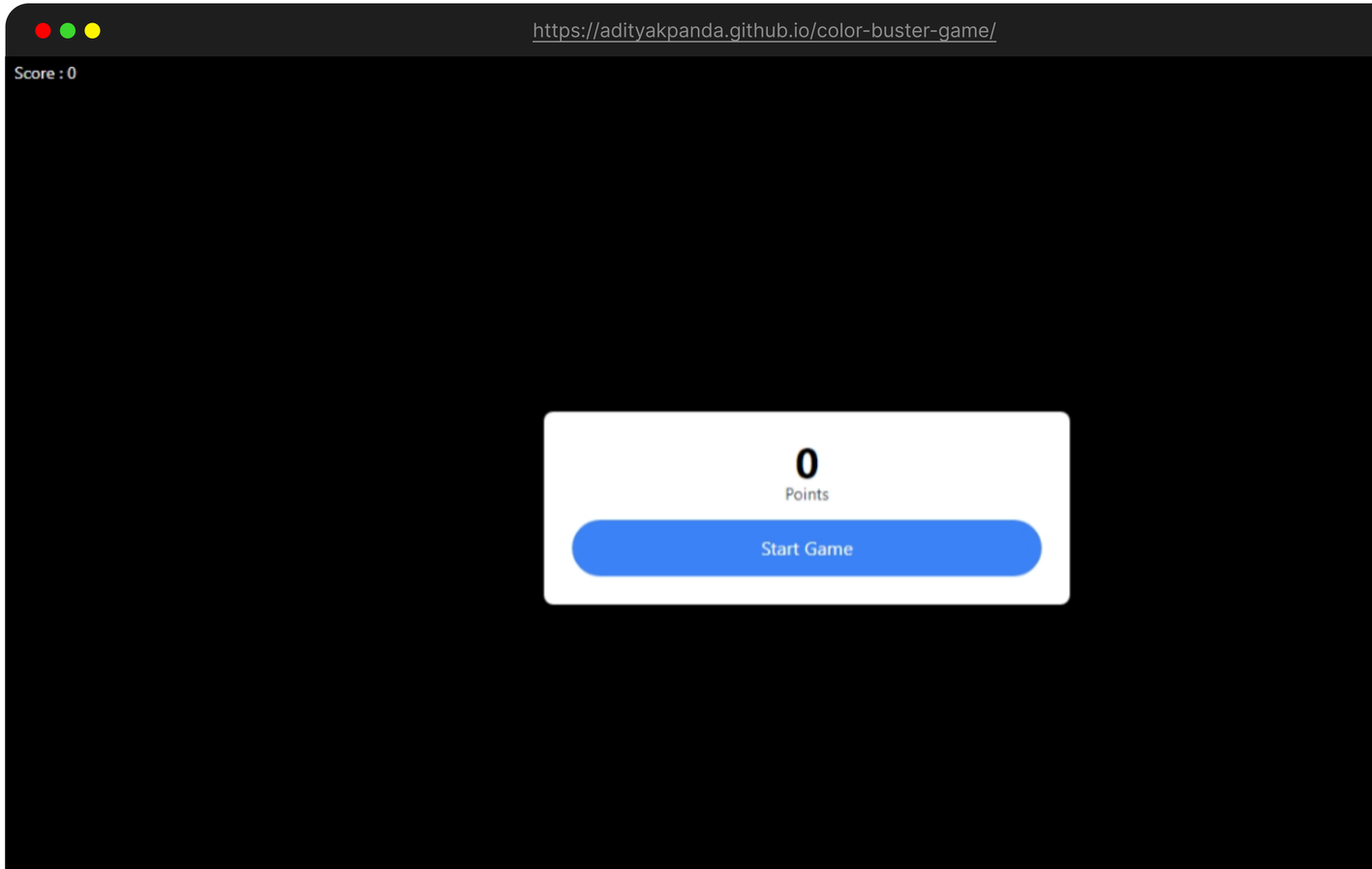
ICFAI GROUP

Special Project For Web Enabled Technologies (CS205)

Under the guidance of Madhu Bandari Sir

Color Blast

Shooting Game



Tools used

Name : Aditya Kumar Panda
En.No : 21STUCHH010139
Section : B

Abstract

What is my project ?	This project uses HTML , CSS and JS to make a shooting game
Theme of my Project	The main theme of this project is to make a shooting game which is fun to play around made using html css and js
Modules of Project	The webpage has mainly 4 main components Player Projectile Enemy Particles
Application of Project	This project can be a small example of how games can be made using html css and js.

