**Part 1**

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Making your first Phaser 3 Game - Part 1</title>

<script src="//cdn.jsdelivr.net/npm/phaser@3.11.0/dist/phaser.js"></script>

<style type="text/css">

body {

margin: 0;

}

</style>

</head>

<body>

<script type="text/javascript">

var config = {

type: Phaser.AUTO,

width: 800,

height: 600,

scene: {

preload,

create,

update

}

};

var game = new Phaser.Game(config);

function preload ()

{

}

function create ()

{

}

function update ()

{

}

</script>

</body>

</html>

**Part 2**

<script type="text/javascript">

var config = {

type: Phaser.AUTO,

width: 800,

height: 600,

scene: {

preload,

create,

update

}

};

var game = new Phaser.Game(config);

function preload ()

{

this.load.image('sky', 'assets/sky.png');

this.load.image('ground', 'assets/platform.png');

this.load.image('star', 'assets/star.png');

this.load.image('bomb', 'assets/bomb.png');

this.load.spritesheet('dude', 'assets/dude.png', { frameWidth: 32, frameHeight: 48 });

}

function create ()

{

this.add.image(400, 300, 'sky');

}

function update ()

{

}

</script>

**Part 3**

<script type="text/javascript">

var config = {

type: Phaser.AUTO,

width: 800,

height: 600,

physics: {

default: 'arcade',

arcade: {

gravity: { y: 300 },

debug: false

}

},

scene: {

preload,

create,

update

}

};

var platforms;

var game = new Phaser.Game(config);

function preload ()

{

this.load.image('sky', 'assets/sky.png');

this.load.image('ground', 'assets/platform.png');

this.load.image('star', 'assets/star.png');

this.load.image('bomb', 'assets/bomb.png');

this.load.spritesheet('dude', 'assets/dude.png', { frameWidth: 32, frameHeight: 48 });

}

function create ()

{

this.add.image(400, 300, 'sky');

platforms = this.physics.add.staticGroup();

platforms.create(400, 568, 'ground').setScale(2).refreshBody();

platforms.create(600, 400, 'ground');

platforms.create(50, 250, 'ground');

platforms.create(750, 220, 'ground');

}

function update ()

{

}

</script>

**Part 4**

<script type="text/javascript">

var config = { … };

var player;

var platforms;

var game = new Phaser.Game(config);

function preload () { … }

function create ()

{

this.add.image(400, 300, 'sky');

platforms = this.physics.add.staticGroup();

platforms.create(400, 568, 'ground').setScale(2).refreshBody();

platforms.create(600, 400, 'ground');

platforms.create(50, 250, 'ground');

platforms.create(750, 220, 'ground');

player = this.physics.add.sprite(100, 450, 'dude');

player.setBounce(0.2);

player.setCollideWorldBounds(true);

this.anims.create({

key: 'left',

frames: this.anims.generateFrameNumbers('dude', { start: 0, end: 3 }),

frameRate: 10,

repeat: -1

});

this.anims.create({

key: 'turn',

frames: [ { key: 'dude', frame: 4 } ],

frameRate: 20

});

this.anims.create({

key: 'right',

frames: this.anims.generateFrameNumbers('dude', { start: 5, end: 8 }),

frameRate: 10,

repeat: -1

});

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

}

function update () { }

</script>

**Part 5**

<script type="text/javascript">

var config = { … };

var player;

var platforms;

var cursors;

var game = new Phaser.Game(config);

function preload () { … }

function create ()

{

…

cursors = this.input.keyboard.createCursorKeys();

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

}

function update ()

{

if (cursors.left.isDown)

{

player.setVelocityX(-160);

player.anims.play('left', true);

}

else if (cursors.right.isDown)

{

player.setVelocityX(160);

player.anims.play('right', true);

}

else

{

player.setVelocityX(0);

player.anims.play('turn');

}

if (cursors.up.isDown && player.body.touching.down)

{

player.setVelocityY(-330);

}

}

</script>

**Part 6**

<script type="text/javascript">

var config = { … };

var player;

var stars;

var platforms;

var cursors;

var game = new Phaser.Game(config);

function preload () { … }

function create ()

{

…

stars = this.physics.add.group({

key: 'star',

repeat: 11,

setXY: { x: 12, y: 0, stepX: 70 }

});

stars.children.iterate(function (child) {

child.setBounceY(Phaser.Math.FloatBetween(0.4, 0.8));

});

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

this.physics.add.collider(stars, platforms);

this.physics.add.overlap(player, stars, collectStar, null, this);

}

function update () { … }

function collectStar (player, star)

{

star.disableBody(true, true);

}

</script>

**Part 7**

<script type="text/javascript">

var config = { … };

var player;

var stars;

var platforms;

var cursors;

var score = 0;

var scoreText;

var game = new Phaser.Game(config);

function preload () { … }

function create ()

{

…

scoreText = this.add.text(16, 16, 'score: 0', { fontSize: '32px', fill: '#000' });

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

this.physics.add.collider(stars, platforms);

this.physics.add.overlap(player, stars, collectStar, null, this);

}

function update () { … }

function collectStar (player, star)

{

star.disableBody(true, true);

score += 10;

scoreText.setText('Score: ' + score);

}

</script>

**Part 8**

<script type="text/javascript">

var config = { … };

var player;

var stars;

var bombs;

var platforms;

var cursors;

var score = 0;

var gameOver = false;

var scoreText;

var game = new Phaser.Game(config);

function preload () { … }

function create ()

{

…

bombs = this.physics.add.group();

scoreText = this.add.text(16, 16, 'score: 0', { fontSize: '32px', fill: '#000' });

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

this.physics.add.collider(stars, platforms);

this.physics.add.collider(bombs, platforms);

this.physics.add.overlap(player, stars, collectStar, null, this);

this.physics.add.collider(player, bombs, hitBomb, null, this);

}

function update () { … }

function collectStar (player, star)

{

star.disableBody(true, true);

score += 10;

scoreText.setText('Score: ' + score);

if (stars.countActive(true) === 0)

{

stars.children.iterate(function (child) {

child.enableBody(true, child.x, 0, true, true);

});

var x = (player.x < 400) ? Phaser.Math.Between(400, 800) : Phaser.Math.Between(0, 400);

var bomb = bombs.create(x, 16, 'bomb');

bomb.setBounce(1);

bomb.setCollideWorldBounds(true);

bomb.setVelocity(Phaser.Math.Between(-200, 200), 20);

bomb.allowGravity = false;

}

}

function hitBomb (player, bomb)

{

this.physics.pause();

player.setTint(0xff0000);

player.anims.play('turn');

gameOver = true;

}

</script>