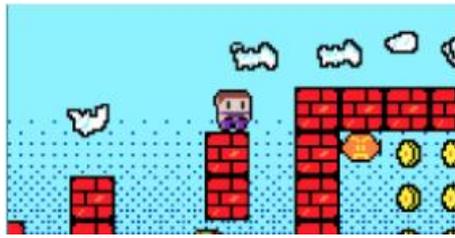


# Jumpy Plataformer

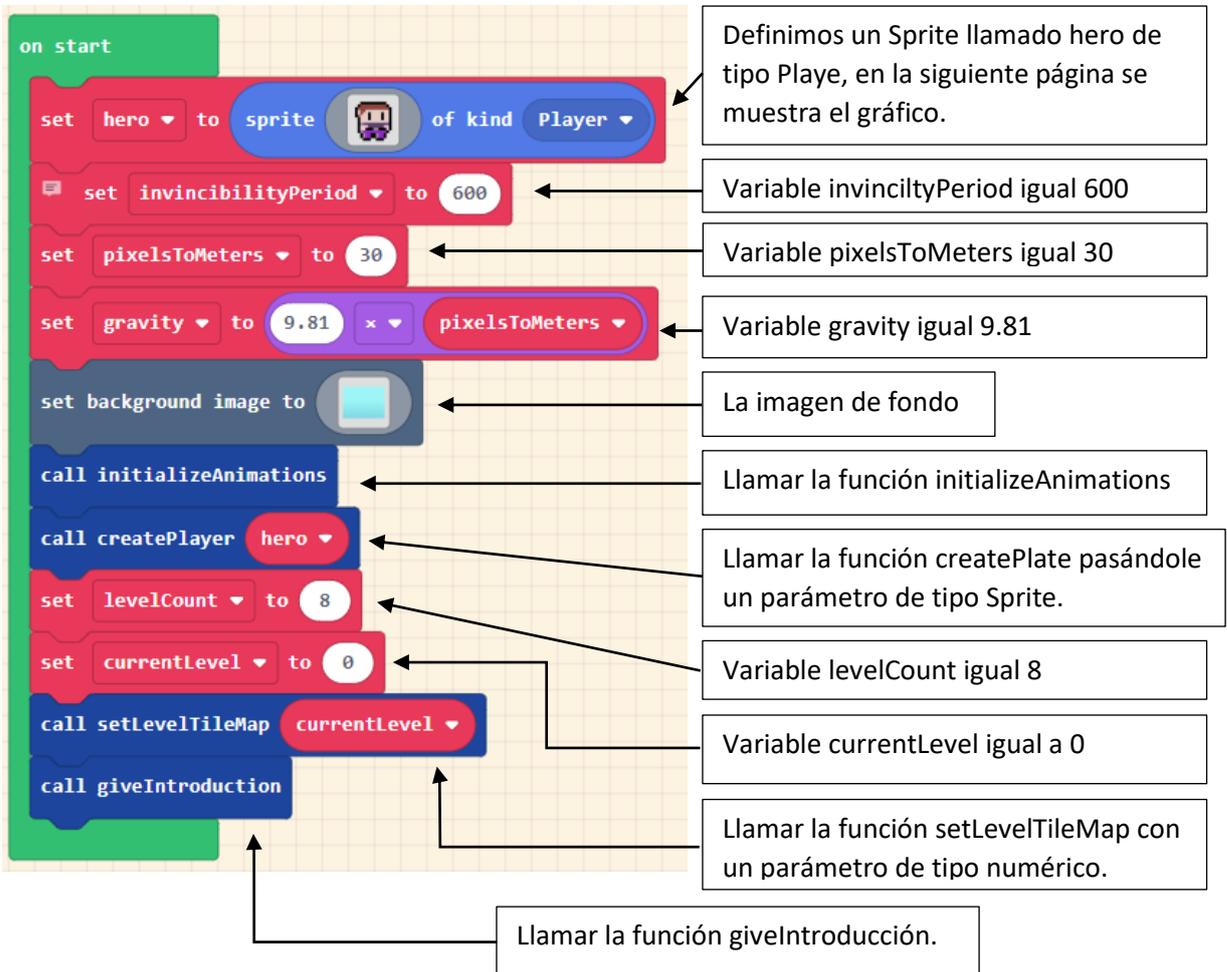


Vamos a importar la siguiente extensión:



En el grupo de bloques Animación tendremos un conjunto de bloques en el apartado de avanzado.

Cuando empieza el programa.



```
on start
  set hero to sprite of kind Player
  set invincibilityPeriod to 600
  set pixelsToMeters to 30
  set gravity to 9.81 x pixelsToMeters
  set background image to
  call initializeAnimations
  call createPlayer hero
  set levelCount to 8
  set currentLevel to 0
  call setLevelTileMap currentLevel
  call giveIntroduction
```

Definimos un Sprite llamado hero de tipo Playe, en la siguiente página se muestra el gráfico.

