CSE334-Internet programming

Bouncy Bird remastered

Introduction:

We remaster the famous most addicted game Bouncy Bird, A game where u play as bird and trying to fly through pipes.
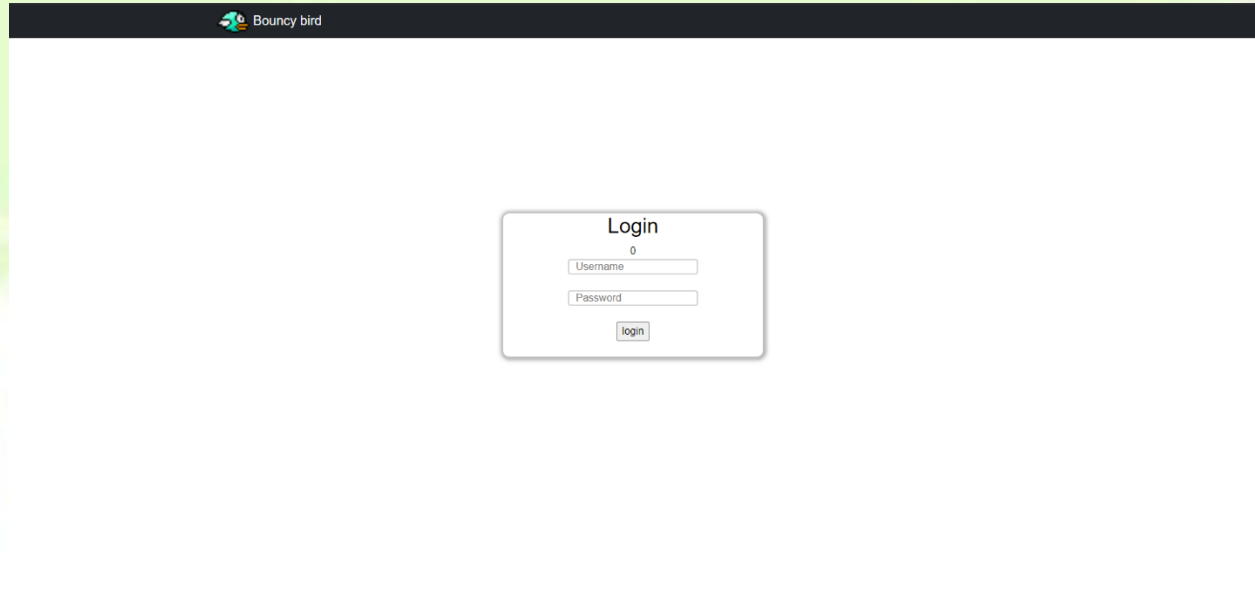
We developed the game using:

1. **HTML**
2. **Css**
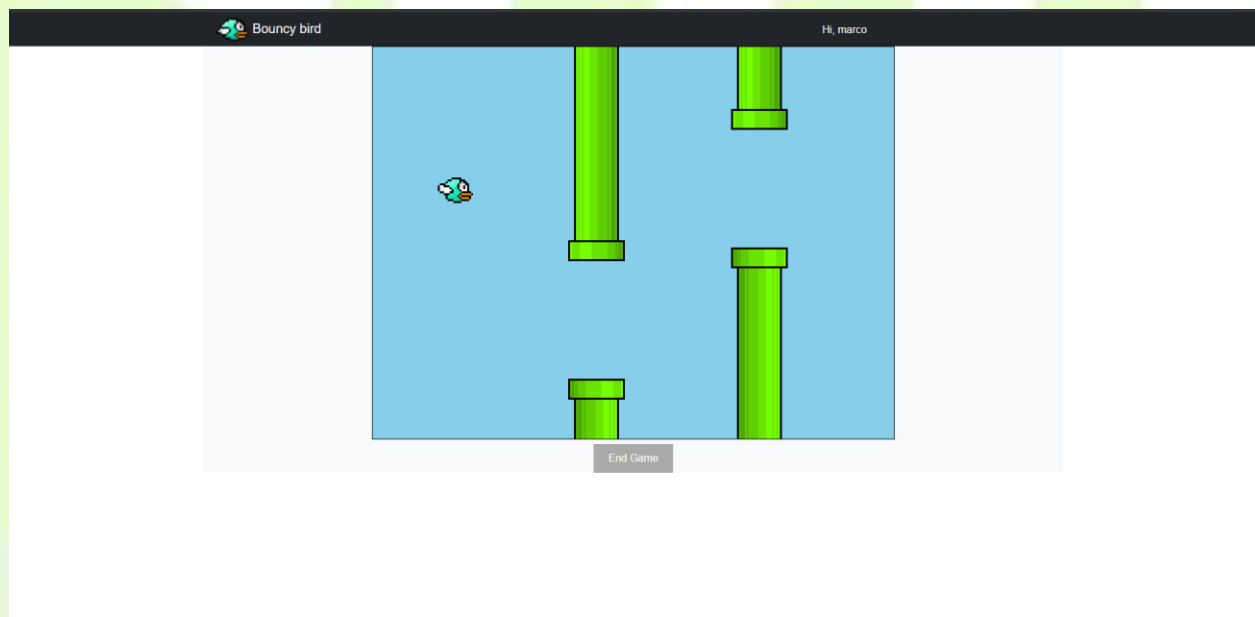3. **Javascript**
4. **Ajax**
5. **Php**

in the following pages we going to take tour around game scripts and screen shots.
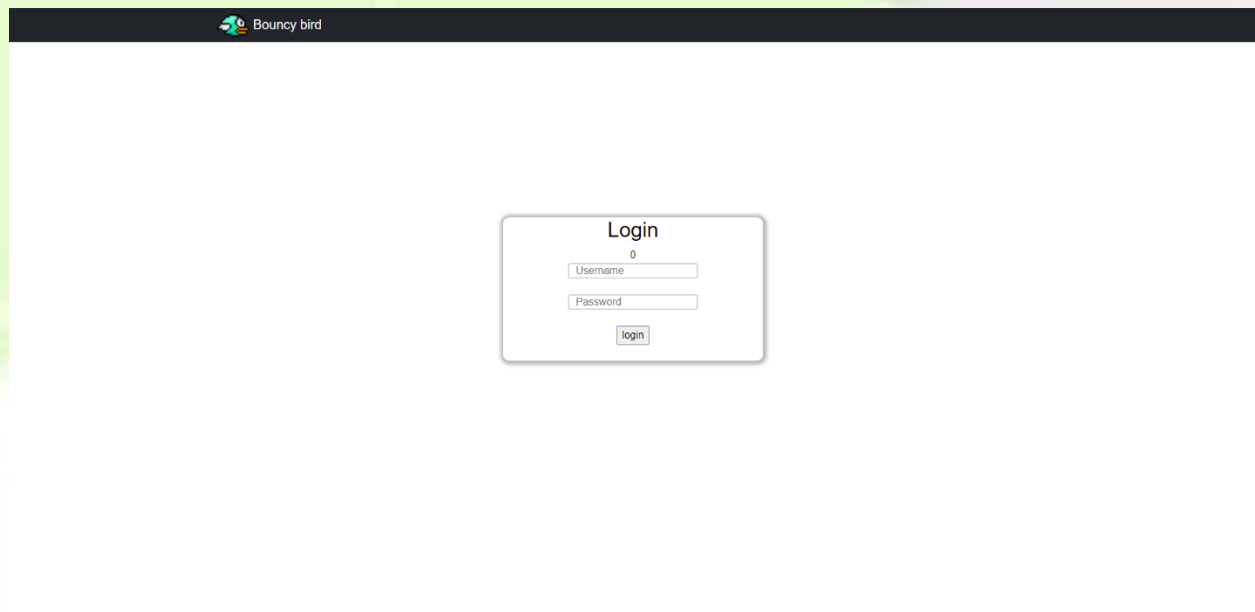
Game Overview:

1-Login Panel

2- Game panel

3-Scoreboard Panel

We going to discuss each of those panels and how this panels interact with each other 😊.

