

            clearInterval(id);
            document.getElementById("myButton").innerHTML =
"Click to start";
            started = false;

        } else {
            //move the box - just a small chance it will move.
            // Every 20 milliseconds, we generate and
accumulate a random number
            // between 1 and 100. When that number reaches
5000, we move the box
            // ahead by updating it's offset. Next time the
timer clicks, the box
            // will be repositioned at its new offset.

            chance += Math.floor(Math.random() * 101);
            randomOffset = 0;
            if (chance > 5000) {
                randomOffset = 25 + Math.floor(Math.random() *
50);

                console.log("Offset " + randomOffset);
                chance = 0;
            }
        }

        pos = pos + 1 + randomOffset;
        if (pos > 350) pos = 350; // Don't extend past the box
        elem.style.top = pos + 'px';
        elem.style.left = pos + 'px';
    }
}
</script>

</body>

</html>
```

The <style> section is important to understand.

```
<style>
  #myContainer {
    width: 400px;
    height: 400px;
    position: relative;
    background: yellow;
  }

  #myAnimation {
    width: 50px;
    height: 50px;
    position: absolute;
    background-color: red;
  }
</style>
```

“Mycontainer” is a CSS “selector.” Selectors define areas to apply styles to – in this case, width, height, position relative to the web page and it’s background color. Selectors starting with ‘#’ are called ‘id selectors’. They can be addressed like HTML divs, using their ID (in this case, Mycontainer.) “MyAnimation”, another id selector, is used in the code later:

```
var elem = document.getElementById("myAnimation");
```

elem is later used in the code –

```
elem.style.top = pos + 'px';
elem.style.left = pos + 'px';
```

So, when those statements are run, ‘myAnimation’'s top and left margin position are updated to ‘pos’ + px.

Challenges:

Easy: change the background color of the animation or the sliding box.

Medium: Change the background color of the sliding box when you’ve clicked it successfully. Switch between ‘red’ and blue – if the box is red, turn it blue, if blue, turn it red

Hard: Slide the box from the lower-right, to the upper left.