

UNIVERSIDAD DE GUADALAJARA  
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS  
DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN  
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES



**ACTIVIDAD 3**

JavaScript – Phaser 3

Landeros Morales Jairo Mauricio



PROGRAMACIÓN PARA INTERNET

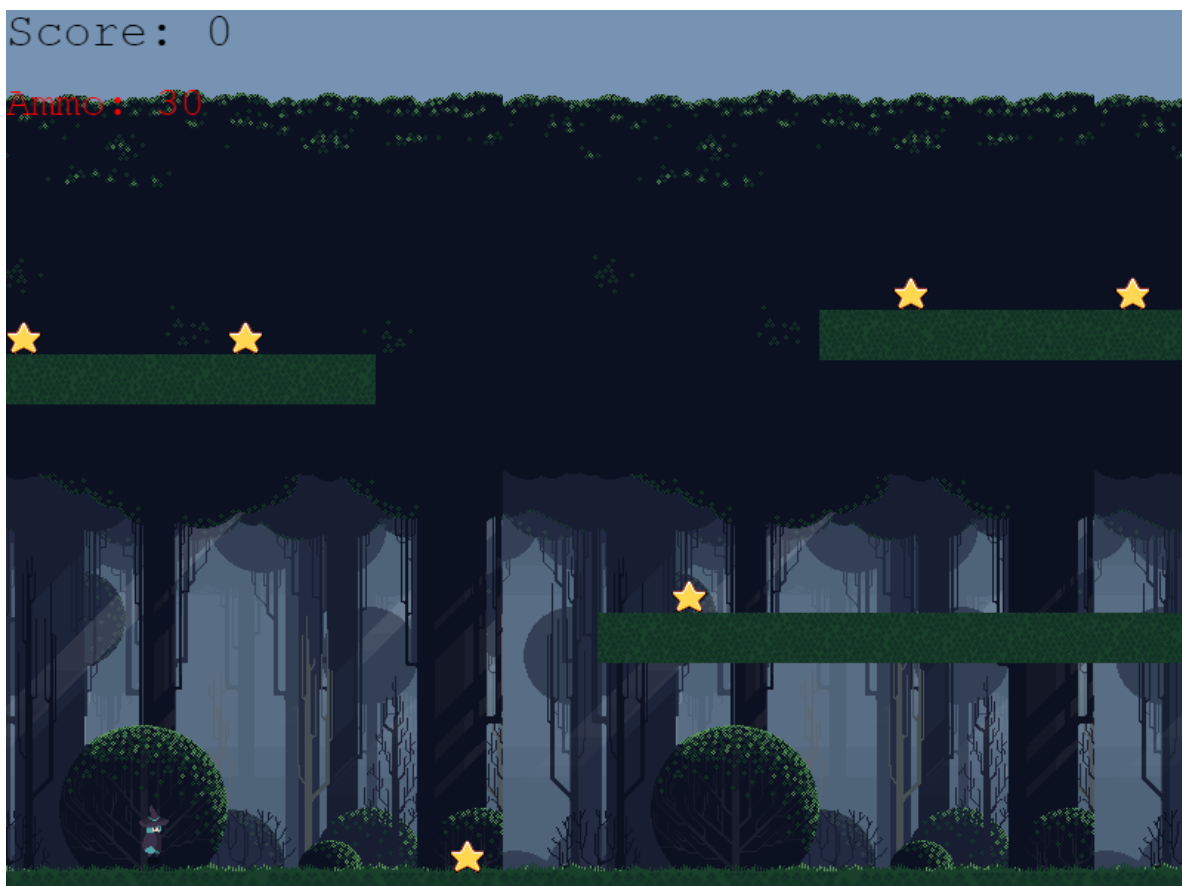
D03          2022A

17 de agosto del 2021

## JavaScript y Phaser 3

Phaser 3 es un framework utilizado para desarrollar aplicaciones interactivas de gráficos en tiempo real para navegadores web. Es compatible con la mayoría de los navegadores existentes y es integrable con cualquier proyecto web, ya que solo proporciona un elemento canvas con una vista al view port de la aplicación. Phaser 3 utilizar WebGL como renderizador. El objetivo de esta práctica es el de desarrollar un minijuego tomando como base el ejemplo introductorio disponible en la página oficial de Phaser.

