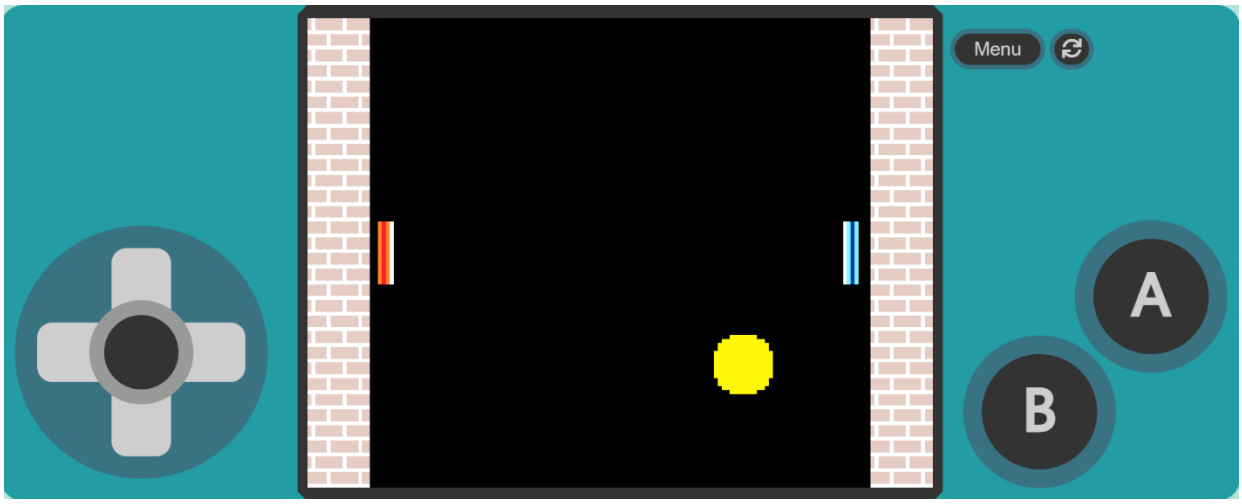
Cuarto Ejercicio Motion



Descripción

En este proyecto realizaremos un juego basado en uno de los primeros videojuegos de la historia **Pong**.