Requirements

Hardware Requirements	Software Requirements
4 GB RAM Any Operating System Atleast 40 GB storage	Visual Studio Code Any Web Browser

HTML code

ICFAI GROUP

```
<style>
    body
    {
        margin: 0;
        background: black;
    }
</style>

<div class="fixed text-white text-sm ml-2 mt-1 select-none"><span>Score : </span><span id="scoreEl">0</span></div>
<div class="fixed inset-0 flex items-center justify-center text-center" id="box">
    <div class="bg-white max-w-md w-full p-6 rounded-lg">
    <h1 class="text-4xl font-bold leading-none select-none" id="boxscore">0</h1>
    <p class="text-sm text-gray-700 mb-3 select-none">Points</p>
    <div><button class="bg-blue-500 text-white w-full py-3 rounded-full select-none" id="StartButton">Start Game</button></div>
</div></div>
<canvas></canvas>
<script src="https://cdn.tailwindcss.com"></script>
<link rel="stylesheet" type="text/css" href="style.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.11.5/gsap.min.js" integrity="sha512-cOH8ndwGgPo+K7pTvMrqYbmI8u8k6Sho3js0gOqVWTmQ
<script src="script.js"></script>
```

Work Flow of my Project.

Define Canvas

```
<canvas></canvas>
```

This game is defined on html element canvas
and all the elements of the game are drawn on this canvas element.

Create Player

```
/* to define player properties */
class Player
{
    constructor(x , y , radius , color)
    {
        this.x = x
        this.y = y
        this.radius = radius
        this.color = color
    }
    draw()
    {
    c.beginPath()
    c.arc(this.x , this.y , this.radius , 0 , Math.PI * 2 , false); // to draw the circle
    // thi.x - x cordinate | this.y - y cordinate | this.radius for radius | start angle - 0 | arc
    c.fillStyle = this.color; // to give  the style of fill
    c.fill(); // to fill the circle
}
}
```

what are those functions about ?

what is this constructor about ?

```
// creating an instance of player
let player = new Player(x , y , 10 , 'white' )
```

What is c ?

```
const canvas = document.querySelector('canvas');
```

```
/* api to use the canvas */
const c = canvas.getContext('2d');
```

ICFAI GROUP

Projectiles

How is it different from player ?

```
// creating a class for projectile to define its properties
class Projectile
{
    constructor( x , y , radius , color , velocity)
    {
        this.x = x
        this.y = y
        this.radius = radius
        this.color = color
        this.velocity = velocity
    }
    draw()
    {
    c.beginPath()
    c.arc(this.x , this.y , this.radius , 0 , Math.PI * 2 , false); // to draw the circle
    // thi.x - x cordinate | this.y - y cordinate | this.radius for radius | start angle - 0 | arc - math.pi *
    c.fillStyle = this.color; // to give  the style of fill
    c.fill(); // to fill the circle
    }
    update()
    {
        this.draw();// this would allow us to call the above function draw along when update is called
        this.x = this.x + this.velocity.x;
        this.y = this.y + this.velocity.y;
    }

}
```

What is this update function about ?

```
//when there is a click these set of instructions are executed
addEventListener('click',(event) =>
{
    const angle = Math.atan2(event.clientY-y,event.clientX-x)
    const velocity = {x:Math.cos(angle) * 6 ,y:Math.sin(angle) * 6 }
    // to create dynamic projectiles instead of static like in that of projectile , projectile1
    projectiles.push(new Projectile(x , y, 5,'white',velocity))
```

