Internet and Website Development

ASSIGNMENT 2

PART 2 DOCUMENTATION

ALEXANDRE ANGRIGNON 1502511

SUMMER XIA 1513551

Ship Design

The javascript for the ship design should look as follows:

```
function spaceship(x, y) {
    this.dead = false;
    this.x = x;
    this.y = y;

    this.shouldBeDrawn = true;
    this.goLeft = function() {
        spaceshipImage.src = spaceshipImageLeft; // Use Left image
        this.x -= 6; //move left
    }
    this.goRight = function() {
        spaceshipImage.src = spaceshipImageRight; // Use right image
        this.x += 5;
    }

    this.goUp = function() {
        // Use right image
        this.y -= 5;
    }

    this.goDown = function() {
        // Use right image
        this.y += 5.1;
    }

    this.draw = function() {
        if (this.shouldBeDrawn)
            c.drawImage(spaceshipImage, this.x, this.y, 100, 50);
        switch (lives) {
            case 3:
                c.drawImage(spaceshipImageStatic, canvas.width - 70, 10, 60, 30);
                c.drawImage(spaceshipImageStatic, canvas.width - 140, 10, 60, 30);
                c.drawImage(spaceshipImageStatic, canvas.width - 210, 10, 60, 30);
                break;
            case 2:
                c.drawImage(spaceshipImageStatic, canvas.width - 70, 10, 60, 30);
                c.drawImage(spaceshipImageStatic, canvas.width - 140, 10, 60, 30);
                break;
            case 1:
                c.drawImage(spaceshipImageStatic, canvas.width - 70, 10, 60, 30);
                break;
        }
```

Ship Movement

The javascript to move the ship should look as follows. The left, right, up and down arrows should all serve appropriate functionality:

```
window.addEventListener("keydown", doKeyDown, true);

function doKeyDown(event) {
    if (Spaceship1.dead == false) {

        if (!over) {

            switch (event.keyCode) {
                case 32:
                    var dx = 1;
                    if (spaceshipImage.src.includes(spaceshipImageRight)) {
                        dx = 1;

                    } else if ((spaceshipImage.src.includes(spaceshipImageLeft))) {
                        dx = -1;
                    }
                    Spaceship1.shoot(dx);
                    break;
                case 37: //left arrow
                    if (Spaceship1.x > -10) {
                        if (spaceshipImage.src.includes(spaceshipImageRight)) {
                            spaceshipImage.src = spaceshipImageLeft;
                        }
                        clearBg();
                        Spaceship1.goLeft();
                        Spaceship1.draw();
                        animateScene();
                    }
                    break;
                case 38: // Up Arrow
                    if (Spaceship1.y > -10) {

                        clearBg();
                        Spaceship1.goUp();
                        Spaceship1.draw();
                        animateScene();
                    }
                    break;
                case 39: // Right Arrow
                    if (Spaceship1.x < (canvas.width - 150)) {
                        if (spaceshipImage.src.includes(spaceshipImageRight)) {
                            spaceshipImage.src = spaceshipImageLeft;
                        }
                        clearBg();
                        Spaceship1.goRight();
                        Spaceship1.draw();
                        animateScene();
                    }
                    break;
                case 40: // Down Arrow
                    if (Spaceship1.y < (canvas.height - 100)) {

                        clearBg();
                        Spaceship1.goDown();
                        Spaceship1.draw();
                        animateScene();
                    }
                    break;

            }

        }
    }
}
```

Ship Shooting Beam

The javascript to make the ship shoot should look as follows. The space bar should serve appropriate functionality:

```
this.shoot = function(dx) {
    if (spaceshipImage.src.includes(spaceshipImageRight)) {
        var b = new bullet(this.x + 82, this.y + 19, 5, dx);

    } else {
        var b = new bullet(this.x - 25, this.y + 19, 5, dx);
    }
    bulletArray.push(b);
    fxLaser.play();



}
```

Enemy Explosion

```
function showExplosion() {
    if (counter < 101) {
        c.fillStyle = "orangered";
        c.beginPath();
        c.arc(globalmx, globalmy, globalmr * 0.75, 0, Math.PI * 2, false);
        c.fill();
        c.fillStyle = "salmon";
        c.beginPath();
        c.arc(globalmx, globalmy, globalmr * 0.5, 0, Math.PI * 2, false);
        c.fill();
        c.fillStyle = "yellow";
        c.beginPath();
        c.arc(globalmx, globalmy, globalmr * 0.25, 0, Math.PI * 2, false);
        c.fill();
        counter++;
    }
}


var counter;
var globalmx;
var globalmy;
var globalmr;
```