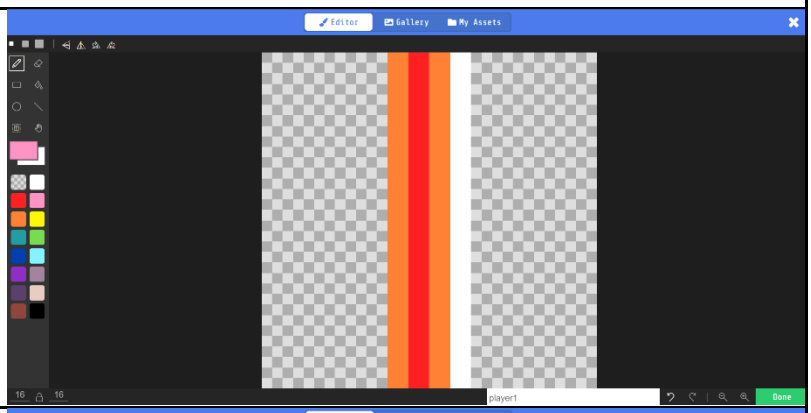
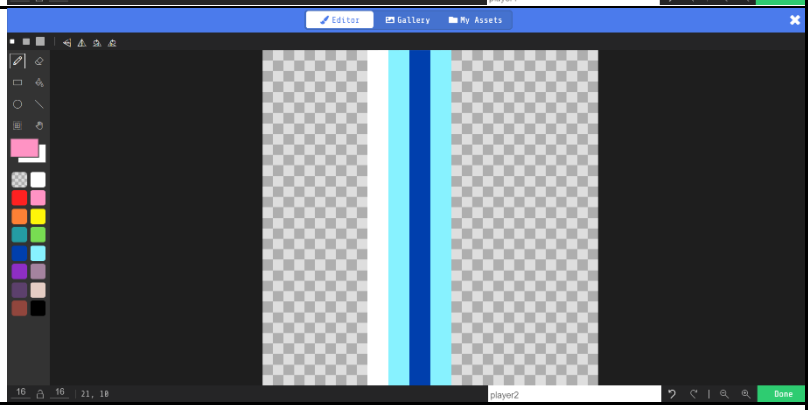
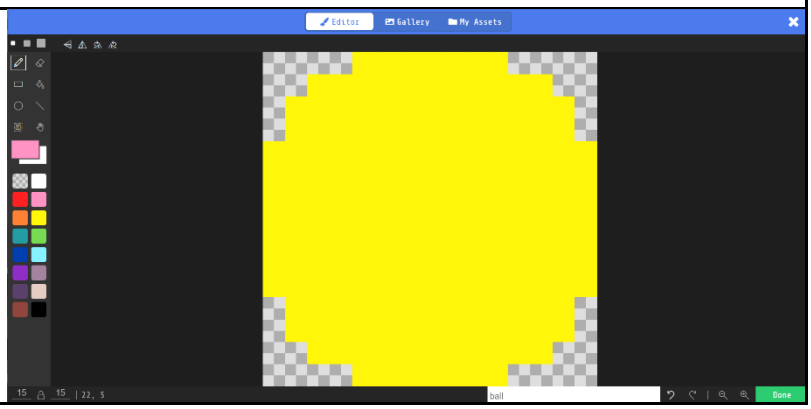
El objetivo principal del videojuego es evitar que la pelota colisione con nuestro muro (nos quita un punto) devolviéndola y haciendo que colisione con muro del oponente para ganar puntos y el que consiga más puntos gana.

Para ello accederemos a **MakeCode Arcade** y realizaremos las operaciones necesarias.

Objetivos de programación y diseño

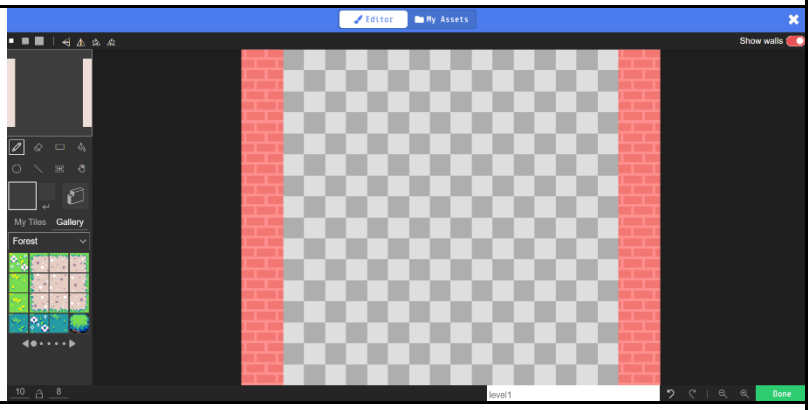
- Crear un Sprite de la barra del jugador 1 y jugador 2.
- Crear un Sprite de la pelota.
- Crear un Tilemap con muros en los laterales.
- Crear Mecánica de rebote de la pelota en los muros y hacer perder un punto.
- Crear Mecánica de rebote en las barras de los jugadores y ganar un punto.

Programación del juego

Aquí os dejamos el enlace con parte de la programación y Assets hechos. https://makecode.com/_JRj6drRms69z
CREACIÓN DE ASSETS
CREACIÓN SPRITES JUGADORES (BARRAS)
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de player1. 
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de player2. 
CREACIÓN SPRITE BALL
Te recomendamos utilizar una matriz de 15x15 px para el Sprite de ball. 

