THE BURNOUT

TEAM

AIDA

4idaJV

FREDDY ESCALADA

tajamajaka

ÁLVARO EG

AEG09

THE BURNOUT

INTRODUCTION

THE GAME IS INSPIRED BY FLAPPY BIRD.

A FUN GAME, BUT A TRUE NIGHTMARE DURING ITS PROGRAMMING. THAT'S WHY WE ARE PRESENTING OUR VERSION UNDER THE NAME 'THE BURNOUT'.

IN OUR VERSION, THE CHICKEN IS PLACED IN A MORE CHALLENGING, ROUGH, INTENSE, AND DANGEROUS ENVIRONMENT, WHERE ITS SITUATION COULD LEAD IT TO GET BURNED.

THE RULES OF THE GAME

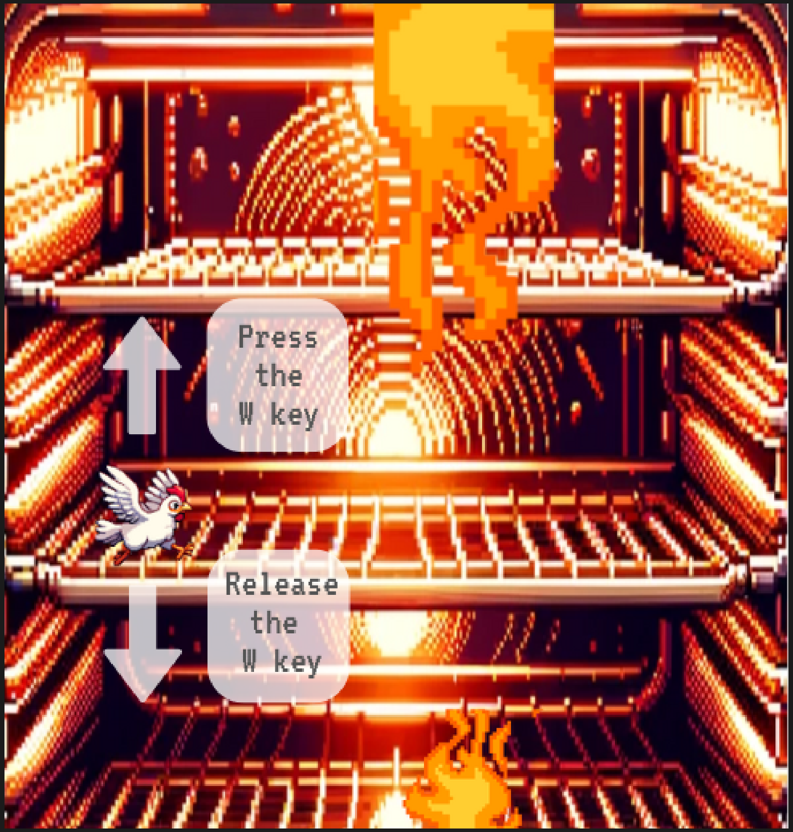
1.FLY BETWEEN THE FLAMES WITHOUT CRASHING OR YOU WILL DIE.

2.FOR EVERY FLAME YOU PASS, YOU WILL EARN ONE POINT.

3.IF YOU DIE, YOU'LL TURN INTO A ROASTED CHICKEN.