Variable invinciltyPeriod igual 600

Variable pixelsToMeters igual 30

Variable gravity igual 9.81

La imagen de fondo

Llamar la función initializeAnimations

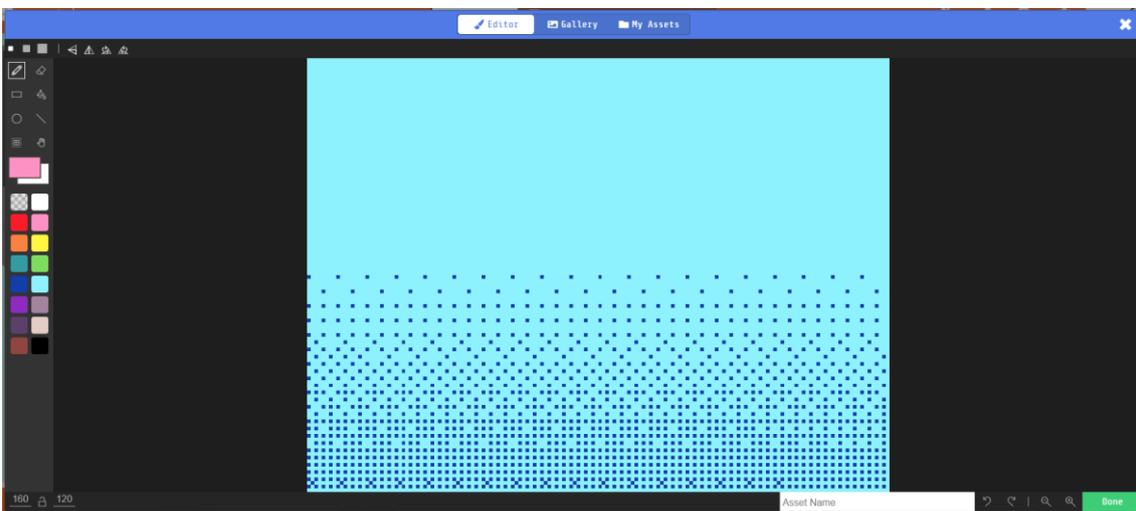
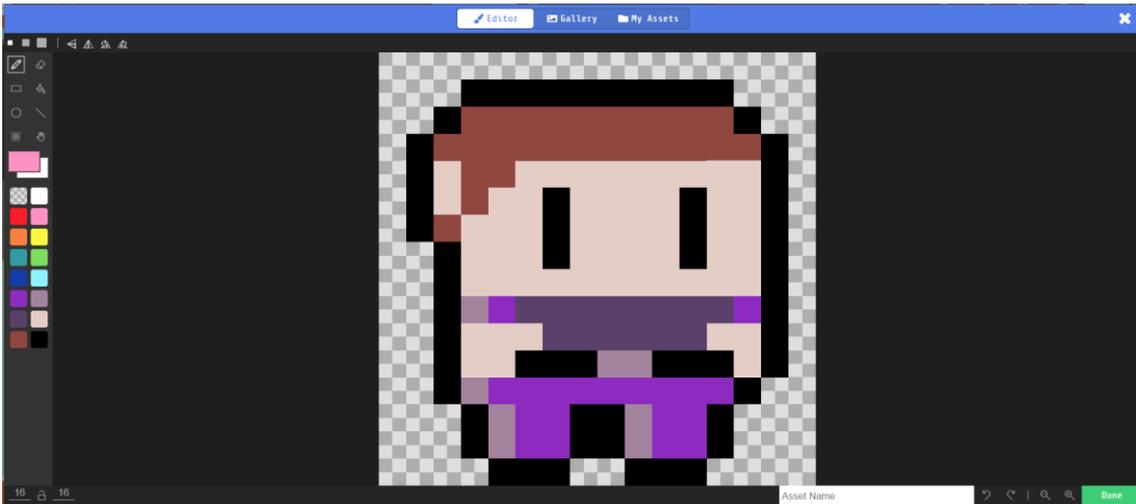
Llamar la función createPlate pasándole un parámetro de tipo Sprite.

Variable levelCount igual 8

Variable currentLevel igual a 0

Llamar la función setLevelTileMap con un parámetro de tipo numérico.

Llamar la función giveIntroducción.



Vamos por la primera función:

```
Inicializar animaciones  
function initializeAnimations  
  call initializeHeroAnimations  
  call initializeCoinAnimation  
  call initializeFlierAnimations
```

Esta función a su vez llama a tres funciones más.

Llamar la función initializeHeroAnimations

Llamar la función initializeCoinAnimation

Llamar la función initializeFlierAnimations.

```

function initializeHeroAnimations
  call animateRun
  call animateIdle
  call animateCrouch
  call animateJumps

```

Llama la función animateRun  
 Llama la función animateIdle  
 Llama la función animateCrouch  
 Llama la función animateJumps

```

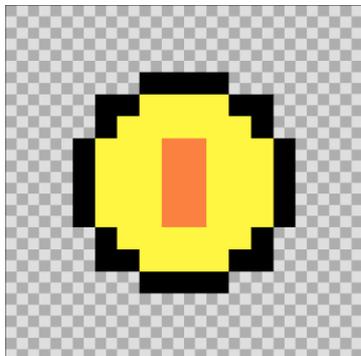
function initializeCoinAnimation
  set coinAnimation to create animation of Idle with interval 200 ms
  add frame [coin frame 1] to coinAnimation
  add frame [coin frame 2] to coinAnimation
  add frame [coin frame 3] to coinAnimation
  add frame [coin frame 4] to coinAnimation
  add frame [coin frame 5] to coinAnimation
  add frame [coin frame 6] to coinAnimation
  add frame [coin frame 7] to coinAnimation

```

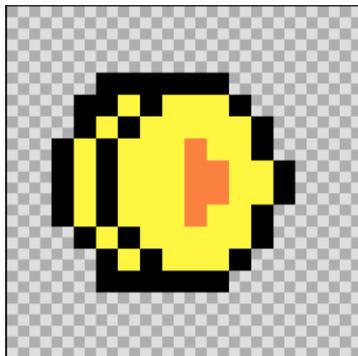
Definimos una variable de tipo animación de modo inactivo con intervalos de 200 ms.

Agregamos 7 frames de una moneda que gira.

