



PHASER

Emanuel Alejandro Gutierrez Romero



JANUARY 29, 2024

PROGRAMACION PARA INTERNET
Prof. Michel Emanuel Lopez Franco

Introducción.

Para la creación de este juego se utilizaron las herramientas de phaser y Brackets el primero es el que permite crear todo el juego y en Brackets es donde lo desarrollamos.

Codigo.

Este sería el código para el juego que intenta ser un asteroids pero en el que solo se esquivan los asteroides.

```
1 var config = {
2   type: Phaser.AUTO,
3   width: 800,
4   height: 600,
5   physics: {
6     default: 'arcade',
7     arcade: {
8       gravity: { y: 0 },
9       enableBody: true,
10    }
11  },
12  scene: {
13    preload: preload,
14    create: create,
15    update: update
16  }
17 };
18
19 var player;
20 var asteroids;
21 var score = 0;
22 var scoreText;
23 var timer;
24
25 var game = new Phaser.Game(config);
26
27 function preload() {
28   this.load.image('ship', 'images/ship.png');
29   this.load.image('asteroid', 'images/asteroid.png');
30 }
31
32 function create() {
33   player = this.physics.add.image(400, 300, 'ship');
34   player.setCollideWorldBounds(true);
35   player.setScale(0.03);
36
37   asteroids = this.physics.add.group({
38     key: 'asteroid',
39     repeat: 10,
40     setXY: { x: 12, y: 0, stepX: 70 }
41   });
42
43   asteroids.children.iterate(function (asteroid) {
44     asteroid.setBounceY(Phaser.Math.FloatBetween(0.4, 0.8));
45     asteroid.setScale(0.01);
46   });
47
48   this.physics.add.collider(asteroids, asteroids);
49   this.physics.add.collider(player, asteroids, gameOver, null, this);
50
51   scoreText = this.add.text(16, 16, 'Score: 0', { fontSize: '32px', fill: '#fff' });
52 }
```