CREACIÓN TILEMAP

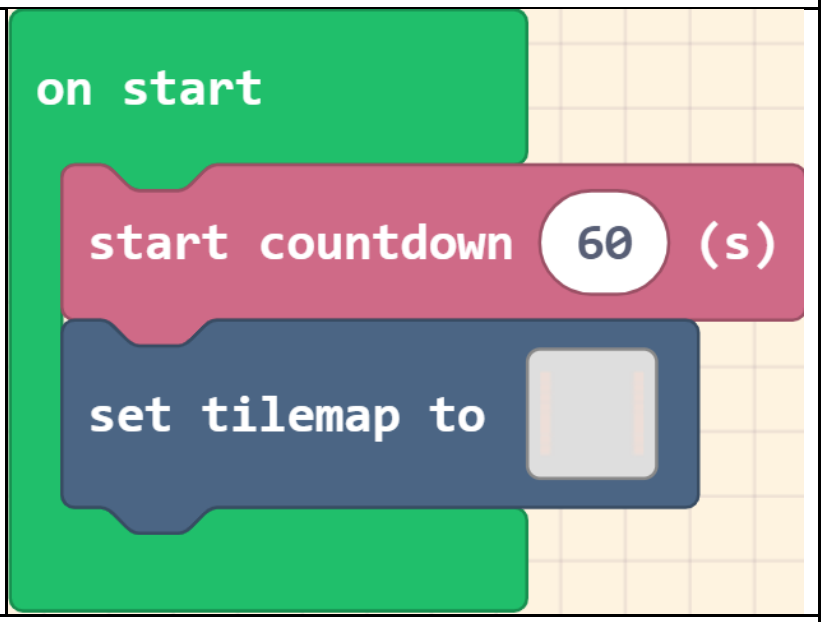
Te recomendamos utilizar una matriz de 10X8 px para el Tilemap level1. Le colocamos los muros encima de los ladrillos para que la pelota rebote.



PROGRAMACIÓN PRINCIPAL

CREACIÓN INICIO DEL JUEGO

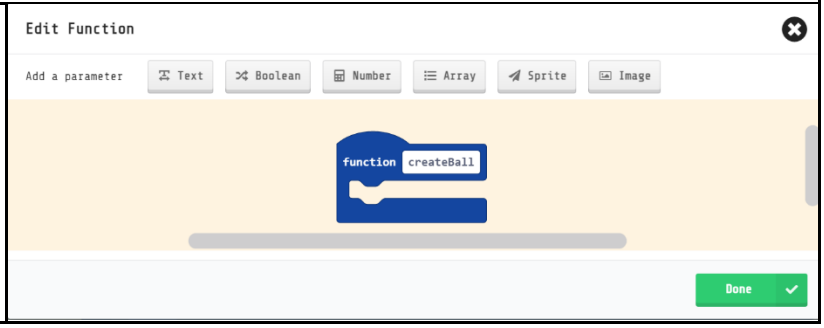
Para comenzar, crearemos un bloque on start y colocaremos en su interior start conutdown lo establecemos en 60 segundos, set tilemap to y colocamos nuestro tilemap level1.