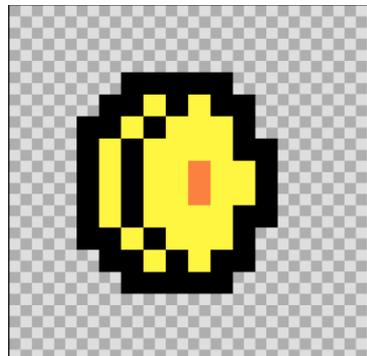
Como tienes que hacer cada imagen.



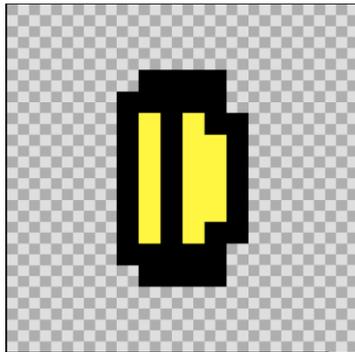
Fotograma 1



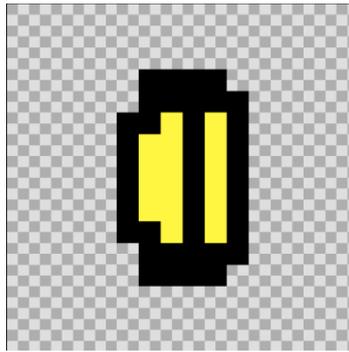
Fotograma 2



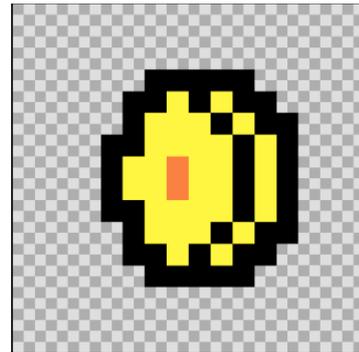
Fotograma 3



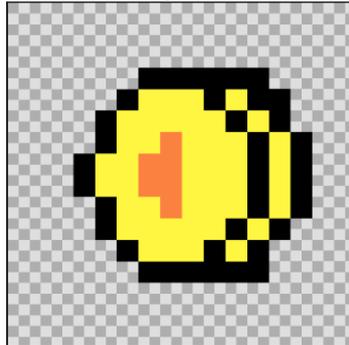
Fotograma 4



Fotograma 5



Fotograma 6



Fotograma 7

Inicializar animaciones de Flier

```
function initializeFlierAnimations
  set flierFlying to create animation of Walking with interval 100 ms
  add frame [frame] to flierFlying
  add frame [frame] to flierFlying
  add frame [frame] to flierFlying
  set flierIdle to create animation of Idle with interval 100 ms
  add frame [frame] to flierIdle
```

Definimos flierFlying como una variable de animación como Walking (caminado) con intervalos de 100 ms.

Definimos flierIdle como una variable de animación como Idle (inactiva) con intervalos de 100 ms.

Agregamos 3 frames personaje que mueve las orejas

Agregamos un fotograma.



Fotograma 1



Fotograma 2



Fotograma 3



Fotograma 4

Animación corriendo.

```
function animateRun  
  set mainRunLeft to create animation of Walking with interval 100 ms  
  attach animation mainRunLeft to sprite hero  
  add frame [hero] to mainRunLeft  
  set mainRunRight to create animation of Walking with interval 100 ms  
  attach animation mainRunRight to sprite hero  
  add frame [hero] to mainRunRight  
  add frame [hero] to mainRunRight  
  add frame [hero] to mainRunRight  
  add frame [hero] to mainRunRight
```

Definimos mainRunLeft como animación walking (caminado) con intervalos de 100

Adjunta animación mainRunLeft a él Sprite hero.

Agregamos los siguientes fotogramas.

Definimos mainRunRight como animación walking (caminado) con intervalos de 10 ms.

Adjunta animación mainRunRight a él Sprite hero.

Agregamos los siguientes fotogramas.



Animar inactivo

```

function animateIdle
  set mainIdleLeft to create animation of Walking with interval 100 ms
  attach animation mainIdleLeft to sprite hero
  add frame [hero frame] to mainIdleLeft
  set mainIdleRight to create animation of Walking with interval 100 ms
  attach animation mainIdleRight to sprite hero
  add frame [hero frame] to mainIdleRight
  
```

Definimos mainIdleLeft como animación walking (caminado) con intervalos de 100

Adjunta animación mainIdleLeft a él Sprite hero.

Agregar fotogramas

Adjunta animación mainIdleRight a él Sprite hero.

Agregar fotogramas

Definimos mainIdleRight como animación walking (caminado) con intervalos de 100



Animación agacharse

```

function animateCrouch
  set mainCrouchLeft to create animation of Walking with interval 100 ms
  attach animation mainCrouchLeft to sprite hero
  add frame [hero] to mainCrouchLeft
  set mainCrouchRight to create animation of Walking with interval 100 ms
  attach animation mainCrouchRight to sprite hero
  add frame [hero] to mainCrouchRight

```

Definimos mainCrouchLeft como animación walking (caminado) con intervalos de 100

Adjunta animación mainCrouchLeft a él Sprite hero.

Agregar fotogramas

Adjunta animación mainCrouchRight a él Sprite hero.

Agregar fotogramas

Definimos mainCrouchRight como animación walking (caminado) con intervalos de 100



Saltos animados

```

function animateJumps
  set mainJumpLeft to create animation of Walking with interval 100 ms
  attach animation mainJumpLeft to sprite hero
  add frame [hero] to mainJumpLeft
  add frame [hero] to mainJumpLeft
  repeat 30 times
  do
    add frame [hero] to mainJumpLeft
  set mainJumpRight to create animation of Walking with interval 100 ms
  attach animation mainJumpRight to sprite hero
  add frame [hero] to mainJumpRight
  add frame [hero] to mainJumpRight
  repeat 30 times
  do
    add frame [hero] to mainJumpRight
  
```

Definimos mainJumpLeft como animación walking (caminado) con intervalos de 100

Adjunta animación mainJumpLeft a él Sprite hero.