OBJECTIVE

EXIT THE OVEN ALIVE

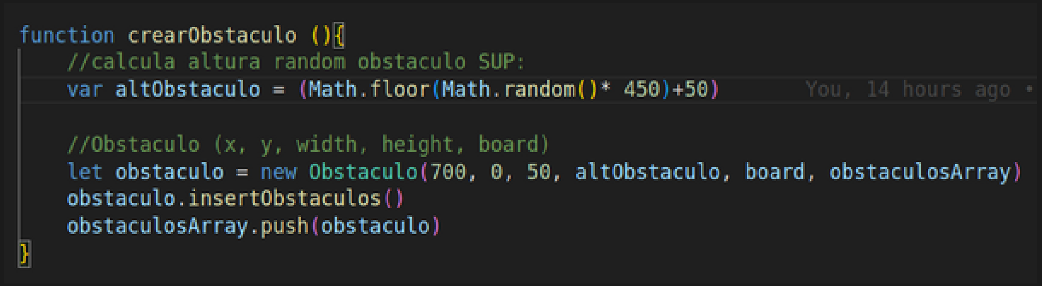


DEMO

THE BURNOUT

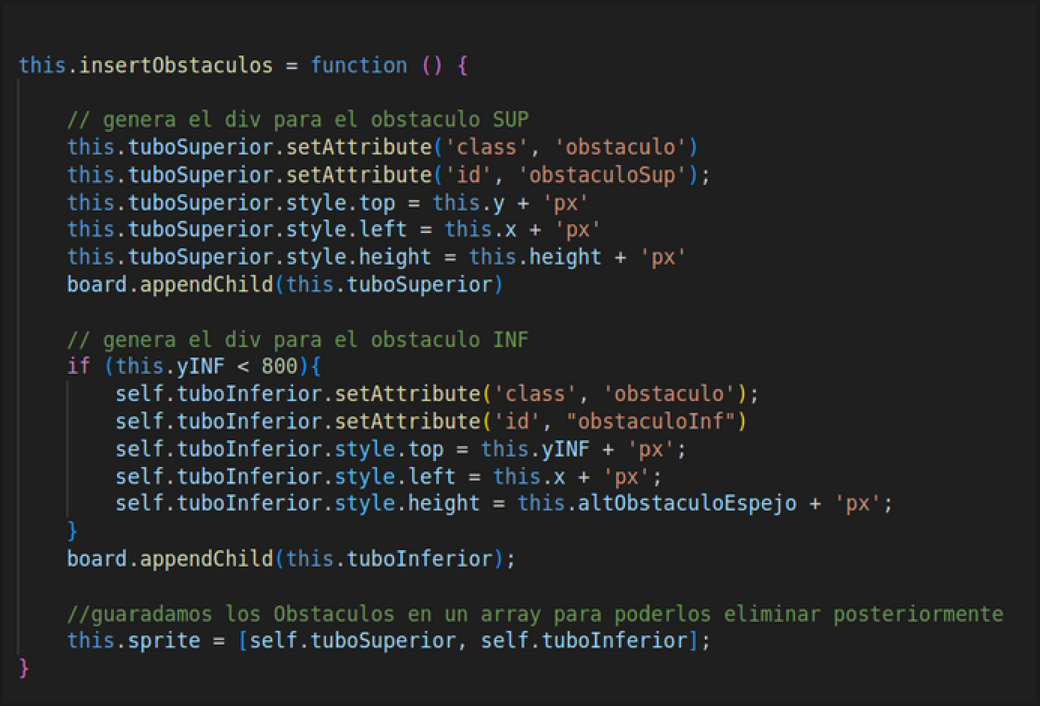
THE CODE

CREATION OF OBSTACLES



```
function crearObstaculo (){
    //calcula altura random obstaculo SUP:
    var altObstaculo = (Math.floor(Math.random()* 450)+50)        You, 14 hours ago •

    //Obstaculo (x, y, width, height, board)
    let obstaculo = new Obstaculo(700, 0, 50, altObstaculo, board, obstaculosArray)
    obstaculo.insertObstaculos()
    obstaculosArray.push(obstaculo)
}
```

