Description: Logo

Description automatically generated Description: Logo

Description automatically generated

CSY1018

Web Development – I

Assignment II

Source Code

**Submitted By**

Student Name: Aashish karki

Uni. ID: ­­ 23834873

Group: L4 (C)

Date: 14-MAY-2023

Table of Contents

[1 Source Code 2](file:///C:\Users\ASUS\Desktop\web%202\CSY1018-Web-Developement-Assignment-II-23834873-Aashish-karki-Project-Source-Code.docx#_Toc134521537)

[1.1 Game.js 2](file:///C:\Users\ASUS\Desktop\web%202\CSY1018-Web-Developement-Assignment-II-23834873-Aashish-karki-Project-Source-Code.docx#_Toc134521538)

# Source Code

## Game.js

var upPressed = false;

var downPressed = false;

var leftPressed = false;

var rightPressed = false;

var lastPressed = false;

//------------------------------------------All the variables are made by user------------------------------------------

var start;

var HP = 3;   //life points

var player;   //player

var Missiles = [];    //all the missile generated

var MovementIncrese = 1; // speed multiplier to increase difficulty

var PlayTime = 0; // game played time

var GameOver = false; // defines game state

var Points = 0; // game Points

var playerHit = false; // player animation when hit

var life; // Life/Health HUD

var PlayerShoot = false; // player animation when fired arrow

var Movements = [];

var high;

var LeftSpeed = [];

var DownSpeed = [];

//-----------------------------------------------------------------------------------------------------------------------------------

function startGame() {

    start[0].style.display = 'none';

    move();

    Toop();

    sarAlien();

}

function sarAlien() {

    const alien = document.getElementById("alien");

    const maxLeft = document.body.clientWidth - alien.offsetWidth;

    const currentLeft = parseInt(alien.style.left) || 0;

    const direction = alien.dataset.direction === "right" ? 1 : -1;

    let newLeft = currentLeft + 10 \* direction;

    if (newLeft > maxLeft) {

        newLeft = maxLeft;

        alien.dataset.direction = "left";

    } else if (newLeft < 0) {

        newLeft = 0;

        alien.dataset.direction = "right";

    }

    alien.style.left = newLeft + "px";

}

//---------------------The function is to genertae the missile ---------------------------------------------------------------------------------------------------------------

function Misile() {

    var missile = document.createElement('Div');

    missile.className = 'bomb'

    document.body.appendChild(missile);

    span(missile);

    Missiles.push(missile);

    LeftSpeed.push(0);

    DownSpeed.push(1);

}

function span(missile) {

    var ScreenWidth = window.innerWidth;

    var RandoWidth = Math.ceil(Math.random() \* ScreenWidth);

    var RandoHeight = Math.random() \* (1000) + 1;

    missile.style.top = -1 \* RandoHeight + 'px';

    missile.style.left = RandoWidth + 'px';

}

// if the game is over

function Toop() {

    // if the game is over don't need to do anything

    if (GameOver) {

        return;

    }

    // when the game time is 0 chanage of speed

    if (PlayTime > 100) {

        if (MovementIncrese < 6) {

            for (var i = 0; i < 5; i++) {

                Misile();   //creates more bomb

            }

            MovementIncrese = MovementIncrese + 0.75;

            PlayTime = 0;

        }

    }

    //---------------this code is used to genertaed the missile from the top direction -------------

    for (var i = 0; i < Missiles.length; i++) {

        var BombTopPos = parseFloat(Missiles[i].offsetTop);

        ExplodeMissile = Math.floor(Math.random() \* (window.innerHeight - sky[0].offsetHeight)) + sky[0].offsetHeight;

        //-------drop bomb on the bottom portion  on the screen--------------------------------

        if (Missiles[i].offsetTop == ExplodeMissile || Missiles[i].offsetTop > window.innerHeight) {

            Pataka(Missiles[i]);

            span(Missiles[i]);

            SpeedLeft(i);

            DownSpeed[i] = ((Math.ceil(Math.random() \* 2))) + 1;

        }

        else {

            if (ElementCollide(player, Missiles[i]) != 0) {

                Pataka(Missiles[i]);    // when missile hits the payer explode the bomb

                span(Missiles[i]);  // rebot the exploded missile

                SpeedLeft(i);

                DownSpeed[i] = ((Math.ceil(Math.random() \* 2))) + 1;