Agregar fotogramas

Repite 30 veces

Agregar fotograma

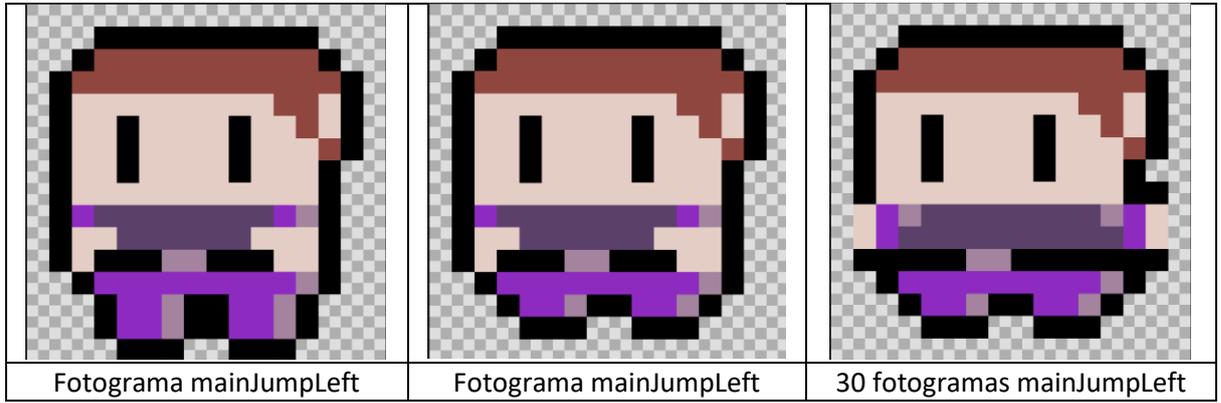
Adjunta animación mainJumpRight a él Sprite hero.

Agregar fotogramas

Repite 30 veces

Agregar fotograma

Definimos mainJumpRight como animación walking (caminado) con intervalos de 100





```

function showInstruction text
  show long text text bottom
  play sound ba ding
  change score by 1
  
```

Muestra una ventana con una mayor cantidad de texto

Se reproduce un sonido.

La variable score (puntuación) se incrementa en 1.

```

function createPlayer player2
  set player2 ay (acceleration y) to gravity
  camera follow sprite player2
  move player2 with buttons vx 100 vy 0
  set player2 z (depth) to 5
  set life to 3
  set score to 0
  
```

Al nuevo player2 le asignamos una aceleración en y del de la variable gravity.

La cámara sigue a player2.

Lo podemos mover con los botones de derecha e izquierda a una velocidad de 100.

Player 2 en z a una profundidad de 5.

La variable score (puntuación) su valor a 0.

```

function hasNextLevel
  return currentLevel <= levelCount
  
```

Esta función retorna un valor lógico, si el valor de la variable currentLevel es distinta a levelCount, en caso afirmativo será True de lo contrario será False.

### Nivel de inicialización

```

function initializeLevel level
  start screen clouds effect
  set playerStartLocation to array of all locations get value at 0
  place hero on top of playerStartLocation
  set [sprite] at playerStartLocation
  call createEnemies
  call spawnGoals
  
```

Empieza un efecto de nubes en la pantalla.

playerStartLocation se le asigna al conjunto de todas las ubicaciones a obtener el valor 0.

A la imagen se le asigna a playerStartLocation.

Se llaman a las funciones createEnemies y spawnGoals.

### Objetivo de generación

```

function spawnGoals
  for element value7 of array of all locations
  do
    set coin to sprite of kind Coin
    place coin on top of value7
    attach animation coinAnimation to sprite coin
    activate animation Idle on coin
    set [sprite] at value7
  
```

Un bucle que se repetirá por el número de monedas que localiza.

Sprite coin se le asigna la imagen de la moneda de tipo Coin.

Lugar donde hay una moneda pon encima un Sprite llamado coin.

Adjuntar animación coinAnimation para el Sprite coin.

Inactivar la animación del Sprite coin.

