F28WP Web Programming
Lab report
Lab 2

Student full name: Abhinav Bhudarapu

Student HWU username: ab2025, H00348223

Student's GitHub URL of the Lab: https://github.com/AbhinavBhudarapu/F28wp-Lab02

Demonstrated to Lab helper: Gosain, Gaurav

Mode of demonstration: Online

Date of demonstration: 02/11/2021

Time of demonstration: 10:51 AM

Part 1: **HTML/CSS**

Final Web Page:

Fig - 1.1

Changes in HTML:

- Added the Start, Restart, Add bee Buttons
- Added an input field for adjusting the bear speed

Changes in CSS:

- Changes the background colour
- Resized the board and changed the colour

Part 2: **Bear motion**

```
function setSpeed() {
    bear.dBear = parseInt(document.getElementById("speedBear").value);
    //Update the speed to the value we get from the input field
}
```

Fig – 2.1

First we make a function setSpeed() and get the values of speedBear by using getElementById,

```
document.getElementById("speedBear").addEventListener("change", setSpeed)
//Add event listener to the input field for changes
```

Fig – 2.2

Then we call the function set speed in the function start()

Bear moves by the keys user presses and the speed user enters in the speedBear input field.

Bees moves by the speed user enters and refreshes by the input given in speedBees

Part 3: Bee creation

```
function createBeeImg(wNum) {
    //Get the dimension and position the board
    let boardDiv = document.getElementById("board"); //Set this variable to be the board div
    let boardDivW = boardDiv.offsetWidth; //Width of the board div
    let boardDivH = boardDiv.offsetHeight; //Height of the board div
    let boardDivX = boardDiv.offsetLeft; //X axis position of the board div
    let boardDivY = boardDiv.offsetTop; //Y axis position of the board div

    //Create the img
    let img = document.createElement("img");
    img.setAttribute("src", "images/bee.gif") //Set the src of the been image to the bee gif in the images file
    img.setAttribute("width", "100"); //Set the width to be 100
    img.setAttribute("alt", "A bee!"); //Alt text of the image
    img.setAttribute("id", "bee" + wNum); //The id of the bee image
    img.setAttribute("class", "bee"); //The class of the bee image

    //Add the image to the DOM as a child of the board div
    img.style.position = "absolute";
    boardDiv.appendChild(img);

    //Set the inital position of the bee
    let x = getRandomInt(boardDivW);
    let y = getRandomInt(boardDivW);
    img.style.left = (boardDivX + x) + "px";
    img.style.top = (y) + "px";

    //Return the img object
    return img;
}
```

Fig – 3.1

First we get the board div and get the Height, Width, X axis position and Y axis position.
Then we create an element img then set the src to bee img after we set the width of the img a text and an id and a class bee.
Then we add the image to DOM as child of the board div and then set the initial positions od the bee img by using getRandomInt function and then we return img.

```
function getRandomInt(max) {
    return Math.floor(Math.random() * max); //Generates a random number between 0 and max
}
```

Fig – 3.2

First we create a function getRandomInt and return a random number by using Math.random() and by using Math.floor we get highest integer possible

Part 4: Bee animation

First we move bees in the moveBees function, then we use document.getElementById
For the input field periodTimer after getting the value we give the value to period in the updateBees function

```
//move the bees randomly
moveBees();
//use the input field periodTimer instead of the fixed period
let period = document.getElementById("periodTimer").value;
```

Fig – 4.1

Part 5: Scores

First, we make it function overlap by adding arguments element 1 an element 2
Then we get the rectangle around the first element and the second element assigning as left top right bottom to both element 1 and element 2.
Then we calculate the intersection between the rectangle by using Math.max, Math.min methods ,If the intersection is null it gives false if not it returns true

```
function overlap(element1, element2) {

    //The rectangle around the first element
    left1 = element1.htmlElement.offsetLeft;
    top1 = element1.htmlElement.offsetTop;
    right1 = element1.htmlElement.offsetLeft + element1.htmlElement.offsetWidth;
    bottom1 = element1.htmlElement.offsetTop + element1.htmlElement.offsetHeight;

    //The rectangle around the second element
    left2 = element2.htmlElement.offsetLeft;
    top2 = element2.htmlElement.offsetTop;
    right2 = element2.htmlElement.offsetLeft + element2.htmlElement.offsetWidth;
    bottom2 = element2.htmlElement.offsetTop + element2.htmlElement.offsetHeight;

    //Calculate the intersection of the 2 rectangles
    x_intersect = Math.max(0, Math.min(right1, right2) - Math.max(left1, left2));
    y_intersect = Math.max(0, Math.min(bottom1, bottom2) - Math.max(top1, top2));
    intersectArea = x_intersect * y_intersect;

    //If intersection is null
    if (intersectArea == 0 || isNaN(intersectArea)) {
        return false;
    }

    return true;
}
```

Fig – 5.1

If the score is less than 1000 then this updates the bee movement by the specified time interval given the user, if it hits 1000 then its score = “Game Over” and updateTimer = clearTimeout()

```
let score = hits.innerHTML;
if (Number(score) < 1000) {
    updateTimer = setTimeout('updateBees()', period);
    //Update the bees movement after the specified interval
} else {
    score = "Game Over"
    hits.innerHTML = score;
    updateTimer = clearTimeout();
}
```

Fig – 5.2

Part 6: **Best duration**

```
//Calculate longest duration
let newStingTime = new Date();
let thisDuration = newStingTime - lastStingTime;
lastStingTime = newStingTime;
let longestDuration = Number(duration.innerHTML);
if (longestDuration === 0 || isNaN(longestDuration)) {
    longestDuration = thisDuration;
} else {
    if (longestDuration < thisDuration) longestDuration = thisDuration;
}

//Update the longest duration display
document.getElementById("duration").innerHTML = longestDuration;
```

Fig – 6.1

We create a newStingTime and thisDuration, thisDuration = newStingTime – lastStingTime Then lastStingTime is equal to newStingTime, then we give longest duration is a number from duration.innerHTML, If the longestDuration is 0 or null then longestDuration = thisDuration, else if longestDuration is less than this Duration of longestDuration = thisDuration, then we use getElementById of duration and it is equal to longestDuration

Part 7: Additional features

1. Restarting the game

```
function restart() {
    score = 0; // sets the score to 0
    hits.innerHTML = score;
    duration.innerHTML = 0;// sets the duration to 0
    updateTimer = clearTimeout();// clear the timer
    removeBees();// clears the bees array by removing the bees
    start();//by using the onclick to call start
}
```

Fig – 7.1

By using onclick function given to the button restart it calls the function Restart() then its sets the score to 0 and hits.innerHTML to score, duration.innerHTML to 0 , its cleat the timer using clearTimeout() function then removebees function is called which clears the bees array by removing the bees and then calls the start() function to start the game.

2. Add a bee

```
function addBee() {
    let nbBees = document.getElementById("nbBees").value; //Gets the number of bees specified by the user
    nbBees = Number(nbBees);
    //Convert the content of the input to a number
    nbBees++;
    var bee = new Bee(nbBees); //Creates a bee
    bee.display(); //Displays the bee
    bees.push(bee);
    //Adding the bee to the bees array
}
```

Fig – 7.2

In the addBee function first we the get the number of bees specified the by the user in the input field nbBees by using getElementById, then nbBees = Number(nbBees) this covert the content given in the input filed to a number, then nbBees++ which adds the bees,
Then var bee = new Bee(nbBees) this create a new bee, then we display the bee created, after push the created bee to the bee array