            }

            else {

                var TopPos = BombTopPos + DownSpeed[i] \* MovementIncrese;

                var LeftPos = parseFloat(Missiles[i].offsetLeft);

                Missiles[i].style.top = TopPos + 'px';

                if (LeftPos > window.innerWidth - Missiles[i].offsetWidth || LeftPos < 0) {

                    SpeedLeft(i);

                }

                Missiles[i].style.left = LeftPos + LeftSpeed[i] + 'px';

            }

        }

    }

}

//---------------------This function is stop the game and show the score board after die-------------------------------------------------------------------------------------------------------------

function Maro() {

    GameOver = true;

    player.className = 'character dead';

    var btn = document.createElement('div');

    btn.className = 'start';

    var DeadPromt = document.createTextNode('OMG YOU LOST, try again?');

    btn.appendChild(DeadPromt);

    btn.addEventListener('click', () => location.reload());

    document.body.appendChild(btn);

    //----------------------------This code will display the score  --------------------------------------------------------------------------------------------------------

    var ProPlayer = localStorage.key(0);

    var MaxPoints = parseInt(localStorage.getItem(ProPlayer));  //collect total point

    var MaxPoints = document.createElement('div');

    document.body.appendChild(MaxPoints);

    MaxPoints.className = 'Points';

    if (!MaxPoints || Points > MaxPoints) {

        localStorage.clear(); MaxPoints

        var PlayerName = prompt("You've scored the highest, save your score with a name");      //score is new ask the player name

        ProPlayer = PlayerName;     //changes the name of player

        MaxPoints = Points; // changes  the player gained score

    }

    localStorage.setItem(ProPlayer, MaxPoints);     //stores player name and score

    MaxPoints.innerHTML = 'Player Name: ' + ProPlayer + '<br> ' + 'HighScore: ' + MaxPoints;    //shows player name and score at the end after die.

    return;

    //---------------------------------------------------------------------------------------------------------------------------------------------------------

}

//--------------------------------this code is for collision--------------------------------------------------------------------------------------------------------------------

function ElementCollide(ElementA, ElementB) {

    //to detect height,width and top/left of an object for collision

    if (ElementA.offsetTop < ElementB.offsetTop + ElementB.offsetHeight) {

        if (ElementA.offsetTop + ElementA.offsetHeight > ElementB.offsetTop) {

            if (ElementA.offsetLeft < ElementB.offsetLeft + ElementB.offsetWidth) {

                if (ElementA.offsetLeft + ElementA.offsetWidth > ElementB.offsetLeft) {

                    return true;

                }

            }

        }

    }

    return false;

}

// this function is used to fire key

function ArrowAayo(arrows, FirInterval) {

    var ArrowPosTop = arrows.offsetTop;

    arrows.style.top = ArrowPosTop - 1 + 'px';

    for (var i = 0; i < Missiles.length; i++) {

        if (ElementCollide(Missiles[i], arrows)) {

            arrows.remove();        //removes arrow

            PlayTime = PlayTime + 2;        //calculates gameplay timers

            span(Missiles[i]);  //span bomb's position

            SpeedLeft(i);

            DownSpeed[i] = ((Math.ceil(Math.random() \* 2))) + 1;

            Points = Points + 10;           //updates the Points with arrow collides

            high.innerHTML = 'Points:' + Points;

            clearInterval(FirInterval);

        }

        if (ArrowPosTop < 0) // removes arrow if arrow goes outside the screen

        {

            arrows.remove();

            clearInterval(FirInterval);

        }

    }

}

function hola() {

    var arrows = document.createElement('div');

    player.classList.add('fire');

    arrows.className = 'arrow up';

    document.body.appendChild(arrows);

    arrows.style.top = player.offsetTop + 'px';

    arrows.style.left = player.offsetLeft + 'px';

    PlayerShoot = true;

    setTimeout(

        function () {

            player.classList.remove('fire');

            PlayerShoot = false;

        }, 500);

    //go arrow in top direction.

    var FirInterval = setInterval(

        function () {

            ArrowAayo(arrows, FirInterval)

        }

        , 5);