Login Panel:

This is the login in panel where you enter your username/ password to gain access to game (Currently doing a hardcode where u choose only one of the given usernames that already in database).

Of course, we started by creating the form by using HTML.

```html
<div  id="Login-Body">
    <form action="#"  onsubmit="return log(event)">
    <h2>
        Login
    </h2>
    <?php include('Bouncy_dataBase.php');?>
    <br><input type="text" placeholder="Username" class="inputs" id="name-input" name="username">
    <br><input type="password" placeholder="Password" class="password-input inputs" id="Pass" name="password">
    <br><input id="input6" type="submit" value="login">

    </form>
```

Then we started to create our database to enter database to save username/password/Score
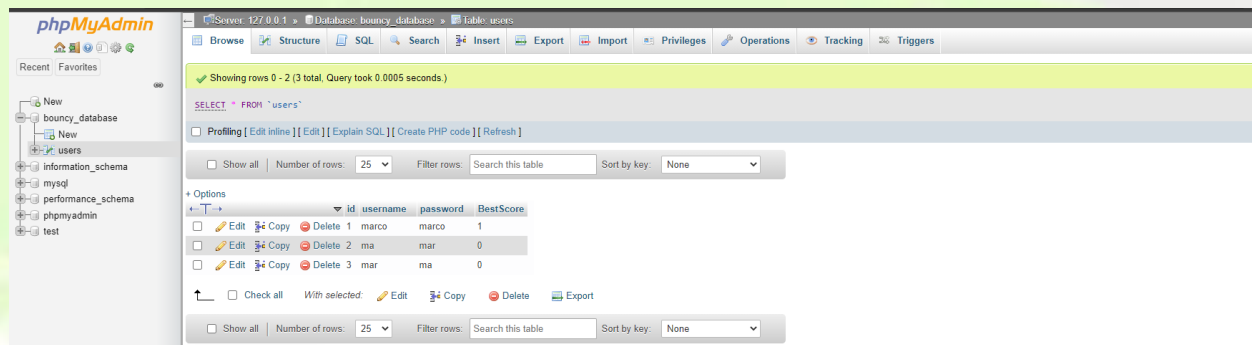
Once u hit the login button the Php role starts where it checks if the following username is in database or not.

```php
<?php

$x=0;
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$db = "Bouncy_dataBase";
$conn =  mysqli_connect($dbhost, $dbuser, $dbpass,$db) or die("Connect failed: %s\n". $conn -> error);

$query = 'SELECT * FROM users';
$results=mysqli_query($conn,$query);

function CloseCon($conn)
{
$conn -> close();
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST["username"];
    //echo $name;
    $pass = $_POST["password"];
    while($row = mysqli_fetch_assoc($results)) {
     if($row["username"]==$name && $row["password"]==$pass)
     {
       $x=$row["id"];
       break;
     }
     else{
     }
    }
  }
echo json_encode($x);

CloseCon($conn);
?>
```

Once your username matches the ones in database Javascript and Ajax roles starts to interfere where we echo encoded Json Object Boolean($x) from our Php file:

```php
    }
 echo json_encode($x);
```
and then receive the response in the Javascript where it move us to the Game Panel

```
13      <script >
14 ∨    const user= {
15          id: null
16      };
17      function log(e){
18      e.preventDefault();
19      $.post("Bouncy_dataBase.php",
20 ∨    {
21        username: $("#name-input").val(),
22        password: $("#Pass").val()
23
24      },
25 ∨    function(data, status){
26        alert("Data: " + data + "\nStatus: " + status);
27 ∨      if(status=="success"){
28            user.id= JSON.parse(data);
29            var id= user.id;
30          console.log({id})
31          let player_name=document.getElementById("name-input").value
32
33          player_span.innerHTML=("Hi, "+player_name)
34
35          Login.remove()
36          document.body.append(Game)
37        }
38      });
39      return false;
40  }</script>
```