Se le asigna el valor de la variable value7

### Dar instrucción

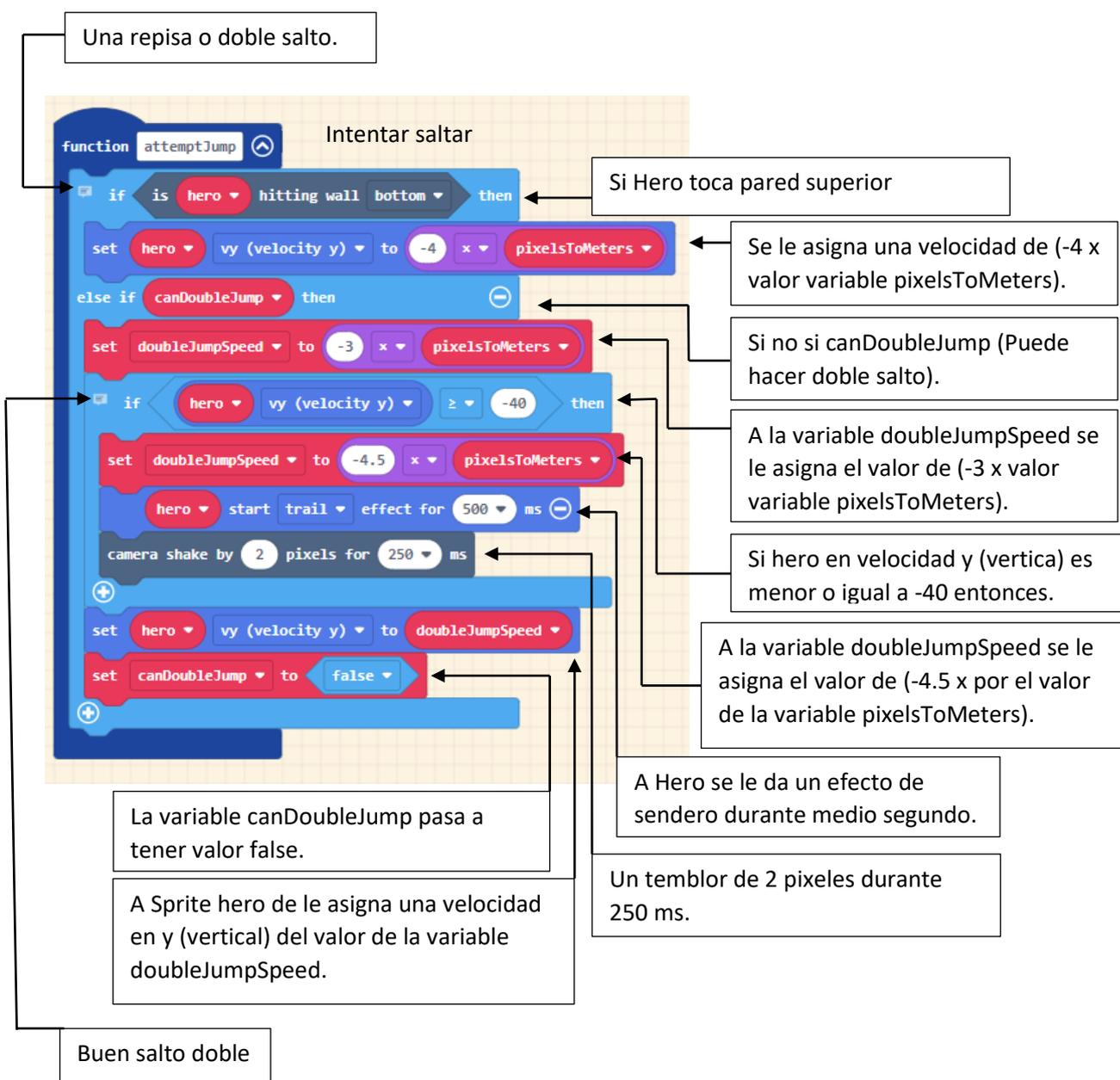
```

function giveIntroduction
  set dialog frame to [frame]
  set dialog cursor to [coin]
  call showInstruction "Muévete con los botones izquierdo y derecho."
  call showInstruction "Salta con el botón A hacia arriba."
  call showInstruction "Doble salto presionando saltar de nuevo."
  
```

Agrega un marco al mensaje.

El cursor lo simboliza con una moneda.

Los mensajes largos.



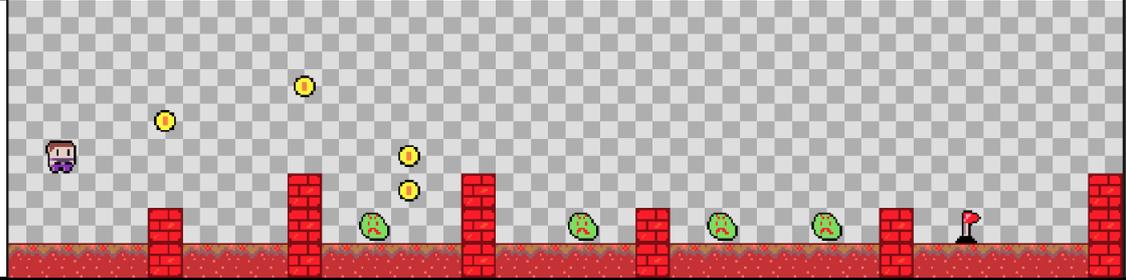
### Establecer nivel de mapa de azulejos

```
function setLevelTileMap level
  call clearGame
  if level = 0 then
    set tilemap to [tilemap_0]
  else if level = 1 then
    set tilemap to [tilemap_1]
  else if level = 2 then
    set tilemap to [tilemap_2]
  else if level = 3 then
    set tilemap to [tilemap_3]
  else if level = 4 then
    set tilemap to [tilemap_4]
  else if level = 5 then
    set tilemap to [tilemap_5]
  else if level = 6 then
    set tilemap to [tilemap_6]
  else if level = 7 then
    set tilemap to [tilemap_7]
  call initializeLevel level
```

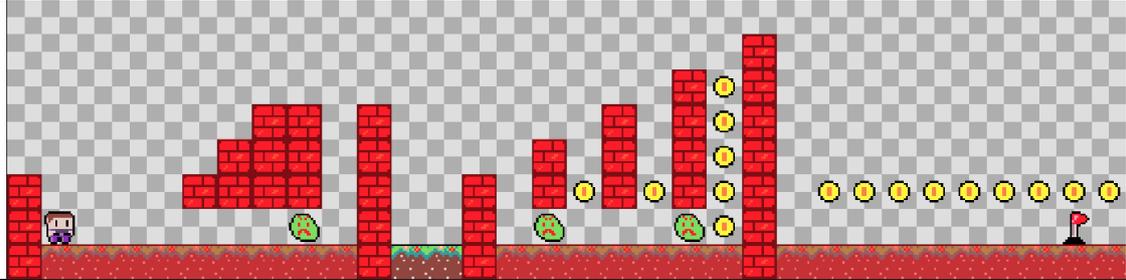
Según el valor de la variable level que puede tener un valor desde 0 hasta 7, este es el fondo que veremos en el juego, en la siguiente página se detallan los fondos.

