

Este es un sencillo juego de navegador hecho con [Phaser 3](#) una librería de javascript que permite crear juegos

Introducción a phaser 3

Primero tenemos que agregar los scripts de phaser 3 en index.html

```
<script src="phaser.js"></script>
<script src="phaser.min.js"></script>
```

Y un archivo Main.js

```
<script src="main.js"></script>
```

Dentro de El archive Main.js hay que crear una variable config y tres funciones Preload() Create() y Upload()

```
// la variable game nos permite crear el canvas que contendrá el juego//
var game = new Phaser.Game(config);

var config = {
  type: Phaser.AUTO,
  width: 800,
  height: 600,
  scene: {
    preload: preload,
    create: create,
    update: update
  }
};

function preload (){}

function create (){}

function update (){}

la variable config contiene todas las configuraciones del CANVAS en el que se Cargara el juego
```

```
var config = {
// Tipo de cava que se cargara//
  type: Phaser.AUTO,
  // crear un contenedor//
  parent: 'container',
  // ancho del canvas
  width: 800,
  // altura del canvas//
  height: 600,
  // ajustar canvas al centro //
  mode: Phaser.Scale.FIT,
  autoCenter: Phaser.Scale.CENTER_BOTH,
```

```
// Gestor de escenas//
scene: {
preload: preload,
create: create,
update: update },
// Fisicas Arcade Para juegos de plataformas //
physics:{
default:'arcade',
arcade:
},,};
```

La Función Preload() precargará las imágenes videos y audios que utilizaremos para el juego pero no los mostrara solo precargara para que estén listos en el momento en que queramos utilizarlos para ello solo tenemos que usar This.load

```
this.load.image('Nombre del Objeto', 'Url de la imagen o Audio');
```

```
function preload(){
// -----pelota de el juego -----//
this.load.image('ball', 'assets/Ball.png');
//-----imagen de fondo -----//
this.load.image('background', 'assets/img/lofi-1.jpg');
// -----plataformas del juego -----//
this.load.image('plataform', 'assets/plataform.png');
//----- Musica del menú pricipal -----//
this.load.audio('menuMusic', 'assets/audio/menu-music.mp3');
}
```

La función Create() es la más importante porque Crea los objetos del juego. Para crearlos solo tenemos que usar una variable que dentro tenga this.physics.add.sprite y especificando Su posion en los eje X y Y luego llamarlo por su nombre

```
player = this.physics.add.sprite( Posicion en el eje X, Posición en el eje
Y, 'Nombre del obleto')
```

Pantalla de inicio

las Variables que usaremos en el juego

```
// Contrles del teclado//
var cursors;
//pelota del JUego//
var player;
// puntuación del juego//
```

```
var score = 0;  
// plataformas //  
var plataformas;  
array que contiene la música del juego
```

```
//----- Music Game -----//  
var songs = [  
  new Audio("assets/audio/menu-music.mp3"),  
  new Audio("assets/audio/level-1.mp3"),  
  new Audio("assets/audio/level-2.mp3"),  
  new Audio("assets/audio/level-3.mp3"),  
  new Audio("assets/audio/level-4.mp3"),  
  new Audio("assets/audio/level-5.mp3"),  
  new Audio("assets/audio/level-6.mp3"),  
  new Audio("assets/audio/level-7.mp3"),  
  new Audio("assets/audio/level-8.mp3"),  
  new Audio("assets/audio/level-9.mp3"),  
  new Audio("assets/audio/level-10.mp3"),  
  new Audio("assets/audio/level-11.mp3"),  
  new Audio("assets/audio/level-12.mp3"),  
];
```

Música del menú principal

```
//----- menu Music -----//  
menuMusic = this.sound.add('menuMusic',{loop:true}); menuMusic.play();  
Imagen de fondo
```

```
this.add.image(400, 300, 'background').setDisplaySize(800, 600);
```

letras del título de la pantalla



```
bounce = this.add.text(290, 80, 'Bounce',{ fontSize:'60px', fill:'#e601c0',fontFamily: 'verdana, arial, sans-serif'});
bounce.setStroke('#000', 10);
pop = this.add.text(180, 100, 'pop ',{ fontSize:'15rem', fill:'#07bccc',fontFamily: ' Meta, sans-serif'});
popStroke = this.add.text(180, 100, 'pop ',{ fontSize:'15rem', fill:'transparent',fontFamily: ' "Meta", sans-serif'});
popStroke.setStroke('#e601c0', 8);
pop.setShadow(-10, 20, '#482896', 2, true, true);
```

