

Primer Ejercicio Functions



Descripción

En este proyecto se trabaja en un juego que usa distintos conceptos de programación y videojuegos, cómo el trato a los sprites, interacciones con distintos elementos del juego o el uso de funciones.

El juego se basará en esquivar enemigos mientras se va cogiendo comida. Se podrá eliminar a los malos usando el ataque del personaje controlado.

Para ello accederemos a [MakeCode Arcade](#) y realizaremos las operaciones necesarias.

Objetivos de programación y diseño

- Trabajar el movimiento de sprites.
- Programar interacciones entre elementos.
- Establecer una puntuación que cambie dependiendo de la situación.
- Usar funciones para programar distintos eventos.
- Programar la entrada en escena de los enemigos, su velocidad y aceleración.

Programación del juego

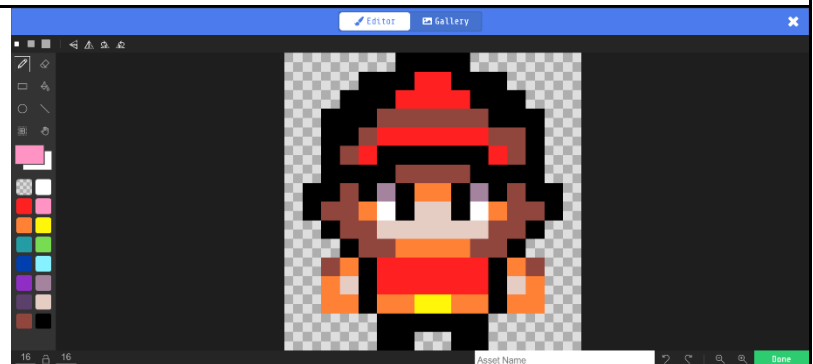
Aquí os dejamos el enlace con parte de la programación y Assets hechos.

<https://makecode.com/7KvXtMcvX3Ej>

CREACIÓN DE ASSETS

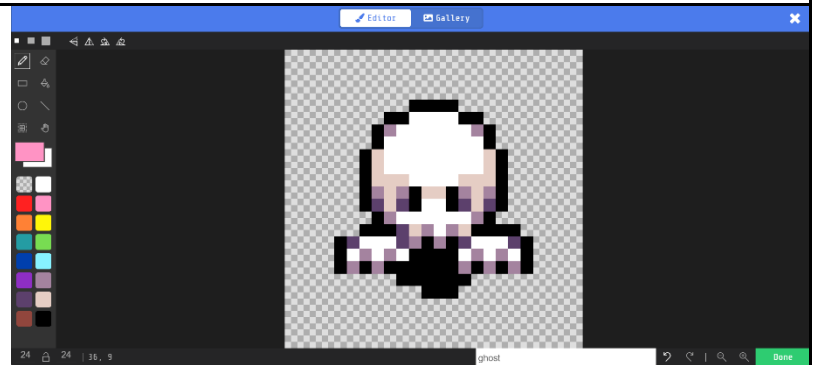
CREACIÓN SPRITE PLAYER

Te recomendamos utilizar una matriz de 16x16 px para el Sprite de **mySprite2**.

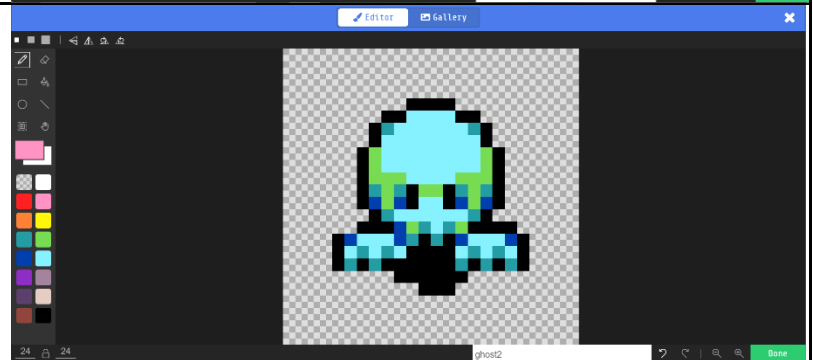


CREACIÓN SPRITE ENEMIGOS

Te recomendamos utilizar una matriz de 24x24 px para el Sprite de **ghost**.