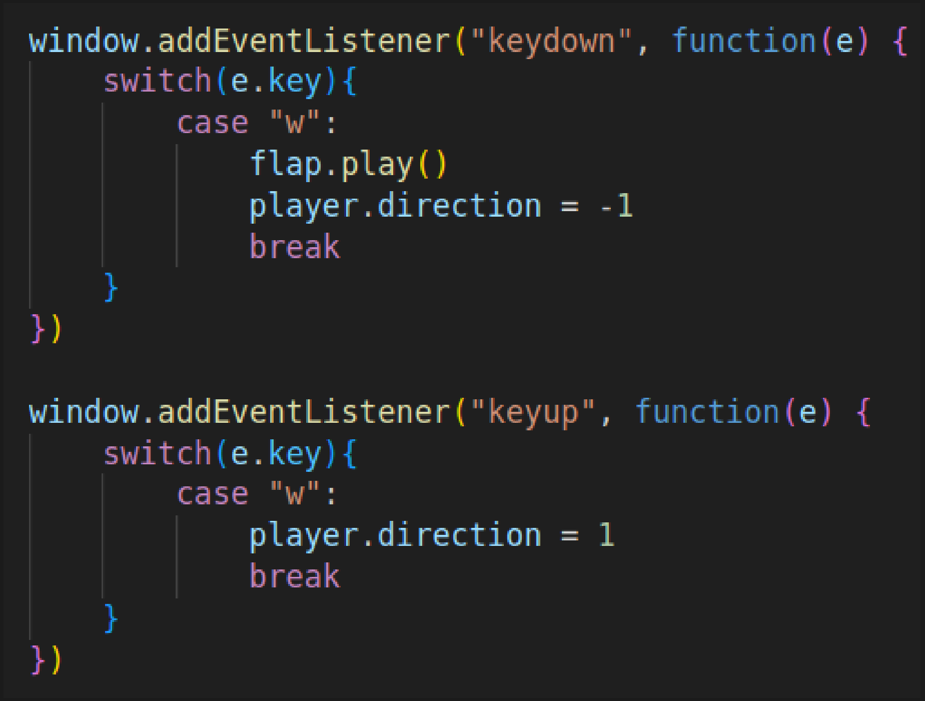
INSERTION OF OBSTACLES INTO THE GAME



```
this.insertObstaculos = function () {

    // genera el div para el obstaculo SUP
    this.tuboSuperior.setAttribute('class', 'obstaculo')
    this.tuboSuperior.setAttribute('id', 'obstaculoSup');
    this.tuboSuperior.style.top = this.y + 'px'
    this.tuboSuperior.style.left = this.x + 'px'
    this.tuboSuperior.style.height = this.height + 'px'
    board.appendChild(this.tuboSuperior)

    // genera el div para el obstaculo INF
    if (this.yINF < 800){
        self.tuboInferior.setAttribute('class', 'obstaculo');
        self.tuboInferior.setAttribute('id', "obstaculoInf")
        self.tuboInferior.style.top = this.yINF + 'px';
        self.tuboInferior.style.left = this.x + 'px';
        self.tuboInferior.style.height = this.altObstaculoEspejo + 'px';
    }
    board.appendChild(this.tuboInferior);

    //guaradamos los Obstaculos en un array para poderlos eliminar posteriormente
    this.sprite = [self.tuboSuperior, self.tuboInferior];
}
```