Botón de inicio



estilos de el botón de texto

```
FullScreen=this.add.text(100, 400, ' Click To Start Game ',{ fontSize:'60px',fill:'#FFF',fontFamily: 'verdana, arial, sans-serif'})
.setStroke('#000', 10).setInteractive();
```

Con .setInteractive(); puedes convertir cualquier objeto de el canvas en un botón

```
FullScreen=this.add.text(100, 400, ' Click To Start Game ').setInteractive()
```

Con `.on('pointerdown')` podemos especificar que acción queremos que ocurra al hacer clic sobre el botón

```
Boton = this.add.text(100, 400, ' Click To Start Game ').setInteractive()  
Boton.on('pointerdown', () => aquí la Acción que queremos que se realice)
```

Manejo de escenas en Phaser 3

Las escenas en phaser 3 son las diferentes secciones en que se divide el juego, en el caso de este juego hay tres escenas.

PANTALLA DE TITULO



LEVEL SELECT

1

2

3

4

5

6

7

8

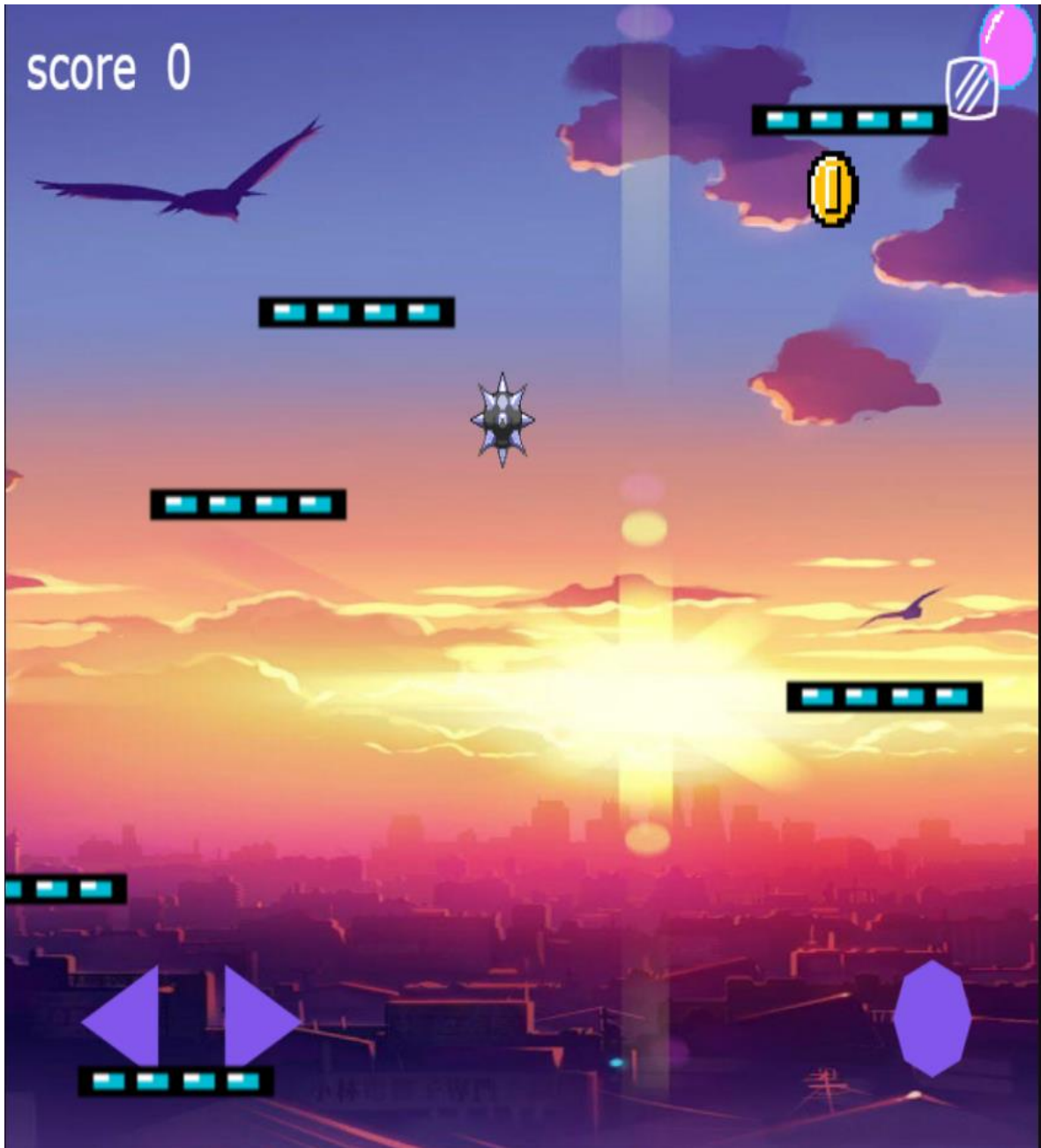
9

10

11

12

Y LA PANTALLA DE EL JUEGO



Para crear nuevas escenas en phaser a  ades nuevos scripts en el index.HTML

```
<script src="sceneA.js"></script>
<script src="sceneB.js"></script>
```

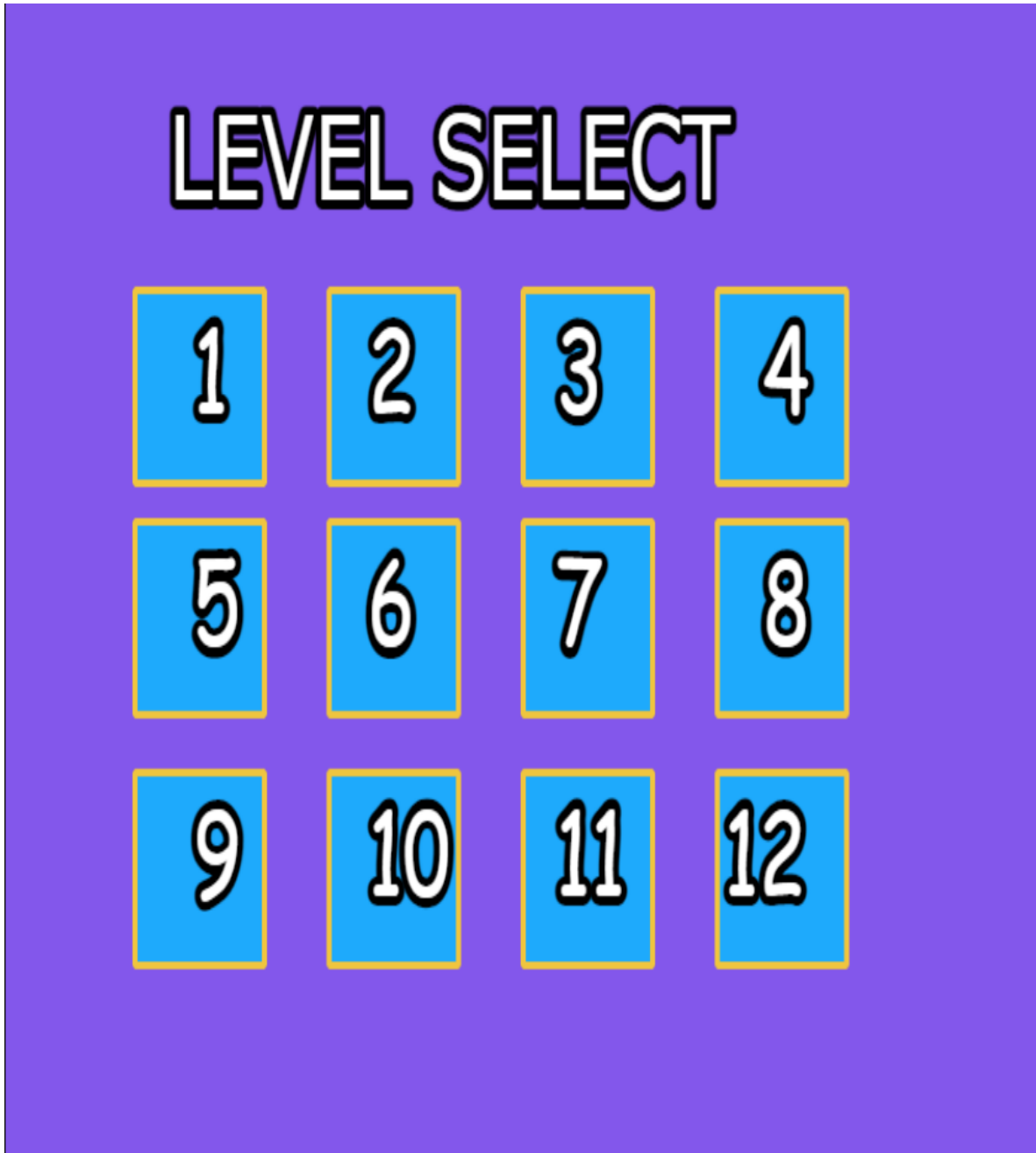
Nuevos a  ades

```
class SceneA extends Phaser.Scene { constructor() {
  super({key: "SceneA", active: true});}
```


al hacer clic en el Botón de inicio se iniciará la escena A

```
Startgame =this.add.text(100, 400, ' Click To Start Game ').setInteractive()  
Statgame.on('pointerdown', () => this.scene.add("SceneA", new SceneA));
```

La escena A será la escena de selección de niveles



background de la escena A

```
//----- Background -----//
```

```
this.add.rectangle(400, 300, 800, 600, 0x8357eb);
```

para crear cuadrados usamos this.add.rectangle

```
//----- Background -----//
```

```
this.add.rectangle(ejeX, ejeY, ancho, alto, color);  
crear cuadrados
```

```
var r1 = this.add.rectangle(200, 200, 100, 100, 0x1eaafc)  
//color y grosor del borde//  
.setStrokeStyle(4, 0xefc53f)  
// convertir cuadrado en boton//  
.setInteractive();
```

```
// cambiar posición de el cuadrado//  
r1.setPosition(150,200)  
  
// iniciar la escena B al hacer clic en el cuadro//  
r1.on('pointerdown', () => this.scene.add("SceneB", new SceneB))  
// al hacer clic en el cuadro carga una el backgroud del nivel/  
r1.on('pointerdown', () => this.add.image(400, 300, 'img1')  
.setDisplaySize(800, 600));
```

Añadimos números a los niveles

```
this.add.text(140,150, '1',  
// estilos de los numeros //  
{ fontSize:'60px',  
fill:'#FFF',  
fontFamily: 'Cursive'})  
//Borde de los numeros//  
.setStroke('#000',10);
```

Musica del juego

array que contiene la música del juego

```
//----- Music Game -----//  
var songs = [  
  new Audio("assets/audio/menu-music.mp3"),  
  new Audio("assets/audio/level-1.mp3"),  
  new Audio("assets/audio/level-2.mp3"),  
  new Audio("assets/audio/level-3.mp3"),  
  new Audio("assets/audio/level-4.mp3"),  
  new Audio("assets/audio/level-5.mp3"),  
  new Audio("assets/audio/level-6.mp3"),  
  new Audio("assets/audio/level-7.mp3"),  
  new Audio("assets/audio/level-8.mp3"),  
  new Audio("assets/audio/level-9.mp3"),  
  new Audio("assets/audio/level-10.mp3"),
```

```

    new Audio("assets/audio/level-11.mp3"),
    new Audio("assets/audio/level-12.mp3"),
];
// Borrar los elementos del array//
r1.on('pointerdown', () => songs.splice(0,1))
// play a la Música del nivel//
r1.on('pointerdown', () => songs[0].play())
//pausa la música del menú //
r1.on('pointerdown', () => menuMusic.destroy())
//reproduce la Música en bucle
r1.on('pointerdown', () => songs[0].loop= true);

```

Escena principal del juego

Variables globales

```

//Controles del teclado//
var cursors;
// Pelota del juego//
var player;
// puntuacion//
var score = 0;
// plataformas//
var plataforms;

```

La ESCENA B ES LA ESCENA DEL JUEGO

```

class SceneB extends Phaser.Scene { constructor() {
super({key: "SceneB", active: true});}

```

preloads

```

preload(){
//----- pelota -----//
this.load.image('ball', 'assets/Ball.png');
//----- plataformas -----//
this.load.image('plataform', 'assets/plataform.png');
//----- PUAS -----//
this.load.image('spike', 'assets/spike.png');
//----- MONEDAS -----//
this.load.image('coin', 'assets/coin.png');
//----- EFECTOS DE SONIDO DE LAS MONEDAS -----//
this.load.audio('soundCoin', 'assets/audio/coin.wav');

```

FUNCION CREATE

FISICAS DE LA PELOTA

```
//----- FISCAS DE LA BOLA -----//  
// CREAR LA PELOTA //  
player = this.physics.add.sprite(Phaser.Math.Between(0,600), 0, 'ball');  
// Has que la Bola APAREZCA en un punto al AZAR//  
Phaser.Math.Between(0,600  
// la bola colisiona con los bordes del escenario//  
player.body.setCollideWorldBounds(true);  
// tamaño de la bola//  
player.setDisplaySize(50,50);
```

crear plataformas

antes de crear las plataformas primero hay que crear un timeEvent de un evento de tiempo que se ejecuta en bucle y llama a la función CreatePlataforms() CADA 1000 Milisegundos

```
timedEvent = this.time.addEvent({ delay: 1000, callback: createPlataforms, c  
allbackScope: this, loop: true })
```

creamos una función que genera las plataformas

```
function createPlataforms(){  
// CRAR plataformas//  
plataforms = this.physics.add.sprite(Phaser.Math.Between(0,800), 0, 'platafo  
rm');  
// que la plataformas sea inmóvil //  
plataforms.body.immovable = true;  
// tamaño de la plataforma//  
plataforms.setDisplaySize(200, 200);  
// Velocidad con la que Bola cae //  
plataforms.setVelocityY(100)  
// Tamaño de la caja de colision//  
plataforms.setSize(80,8);  
// que la bola colisiona con la plataforma//  
this.physics.add.collider(player, plataforms, Impact, null, this);  
// colisión de abajo falsa //  
plataforms.body.checkCollision.down = false;  
// colisión de de la izquierda falsa //  
plataforms.body.checkCollision.left = false;  
// colisión de la deracha falsa //  
plataforms.body.checkCollision.right = false;
```

```
// Evita que la bola colisione con los bordes de la escena//
this.physics.world.setBoundsCollision(true,true,true, false);
```

CONTRLES

Para crear los controles usamos la variable cursors en la función CREATE()

```
cursors = this.input.keyboard.createCursorKeys();
para mover nuestro personaje utilizamos función UPLOAD()
```

```
// lafuncion UPLOAD mueve al personaje//
function update(){
// Cuando el persona este inmóvil tiene velocidad 0//
player.setVelocityY(0);
// moverse a la izquierda al presionar ← //
if (cursors.left.isDown){
player.setVelocityX(-250);}
// moverse a la izquierda al presionar →//
else if(cursors.right.isDown){
player.setVelocityX(250);}
// saltar al presionar la barra espaciadora//
if(cursors.space.isDown){
// empujar la bola hacia arriba//
player.setVelocityY(-350).setGravityY(0)
// empujar la bola hacia abajo//
setTimeout(function(){player.setGravityY(14000).setVelocityY(0)}, 400);}
}
```

CREAR BOTONES TACTILES

```
// Botón izquierdo//
LeftButton=this.add.text(16, 480, ' ◀ ',
// Estilos de botón //
{fontSize:'85px',
fill:'#8357eb',
fontFamily: 'verdana, arial, sans-serif'})
// que el botón sea táctil //
).setInteractive();
// Botón derecho //
RightButton=this.add.text(128, 480, ' ▶ ',
// Estilos//
{fontSize:'85px',
fill:'#8357eb',
fontFamily: 'verdana, arial, sans-serif'})
// Convertir texto en boton//
).setInteractive();
```

```
// botón de salto //
jumpButton=this.add.text(650, 480, ' ● ',
// estilos de el boton//
{
fontSize:'85px',
fill:'#8357eb',
fontFamily: 'verdana, arial, sans-serif'}
// convertir texto en botón //
).setInteractive();
```

Perder el juego al Caer al vacío

Antes de crear la función GAMEOVER() tenemos que crear una plataforma invisible para que cuando el personaje choque con la plataforma llame a la función gameOver()

```
// Crea una barra de Game over debajo el canvas //
this.gameOverBar = this.physics.add.sprite(400, 1200, 'plataform')
// Tamaño sobre caja de colisión//
.setSize(190,8)
// Tamaño de labarra de colision//
.setDisplaySize(1100,400);
// cuando el personaje colisione con la barra llama a la fucion GameOver//
this.physics.add.collider(this.gameOverBar, player, GameOver, null, this);
```

Función gameOVER()

Game Over



```
function GameOver(){  
  // Crea un rectángulo negro//  
  this.add.rectangle(400, 300, 800, 600, 0x111);  
  // Borra las plataformas//  
  timedEvent.destroy(), this.timeSpike.destroy()
```

```
// crea un texto de Game Over//
this.add.text(200, 200, 'Game Over',
// estilos del texto de game over//
{
fontSize:'60px',
fill:'#000',
fontFamily: '"Press Start 2P", Arial, sans-serif'}
// Ponle una sombra rosada //
).setShadow(5, 5, '#e601c0', 10, true, true);
```

Botón de Home

```
// crea un boton de HOME //
this.add.text(300, 300, '△',
// Estilos de el boton//
{
fontSize:'70px',
fill:'#07bccc',fontFamily: '"Press Start 2P", Arial, sans-serif'
})
//ponle una sombra rosada para el boton//
.setShadow(5, 5, '#e601c0', 5, true, true)
// convierte el Texto botón //
.setInteractive()
// al hacer clic en el botón reinicia la pagina//
.on('pointerdown',() => location.reload())
```

Botón de repeat

```
// Crea un botón de para repetir la escena//
this.repeatButton = this.add.text(450, 300, '↺',
// Estilo de boton//
{
fontSize:'60px',
fill:'#07bccc',
fontFamily: '"Press Start 2P", Arial, sans-serif'
})
//ponle una sombra rosada para el boton//
.setShadow(5, 5, '#e601c0', 5, true, true)
// convierte el Texto botón //
.setInteractive()
// Reiniciar escena B al hacer clic en el boton//
.on('pointerdown', () => this.scene.restart())
//Reiniciar Musica de el nivel//
.on('pointerdown', () => songs[0].currentTime = 0)
// dar play a la música de el nivel//
.on('pointerdown', () => songs[0].play())
```



```
//ponle pausa a la música Game over
.on('pointerdown', () =>GameOverSong.pause())
// reiniciar puntaje a 0//
.on('pointerdown', () => score = 0);
```

Música de game over

```
//ponle pausa al botón //
songs[0].pause();
// reinicia la música de game over//
GameOverSong.currentTime = 0;
// dar play a la música de game over //
GameOverSong.play();
```

Crear Bolas puntiagudas



```
// llamar a la función Spike cada 7500 milisegundos//
this.timeSpike= this.time.addEvent({ delay:7500, callback: spikes, callbackS
cope: this, loop: true });
// la spikes crea bolas puntiagudas//
function spikes() {
// crear púas en un punto al azar del eje X//
this.spike = this.physics.add.sprite(Phaser.Math.Between(0,800), 0, 'spike')
// velocidad a la que caen las púas//
.setVelocityY(100)
// tamaño de las púas//
.setDisplaySize(50,50);
// llama a la función GameOver() cuando la bola choque con las púas//
this.physics.add.collider(this.spike, player, GameOver, null, this);}
```

Puntuación

```
//-----Texto de puntuación -----//
this.scoreText = this.add.text(16, 16, 'score ' + score,
//----- Estilos de Texto -----//
{ fontSize:'32px',
```

```
fill:'#fff',  
fontFamily:'verdana, arial, sans-serif'  
});
```

Crear Monedas

```
//----- Crear Monedas -----//  
this.coin = this.physics.add.image(Phaser.Math.Between(0,800), 100, 'coin')  
.setDisplaySize(40, 40)  
// llama a la función getCoins Cuando la Bola colisione Con la moneda //  
this.physics.add.collider(this.coin, player, getCoins, null, this);  
// Conseguir monedas //  
function getCoins(){  
    // suma un punto Cuando toques la moneda//  
    this.scoreText.setText('score ' + score++)  
    // la moneda choca con los bordes //  
    this.coin.body.setCollideWorldBounds(true);  
    // efecto de sonido de monedas //  
    this.sound.add('soundCoin').play();  
    // evita que la moneda caiga al vacío//  
    this.coin.setVelocityY(-1)  
    // has que la moneda se aparezca en un punto al azar//  
    this.coin.setPosition(Phaser.Math.Between(0,800), Phaser.Math.Between(0,600)  
    , 'coin')}
```