Taller de Desempeño Profesional

Technical Expert

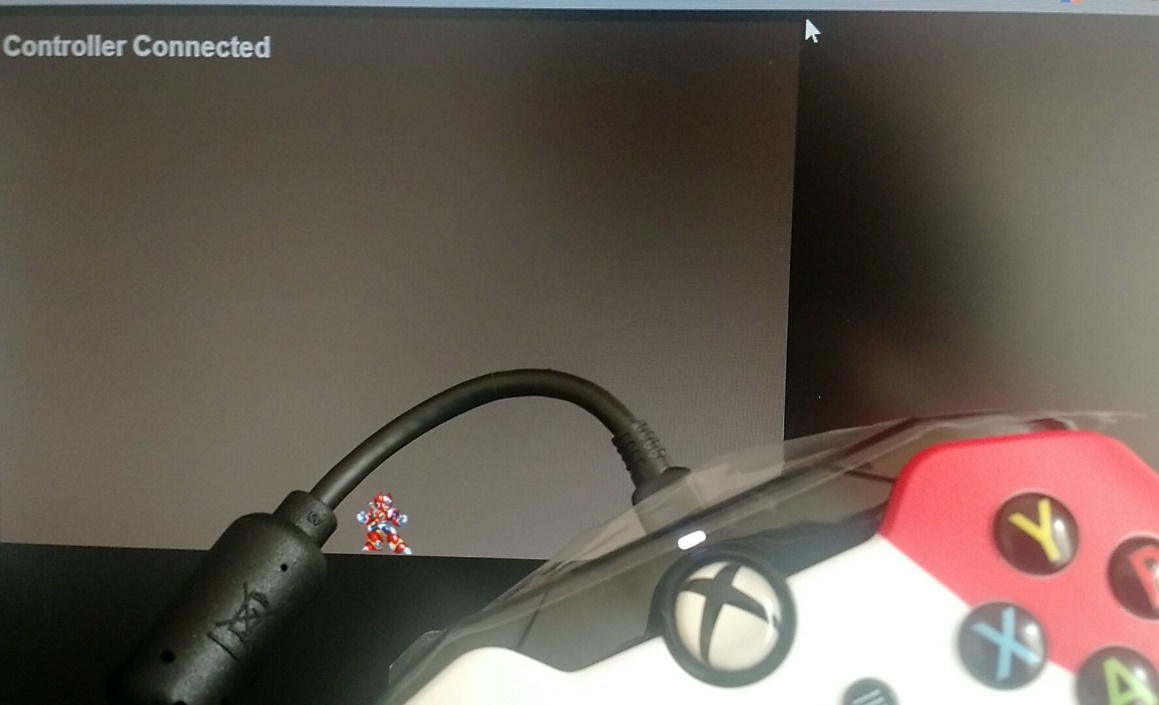
(Phaser)



Gamepad

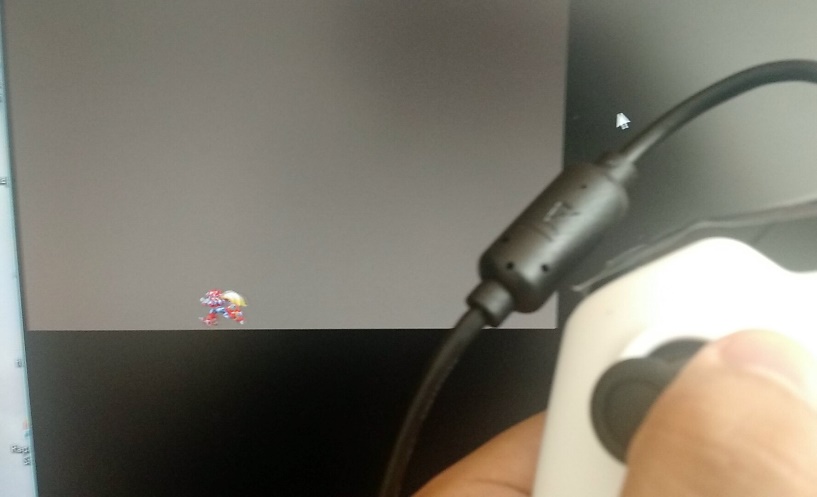
Eduardo Chiang

Para esta demo implementaremos el uso de un control de Xbox One para un juego web. Se puede comprobar que se registra si el control se encuentra conectado o no a la computadora.