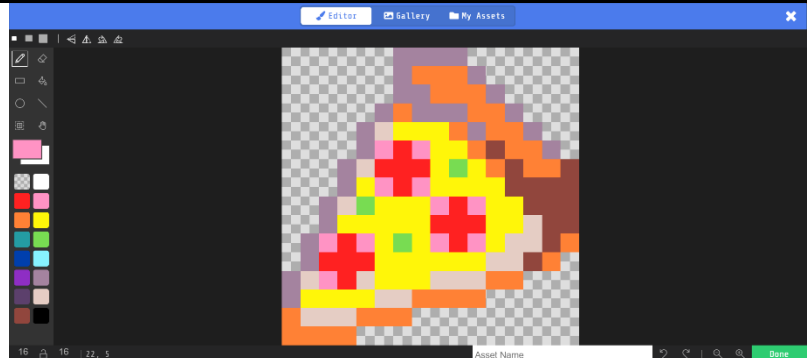


Te recomendamos utilizar una matriz de 24x24 px para el Sprite de **ghost2**.

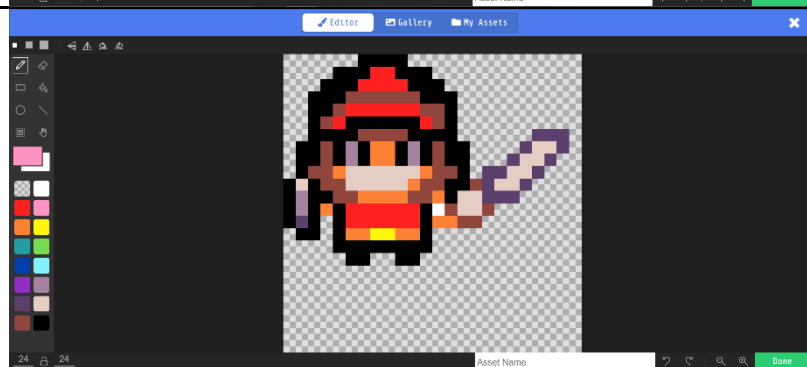


CREACIÓN SPRITE ADICIONALES

Te recomendamos utilizar una matriz de 16x16 px para el **Sprite** de **block2**.



Te recomendamos utilizar una matriz de 24x24 px para el **Sprite** de **block3**.

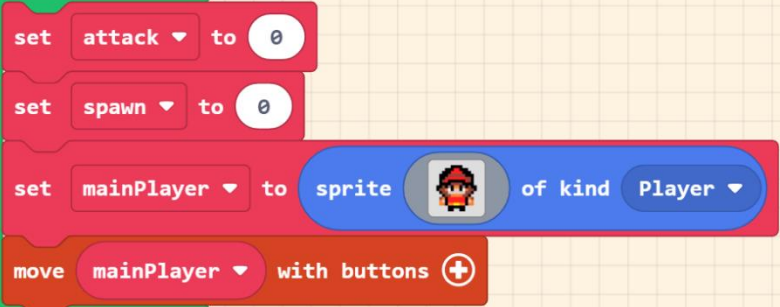


PROGRAMACIÓN PRINCIPAL

CREACIÓN INICIO DEL JUEGO

Comenzaremos estableciendo las **variables** **"attack"** y **"spawn"** que usaremos más adelante. También crearemos a nuestro **personaje** y le daremos movimiento usando la **cruce**.

on start



CREACIÓN MECÁNICA ATAQUE PLAYER

Programaremos que, al pulsar el botón A, nuestro personaje se defienda de los enemigos, utilizando la variable “ataque” y dos imágenes distintas del sprite para simular un golpe. Más adelante lo concretaremos más.

```

on A button pressed
  set mainPlayer image to [Mario]
  set attack to 1
  pause 2000 ms
  set mainPlayer image to [Mario with sword]
  set attack to 0
  pause 1000 ms
  
```

Con esta programación nos podremos defender de los enemigos. Cuando la variable “ataque” sea 1, al pulsar A como ya vimos, podremos destruir todo aquel sprite enemigo que toquemos, lo que nos sumara un punto.

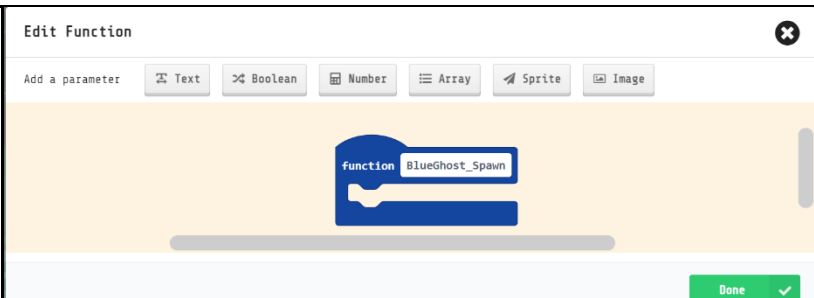
```

on sprite of kind Player overlaps otherSprite of kind Enemy
  if attack = 1 then
    destroy otherSprite
    change score by 1
  
```

CREACIÓN MECÁNICA CREACIÓN DE ENEMIGOS

CREACIÓN FUNCIÓN BlueGhost_Spawn

Vamos a crear nuestro enemigo, para ello vamos a crear una función BlueGhost_Spawn.