Collision Detection

```
function update() {
    formatTime();
    var mx, my, mr, bx, by;
    for (var i = enemyArray.length - 1; i >= 0; i--) {
        //get Enemy properties
        mx = enemyArray[i].x;
        my = enemyArray[i].y;
        mr = enemyArray[i].radius;

        //loop over beams
        for (var j = bulletArray.length - 1; j >= 0; j--) {
            //Get the bream properties
            bx = bulletArray[j].x;
            by = bulletArray[j].y;

            //Detect Collisions
            if (bulletArray[j].destroyTime == 0 && lengthBetweenPosition(mx, my, bx, by) < mr + 5) {
                bulletArray.splice(j, 1);
                points++;
                destroyEnemy(i);
                globalmx = mx;
                globalmy = my;
                globalmr = mr;
                counter = 0;
                setInterval(showExplosion, 1);
                fxExplodeEnemy.play();



            }

        }


        //Spaceship Collision
        if (lengthBetweenPosition(mx, my, Spaceship1.x + 50, Spaceship1.y + 50) < mr + 15) {
            if (points > 0)
                points--;
            Spaceship1.destroyShip();
            fxExplodeShip.play();
            destroyEnemy(i);
            lives--;
            if (lives == -1) {
                gameOver();

            }
        }
    }

}
```

Sound Effects and function

```
//Set up Audio
var fxLaser = new Sound("shootingSound.mp3", 20, 0.5);
// var fxExplodeShip = new Sound("beepSound.mp3");
var fxExplodeEnemy = new Sound("happySound.mp3", 20, 0.5);
// var fxGameOverSound = new Sound("gameOver.mp3");
var fxExplodeShip = new Sound("gameOver.mp3");
var fxBeginSound = new Sound("startSound.mp3", 1, 0.5);
fxBeginSound.loop = false;
```

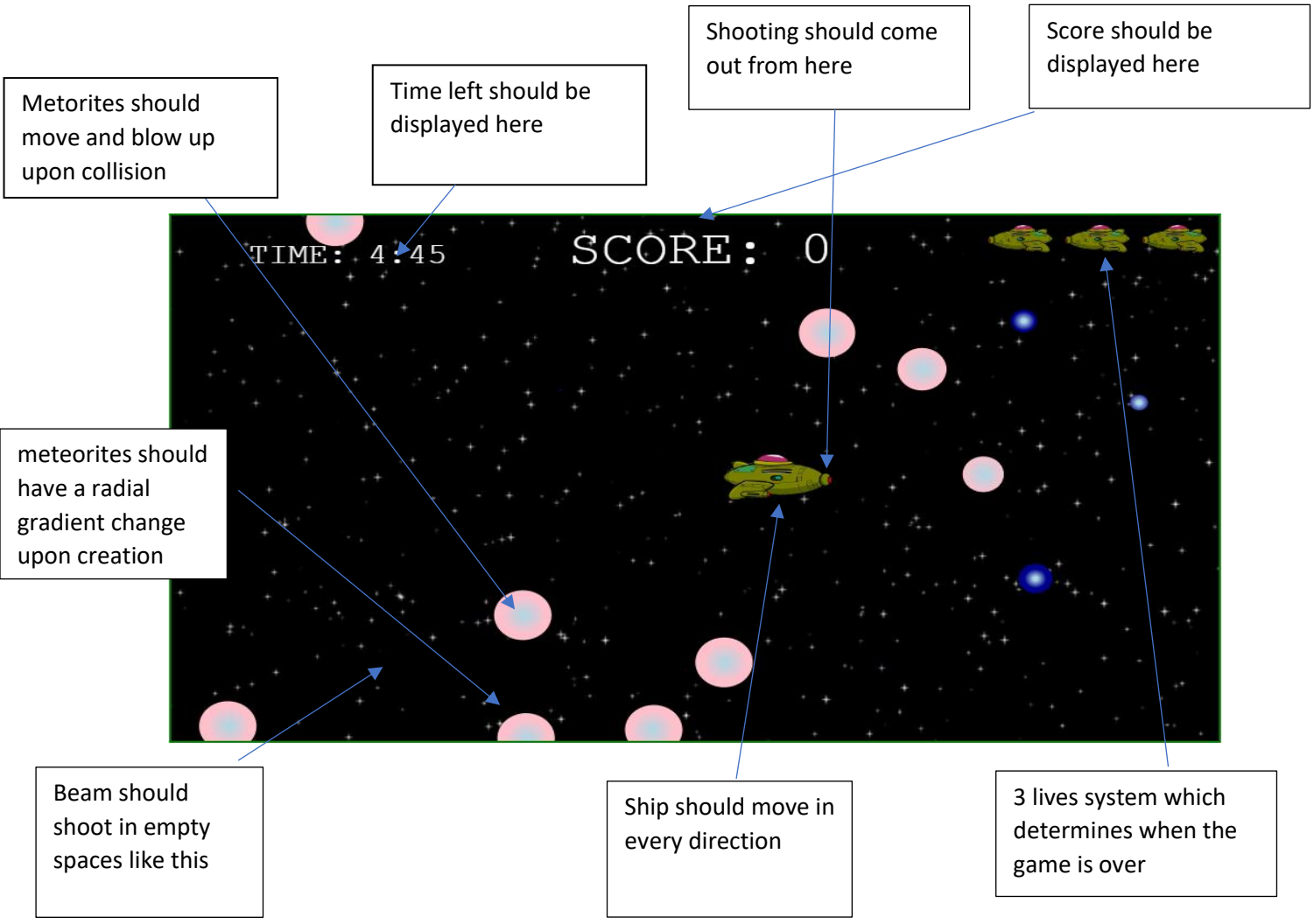
```
function Sound(src, maxStreams = 1, vol = 1.0) {
    this.streamNum = 0;
    this.streams = [];
    for (var i = 0; i < maxStreams; i++) {
        this.streams.push(new Audio(src));
        this.streams[i].volume = vol;

    }

    this.play = function() {
        for (var i = 0; i < maxStreams; i++) {
            this.streams.push(new Audio(src));
            this.streams[i].volume = volumevariable;

        }
        this.streamNum = (this.streamNum + 1) % maxStreams;
        this.streams[this.streamNum].play();
    }
}
```