Enemies

```
// creating a class for enemies to define its properties
class Enemy
{
    constructor( x , y , radius , color , velocity)
    {
        this.x = x
        this.y = y
        this.radius = radius
        this.color = color
        this.velocity = velocity
    }
    draw()
    {
    c.beginPath()
    c.arc(this.x , this.y , this.radius , 0 , Math.PI * 2 , false); // to draw the circle
    // thi.x - x cordinate | this.y - y cordinate | this.radius for radius | start angle - 0 | arc - math.pi *
    c.fillStyle = this.color; // to give  the style of fill
    c.fill(); // to fill the circle
    }
    update()
    {
        this.draw();// this would allow us to call the above function draw along when update is called
        this.x = this.x + this.velocity.x; // it means the x term defined for velocity - velocity {x: , y:}
        this.y = this.y + this.velocity.y; // it means the y term defined for velocity - velocity {x: , y:}
    }

}
```

```
function spawnEnemies()
{
    setInterval(() =>{
        // defining the value of the enemy
        let x ; // these are defined outside so that it can be used even outside the if cond
        let y ;
        //const radius = 30; - IT WOULD CREATE ONLY ONE SIZE OF ENEMIES
        // const radius = Math.random() * 30 ; // this would create of diff sizes but there
        const radius = Math.random() * (40 - 6) + 6 ; // this means we would get enemies of
        //const x = Math.random() * canvas.width;
        //const y = Math.random() * canvas.height;
        //using the above values would create a chance for enemy too close to the player but
        //-----------------------------------------------------------------------------
        //const x = Math.random() < 0.5 ? 0- radius : canvas.width + radius ; //this would m
        //in the above case if the value is < 0.5 that is on left side it would spawn from l
        //const y = Math.random() < 0.5 ? 0- radius : canvas.height + radius ; //this would
        // the above case has a problem that the enemies spawn only from corners since the x
        //-----------------------------------------------------------------------------
        if(Math.random()< 0.5)  // that is on the left side of screen
        {
            x = Math.random() < 0.5 ? 0- radius : canvas.width + radius ; //this would make
            y = Math.random() * canvas.height;
        }
        else
        {
        x = Math.random() * canvas.width;
        y = Math.random() < 0.5 ? 0 - radius : canvas.height + radius ; //this would make th
    }
        const color = `hsl(${Math.random() * 360} , 50% , 50%)`;
        // since it comes from diff positions to center player it is subttracted from center
        //math.atan2 returns the angle , and it takes (y,x)
        const angle = Math.atan2(canvas.height / 2 - y, canvas.width/2 - x  )
        const velocity = {  x: Math.cos(angle), y: Math.sin(angle) }
        enemies.push(new Enemy(x,y,radius, color,velocity))


    } ,1000) //1000 - 1s it is at what interval it should call enemies
    //this means this functions defined inside set interval will execute after every 1000ms
}
```

Particle

```
constructor( x , y , radius , color , velocity)
{
    this.x = x;
    this.y = y;
    this.radius = radius;
    this.color = color;
    this.velocity = velocity;
    this.alpha = 1;
}
draw()
{
c.save();
c.globalAlpha = this.alpha;
c.beginPath()
c.arc(this.x , this.y , this.radius , 0 , Math.PI * 2 , false); // to draw the circle
// thi.x - x cordinate | this.y - y cordinate | this.radius for radius | start angle - 0 | arc - math.pi * 2 (360) | false
c.fillStyle = this.color; // to give  the style of fill
c.fill(); // to fill the circle
c.restore();
}
update()
{
    this.draw();// this would allow us to call the above function draw along when update is called
    this.velocity.x *= friction;
    this.velocity.y *= friction;
    this.x = this.x + this.velocity.x; // it means the x term defined for velocity - velocity {x: , y:}
    this.y = this.y + this.velocity.y; // it means the y term defined for velocity - velocity {x: , y:}
    this.alpha -= 0.01
}
```

```
particles.forEach((Particle , index) => {
    if (Particle.alpha <= 0) {
        particles.splice(index , 1);}
        else
        {
            Particle.update();
        }});
```

