



Project 2

HTML Game Development

Name:

Dina Ahmed Abbas Mohamed – 921

Karim Yasser-1049

Date:

May 7, 2013

About the game:

- A 2D web game implemented using HTML5, it consists of a shooting object trying to hit as many of the attacking enemies without being killed by them.
- The game consists of 2 levels of difficulty , hitting an enemy increments score by 10, reaching a certain score (50 in our case to ease the game testing) takes the player to a harder level with more powerful enemies in speed and number
- Missing an enemy that reaches the shooter and hits it results in losing a life, 3 lives are available, losing them leads to “game over”, and the game is reset to give it another try.

Main game components:

1) The shooter:

A sprite object, moving with the mouse cursor on the screen, shooting bombs, to hit enemies attacking from above.

2) The enemies:

Sprite animated objects attacking from above down towards the shooter to destroy it.

3) Background music:

An optional sound track, playing in the background.

4) Levels of easiness:

After reaching some score , the game gets more difficult by speeding and making the enemies more numerous.

Technologies used:

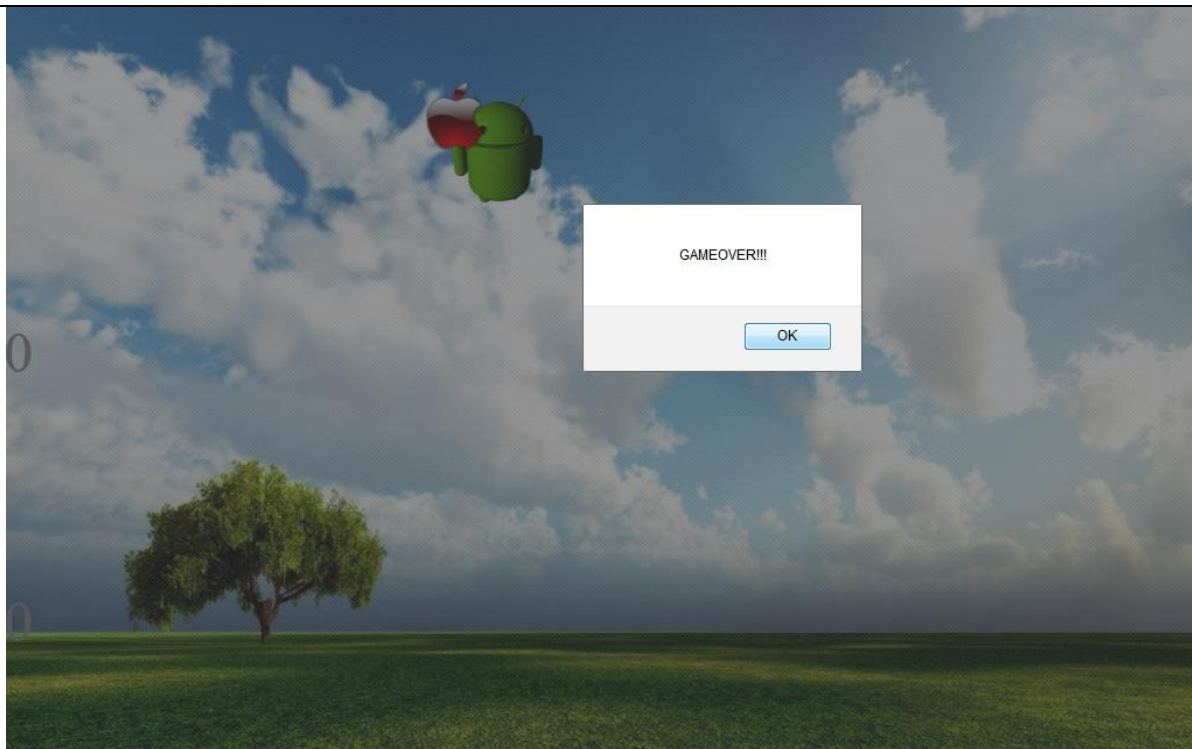
- HTML5
- Javascript
- Pixi.js javascript library
- Css

Use of “pixi.js”:

Pixi.js was useful in:

- Creating sprite objects
- Animation
- Rendering





HTML5:

```
<head>
<title>pixi.js example 2 loading a sprite sheet</title>
<meta charset="utf-8">
<meta name="viewport" content="initial-scale=1 maximum-scale=1 user-scalable=0" />
<link rel="stylesheet" href="storm.css">
<script src="jquery-1.8.3.min.js"></script>
<script src="pixi.js"></script>
<style>
body {
margin: 0;
padding: 0;
background-color: #000000;
}
</style>
<script src="pixi.js"></script>
</head>
```

```
//-----
```

soundtrack :

```
var music=document.getElementById("music1");
    music.autoplay=true;
    music.load();
    music.loop=true;
```

/soundtrack mute

```
//-----
```

```
function mute() {
    if (mutesound == 1) {

        music.pause();
        mutesound = 0;
    } else {

        music.play();
        mutesound = 1;
    }
}
```

```
//-----
```

```
ar mySprite = new PIXI.Sprite(PIXI.Texture.fromImage("sky.jpg")); //background
var texture = PIXI.Texture.fromImage("xtb1.gif"); //shooter
```

```
//-----
```

display function:

```
function update(){
```

```
if ( score > 50 ){
    aliens_speed = 80;
    starCount = 2;
}
```

```
collision();
spaceShipCollision();
```

```
delta = (new Date().getTime() - lastCalledTime)/1000;
lastCalledTime = new Date().getTime();
```

```
var current = new Date().getTime() - timer;
//alert(current);
if ( current>=500){
    clearExplosions();
    timer = new Date().getTime();
}
```

```
var disp = aliens_speed*delta;
```

```
if ( stars.length > 0){
    for (var i = 0; i < stars.length; i++){

        stars[i].position.y += disp;

    }
}
```

```
var speed = 150;
```

```
var disp2 = speed*delta;
if (M.length > 0){
    for ( var i = 0 ; i < M.length; i++){
```

```
M[i].position.y -= disp2;
```

```
}  
}
```

```
requestAnimationFrame(update);  
renderer.render(stage);
```

```
}
```

```
//-----
```

event handlers:

```
PIXI.InteractionManager.prototype.onMouseMove = function(event) {
```

```
SpaceShip.position.x = event.pageX;  
SpaceShip.position.y = event.pageY;
```

```
}
```

```
//-----
```

```
function spaceShipCollision(){
```

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

```
        if((stars[i].position.x + stars[i].width/4>=SpaceShip.position.x -  
SpaceShip.width/4)&&(stars[i].position.x - stars[i].width/4<=SpaceShip.position.x+SpaceShip.width/4) &&  
(stars[i].position.y + stars[i].height/4>=SpaceShip.position.y - SpaceShip.height/4)&&(stars[i].position.y -  
stars[i].height/4<=SpaceShip.position.y+SpaceShip.height/4)){
```

```
            //alert("BOMMM!!!");
```

```
            //alert(O);
```

```
            O--;
```

```
            document.getElementById('lives').innerHTML=O;
```

```
            var boom = new PIXI.Sprite(explosion);
```

```
            boom.anchor.x = 0.5;
```

```
            boom.anchor.y = 0.5;
```

```
            boom.position.x = stars[i].position.x;
```

```
            boom.position.y = stars[i].position.y;
```

```
            stage.addChild(boom);
```

```
            explosions.push(boom);
```

```
            stage.removeChild(stars[i]);
```

```
            //stage.removeChild(SpaceShip);
```

```
            stars.splice(i,1);
```

```
            if(O==0){
```

```
                alert("GAMEOVER!!!");
```

```
reset();
```

```
}
```

```
}
```

```
}
```

```
}
```

Reset to initial game state:

```
function reset(){
    for ( var i = 0 ; i<M.length ; i++){
        stage.removeChild(M[i]);
    }

    for ( var i=0; i<stars.length ; i++){
        stage.removeChild(stars[i]);
    }
    stage.removeChild(SpaceShip);
    score = 0;
    document.getElementById('score').innerHTML=score;
    O=3;
    document.getElementById('lives').innerHTML=O;
    loader.load();
}
```

```
}
```

```
//-----
```