Dentro de la misma crearemos el **sprite** del enemigo, estableceremos que se **autodestruya** al salir de la escena y que la **variable** “**salida**” escoja un **numero aleatorio** entre 0 y 1.

Por medio de este “**if else**” y de los bloques que encierra, estableceremos que los enemigos salgan de un lado u otro del mapa en función de si la **variable** es 1 o 0.

```
function BlueGhost_Spawn
  set spawn to pick random 0 to 1
  set blueGhost to sprite of kind Enemy
  set ghost auto destroy ON
  if spawn = 0 then
    set blueGhost position to x 160 y pick random 40 to 100
    set blueGhost velocity to vx pick random -60 to -20 vy 0
  else
    flip blueGhost image horizontally
    set blueGhost position to x 0 y pick random 40 to 100
    set blueGhost velocity to vx pick random 20 to 60 vy 0
```

CREACIÓN FUNCIÓN Ghost_Spawn

Utilizaremos otra **función** para crear a nuestro segundo **tipo de enemigo**.

Dentro de la misma crearemos el **sprite** y le estableceremos una **velocidad y posición aleatoria** para que sea impredecible. Cuando salga de la escena se **autodestruirá**.

```
function Ghost_Spawn
  set ghost to sprite of kind Enemy
  set ghost position to x pick random 20 to 120 y 20
  set ghost velocity to vx pick random -40 to 100 vy pick random 40 to 100
  set ghost auto destroy ON
```

En este “**game update**”, llamaremos a las **funciones** anteriormente creadas para que se lleven a cabo cada 1.5 segundos, creando así a los **enemigos**.

on game update every 1500 ms

call Ghost_Spawn

call BlueGhost_Spawn

CREACIÓN MECÁNICA CREACIÓN DE COMIDA

Con este “game update”, estableceremos que vayan apareciendo estos sprites de tipo comida, en una posición aleatoria cada segundo.

```
on game update every 1000 ms
  set pizza to sprite of kind Food
  set pizza position to x pick random 30 to 130 y pick random 40 to 120
```

CREACIÓN MECÁNICA OPTENCIÓN DE PUNTOS Y SUBIDA DE DIFICULTAD

Con el “overlaps” establecemos que, si nuestro personaje toca un sprite de tipo comida, este último se destruya y nos dé un punto.

Con el bloque de abajo y llamando a la función, haremos que por cada sprite de comida destruido aparezca un fantasma nuevo, aumentando así la dificultad.

```
on sprite of kind Player overlaps otherSprite of kind Food
  destroy otherSprite
  change score by 1

on destroyed sprite of kind Food
  call Ghost_Spawn
```

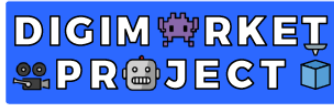
Para aumentar aún más la dificultad, llamaremos a la función del fantasma dentro del “on start”, haciendo que aparezcan dos fantasmas nada más empezar.

```
on start
  set attack to 0
  set spawn to 0
  set mainPlayer to sprite of kind Player
  move mainPlayer with buttons
  call Ghost_Spawn
```

CREACIÓN MECÁNICA DE FINAL DEL JUEGO

Cuando nuestro ataque este en reposo, la variable estará en 0. Si en este momento nos impacta un enemigo, perderemos la partida.

```
on sprite of kind Enemy overlaps otherSprite of kind Player
  if attack = 0 then
    game over LOSE
```



Con esta programación usaremos **funciones** y **variables** para ayudar a nuestro **sprite de tipo "Player"** a obtener **sprites de tipo "Food"** y eliminar **sprites de tipo "Enemy"**, para conseguir así la mayor cantidad de **puntos posibles**. Si **impactamos** con un enemigo cuando nuestra **variable** nos haga vulnerables, la **partida terminará**.

Ahora, es tu turno de personalizarlo y añadirle contenido. Aquí te dejamos el nuestro para que te inspires un poco: <https://makecode.com/JJp1fyYmf52q> o <https://makecode.com/hrHKfA07saA2>

Glosario

Variables: Es un espacio asociado a un identificador, en ese espacio hay un valor que puede ser modificado.

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

Evento: Ejecuta una secuencia de instrucciones cuando ocurre un suceso externo al sistema.

If-Else: Sentencia condicional que si se cumple ejecuta una secuencia de instrucciones y otra que omite.

Aceleración: Es la variación de velocidad por unidad de tiempo.

Velocidad: Es una magnitud física que relaciona la posición con el incremento de tiempo.

Ciclo de Vida: Duración de un elemento de un programa desde su creación hasta su destrucción.

Aleatoriedad: La generación de números que tienen la misma probabilidad de generarse.

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

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