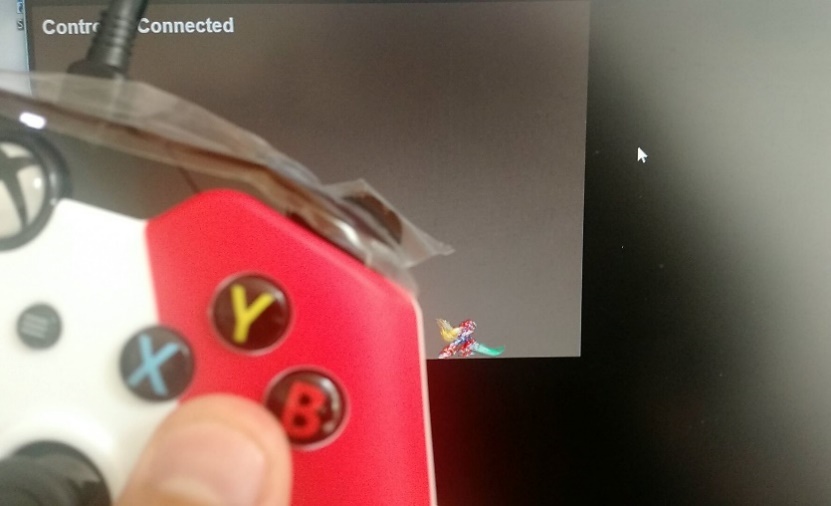


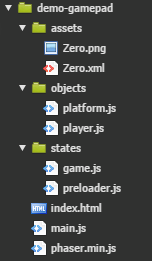
Utilizando el joystick izquierdo, si se mueve hacia los lados, el jugador se moverá en la dirección deseada. Si se mueve hacia arriba, el jugador saltará.





Por otro lado, si se presiona el botón “A” el jugador realizará un ataque.



El directorio del proyecto tendrá la siguiente estructura:

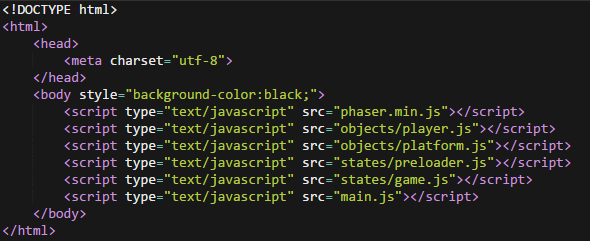
Carpeta de *Assets* Se guardarán todos los archivos multimedia a utilizar en el proyecto.

Carpeta de *Objects* Se guardarán los archivos .js de clases extendidas.

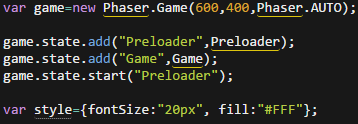
Carpeta de *States* Se guardarán los archivos .js de estados a utilizar en el proyecto.

Raíz En la raíz se encontrará el *index*, el *main* y el archivo que contiene Phaser.

En el *index* sólo se agregan los scripts de javascript como Phaser, *main*, estados y clases extendidas.



