**Phaser – Resumen de la sesión 5: Convirtiendo nuestro proyecto en un juego**

En esta sesión, se aprendió a añadir input o entradas como el Mouse y el teclado, archivos locales de formato imagen y audio, como también agregar texto, y hacer que la página sea responsiva.

**Conceptos claves**

* **Cómo colocar la página responsiva**

Modificando las configuraciones de modo que aquellas referidas al tamaño, al contenedor y a la renderización estén dentro de “scale”. Además, se añaden 2 líneas nuevas que son

* **mode: Phaser.Scale.FIT,**
* **autoCenter: Phaser.Scale.CENTER\_BOTH**

quedando de la siguiente forma:

const config = {

title: "",

scale:{

mode: Phaser.Scale.FIT,

autoCenter: Phaser.Scale.CENTER\_BOTH,

type: Phaser.AUTO,

parent: "contenedor",

width: 800,

height: 600,

},

* **Agregar texto**
* Indicando el método text con el cual se usan 3 parámetros: posición X, posición Y, y contenido del texto, quedando de la siguiente forma:

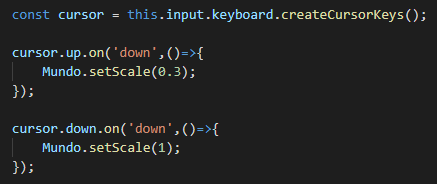
**This.add.text(200,200,” Hola Mundo”);**

* **Agregando recursos locales**
* Para esto fue necesario crear una carpeta llamada Assets y eliminar la línea de la función preload donde seteamos una Url (setBaseURL), quedando de la siguiente forma:

**this.load.image(‘raqueta’, ’./Assets/raqueta.png’);**

* **Cómo crear un cursor predeterminado**
* **const cursor = this.input.keyboard.createCursorKeys();**

un ejemplo de esto sería el siguiente:

****

En este ejemplo, al presionar tecla flecha hacia arriba (up), el objeto mundo fija su tamaño a 0.3, mientras que al presionar tecla flecha hacia abajo (down) fija su tamaño a 1 (tamaño normal).

* Cómo agregar un sonido
* Para esto, entendemos que el sonido debe estar previamente cargado en la función preload, para luego llamarlo en la función create de la siguiente forma

**const nombre = this.sound.add(‘nombre\_sonido');**

Para reproducir el sonido será suficiente con colocar  
**nombre\_sonido.play();p**

* **Colisiones**

Las colisiones son parte importante para la creación de un juego, ya que es la forma que tenemos que un objeto reconozca a otro. Esto lo podemos realizar de la siguiente forma:

this.physics.add.collider(Kaze, Plataforma);