Game Panel:
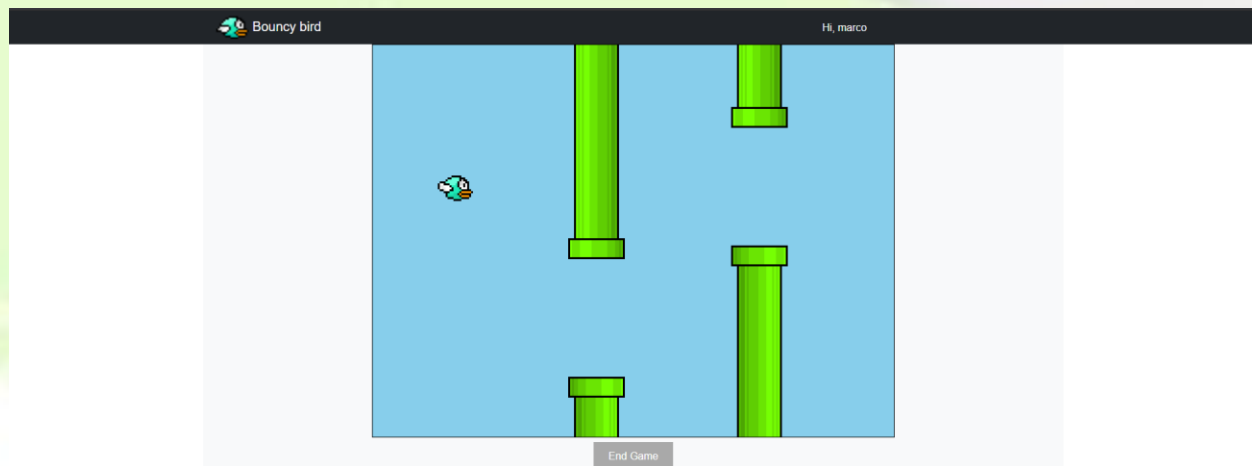
The game is simple where u just try not to hit the pipes by pressing Space to fly though them.

Javascript was our main scope here, Where we created functions for control

```
1  ∨ document.onkeydown = function userInput() {
2
3  ∨     switch (event.keyCode) {
4  ∨         case 32:
5  ∨             if (state.current == state.ready) {
6                   state.current = state.play;
7                   birdFlap();
8  ∨             } else if (state.current == state.play) {
9                   birdFlap();
10 ∨             } else {
11                  state.current = state.ready;
12                  reset();
13                  }
14                  break;
15 ∨         default:
16                  break;
17         }
18 }
```