}

//------------------------------------------All the function is to detect bomb location and  hp loss.----------------------------

function Pataka(bomb) {

    var explosion = document.createElement('div');

    explosion.className = 'explosion';

    document.body.appendChild(explosion);

    explosion.style.top = bomb.offsetTop + 'px';

    explosion.style.left = bomb.offsetLeft + 'px';

    console.log("COOL99")

    if (ElementCollide(player, explosion)) {

        console.log("COol")

        plauyerHit = true; //sets player collision to true

        player.classList.add('hit');    //plays human's animation when hit

        HP--; //decrease one health point

        life[0].remove(); //removes life from top left of the screen

        if (HP == 0) //if all 3 lives are gone call die function

        {

            Maro();

        }

        setTimeout(function () {

            playerHit = false; //sets player's state to normal

            player.classList.remove('hit'); //also removes the exploded animation

        }, 1000);

    }

    else {

        high.innerHTML = 'Points:' + Points;

        Points = Points + 5; //if player didnt get hit increment score points

        PlayTime++; //increment play time

    }

    setTimeout(function () {

        explosion.remove(); //explosion is removed after 100ms

    }, 100);

}

//----------------------------------The following function is for giving random speed in different  random direction-----------------------------------------------------------------------------------------------------------------------

function SpeedLeft(Index) {

    var rando = Math.floor(Math.random() \* 2);

    if (rando == 0) {

        LeftSpeed[Index] = ((Math.ceil(Math.random() \* 3)));

    }

    else {

        LeftSpeed[Index] = ((Math.ceil(Math.random() \* 3))) \* -1;

    }

}

function keyup(event) {

    if (GameOver) {

        return;

    }

    if (event.keyCode == 37) {

        leftPressed = false;

        lastPressed = 'left';

    }

    if (event.keyCode == 39) {

        rightPressed = false;

        lastPressed = 'right';

    }

    if (event.keyCode == 38) {

        upPressed = false;

        lastPressed = 'up';

    }

    if (event.keyCode == 40) {

        downPressed = false;

        lastPressed = 'down';

    }

    player.className = 'character stand ' + lastPressed;

}

function move() {

    if (GameOver || PlayerShoot) {

        return;

    }

    var positionLeft = player.offsetLeft;

    var positionTop = player.offsetTop;

    if (downPressed) {

        if (player.offsetTop < window.innerHeight - 30) {

            var TopPos = positionTop + 2;

            player.style.top = TopPos + 'px';

        }

        if (leftPressed == false) {

            if (rightPressed == false && !playerHit) {

                player.className = 'character walk down';

            }

        }

    }

    if (upPressed) {

        var TopPos = positionTop - 2;

        var element = document.elementFromPoint(0, TopPos);

        if (element.classList.contains('sky') == false) {

            player.style.top = TopPos + 'px';

        }

        if (leftPressed == false) {

            if (rightPressed == false && !playerHit) {

                player.className = 'character walk up';

            }

        }

    }

    if (leftPressed) {

        var newLeft = positionLeft - 2;

        if (newLeft > 0) {

            player.style.left = newLeft + 'px';

        }

        if (!playerHit) { player.className = 'character walk left'; }

    }

    if (rightPressed) {

        var newLeft = positionLeft + 2;

        var element = document.elementFromPoint(0, player.offsetTop);

        if (newLeft < window.innerWidth - player.offsetWidth) {

            player.style.left = newLeft + 'px';

        }

        if (!playerHit) {

            player.className = 'character walk right';

        }

    }

}

function keydown(event) {

    if (GameOver || PlayerShoot) {

        return;

    }

    if (event.keyCode == 32) {

        hola();

    }

    if (event.keyCode == 37) {

        leftPressed = true;

    }

    if (event.keyCode == 39) {

        rightPressed = true;

    }

    if (event.keyCode == 38) {

        upPressed = true;

    }

    if (event.keyCode == 40) {

        downPressed = true;

    }

}

// keeps on calling start game function 60times/1sec

function pap() {

    setInterval(startGame, 1000 / 60);

}

// When the page is load player calls 20 bombs

function myLoadFunction() {

    high = document.createElement('div');

    high.className = 'Points';

    document.body.appendChild(high);

    player = document.getElementById('player');

    document.addEventListener('keydown', keydown);

    document.addEventListener('keyup', keyup);

    start = document.getElementsByClassName('start');

    start[0].addEventListener('click', pap);

    sky = document.getElementsByClassName('sky');

    for (var i = 0; i < 14; i++) {

        Misile();

    }

    life = document.getElementsByClassName('health')[0].getElementsByTagName('li');

}

document.addEventListener('DOMContentLoaded', myLoadFunction);