Por último se llama a la función initializeLevel con el parámetro del valor de la variable level.

Nivel 0



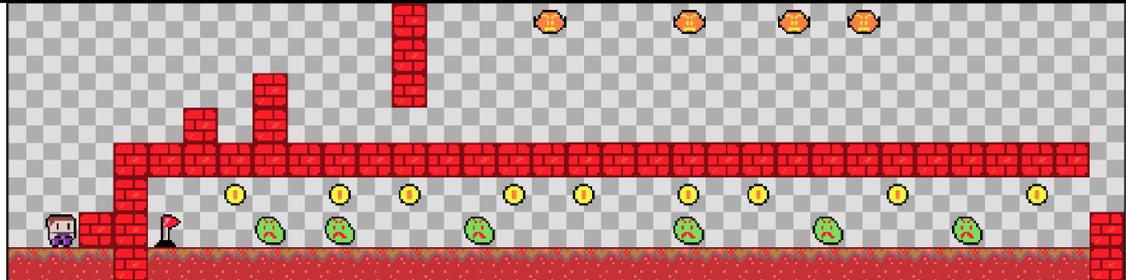
Nivel 1



Nivel 2

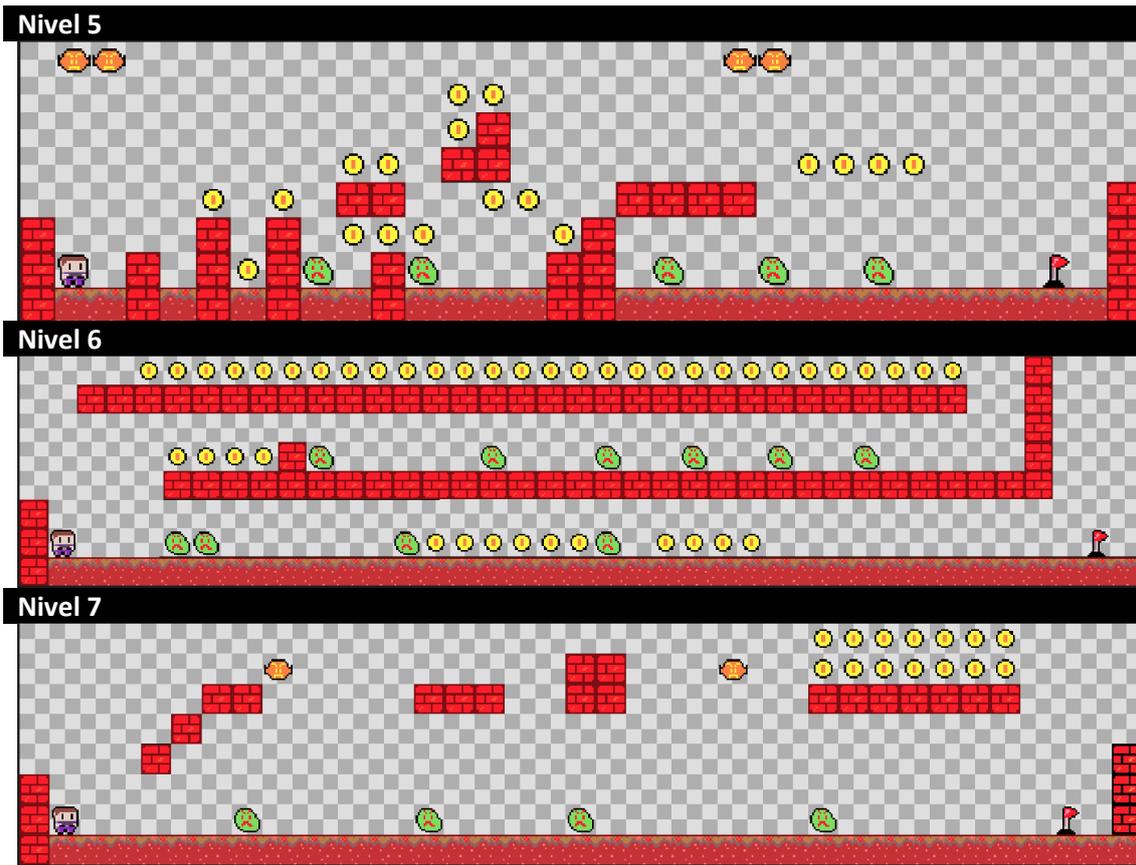


Nivel 3



Nivel 4





function clearGame

```

for element value of array of sprites of kind Bumper
do
  destroy value
for element value2 of array of sprites of kind Coin
do
  destroy value2
for element value3 of array of sprites of kind Goal
do
  destroy value3
for element value4 of array of sprites of kind Flier
do
  destroy value4

```

Limpiar juego

Un ciclo for donde hará un recorrido por todos los sprites de tipo Bumper.

Y destruye cada uno de los sprites.

Esto lo repetimos con el resto de sprites de tipo Coin, Goal y Flier.

Enemigo que se mueve de un lado a otro.

```
function createEnemies
  for element value5 of array of all locations
  do
    set bumper to sprite of kind Bumper
    place bumper on top of value5
    set [ ] at value5
    set bumper ay (acceleration y) to gravity
    if 50 % chance then
      set bumper vx (velocity x) to pick random 30 to 60
    else
      set bumper vx (velocity x) to pick random -60 to -30

  for element value6 of array of all locations
  do
    set flier to sprite of kind Flier
    place flier on top of value6
    set [ ] at value6
    attach animation flierFlying to sprite flier
    attach animation flierIdle to sprite flier
```

Un ciclo for de Sprite del conjunto de todos en su localización.