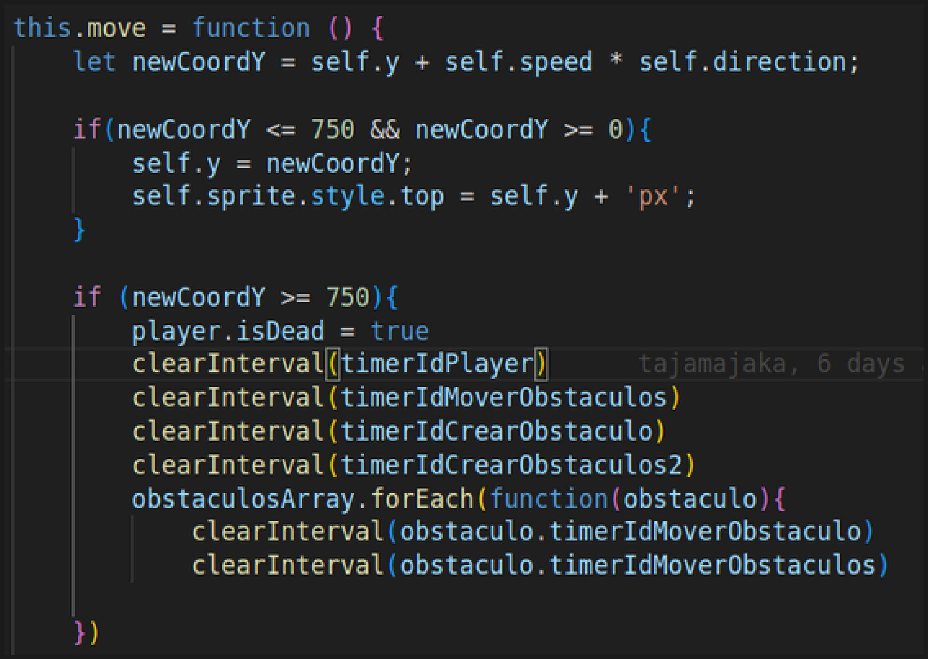
THE CODE

PLAYER'S MOVEMENT BUTTONS



```
window.addEventListener("keydown", function(e) {
    switch(e.key){
        case "w":
            flap.play()
            player.direction = -1
            break
    }
})

window.addEventListener("keyup", function(e) {
    switch(e.key){
        case "w":
            player.direction = 1
            break
    }
})
```

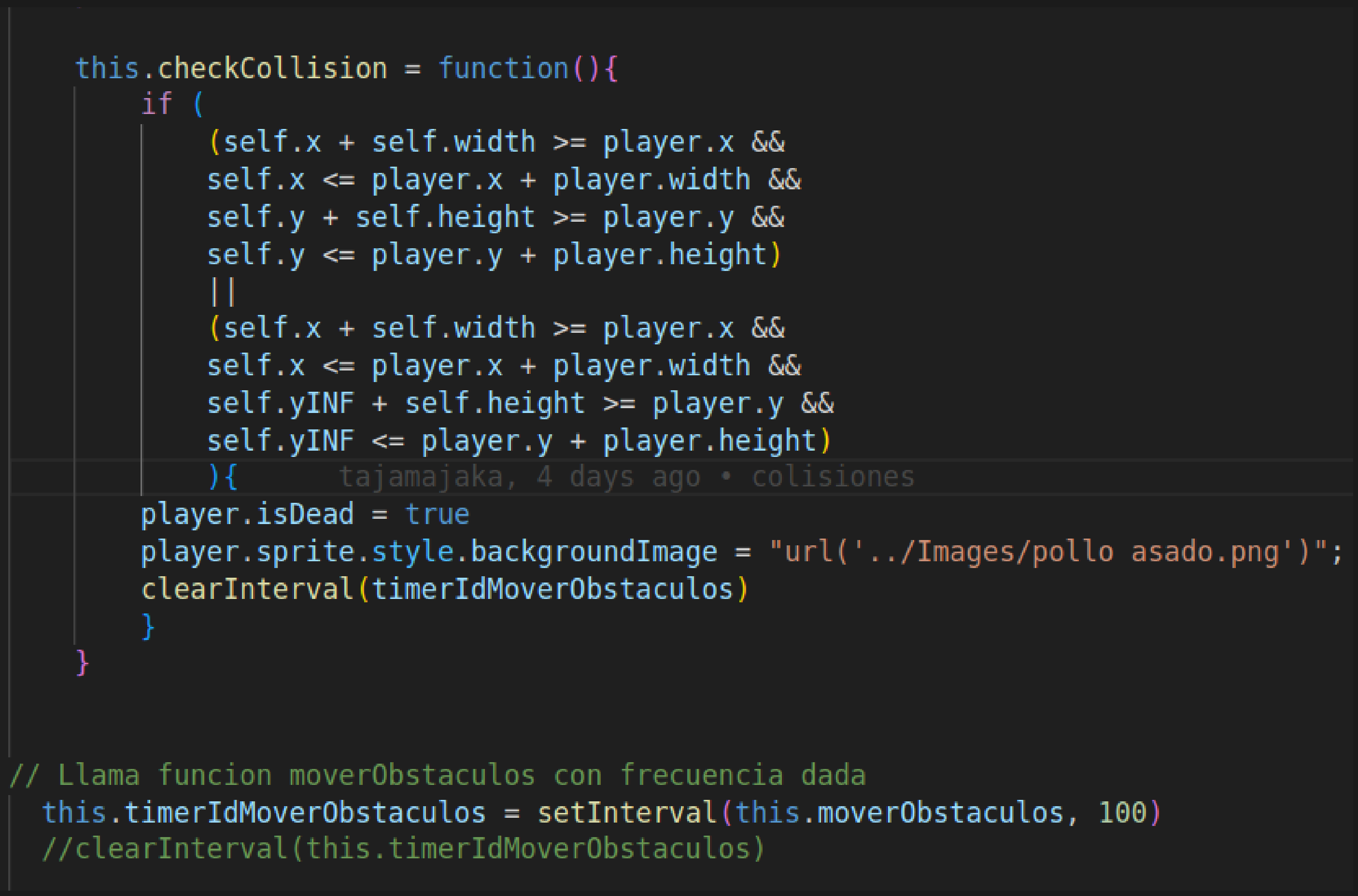
PLAYER MOVEMENT



```
this.move = function () {
    let newCoordY = self.y + self.speed * self.direction;

    if(newCoordY <= 750 && newCoordY >= 0){
        self.y = newCoordY;
        self.sprite.style.top = self.y + 'px';
    }

    if (newCoordY >= 750){
        player.isDead = true
        clearInterval(timerIdPlayer)      tajamajaka, 6 days
        clearInterval(timerIdMoverObstaculos)
        clearInterval(timerIdCrearObstaculo)
        clearInterval(timerIdCrearObstaculos2)
        obstaculosArray.forEach(function(obstaculo){
            clearInterval(obstaculo.timerIdMoverObstaculo)
            clearInterval(obstaculo.timerIdMoverObstaculos)

    })
```

