Create palyer and enemy :

Index.js

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

const c = canvas.getContext('2d');

canvas.width = 1024;

canvas.height = 576;

c.fillRect(0, 0, canvas.width, canvas.height);

const gravity = 0.2;

class Sprite {

    constructor(position,velocity) {

        this.position = position;

        this.velocity = velocity;

        this.height = 150;

    }

    draw() {

        c.fillStyle = 'red';

        c.fillRect(this.position.x, this.position.y, 50, this.height);

    }

    update() {

        this.draw();

        this.position.y += this.velocity.y;

        if (this.position.y + this.height + this.velocity.y >= canvas.height) {

            this.velocity.y = 0;

    }    else

        this.velocity.y += gravity;

    }

}

const player = new Sprite(

  { x: 0, y: 0 },

  { x: 0, y: 0 }

);

const enemy = new Sprite(

  { x: 400, y: 100 },

  { x: 0, y: 0 }

);

console.log(player);

function animate() {

    window.requestAnimationFrame(animate);

    c.fillStyle = 'black';

    c.fillRect(0, 0, canvas.width, canvas.height);

    player.update();

    enemy.update();}

animate();

index.html:

<canvas></canvas>

<script src="index.js"></script>

Move Characters with Event Listeners:

index.js:

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

const c = canvas.getContext('2d');

canvas.width = 1024;

canvas.height = 576;

c.fillRect(0, 0, canvas.width, canvas.height);

const gravity = 0.7;

class Sprite {

    constructor(position,velocity) {

        this.position = position;

        this.velocity = velocity;

        this.height = 150;

        this.lastKey;

    }

    draw() {

        c.fillStyle = 'red';

        c.fillRect(this.position.x, this.position.y, 50, this.height);

    }

    update() {

        this.draw();

        this.position.x += this.velocity.x;

        this.position.y += this.velocity.y;

        if (this.position.y + this.height + this.velocity.y >= canvas.height) {

            this.velocity.y = 0;

    }    else

        this.velocity.y += gravity;

    }

}

const player = new Sprite(

  { x: 0, y: 0 },

{ x: 0, y: 0 }

);

const enemy = new Sprite(

{ x: 400, y: 100 },

 { x: 0, y: 0 }

);

console.log(player);

const keys = {

    a: {

        pressed: false

    },

    d: {

        pressed: false

    },

    w: {

        pressed: false

    }

    ,    ArrowLeft: {

        pressed: false

    },

    ArrowRight: {

        pressed: false

    }

    ,    ArrowUp: {

        pressed: false

    }

}

function animate() {

    window.requestAnimationFrame(animate);

    c.fillStyle = 'black';

    c.fillRect(0, 0, canvas.width, canvas.height);

    player.update();

    enemy.update();

    player.velocity.x = 0;

    enemy.velocity.x = 0;

    //player movement

    if (keys.a.pressed && player.lastKey==='a') {

        player.velocity.x = -5;

    }else if (keys.d.pressed && player.lastKey==='d') {

        player.velocity.x = 5;

    }

    //enemy movement

    if (keys.ArrowLeft.pressed && enemy.lastKey==='ArrowLeft') {

        enemy.velocity.x = -5;

    }else if (keys.ArrowRight.pressed && enemy.lastKey==='ArrowRight') {

        enemy.velocity.x = 5;

    }

}

animate();

window.addEventListener('keydown', (event) => {

    console.log(event.key);

    switch (event.key) {

        case 'd':

            keys.d.pressed=true

            player.lastKey='d'

        break;

         case 'a':

            keys.a.pressed=true

            player.lastKey='a'

         break;

         case 'w':

            keys.w.pressed=true

            player.velocity.y = -20

            break;

        case 'ArrowRight':

            keys.ArrowRight.pressed=true

            enemy.lastKey='ArrowRight'

        break;

            case 'ArrowLeft':

            keys.ArrowLeft.pressed=true

            enemy.lastKey='ArrowLeft'

         break;

            case 'ArrowUp':

            keys.ArrowUp.pressed=true

            enemy.velocity.y = -20

            break;

    }

    console.log(event.key);

});

window.addEventListener('keyup', (event) => {

    switch (event.key) {

        case 'd':

            keys.d.pressed=false

        break;

        case 'a':

            keys.a.pressed=false

            break;

        case 'w':

            keys.w.pressed=false

            break;

        case 'ArrowRight':

            keys.ArrowRight.pressed=false

        break;

        case 'ArrowLeft':

            keys.ArrowLeft.pressed=false

            break;

        case 'ArrowUp':

            keys.ArrowUp.pressed=false

            break;

    }

    console.log(event.key);

});

Index.html:

<canvas></canvas>

<script src="index.js"></script>