Crea un Sprite llamado bumper de tipo Bumper.

Lugar donde está bumper.

Coloca al Sprite.

A bumper le damos una aceleración en y según valor variable gravity.

(\*) Si (el porcentaje es el 50%) entonces, retorna un valor booleano, Verdadero o Falso.

(\*\*) A la variable bumper en el parámetro velocity en x (vertical) se le asigna un valor aleatorio desde 30 hasta 60, si no un valor aleatorio desde -60 hasta -30.

Hacemos un ciclo for para recorrer todos los elementos según imagen.

Crea un Sprite con el nombre flier de tipo Flier.

Lugar donde está flier.

Coloca el Sprite.

Adjunta animación a flierIdle de tipo flier.

Adjunta animación a flierFlying de tipo flier.

Cuando se está jugando.

The image shows a Scratch script for the 'on game update' event. The code is annotated with Spanish text boxes explaining each block's function. The script is as follows:

```
on game update
  if (hero velocity x < 0) then
    set heroFacingLeft to true
  else if (hero velocity x > 0) then
    set heroFacingLeft to false
  if (is hero hitting wall top) then
    set hero velocity y to 0
  if (is down button pressed) then
    if (heroFacingLeft) then
      activate animation Walking on hero
    else
      activate animation Walking on hero
  else if (hero velocity y < 20 and not is hero hitting wall bottom) then
    if (heroFacingLeft) then
      activate animation Walking on hero
    else
      activate animation Walking on hero
  else if (hero velocity x < 0) then
    activate animation Walking on hero
  else if (hero velocity x > 0) then
    activate animation Walking on hero
  else
    if (heroFacingLeft) then
      activate animation Walking on hero
    else
      activate animation Walking on hero
```

Annotations (from top to bottom):

- Compara si la velocidad de hero en la coordenada x es menor a 0.
- Si se cumple asignamos a la variable heroFacingLeft al valor true (es boolean).
- De lo contrario le asignamos a la variable heroFacingLeft el valor False.
- Si hero está golpeando la parte superior de la pared.
- La velocidad de hero en la coordenada y será 0.
- Si presionamos el botón inferior entonces.
- Si heroFacingLeft es true.
- Activamos la animación caminando de hero.
- Si no sí la velocidad de hero con la coordenada y es menor de 20 y además no está golpeando el fondo de la pared.
- Activamos la animación caminando de hero.
- Si no si velocidad en x de hero menor de 0.
- Si no si velocidad en x de hero mayor de 0.
- Activamos la animación caminando de hero.
- Si no
- Si heroFacingLeft es igual a True entonces.
- Si no
- Activamos la animación caminando de hero.

Cuando un Sprite de tipo Player toca a otro Sprite de tipo Bumper.

Si Sprite velocidad en y (vertical) es mayor de 0, y (no Sprite está golpeando el fondo de la pared o Sprite en la coordenada y es menor a la parte superior del otro Sprite), entonces.

The image shows a Scratch script for a collision event between a Player and a Bumper. The script is as follows:

```
on sprite of kind Player overlaps otherSprite of kind Bumper
  if (sprite vy (velocity y) > 0 and not (is sprite hitting wall bottom or (sprite y < otherSprite top))) then
    destroy otherSprite with ashes effect for 250 ms
    set otherSprite vy (velocity y) to -50
    set sprite vy (velocity y) to -2 * pixelsToMeters
    change score by 1
    play sound power up
  else
    change life by -1
    sprite say "Ow!" for invincibilityPeriod ms
    play sound power down
  pause invincibilityPeriod ms
```

Annotations explaining the script blocks:

- Destroy otherSprite with ashes effect for 250 ms:** Destruye el otro Sprite con un efecto de despojos mortales por una durante 250 ms.
- set otherSprite vy (velocity y) to -50:** A los otros sprites de tipo Bumper se le asigna una velocidad de -50 (vertical) coord. y
- set sprite vy (velocity y) to -2 \* pixelsToMeters:** Al Sprite coordenada y (vertical) en velocidad se le asigna el valor (-2 x el valor de la variable pixelesToMeters).
- change score by 1:** A la variable score se le incrementa en 1. (Puntuación)
- play sound power up:** Se reproduce un sonido.
- else:** Si no
- change life by -1:** A la variable score se le decrementa en 1. (Puntuación)
- sprite say "Ow!" for invincibilityPeriod ms:** Spray dice "Ow!" para el tiempo que tiene la variable invincibilityPeriod en milisegundos.
- play sound power down:** Se reproduce un sonido.
- pause invincibilityPeriod ms:** Una pausa por el tiempo de la variable invincibilityPeriod en milisegundos.

Durante el juego

Un bucle según los elementos que tiene la array de sprites de tipo Flier.

Si el valor absoluto del Sprite en la coordenada x menos el valor de hero en la coordenada x es menor de 60 entonces.

Si el valor del Sprite en la coordenada x menos el valor de hero en la coordenada x es menor de -5 entonces.

Al Sprite (velocidad x) igual a 25

Si no si Sprite posición x menos hero posición x sea mayor a 5 entonces.

Al Sprite (velocidad x) igual a -25

Si Sprite en y – hero en y es menor a -5 entonces.

Activa la animación caminando del Sprite.

Si no

Al Sprite velocidad y -20, velocidad x 0.

Desactivamos la animación del Sprite.

Durante el juego

Si hero golpea con la parte superior en la pared.

La variable canDoubleJump que es de tipo booleano se le asigna el valor True.

Durante el juego

Un bucle for con el número de elementos que tiene el Sprite de tipo Bumper.

Si el Sprite está golpeando la pared por la izquierda entonces.