Pipe generators where it can randomly created on the panel

```
1    const pipeUp = new Image();
2    const pipeDown = new Image();
3    pipeUp.src = "sprite/pipeUp.png";
4    pipeDown.src = "sprite/pipeDown.png";
5    const pipe_width = 29;
6    const pipe_height = 155;
7    const pipe_scaleX = 3;
8    const pipe_scaleY = 2.9;
9    const dx = 0.7;
10   const pipe_gap = 180;
11   class Pipe {
12       constructor(x, y) {
13           this.x = x;
14           this.y = y;
15       }
16       draw() {
17           drawPipeUp(0, 0, this.x, this.y);
18           drawPipeDown(0, 0, this.x, this.y + pipe_height * pipe_scaleY + pipe_gap);
19       }
20       update() {
21           if (state.current == state.play) {
22               this.x -= dx;
23           }
24
25       }
26       getTopAccY() //highest point before collision in up pipe
27           {
28               return this.y + (pipe_height * pipe_scaleY);
29           }
30       getLowAccY() {
31           return this.y + (pipe_height * pipe_scaleY) + pipe_gap;
32       }
33   }
```

```
34
35    let pipe_obj = new Pipe(c.width, -90);
36
37    function drawPipeUp(frameX, frameY, canvasX, canvasY) {
38        ctx.drawImage(pipeUp,
39            frameX, frameY, pipe_width, pipe_height,
40            canvasX, canvasY, pipe_width * pipe_scaleX, pipe_height * pipe_scaleY);
41    }
42
43    function drawPipeDown(frameX, frameY, canvasX, canvasY) {
44        ctx.drawImage(pipeDown,
45            frameX, frameY, pipe_width, pipe_height,
46            canvasX, canvasY, pipe_width * pipe_scaleX, pipe_height * pipe_scaleY);
47    }
48    const pipe_distance = 250;
49
50    let pipes = [];
51
52    function init_PipesList() {
53        let rand = Math.random() * -300 - 100;
54        pipes = [new Pipe(c.width - pipe_distance * 2, rand)];
55        rand = Math.random() * -300 - 100;
56        pipes.push(new Pipe(c.width - pipe_distance, rand));
57    }
58
59
60    function pipeGenerator(frame_count) {
61        let rand = Math.random() * -300 - 100;
62        if (state.current == state.play) {
63            if (c.width - pipes[pipes.length - 1].x > pipe_distance) {
64                pipes.push(new Pipe(c.width, rand));
65            }
66            pipes.forEach(p => {
67                p.update();
68            });
69            //remove the pipe when it goes out of screen
70            if (pipes[0].x < -pipe_width * pipe_scaleX) {
71                pipes.shift();
72
73                score.value += 1;
74                score.best = Math.max(score.value, score.best);
75                localStorage.setItem("best", score.best);
76            }
77        }
78    }
79    function drawPipes() {
80        pipes.forEach(p => {
81            p.draw();
82        });
```

Then we create our Flappy Bird Functios:

```
1    const bird_img = new Image();
2    bird_img.src = "sprite/birdSprite.png";
3
4    const b_width = 18;
5    const b_height = 13;
6    const b_gravity = 0.1;
7    const b_scale = 3;
8
9    class Bird {
10       constructor(x, y) {
11           this.x = x;
12           this.y = y;
13           this.velY = 1;
14       }
15       draw(i) {
16           drawBird(i, 0, this.x, this.y);
17       }
18       update() {
19           if (state.current == state.play || state.current == state.over) {
20               gravityOn();
21           } else {
22               gravityOff()
23           }
24           if (state.current == state.over) {
25               if (this.y + b_height * b_scale < canv_height) {
26                   this.x += 0.7;
27               }
28
29           }
30       }
31   }
32
33   let bird = new Bird(100, 200);
34
35   function init_bird() {
36       bird = new Bird(100, 200);
37   }
38
39   function drawBird(frameX, frameY, canvasX, canvasY) {
40       ctx.drawImage(bird_img,
41           frameX * b_width, frameY * b_height, b_width, b_height,
42           canvasX, canvasY, b_width * b_scale, b_height * b_scale);
43   }
44
```

```
46
47   function birdFlap() {
48       bird.velY *= -0.2;
49       bird.velY -= 3;
50   }
51
52   function gravityOn() {
53       bird.y += bird.velY;
54
55       if (bird.y >= canv_height - b_height * b_scale) {
56           bird.velY = 0;
57           bird.y = canv_height - b_height * b_scale;
58       } else {
59           bird.velY += b_gravity;
60       }
61   }
62
63   function gravityOff() {
64       bird.velY = 0;
65   }
```

And Some Physics for the collision between the bird and Pipe:

```js
1  function isCollide(bird, pipe) {
2      if ((bird.x <= pipe[0].x + pipe_width * pipe_scaleX) && (bird.x + b_width * b_scale >= pipe[0].x)) {
3          if (bird.y < pipe[0].getTopAccY() || bird.y + b_height * b_scale > pipe[0].getLowAccY()) {
4              return true;
5          }
6      } else {
7          return false;
8      }
9  }
```

