**Rich Web Application Development**

*Bryan Nyondo – C14372736*

Lab 1

Part 1 - CSS

HTML

{

background: #f2f2f2

}

header

{

font-style: italic;

font-weight: bold;

}

ul

{

list-style-type: none;

padding-right: 20px;

}

#content

{

display:block;

border-style: solid;

border-width: 1px;

height: 300px;

width: 350px;

float: left;

}

#sidebar

{

display: inline-block;

border-style: solid;

border-width: 1px;

margin-left: 5px;

}

footer

{

padding-top: 220px;

position: relative;

}

Part 2 - Calculator

HTML:

<head>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<div id="calculator">

<!-- Screen and clear key -->

<div class="top">

<div class="screen"></div>

</div>

<div class="keys">

<!-- operators and other keys -->

<span>(</span>

<span>)</span>

<span>+=</span>

<span class="operator">÷</span>

<span>7</span>

<span>8</span>

<span>9</span>

<span class="operator">x</span>

<span>4</span>

<span>5</span>

<span>6</span>

<span class="operator">-</span>

<span>1</span>

<span>2</span>

<span>3</span>

<span class="operator">+</span>

<span>1</span>

<span>2</span>

<span>3</span>

<span class="operator">=</span>

</div>

</div>

CSS:

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font: bold 14px Arial, sans-serif;

}

html {

height: 100%;

background: white;

background-size: cover;

}

/\* 3D shadows on the buttons \*/

#calculator {

width: 325px;

height: auto;

margin: 100px auto;

padding: 20px 20px 9px;

background: #ffffff;

background: linear-gradient(#9dd2ea, #8bceec);

border-radius: 3px;

box-shadow: 0px 4px #009de4, 0px 10px 15px rgba(0, 0, 0, 0.2);

}

/\* Screen above the keys \*/

.top .screen {

height: 40px;

width: 290px;

float: left;

padding: 0 10px;

background: white;

border-radius: 3px;

box-shadow: inset 0px 4px rgba(0, 0, 0, 0.2);

/\* Typography \*/

font-size: 17px;

line-height: 40px;

color: white;

text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.2);

text-align: right;

letter-spacing: 1px;

padding-bottom: 20px;

}

/\* Clear floats \*/

.keys, .top {overflow: hidden;

padding-top: 10px;}

/\* All keys \*/

.keys span, .top span.clear {

float: left;

position: relative;

top: 0;

cursor: pointer;

width: 66px;

height: 36px;

background: white;

border-radius: 3px;

box-shadow: 0px 4px rgba(0, 0, 0, 0.2);

margin: 0 7px 11px 0;

color: #888;

line-height: 36px;

text-align: center;

}

/\* Different style for operator keys\*/

.keys span.operator {

background: #FFF0F5;

margin-right: 0;

}

Part 3 - JS Rules

JS:

<!DOCTYPE html>

<html>

<body>

</body>

<script>

var row = 50;

var col = 101;

var divEven = new Array();

var divOdd = new Array();

var border = "1px solid pink";

var activeColour = "blue";

var inactiveColour = "white";

var randomColour= [inactiveColour, activeColour];

var isActive;

var container = document.createElement("div");

container.style.width = "1010px";

container.style.background = "black";

container.style.height = "100%";

container.id = "container";

//loop through rows(j)

for (var j = 0; j < row; j++) {

for (var k = 0; k < col; k++) {

var div = document.createElement("div");

div.style.width = "8px";

div.style.height = "8px";

div.style.border = border;

div.style.cssFloat= "left";

//if first row colour the first row random colours

if (j === 0) {

var activityRandom=Math.floor(Math.random() \* randomColour.length);

div.style.background = randomColour[activityRandom];

divOdd[k] = activityRandom;

}

// check the ancestors every second row after

else if(j % 2 === 1) {

isActive = rules(divOdd);

div.style.background = randomColour[isActive];

divEven[k] = isActive;

}else if(j % 2 === 0) {

isActive = rules(divEven);

div.style.background = randomColour[isActive];

divOdd[k] = isActive;

}

container.appendChild(div);

}

}

document.body.appendChild(container);

function rules(divActivity) {

var left = k-1;

var right = k+1;

var active= 1;

var inactive = 0;

if (left< 0) {

left = divActivity.length-1;

}

if (right> divActivity.length) {

right = 0;

}

//Rule one

if(divActivity[left] == active && divActivity[right] == active && divActivity[k] == active )

{

div.style.background = active;

isActive = active;

}

//Rule two

if(divActivity[left] == active && divActivity[k] == active && divActivity[right] == inactive){

div.style.background = inactive;

isActive = inactive;

}

//Rule three

if(divActivity[left] == active && divActivity[k] == inactive && divActivity[right] == active)

{

div.style.background = active;

isActive = active;

}

//Rule four

if(divActivity[left] == active && divActivity[k] == inactive && divActivity[right] == inactive){

div.style.background = active;

isActive = active;

}

//Rule five

if (divActivity[left] == inactive && divActivity[k] == active && divActivity[right] == active) {

div.style.background = inactive;

isActive = inactive;

}

//Rule Six

if (divActivity[left] == inactive && divActivity[k] == active && divActivity[right] == inactive) {

div.style.background = active;

isActive = active;

}

//Rule seven

if (divActivity[left] == inactive && divActivity[k] == inactive && divActivity[right] == active) {

div.style.background = active;

isActive = active;

}

//Rule one

if(divActivity[left] == inactive && divActivity[right] == inactive && divActivity[k] == inactive )

{

div.style.background = inactive;

isActive = inactive;

}

return isActive;

}

</script>

</html>