Al Sprite en velocidad x le damos un valor aleatorio desde 30 hasta -60.

Si el Sprite está golpeando la pared por la derecha entonces.

Al Sprite en velocidad x le damos un valor aleatorio desde -60 hasta -30.

```
on game update
  for element value9 of array of sprites of kind Bumper
  do
    if is value9 hitting wall left
    then
      set value9 vx (velocity x) to pick random 30 to 60
    else if is value9 hitting wall right
    then
      set value9 vx (velocity x) to pick random -60 to -30
```

Cuando un Sprite de tipo Player toca a otro Sprite de tipo Flier.

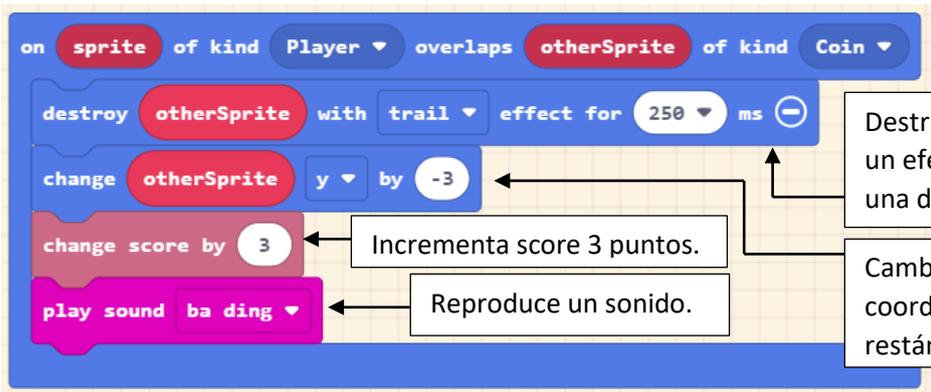
Ha la variable life (vidas) se le resta 1.

Sprite dice "Ow!" por un tiempo de (valor variable invicibilityPerido por 1.5 ms.

Pausa por el mismo tiempo.

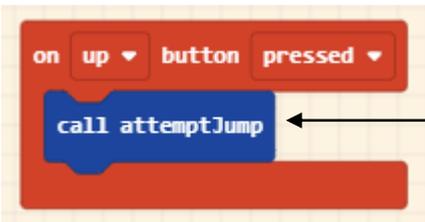
```
on sprite of kind Player overlaps otherSprite of kind Flier
  change life by -1
  sprite say "Ow!" for invicibilityPerido x 1.5 ms
  play sound power down
  pause invicibilityPerido x 1.5 ms
```

Cuando Sprite de tipo Player toca a otro Sprite de tipo Coin.



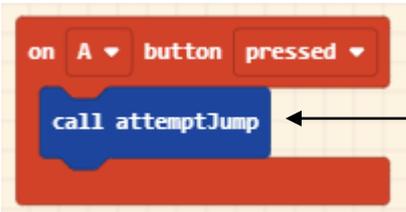
The image shows a Scratch script block starting with 'on sprite of kind Player overlaps otherSprite of kind Coin'. The script contains four sub-blocks: 'destroy otherSprite with trail effect for 250 ms', 'change otherSprite y by -3', 'change score by 3', and 'play sound ba ding'. Three callout boxes with arrows point to these blocks: 'Destruye el otro Sprite con un efecto de sendero por una duración de 250 ms.' points to the destroy block; 'Incrementa score 3 puntos.' points to the change score block; 'Reproduce un sonido.' points to the play sound block. A fourth callout box, 'Cambia el valor de la coordenada y del otro Sprite restándole -3.', has arrows pointing to both the 'change otherSprite y by -3' and 'destroy otherSprite' blocks.

Cuando se presiona el botón Arriba.



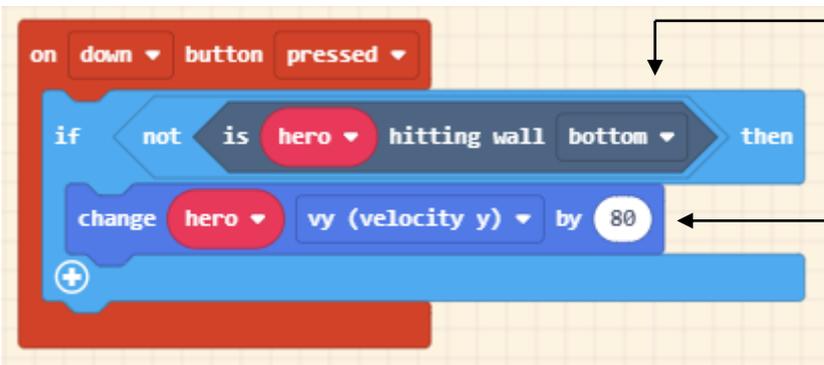
The image shows a Scratch script block starting with 'on up button pressed'. It contains one sub-block: 'call attemptJump'. A callout box with an arrow points to this block with the text 'Llama a la función attemptJump.'

Cuando se presiona al botón A.



The image shows a Scratch script block starting with 'on A button pressed'. It contains one sub-block: 'call attemptJump'. A callout box with an arrow points to this block with the text 'Llama a la función attemptJump.'

Cuando se presiona el botón abajo.



The image shows a Scratch script block starting with 'on down button pressed'. It contains an 'if' block with the condition 'not is hero hitting wall bottom then'. Inside the 'if' block is a 'change hero vy (velocity y) by 80' block. Two callout boxes with arrows point to the script: 'Si (hero con la parte superiores no está tocando la pared) entonces.' points to the 'if' block; 'Incrementa la velocidad de hero en y en 80.' points to the 'change hero vy' block.

Cuando Sprite de tipo Player toca la banderita (final del recorrido)