Final Execution Code

```
StartButton.addEventListener('click',() =>
{
    init();
    animate(); // to call the function
    spawnEnemies();  // to call the function
    boxel.style.display = 'none';
})
```

init( )

```
let  projectiles = []
let  enemies = []
let particles = []

function init()
{
score = 0;
player = new Player(x , y , 10 , 'white' )
 projectiles = []
enemies = []
particles = []
boxscore.innerHTML = score;
ScoreEl.innerHTML = score;
```

animate( )

```
let animationId; // a variable created to assign the animation request to it
let score = 0;
function animate()
{
    //this makes it will call the same fucntion again and again
    animationId = requestAnimationFrame(animate);
    c.fillStyle = 'rgba(0 , 0 , 0 , 0.1)';
    c.fillRect(0,0,canvas.width ,canvas.height);//this is to keep the canvas clear instead of geeting lines we get balls
    //but the above statement will even clear the palyer so we need to call the player.draw() everytime so it is visible
    player.draw();
    particles.forEach((Particle , index) => {
        if (Particle.alpha <= 0) {
            particles.splice(index , 1);}
            else
            {
                Particle.update();
            }});


    projectiles.forEach((projectile , index) => {
        projectile.update();
        // the below function would remove the projectiles which are out of the frame
        if(projectile.x + projectile.radius < 0 || projectile.x - projectile.radius > canvas.width ||
            projectile.y + projectile.radius < 0 || projectile.y -  projectile.radius > canvas.height)
        {
            setTimeout(() => {
                projectiles.splice(index , 1);

            }, 0);
        }


    })
```

continued...

```
enemies.forEach((enemy , index) => {enemy.update()
    const dist = Math.hypot(player.x - enemy.x , player.y - enemy.y);
    if(dist - player.radius - enemy.radius <  1 )
    {
        cancelAnimationFrame(animationId); // to stop the animation id when the palyer and enemy collide.
        boxel.style.display = 'flex';
        boxscore.innerHTML = score;

    }

    // note we need to write the below fucnction inside the main enemies so that the enemy attribute can be used inside the below fucntion.
    projectiles.forEach((projectile , projectileIndex ) => {
        const dist = Math.hypot(projectile.x - enemy.x , projectile.y - enemy.y);
        //the above statement is used to find the dist between enemy and projectile

    if(dist - projectile.radius - enemy.radius <  1 ) // dist - distance between center , so for collision we subtract the radii as well
    {

        for(i = 0 ; i < enemy.radius * 2 ; i ++)
        {
            particles.push(new Particle(projectile.x , projectile.y , Math.random() * 2 , enemy.color ,
            {x : (Math.random()- 0.5) * (Math.random() * 6), y : (Math.random()- 0.5) * (Math.random() * 6)}))
        }
        if (enemy.radius - 10 > 7 )
        {
            score += 10;
        ScoreEl.innerHTML = score;
            gsap.to(enemy,{radius :enemy.radius - 10});
            //enemy.radius -= 10;
            setTimeout(()=> {
                projectile.splice(projectileIndex , 1); // to remove the projectile
                } , 0)
            }
        else
        {
            score += 25;
        ScoreEl.innerHTML = score;
        // on removing the enemy from the array , they all flash , so using this would not do that
        setTimeout(()=> {
        enemies.splice(index , 1); // it is to remove the enemy after collision
        projectile.splice(projectileIndex , 1); // to remove the projectile
        } , 0)
    }
    }
     })
})



}
```

Code Files

All the code files and other resources are provided in the git hub repo.

https://github.com/adityakpanda/color-buster-game.git

Website Link

https://adityakpanda.github.io/color-buster-game/

Limitations in the website

Not mobile optimized

Thank You

Name : Aditya Kumar Panda
En.No : 21STUCHH010139
Section : B