

Segundo ejercicio Sprites



Descripción

En este proyecto se trabaja con un juego que usa distintos conceptos de programación y videojuegos, cómo el trato a los sprites, el uso de funciones o el uso de tilemap de colores y sustituir estos colores por distintos sprites. También se apreciarán animaciones de sprites, interacciones con distintos elementos del juego y más instrucciones que añaden sustancia al videojuego.

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

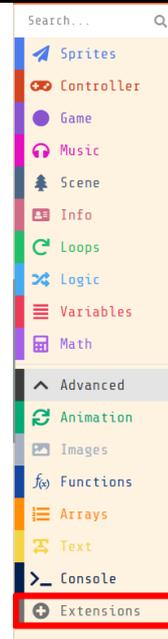
Objetivos de programación y diseño

- Crear un Sprite "Protagonista" que podamos controlar su movimiento y que pueda realizar distintos tipos de salto.
- Dibujar un mapa con colores y por cada color, sustituirlos por distintos elementos para el juego.
- Programar funcionar para distintos apartados del juego.
- Animar las monedas que aparezcan en pantalla.
- Generar distintos niveles.
- Aplicar distintos elementos de física en el Sprite que se controlará.

Antes de empezar

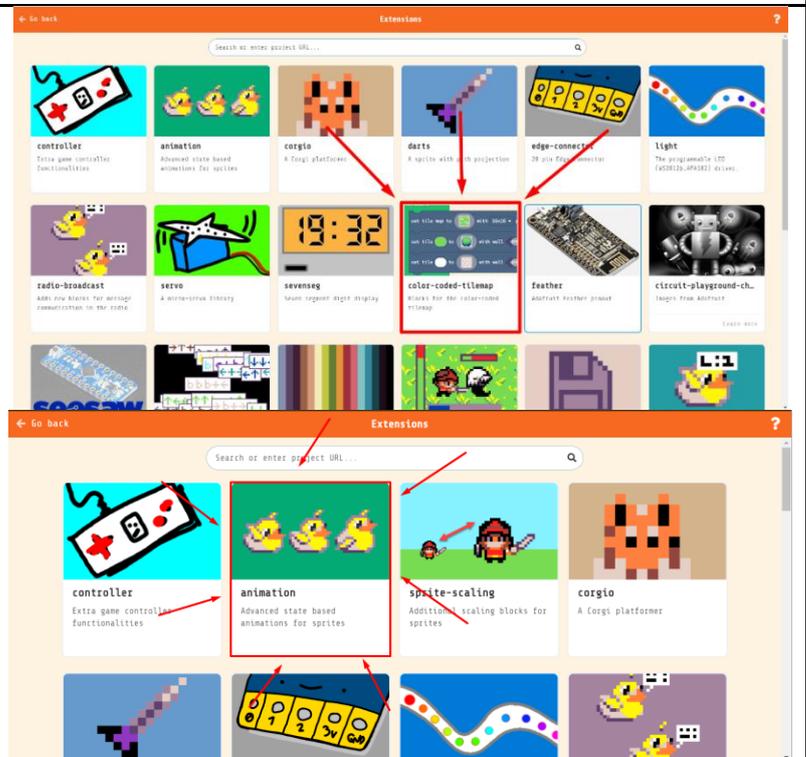
INSTALACIÓN DE EXTENSIONES NECESARIAS

Para instalar una extensión, hacemos clic abajo del todo en **Extensions**.



Una vez en las **Extensions** seleccionaremos “color-coded-tilemap”.

Después realizaremos la misma operación y seleccionaremos “animation”.