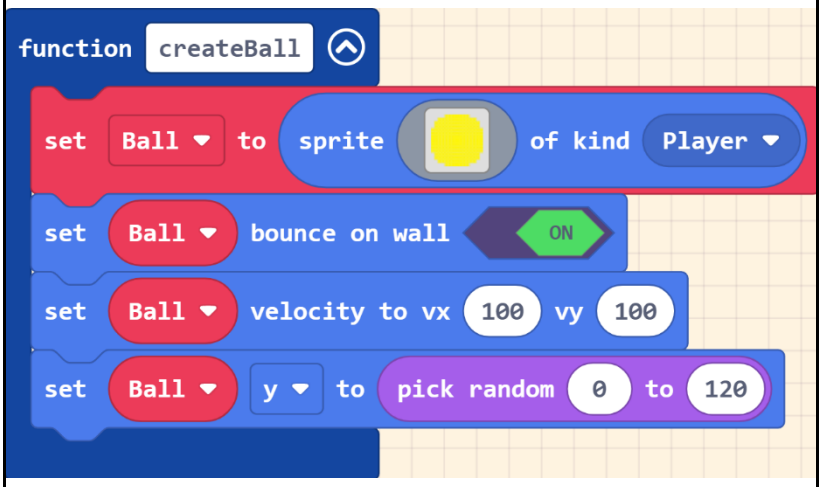
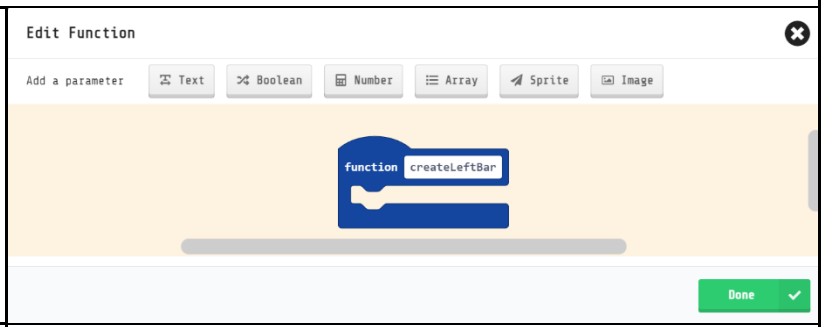
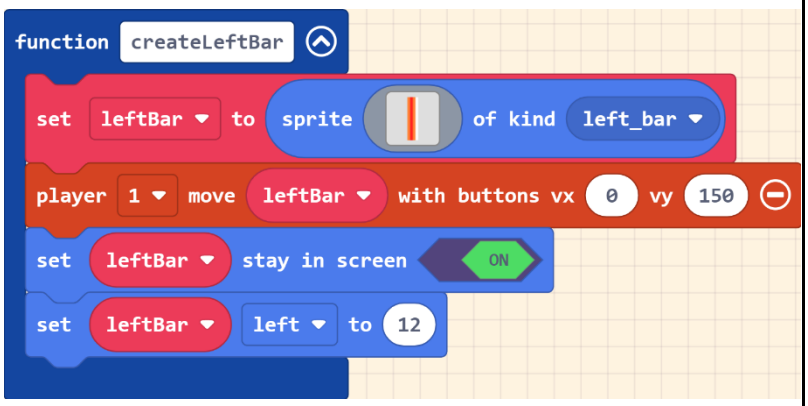


CREACIÓN FUNCIÓN createBall

Acto seguido, vamos a empezar a crear nuestra pelota.

Vamos a crear una función para guardar todo nuestro código en él.