En la siguiente imagen podemos visualizar todos los componentes que forman parte de la escena del juego. Podemos ver un HUD en la esquina superior izquierda donde se nos indica la cantidad de estrellas recolectadas y la cantidad de munición disponible. Podemos observar un fondo, plataformas y al personaje controlable.



Iniciaremos explicando como se estructuró el programa completo. La escena completa está definida como una clase que es una extensión de la escena de Phaser.

```
class Game extends Phaser.Scene
```

Dentro de esta clase encontramos los métodos de preload, create y update, además de los controles del juego. Los controles son los siguientes:

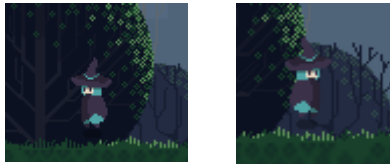
- ↑ para saltar.
- → para correr a la derecha.

- ← para correr a la izquierda.
- SPACE y/o Botón izquierdo del mouse para disparar hechizos.

El movimiento del personaje jugable se implementó fácilmente ya que Phaser cuenta con formas directas de detectar entradas de las flechas del teclado.

```
if(this.cursors.left.isDown)
{
    this.player.flipX = true;
    this.player.setVelocityX(-160);
    this.player.anims.play('left', true);
}
```

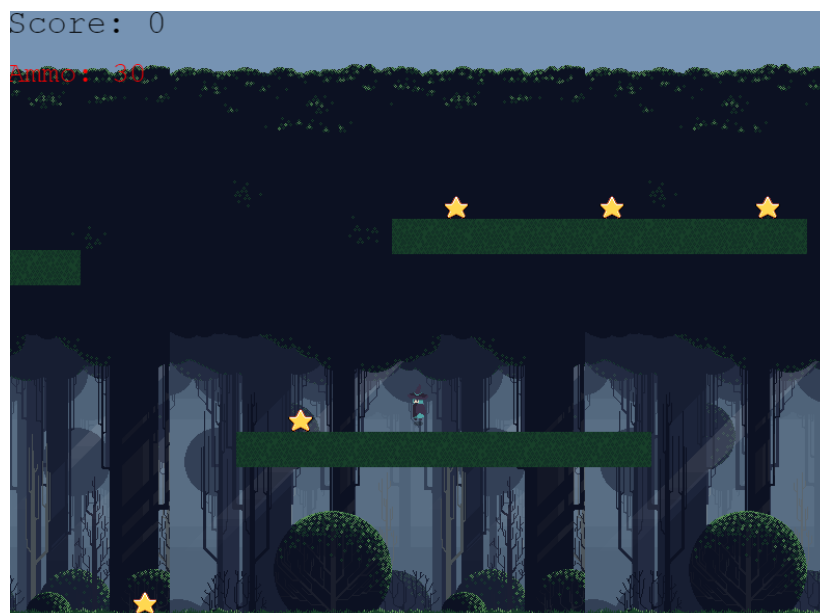
Esta parte de código verifica si la ← está siendo presionada. Si es así podemos ver que se añade velocidad al Sprite del personaje en dirección izquierda y se ejecuta la animación correspondiente a ese movimiento. Podemos notar también el método flipX, este permite espejear el Sprite para así poder reciclar la animación de correr a la izquierda y a la derecha ya que las animaciones proporcionadas solo se presentan en una dirección.



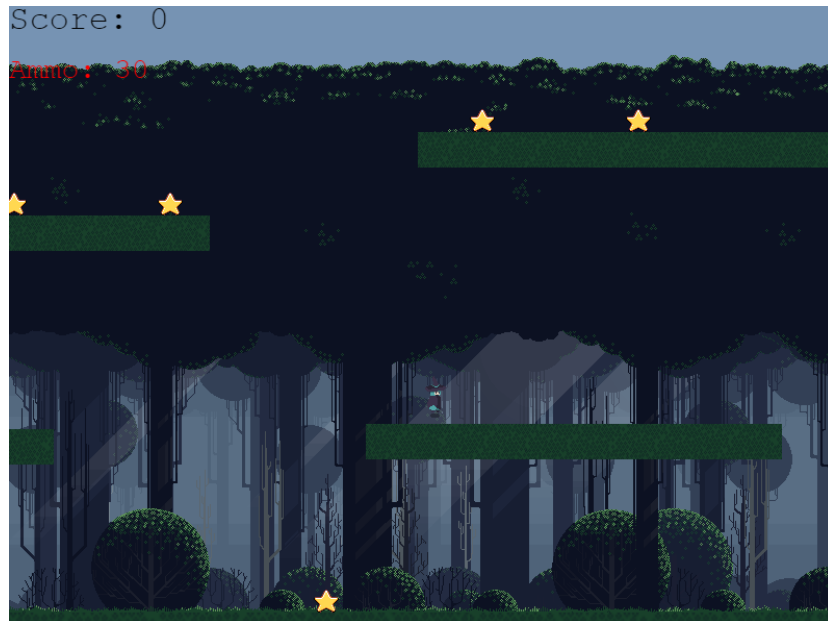
Debido a que la escena ocupa un espacio mayor que el que abarca el view port es necesario contar con una cámara que siga al jugador en todo momento. Son esencialmente las tres líneas siguientes las que permiten realizar este efecto.

```
this.cameras.main.setBounds(0, 0, 800 * 2, 600);
this.physics.world.setBounds(0, 0, 800 * 2, 600);
this.cameras.main.startFollow(this.player, true, 0.05, 0.05);
```