Programación del juego

CREACIÓN DE ASSETS

CREACIÓN SPRITE PROTAGONISTA

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



CREACIÓN ESCENARIO

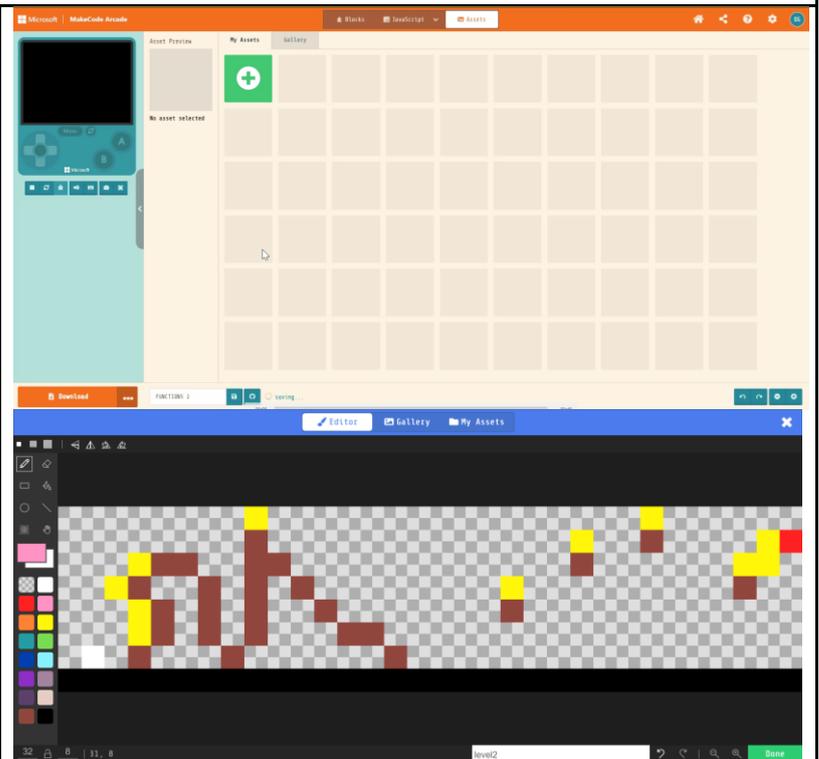
Te recomendamos utilizar una matriz de 64x12 px para el **Sprite** de **level1**.

Para el **Sprite** de **level2** utilizaremos una matriz de 32x8 px.

Colorearemos el nivel para colocar distintos elementos:

- Negro = Suelo
- Marrón = Obstáculos
 - Azul = Agua
 - Amarillo = Moneda
 - Naranja = Trampa
- Blanco = Punto de inicio
 - Rojo = Meta final

Podéis hacer el nivel que queráis, pero os recomendamos hacer estos para poder seguir mejor el ejercicio.

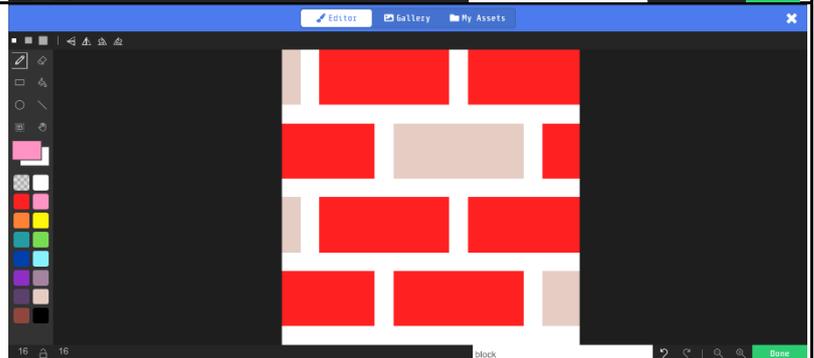


CREACIÓN SPRITES ADICIONALES

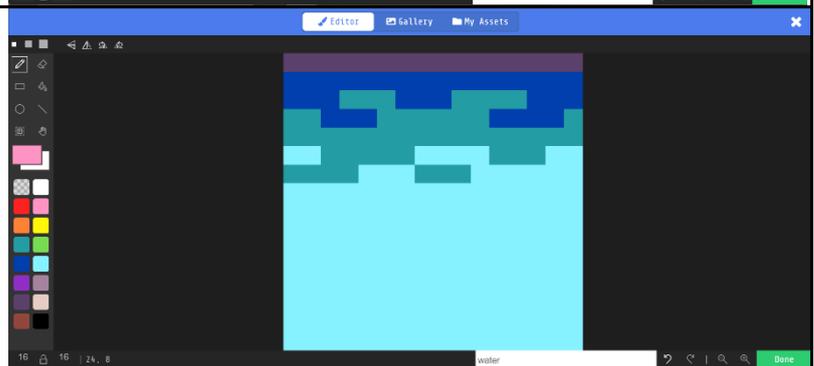
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de floor.