Score and Toggle Game JQuery

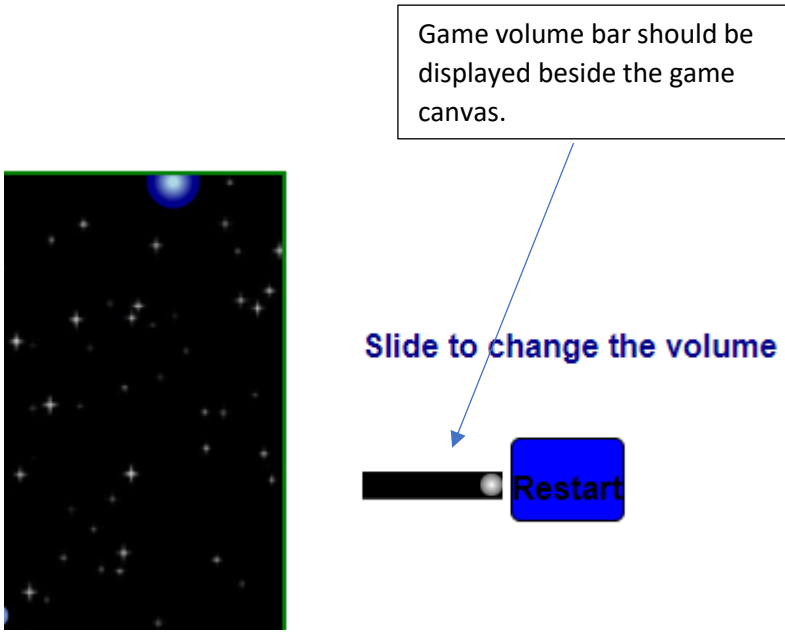
```
function displayScore() {
    c.fillStyle = "white";
    c.font = "small-caps 100px courier";
    c.fillText("GAME OVER", canvas.width - 200, 250);
    c.font = "small-caps 50px courier";
    if (points != 0)
        points++;
    c.fillText("Your score is: " + points, canvas.width - 250, 350);
}
var startTime;
var newEnemyIntervalHolder;
var animateEnemiesIntervalHolder;

function toggleGame(bool, time1) {
    if (bool) {
        endTime = time1;
        timeRemaining = endTime;
        fxBeginSound.play();
        fxBeginSound.loop = false;
        $("#volumeinstruction").show();

        $("#volumeslider").show();
        $("#myCanvas").show();
        $(".gameMenu").hide();
        $('#restartBtn').show();
        animateEnemiesIntervalHolder = setInterval(animateScene, 10);
        newEnemyIntervalHolder = setInterval(makeNewEnemy, 900);
        start = setInterval(update, 1000 / 50);
        startTime = setInterval(updateTimeRemaining, 1000);
        Spaceship1.dead = false;
        lives = GAME_LIVES;
    }
}


$(document).ready(function() {
    volumevariable = 1;
    $("#volumeinstruction").hide();
    $("#restartBtn").hide();
    $("#volumeslider").hide();
    $('#mins1').click(function() { toggleGame(1, 60); });
    $('#mins3').click(function() { toggleGame(1, 180); });
    $('#mins5').click(function() { toggleGame(1, 300); });
    $('#mins6').click(function() { toggleGame(1, 360); });

    $("#restartBtn").click(function() {
        $(".gameMenu").show();
        $('#myCanvas').hide();
        $("#restartBtn").hide();
        points = 0;
        gameOver();
        time = 0;
        enemyArray = [];
        bulletArray = [];
    });
```

Canvas Game

The canvas on which the game is played should look like this:



The Game Over message should look like this and stop the game completely once there are no more lives.


Game volume bar should be displayed beside the game canvas.
Slide to change the volume
Restart