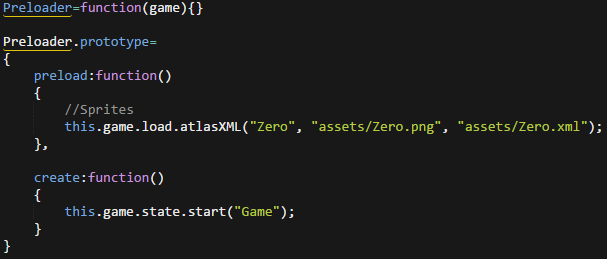
En el *main* declararemos la variable *game* que contendrá nuestro objeto Phaser. Será creado con un tamaño de 600x400 y con detección automática para el renderizador, ya sea WebGL o Canvas. Luego se declaran los estados que serán utilizados en la aplicación y especificamos con qué estado iniciar la aplicación. Le agregaremos también una variable que contenga el estilo del texto que imprimiremos en pantalla.



En el estado “Preloader” declararemos los *assets* que se usarán para precargarlos.

Cargaremos el *spritesheet* como un atlas de XML, el cual recibe un nombre único para identificarlo, la imagen que contiene las animaciones y el archivo .xml que lo mapea.

Finalmente, redirigimos la aplicación al estado “Game”.

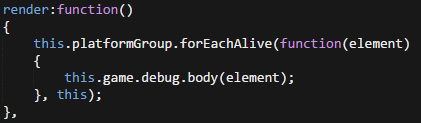


Dentro del estado “Game”, declararemos que se utilizará un fondo marrón y física del tipo Arcade, la cual sólo utiliza colisiones entre cuadrados, y definiremos una gravedad de -1000.

Crearemos también la variable que guardará el control de Xbox, el jugador y un grupo que contendrá las plataformas autogeneradas.

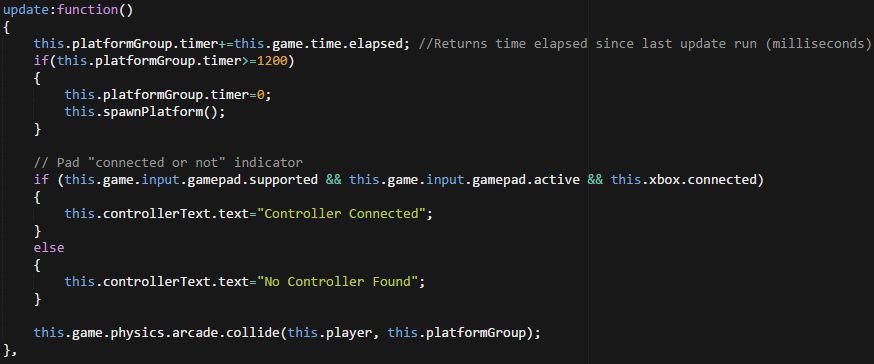


Ya que en este caso no estamos usando un *sprite* para las plataformas, utilizaremos la función nativa de Phaser *render* para pintar el cuerpo físico de las mismas.

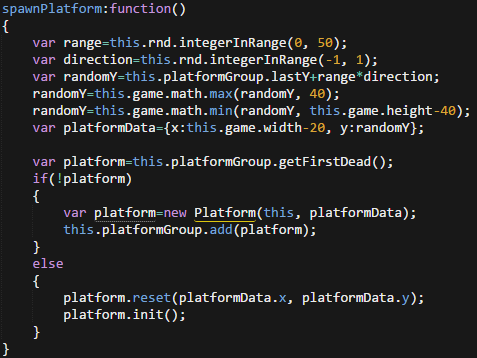


En el *update*, le aumentaremos al contador de tiempo de las plataformas el tiempo transcurrido desde la última llamada de la función, devuelto en milisegundos. De este modo, generaremos una plataforma cada 1.2 segundos.  
Además, validaremos que el control creado anteriormente esté conectado y sea soportado por Phaser.

Finalmente declararemos que habrá colisión entre el jugador y las plataformas.