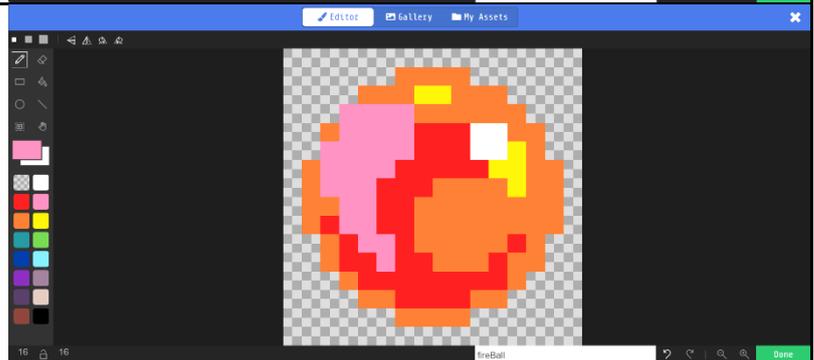
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de block.



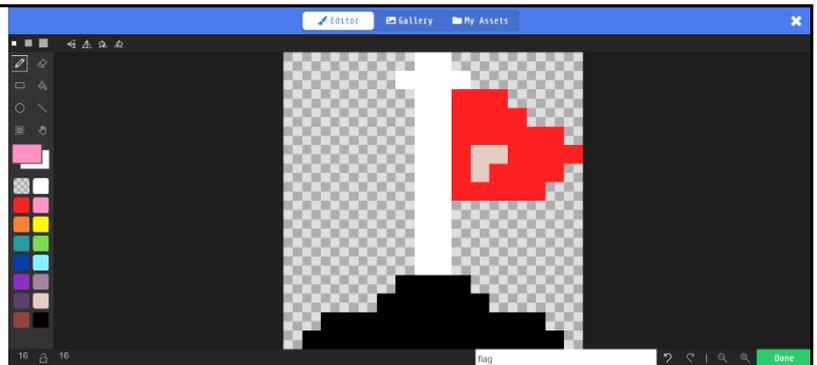
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de water.



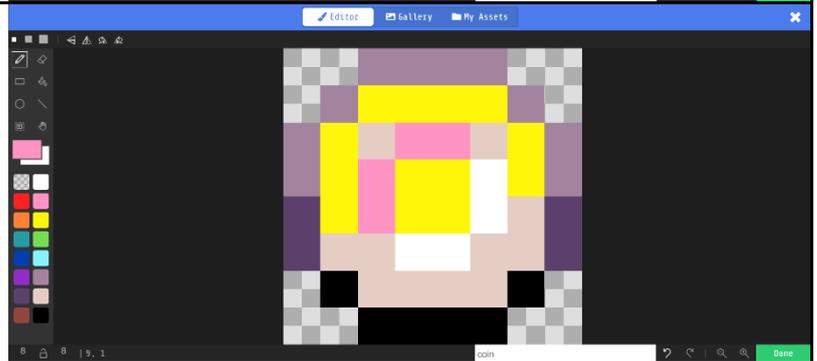
Te recomendamos utilizar una matriz de 16x16 px para el Sprite de fireBall.



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



Te recomendamos utilizar una matriz de 8x8 px para el **Sprite** de **coin**.



CREACIÓN SPRITES ANIMACIÓN

En el editor de la animación, es importante escoger los distintos frames de la animación para realizar la animación de la moneda. Los podemos sacar de "Gallery"



PROGRAMACIÓN PRINCIPAL

CREACIÓN INICIO DEL JUEGO

Con esta programación estableceremos el **nivel actual**, que empieza siendo el nivel cero. Crearemos el **Sprite** que vamos a **controlar** y estableceremos que:

- La **cámara** le siga
- Una **gravedad** en el eje y
- Podamos **controlarlo** en el eje x

Por último, seleccionaremos el nivel en el bloque "set tile map to".

on start

set changeLevel to 0

set hero to sprite of kind Player

camera follow sprite hero

set hero ay (acceleration y) to 300

move hero with buttons vx 100 vy 0

set tile map to

CREACIÓN MECÁNICA DE CREACIÓN DEL ESCENARIO

CREACIÓN DE LA FUNCIÓN startStage

Crearemos una función "startStage" de la pestaña "Functions".

Edit Function

Add a parameter Text Boolean Number Array Sprite Image

function startStage

Done

Estableceremos el color del fondo del escenario y luego, sustuiremos los distintos colores por elementos del escenario. Tenemos que buscar el Sprite adecuado.