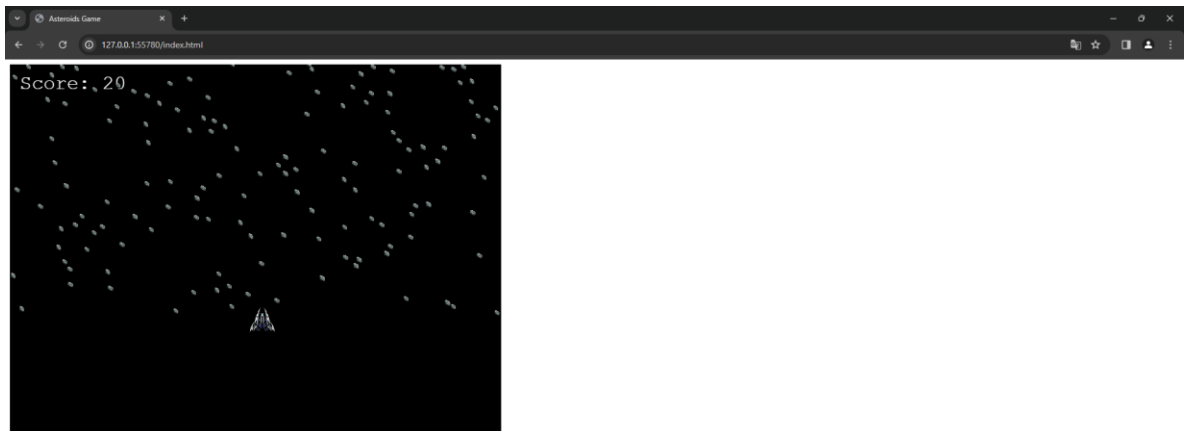
```

53 ▼ timer = this.time.addEvent({
54     delay: 1000, // 1000 ms = 1 segundo
55     loop: true,
56 ▼     callback: function () {
57         score += 10;
58     }
59 });
60
61 ▼ this.input.on('pointermove', function (pointer) {
62     player.x = pointer.x;
63     player.y = pointer.y;
64 });
65 }
66
67 ▼ function update() {
68 ▼     if (player.active) {
69 ▼         if (score % 10 === 0) {
70             spawnAsteroid();
71         }
72
73         scoreText.setText('Score: ' + score);
74
75 ▼         asteroids.children.iterate(function (asteroid) {
76 ▼             if (asteroid.y > config.height) {
77                 asteroid.y = 0;
78                 asteroid.x = Phaser.Math.Between(0, config.width);
79                 score += 1;
80             }
81         });
82
83 ▼         this.input.on('pointermove', function (pointer) {
84             player.x = pointer.x;
85             player.y = pointer.y;
86         });
87     }
88 }
89
90 ▼ function spawnAsteroid() {
91     var asteroid = asteroids.create(Phaser.Math.Between(0, config.width), 0, 'asteroid');
92     asteroid.setBounce(1);
93     asteroid.setCollideWorldBounds(true);
94     asteroid.setVelocity(Phaser.Math.Between(-200, 200), 200);
95     asteroid.setScale(0.01);
96 }
97
98 ▼ function gameOver() {
99     player.setActive(false).setVisible(false);
100     asteroids.clear(true, true);
101     this.physics.pause();
102     alert('Game Over! Your score: ' + score);
103     resetGame();
104 }
105
106 ▼ function resetGame() {
107     player.setActive(true).setVisible(true);
108     score = 0;
109     timer.reset({ delay: 1000, loop: true }); |
110     this.physics.resume();
111 }

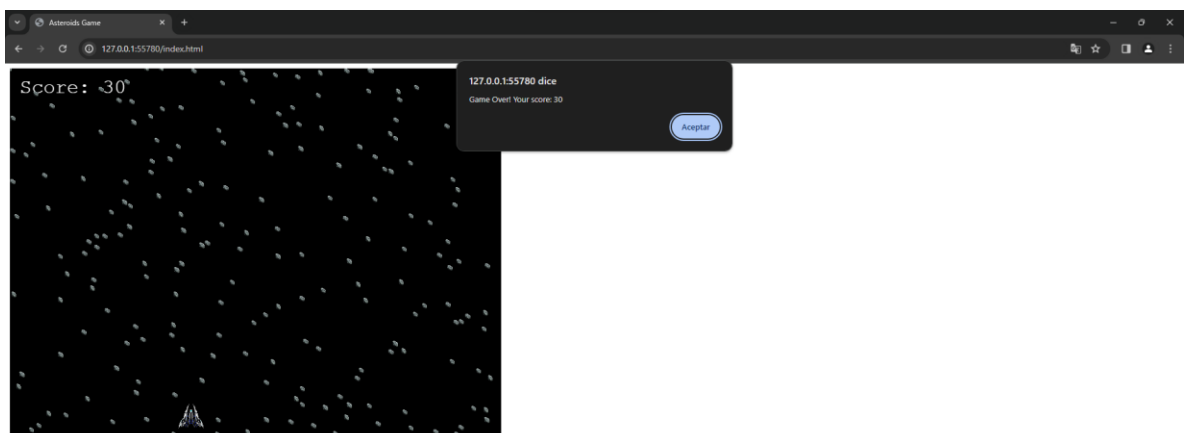
```

Tenemos las diferentes variables que se usan para el funcionamiento del juego una para el jugador otra para los asteroides, unas mas para el marcador y una última para el temporizador con el cual se relaciona con el score ya que cada segundo que pasa se registran 10 puntos al marcador, seguido a ello tenemos múltiples funciones que controlan el movimiento de la nave, el spawn de los asteroides, las físicas para que se termine el juego etc.

Ejecución del código.



Así se ve el juego en ejecución 2 segundos después de que iniciara



Así se mira cuando un Asteroide choca con la nave podemos ver que da un game over con la puntuación adquirida.

Para reiniciar solo se da refresh a la pagina y podemos volver a jugar.

Conclusión.

Me pareció una actividad bastante divertida aprendí mucho sobre el desarrollo en Brackets y me pareció increíble el resultado del juego, comprendiendo que le fallan o faltan cosas aun así me gusto el resultado obtenido espero en un futuro continuar aprendiendo sobre Brackets y como usarlo de mejor manera.