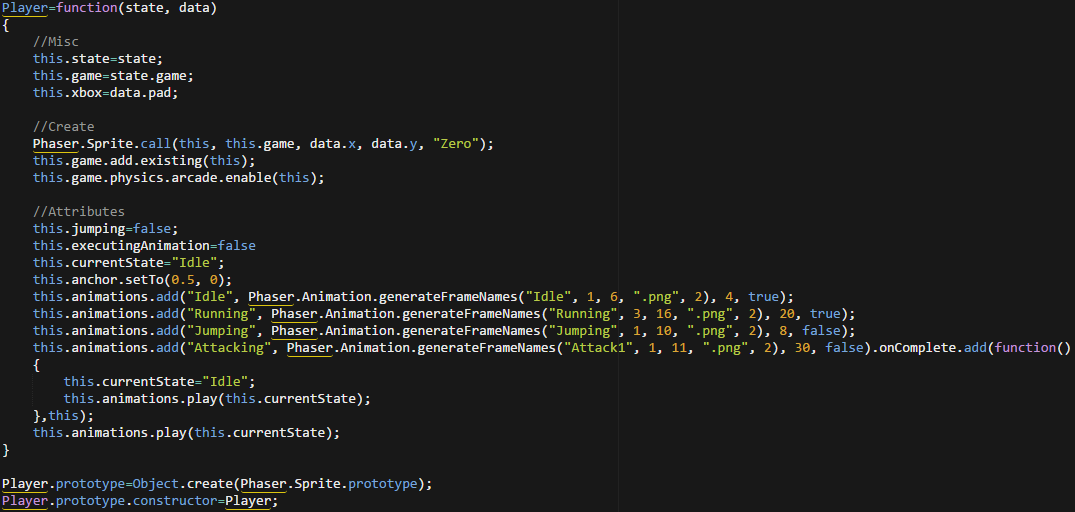


Para autogenerar las plataformas, utilizaremos un valor de y aleatorio dentro de un rango alrededor de la plataforma anterior de modo que siempre se pueda alcanzar la siguiente plataforma. Utilizaremos también la función getFirstDead() para recuperar un objeto que ha sido desactivado con la función kill() y reutilizarlo, ahorrando así la creación del mismo.



En la clase “Player” declararemos que tendrá un cuerpo sujeto a la física de la aplicación.

Para cada animación crearemos un identificador único y entregaremos un arreglo que contenga la secuencia de imágenes que representa dicha animación, además de la velocidad a la que debe ser ejecutada y si es que debe o no continuar reproduciéndose después de completarse.



Si el jugador sale de los límites de la pantalla por los lados o por abajo, se reiniciará el juego.

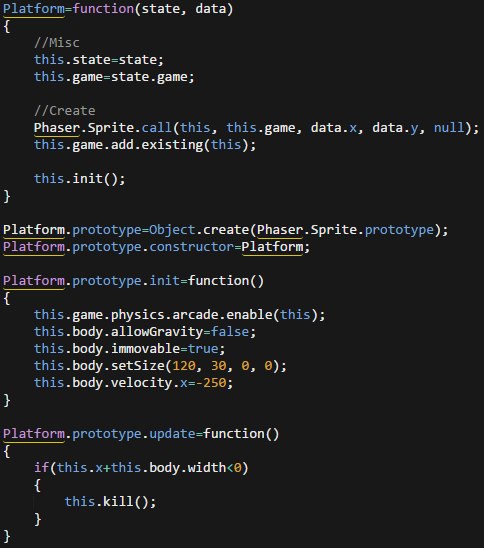
El jugador utilizará el joystick izquierdo de; control de Xbox para moverse y el botón “A” para atacar.



En la clase “Platform” declararemos que tendrá un cuerpo sujeto a la física de la aplicación con un tamaño de 120x30 y una velocidad constante hacia la izquierda de 250.

La plataforma no se verá afectada por la gravedad ni recibirá una fuerza opuesta cuando reciba una colisión.

Finalmente, se desactivará la plataforma si sale de la pantalla por el límite izquierdo.



Imágenes de la demo:





