

Apuntes generales PVLI

ÍNDICE

[Escenas](#)

[Texto](#)

[Botones](#)

[Botones por ratón](#)

[Botones por teclado](#)

Escenas

Para crear una escena añadimos el archivo de la escena al src y lo importamos al game, el orden de pase de las escenas se determina en "game.js" en el array Scenes.

Todas las escenas deben tener:

1. export default class "NombreEscena" extends Phaser.Scene
2. Una constructora de la escena con la key
3. Un create() donde se hará la inicialización de la escena
4. Funciones que tenga la escena

Texto

```
this.add.text(x,y,"texto", {})
```

Primero añadimos la posición x,y, luego escribimos el texto en cuestión y por último los parámetros.

fontFamily: Tipografía del texto

fontSize: Tamaño del texto

stroke: Borde del texto (Le pasamos el color en hexadecimal)

strokeThickness: Grosor del borde

color: Color de la fuente

align: Alineamiento del texto dentro de su posición

Botones

```
this.Button = this.add.text()
```

Guardamos el texto del botón como una variable, después lo hacemos interactuable y le añadimos los callbacks correspondientes.

Botones por ratón

```
this.Button.setInteractive();  
this.Button.on  
('pointerdown', ()=> this.ChangeScene  
('Escena siguiente', Escena actual, 'Variables de la escena'));  
this.Button.on('pointerover', ()=> this.OnButton(Botón sobre el que se actúa));  
this.Button.on('pointerout', ()=> this.OutButton(Botón sobre el que se actúa));
```

Hacemos que el texto sea interactuable para el ratón. Después le indicamos que debe hacer según las distintas acciones del ratón, algunos feedback otros acciones.

pointerdown: Cuando el ratón clicka

pointerover: Cuando el ratón está encima

pointerout: Cuando el ratón sale del collider

Botones por teclado

```
this.teclas = scene.input.keyboard.addKeys({  
up: Phaser.Input.Keyboard.KeyCodes.W,  
down: Phaser.Input.Keyboard.KeyCodes.S,  
select: Phaser.Input.Keyboard.KeyCodes.SPACE})  
scene: referencia a la escena actual
```

Primero registramos las teclas que vamos a utilizar como input en una variable, luego en el `update()` se comprueba si estas teclas están siendo pulsadas o no.

```
if (this.teclas.up.isDown) {}
```

Para el cambio de escena emitimos un evento creado en el `create`.