//Chuletaca gorda

COMENZAR CON CONSOLA:

(Si no empiezas en el sitio correcto cd para moverte)

-ESTAR EN EL SITIO DE LA CARPETA Y npx http-server / npx live-server

this.scene.add.existing(this); //Para que se renderice (Sprites)

this.scene.physics.add.existing(this, true); (Si no tira el anterior)

//CARGAR UN TILEMAP

this.load.tilemapTiledJSON('tilemap\_level1', 'src/assets/tiles/level1.json');

this.load.image('tiles', './assets/sprites/tilesets/slates\_tileset.png');

(LevelBoss linea 34)

//CARGAR UNA IMAGEN

this.load.image('feather', 'src/assets/sprites/unamuno/feather.png');

//CARGAR UN AUDIO

this.load.audio('bandaSonora','src/assets/sonido/bandasonoracompr.mp3');

(Comienzo de sonido en gameScene.js linea 104-114)

//Timer

//Timer de phaser

var timer = scene.time.addEvent({

delay: this.time, // ms

callback: () => {

Tu puta función con ;

}

},

loop: true

});

Si le metes un repeat lo hace x veces

//PARA LAS ANIMACIONES HACER EL SUPER.PREUPDATE

//CREAR UNA ANIMACION

this.anims.create({

key: 'Nombre que quieres para la anim',

frames: this.anims.generateFrameNumbers('Nombre del SpriteSheet', { start: 0, end: x }),

frameRate: 8,

repeat: 0

});

//PLAYEAR UNA ANIMACION

this.anims.play('Boss\_Death',true);

//CARGAR UN SpriteSheet

this.load.spritesheet({

key: 'Nombre que quieres que tenga el spritesheet',

url: 'Dirección del motherfucking asset',

frameConfig: {

frameWidth: 120,

frameHeight: 120

}

});

//HACER COSAS CUANDO TERMINA LA ANIMACION

this.boss.on('animationcomplete', function (anim, frame) {

this.boss.emit('animationcomplete\_' + anim.key, anim, frame);

}, this);

//ASÍ ES PARA QUE SEA ANÓNIMA LA FUNCIÓN YO QUE SE

this.boss.on('animationcomplete-Boss\_attk1', () => {

this.boss.states.atacando = false;

this.boss.states.idle = true;

});

//ANIMACION REPETIDA

this.boss.on('animationrepeat', function (anim, frame) {

this.boss.emit('animationrepeat\_' + anim.key, anim, frame);

}, this);

//Partículas

let leaves = this.add.particles(particleSprite);

leaves.createEmitter({

frames: [{key: particleSprite, frame: 0}],

x: -400,

y: { min: 500, max: 1000},

speedX: { min: 100, max: 300 },

speedY: { min: -50, max: 50 },

lifespan: 20000,

scale: {start: 0.7, end: 0.1},

rotate: {start: 0, end: 360},

frequency: 600

});

}

//CARGAR UNA FUENTE

this.load.bitmapFont('dialogue\_font','src/assets/fonts/dialogue.png','src/assets/fonts/dialogue.xml');

//HACER ALGO CUANDO LA CÁMARA TERMINE DE HACER FADEOUT

this.cameras.main.once(Phaser.Cameras.Scene2D.Events.FADE\_OUT\_COMPLETE, () => {

this.scene.start('level1');

});

//AÑADIR TECLAS

this.keycodeC = this.scene.input.keyboard.addKey(Phaser.Input.Keyboard.KeyCodes.C);

this.keycodeV = this.scene.input.keyboard.addKey(Phaser.Input.Keyboard.KeyCodes.V);

this.keycodeF = this.scene.input.keyboard.addKey(Phaser.Input.Keyboard.KeyCodes.F);

//INPUT SOLO UNA VEZ

Phaser.Input.Keyboard.JustDown(this.keycodeV);

//Plantillas

https://github.com/cleongh/phasertemplate.git (Plantilla sin nada)

https://github.com/cleongh/simplephasergame.git (Plantilla sencillita de señor que ya se mueve)

// CREAR EVENTOS CUANDO UNA TECLA ES PULSADA

this.space.on('down',() => {

this.scene.start('scene');

//this.space.resetkey(); (ESO IGUAL SI IGUAL NO)

})

//COLISIONES SI QUIERES QUE HAGA ALGO ASÍ (En el create hijo de la grandísima)

this.scene.physics.add.collider(this.player, this, () => {

this.clonateBola(this.x, this.y);

});

lo mismo pero con overlap si no quieres que se muevan las movidas crack

//COLLISIONES ENTRE DOS COSAS SIN HACER NADA MAS

this.scene.physics.add.collider(this, player);

//CREAR GRUPOS FíSICOS

this.walls = this.physics.add.staticGroup(); //ASÍ SE CREA UN GRUPO ESTÁTICO

//UN RANDOM ENTRE DOS MIERDAS

Phaser.Math.RND.pick(from || this.bases.children.entries);

(Lo mismo pero sin autismo)

**Phaser.Math.Between(2000,4000)**

//HACER EL TILEMAP (DRIVE)

//COLISION CON UN LAYER

setCollisionByProperty(atributo que le añadiste en el tiled);

//FOR CHULAZO

for (const objeto of this.map.getObjectLayer('objectLayer').objects) {

// 'objeto.name' u 'objeto.type' nos llegan de las propiedades del

// objeto en Tiled

if (objeto.name === 'spawnPoint') {

this.spawnpoint = objeto;

let savedFaith;

if (this.info !== undefined && this.info.obtainedFaith !== undefined) savedFaith = this.info.obtainedFaith;

else savedFaith = 0;

this.player = new Player(this.matter.world, objeto.x, objeto.y, objeto, savedFaith, 2.2);;

}

}

 justInCase

//DELAYED CALL

this.time.delayedCall(2), () => {

this.scene.restart();

}

//INIT

Para inicializar parámetros cuando se pasa de otra escena

//CAMBIAR COSAS DE COLOR

<https://htmlcolorcodes.com/es/>

this.setTint('0x0c9ead');

//INPUT DE RATÓN

this.on('pointerover',() => {

console.log('Hola soy el chocu y estoy pasando por enicma');

});

this.on('pointerout',() => {

console.log('Hola soy el chocu y he dejado de estar encima');

});

this.on('pointerdown',() => {

console.log('Hola soy el chocu y he clickado');

});

this.on('pointerup',() => {

console.log('Hola soy el chocu y he dejado de clickar');

});