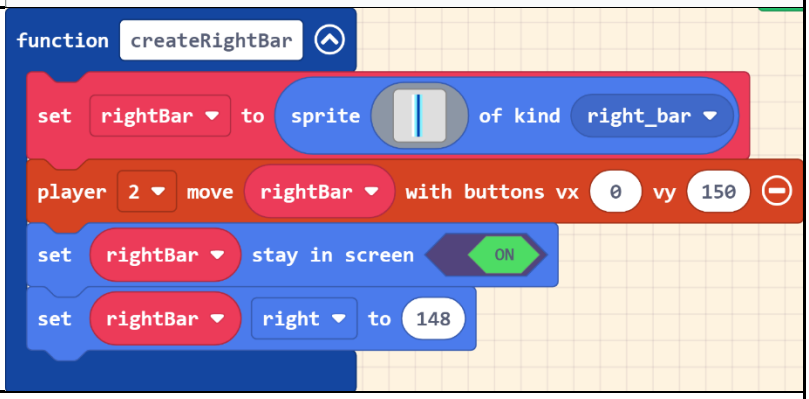


Como ya venimos viendo: set Ball to (Sprite (colocamos el Sprite de nuestra pelota) of kind Player) set Ball bounce on wall. Para que rebote con los muros puestos en el tilemap. set Ball velocity to vx 100 vy 100. Para que tenga una velocidad. set Ball y to (pick random 0 to 120). Para posicionarlo en en un sitio aleatorio en el centro pero a distinta distancia en el eje y.	
CREACIÓN FUNCIÓN createLeftBar	
Vamos a crear al jugador 1 y definir su posición, movimiento y velocidad y lo guardaremos todo en una función.	
Empezaremos definiendo a leftBar como el Sprite player1 y como left_bar. A continuación, le diremos que leftBar se mueve en el eje y a 150 y eje x en 0. Así solo se mueve de arriba abajo. Limitamos el que se pueda salir de la pantalla con set leftBar stay in screen on. Por último le decimos que se coloque a la izquierda: set leftBar left to 12.	

CREACIÓN FUNCIÓN createRightBar

Este paso es más fácil ya solo tenemos que duplicar la anterior función hacer una serie de cambios y entre ellos es cambiar el nombre.

Luego solo tenemos que cambiar las variables por las nuevas que se aprecian en la imagen.

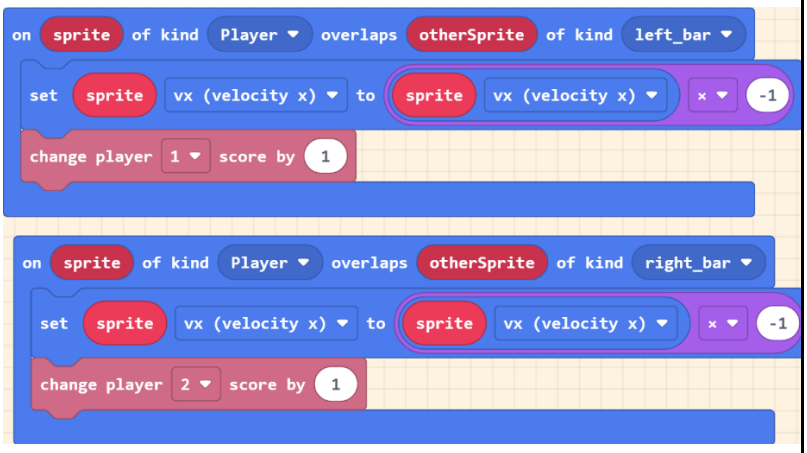


CREACIÓN MECÁNICA DE COLISIÓN CONTRA LAS BARRAS DE LOS JUGADORES Y OBTENCIÓN DE PUNTOS

Para esta mecánica, comprobaremos cuando la pelota choca con una de las barras (left_bar o right_bar).

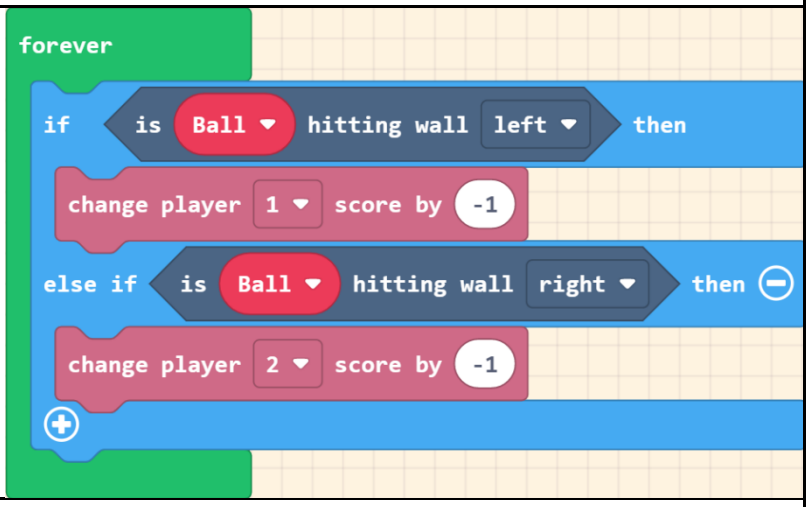
on sprite of kind Player overlaps otherSprite of kind left_bar. Con esto comprobamos si la pelota colisiona con la barra de la izquierda.
Le decimos que cuando colisione, cambie el sentido del exe x y que le sume un punto al player1.

Luego hacemos los mismo con el player2.



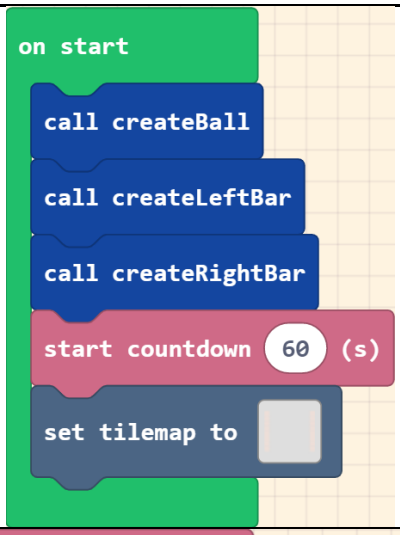
CREACIÓN MECÁNICA DE COLISION CON EL MURO Y PERDIDA DE PUNTOS

Ahora vamos a realizar la mecánica para que cuando la pelota colisione con el muro, le reste puntos al jugador correspondiente.
En un Loops forever, para que compruebe siempre, le vamos a decir lo siguiente. Si la pelota toca el muro de la izquierda, entonces, le restas un punto al jugador 1. En cambio, si toca el muro de la derecha, entonces, le restas un punto al jugador 2.

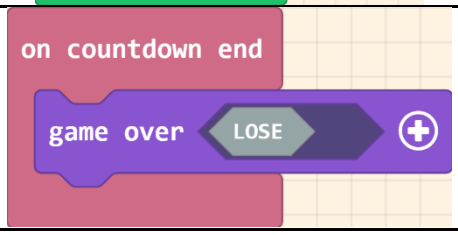


ÚLTIMOS DETALLES

Ahora vamos a llamar a las funciones creadas anteriormente en el inicio del juego para que se cree todo.



Y para que el juego no se haga eterno, vamos a decirle que cuando la cuenta atrás llegue a 0, que se acabe el juego.



Gracias a esta programación hemos hecho un “Ping Pong” para 2 jugadores, en el cual, hemos aprendido a realizar dos jugadores y aplicarles controles, puntuaciones y movimiento independientes. También hemos aprendido a realizar que cuando se acabe el tiempo se acabe el juego, que cuando la pelota colisione con la barra obtenga ese jugador un ponto y si choca con el muro pierda un punto. Ahora, es tu turno de personalizarlo y añadirle contenido. Aquí te dejamos el nuestro para que te inspires un poco: **https://makecode.com/_FpeYTxYLz68A**

Glosario

Bucles: Secuencia de código que se ejecuta repetidas veces.

Ejemplo: Forever, while, for.

Condicionales: Secuencia de instrucciones que se ejecuten en función del valor de una condición.

Ejemplo: If, If…Else

If…Else if: Secuencia de condicionales en la cual, iremos pasando, de manera ordenada, de una condición a otra hasta que se cumpla una de ella.

Funciones: Es un subprograma que recoge un conjunto de instrucciones y pueden ejecutarse desde el programa principal haciendo una llamada a él.

Escenario: Espacio donde se desarrolla el videojuego.

Muros: Objetos o espacios donde los distintos elementos del juego no pueden atravesar.

Puntuación: Puntos totales que obtiene un jugador al realizar ciertas interacciones.

Cuenta atrás: Tiempo que se establece para que vaya corriendo y al terminar suceda algo, como terminar la partida.

Game Over: La partida se ha terminado. Se suele mostrar puntuaciones y te pregunta si quieres jugar otra partida.

Música: Combinación de sonidos y silencios que componen un ritmo.