También activaremos los muros en el color negro y marrón para que el Sprite "player" no pueda atravesarlos. El amarillo, rojo y blanco se dejan sin Sprite porque más adelante ya los sustuiremos de otra manera.

function startStage

set background color to

set tile [black] to [grass] with wall

set tile [brown] to [brick] with wall

set tile [yellow] to [grey] with wall

set tile [red] to [grey] with wall

set tile [white] to [grey] with wall

set tile [cyan] to [water] with wall

set tile [orange] to [fire] with wall

CREACIÓN DE LA FUNCIÓN startTargets

Crearemos una función "startTargets" de la pestaña "Functions".

Edit Function

Add a parameter: Text, Boolean, Number, Array, Sprite, Image

function startTargets

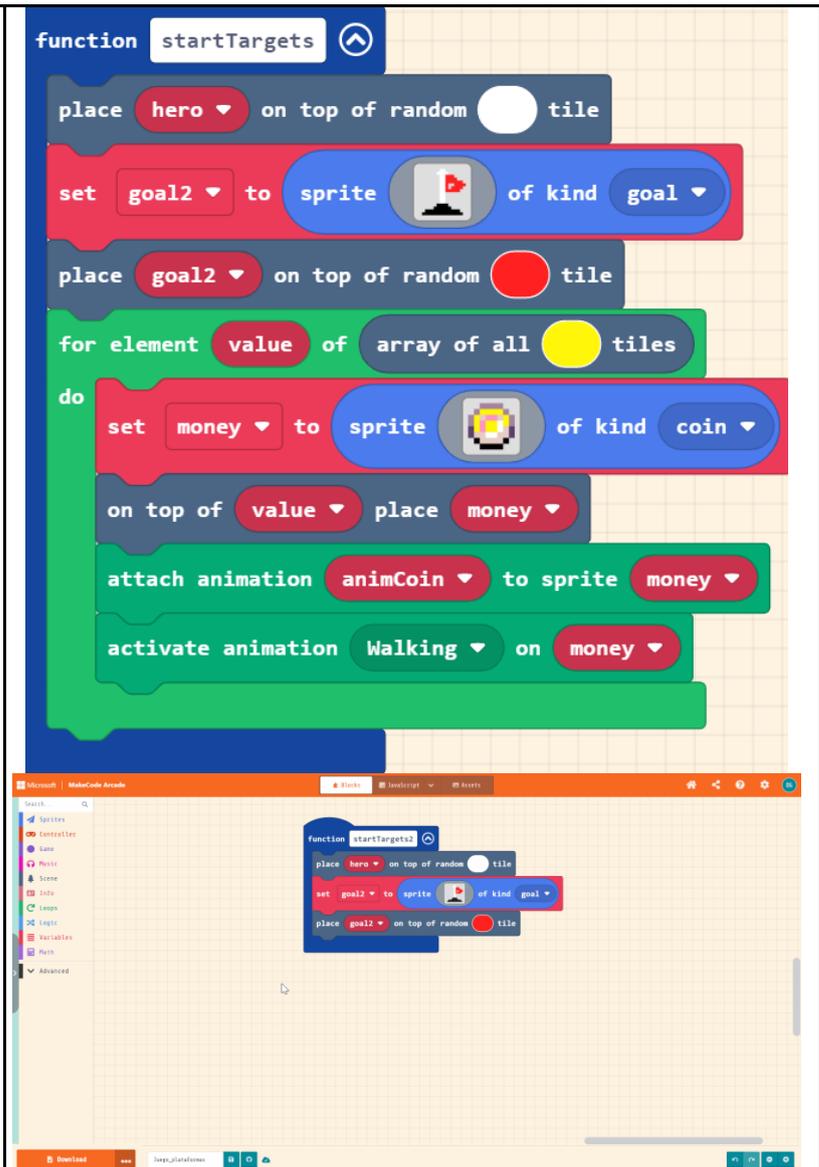
Done ✓

Dentro se coloca a "hero" en el color blanco. También crearemos el Sprite de "flag" dándole un nombre al tipo "goal".

Después en "Loops" cogemos el bloque "for element value of list" y dentro de "list" introduciremos el bloque "array of all tiles" para que en cada área amarilla realice la programación dentro del bucle.