**Posición 1**



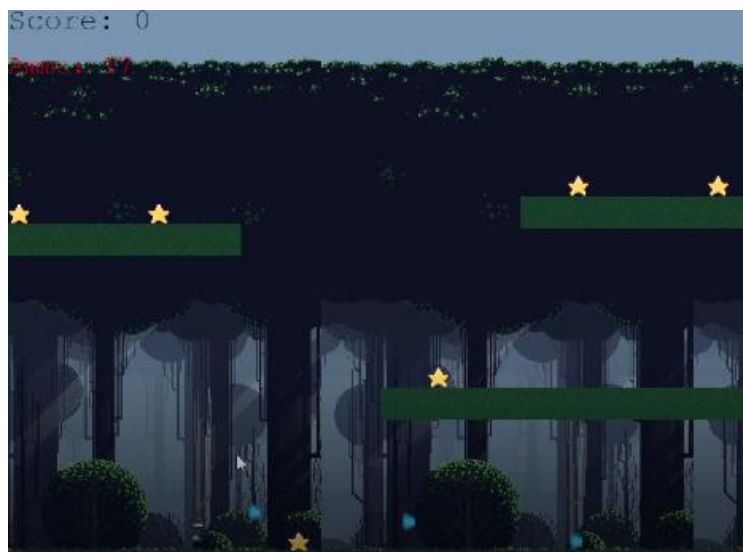
## Posición 2



Podemos notar que el HUD también se mueve junto con la cámara. Este se logró únicamente deshabilitando la opción de scroll del texto en la hora de la creación.

```
this.scoreText = this.add.text( this.player.x - 100,  
                                this.player.y - 450,  
                                'Score: 0',  
                                { fontSize: '32px', fill: '#000'  
}).setScrollFactor(0);
```

Otra mecánica implementada es la de disparar hechizos. Para disparar hechizos es necesario presionar la barra espaciadora o el botón izquierdo del mouse. Para dibujar el Sprite del hechizo se creó un grupo de Phaser. Podemos también apreciar que la munición disminuye.



Otra mecánica importante es que cada vez que se recolecta una estrella aparece un enemigo, el cual, al entrar en contacto con el jugador se termina el juego. El jugador tiene la capacidad de eliminar estos enemigos disparándoles.



Para finalizar debemos incluir el canvas que contiene nuestro juego en nuestro documento de la página web. Justo después del tag <script> que contiene el enlace a nuestro archivo JavaScript se coloca un elemento <canvas> con el contenido de nuestra aplicación de Phaser 3.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Showcase</title>
    <script src="Phaser/phaser.js"></script>
    <link rel="stylesheet" href="css/main.css">
  </head>
  <body>
    <script src="js/GameApp.js"></script>
  </body>
</html>
```

## Conclusión

Las principales modificaciones de mecánicas de juego respecto al provisto en el material son **el seguimiento de cámara y disparos y eliminación de enemigos**. Estos dos elementos permiten mucha más variedad de posibilidades de juego y de exploración. La estructura del código se hizo en

clases para la escena y para el efecto del hechizo y para el grupo de hechizos disparables. Al final el juego es un elemento más en nuestro documento de página web, por lo que es posible agregar más elementos antes o después o de aplicar técnicas y tecnologías vistas anteriormente.