THE CODE

COLLISIONS



```
    this.checkCollision = function(){
        if (
            (self.x + self.width >= player.x &&
            self.x <= player.x + player.width &&
            self.y + self.height >= player.y &&
            self.y <= player.y + player.height)
            ||
            (self.x + self.width >= player.x &&
            self.x <= player.x + player.width &&
            self.yINF + self.height >= player.y &&
            self.yINF <= player.y + player.height)
            ){      tajamajaka, 4 days ago • colisiones
        player.isDead = true
        player.sprite.style.backgroundImage = "url('../Images/pollo asado.png')";
        clearInterval(timerIdMoverObstaculos)
        }
    }


// Llama funcion moverObstaculos con frecuencia dada
  this.timerIdMoverObstaculos = setInterval(this.moverObstaculos, 100)
  //clearInterval(this.timerIdMoverObstaculos)
```

DIFFICULTIES

- GENERATING MIRRORED OBSTACLES RECALCULATED FROM THE ORIGINAL OBSTACLE.

- COLLISION CONTROL (INVOLVING TWO OBJECTS + THE PLAYER).

- MANAGING TIMERS THAT PERSIST AFTER THE GAME ENDS AND 'LOAD' MEMORY.

- GIT - BRANCH // GIT - MERGE.

FUTURE IMPLEMENTATIONS

- END GAME SCREEN WITH SCORE AND HIGHEST RECORD.
- CHARACTER SELECTION.
- VARYING THE DIRECTIONS IN WHICH OBSTACLES CAN COME.
- SCORE-BOOSTING ITEMS.

THANK YOU SO MUCH