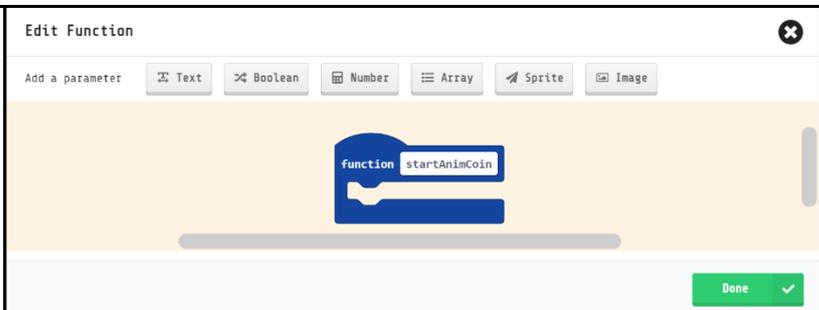
Cuando la programación detecta esta área amarilla, crea un Sprite "coin", lo sitúa en estas áreas y le da una animación.

Gif explicativo si no tenemos claro cómo introducir el bloque "array of all tiles".



CREACIÓN DE LA FUNCIÓN startAnimCoin

Crearemos una función "startAnimCoin" de la pestaña "Functions".





En esta **función** estableceremos cómo será la **animación** de las **monedas**.

```
function startAnimCoin
  set animCoin to create animation of Walking with interval 200 ms
  add frame  to animCoin
  add frame  to animCoin
  add frame  to animCoin
  add frame  to animCoin
  add frame  to animCoin
  add frame  to animCoin
```

Activamos las **funciones** llamándolas dentro del bloque **"on start"** para iniciar las programaciones realizadas anteriormente.

```
on start
  set changeLevel to 0
  set hero to sprite  of kind Player
  camera follow sprite hero
  set hero ay (acceleration y) to 300
  move hero with buttons vx 100 vy 0
  set tile map to 
  call startStage
  call startAnimCoin
  call startTargets
```

CREACIÓN MECÁNICA SALTO Y DOBLE SALTO

Programaremos que **al pulsar el botón "A"**, el personaje que controlamos realice un salto. Siempre podrá saltar en el suelo y en el aire podrá hacerlo una vez más hasta que toque el suelo.

El **control** de poder saltar lo haremos a través de la **variable jump**. Si salto es igual a 0 podremos saltar en el aire, si no lo es, no podremos saltar en el aire.

```

on A button pressed
  if is hero hitting wall bottom then
    set hero vy (velocity y) to -150
  else if jump = 0 then
    set hero vy (velocity y) to -150
    set jump to 1
  
```

Dentro del **bucle forever** estableceremos que al tocar el suelo, la **variable "jump"** se restablezca a 0 para que al volver a estar en el aire podamos hacer un solo salto.

```

forever
  if is hero hitting wall bottom then
    set jump to 0
  
```

CREACIÓN MECÁNICA PERDER LA PARTIDA

Programaremos que al **tocar el agua** perdamos la partida.

```

on sprite of kind Player overlaps [Water] at location
  game over LOSE
  
```

También al **tocar los orbes de fuego** perderemos la partida.

```

on sprite of kind Player overlaps [Fire] at location
  game over LOSE
  
```

CREACIÓN MECÁNICA SUMAR PUNTOS

Cuando el **Sprite** que controlemos **toque** a un **Sprite de tipo coin**, estas **desaparecerán**, **aumentarán la puntuación** y que generarán un **sonido de obtención de moneda**.

```

on sprite of kind Player overlaps otherSprite of kind coin
  destroy otherSprite with spray effect for 500 ms ms
  play sound ba ding
  change score by 1
  
```

CREACIÓN MECÁNICA CAMBIO DE NIVEL Y FINAL DEL JUEGO

Cuando el jugador **toque la bandera** por primera vez se hará un **cambio de nivel**.

```

on sprite of kind Player overlaps otherSprite of kind goal
  if <changeLevel = 0> then
    splash "Nivel superado"
    change changeLevel by 1
    set tile map to [img]
    call startStage
    call startAnimCoin
    call startTargets
  else
    splash "Has terminado el juego"
    game over WIN with confetti effect effect
  
```

Con esta programación usaremos al **sprite de tipo "Player"** para que supere obstáculos hasta llegar a la **meta** y consigamos **puntos** al alcanzar las distintas **monedas** que habrá repartidas en el **escenario**. Si caemos al **agua** o tocamos un **orbe rojo**, la **partida terminará**. Además, al cruzar la meta por primera vez, accederemos a otro **nivel nuevo**.

Ahora, es tu turno de personalizarlo y añadirle contenido. Aquí te dejamos el nuestro para que te inspires un poco: https://makecode.com/_2pDcsCXVHPi2