Then after creating all those files we create the Game lifespan:

```js
js > JS gameScript.js > ...
1   ctx.imageSmoothingEnabled = false;
2   const state = {
3       current: 0,
4       ready: 0,
5       play: 1,
6       over: 2
7   }
8   state.current = state.ready;
9   window.requestAnimationFrame(loop);
10  let frame_count = 0; //frames counter
11  let i = 0; //bird animation index
12
13  init_PipesList();
14  //The Game Loop
15  function loop() {
16      //1- Draw
17      //Draw background
18      ctx.clearRect(0, 0, c.width, c.height);
19      ctx.fillStyle = "skyblue";
20      ctx.fillRect(0, 0, c.width, c.height);
21      //Draw pipes and bird and score
22      drawPipes();
23      bird.draw(i);
24      score.draw();
25
26      //2- Update
27      bird.update();
28      pipeGenerator(frame_count);
29
30      if (isCollide(bird, pipes)) {
31          state.current = state.over;
32      }
33
34      //3- increment the frame counter
35      //frame count and index of bird to animate it
36      frame_count++;
37      if (frame_count % 10 == 0 && state.current != state.over) {
38          i ^= 1; // toogle the animation of bird
39      }
40      if (frame_count > 1000000) {
41          frame_count = 0;
42      }
43
44      window.requestAnimationFrame(loop);
45  }
46
47
```

```
function reset() {
    //CLear the pipe array
    pipes.length = 0;
    init_PipesList();
    //clear the bird
    init_bird()
        //set game state
    state.current = state.ready;

    score.reset();
}
```

The game keep reseting itself everytime you die and saving your highest score. You can end the game by pressing the End game button and then it moves you to our Scoreboard Panel where it show your score with all users in the database

ScoreBoard Panel:

The score board was confusing but we managed it we basically receive the last score u did in the game panel then once you hit end game button the score is sent to database, so then again we started using our Php and ajax skills to receive the score from the

database and print it on scree:

```
5
6  function end_click(){
7      var id= user.id;
8
9      var currentScore= score.value;
0      console.log({currentScore})
1      console.log({id})
2      $.post("ScoreUpdate.php",
3      {
4        id: id,
5        score: currentScore
6      },
7      function(data, status){
8          Game.remove()
9          document.body.append(board)
0            var users= JSON.parse(data);
1          users_output(users);
2      });
3  }
4
```

Echoing the users info from php file:

```php
ScoreUpdate.php
1   <?php
2
3   $x=0;
4   $dbhost = "localhost";
5   $dbuser = "root";
6   $dbpass = "";
7   $db = "Bouncy_dataBase";
8   $conn = mysqli_connect($dbhost, $dbuser, $dbpass,$db) or die("Connect failed: %s\n". $conn -> error);
9   // $id = 1;
10  // $score = 4;
11
12  $id = $_POST["id"];
13  $score = $_POST["score"];
14  $query = "SELECT * FROM users where id='".$id."'";
15  $results=mysqli_query($conn,$query);
16
17  function CloseCon($conn)
18  {
19  $conn -> close();
20  }
21  if ($_SERVER["REQUEST_METHOD"] == "POST") {
22      if(mysqli_num_rows($results) > 0) {
23          while($row = mysqli_fetch_assoc($results)) {
24              if($score > $row["BestScore"]){
25                  $query2= "update users set BestScore='".$score."' where id='".$id."'";
26                  if(mysqli_query($conn,$query2)){
27                      // echo "success";
28                  }else{
29                      // echo "failed ".mysqli_error($conn);
30                  }
31
32              }
33          }
34      }
35  }
36  $users_query = "SELECT id, username, BestScore FROM users";
37  $users_results=mysqli_query($conn,$users_query);
38  $users= [];
39  while($row= mysqli_fetch_assoc($users_results)){
40      $users[]= $row;
41  }
42  echo json_encode($users);
43
44  CloseCon($conn);
45  ?>
```

Finally :

We hope we explained everything clearly and how the panels interact with each other and how our game works… Hope you enjoy it

Team Members:

Marco Kamal Saad Naoum 17p6018

Ahmed Fawzy Amer 17P6035

Ahmed Walid Elhakim 17p6061

Hassan Ahmed Abd Alglil 17p8011

Noor El-Din Walid Fathy 17p6084