

## Bunny Hop

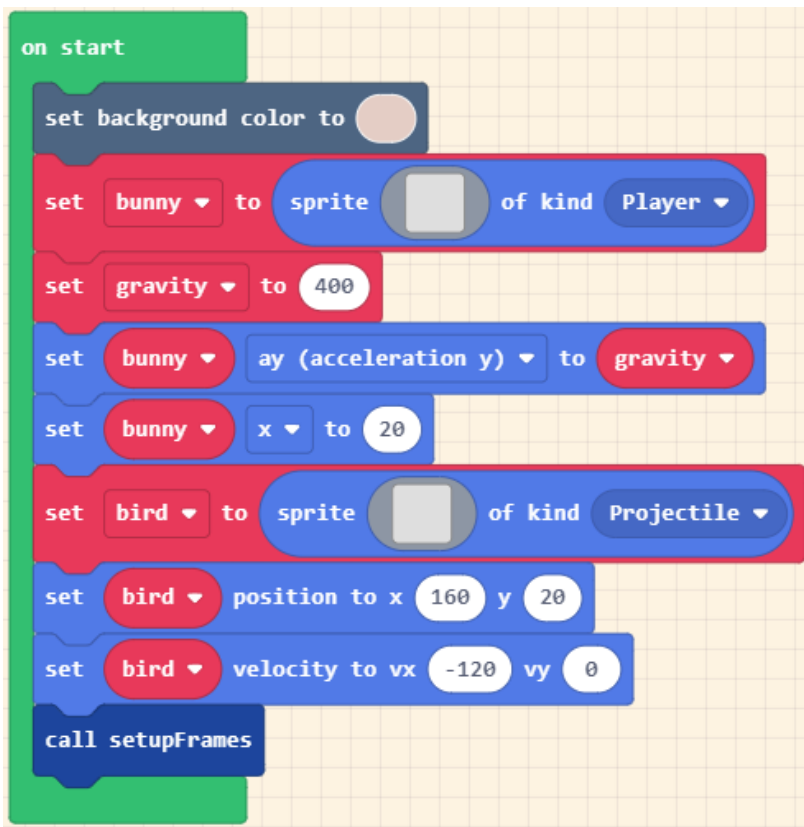


Ayuda a tu conejito a atravesar el bosque saltando por encima de los árboles y los tocones.

Para este proyecto necesitamos la extensión:



Vamos a empezar por bloque on start.



Agregamos un fondo.

Definimos un Sprite llamado bunny de tipo Player.

No hemos dibujado nada, pero tiene unas dimensiones de 16 x 16.

Definimos una variable llamada gravity con un valor de 400.

Al Sprite bunny le damos una aceleración en y según valor gravity.

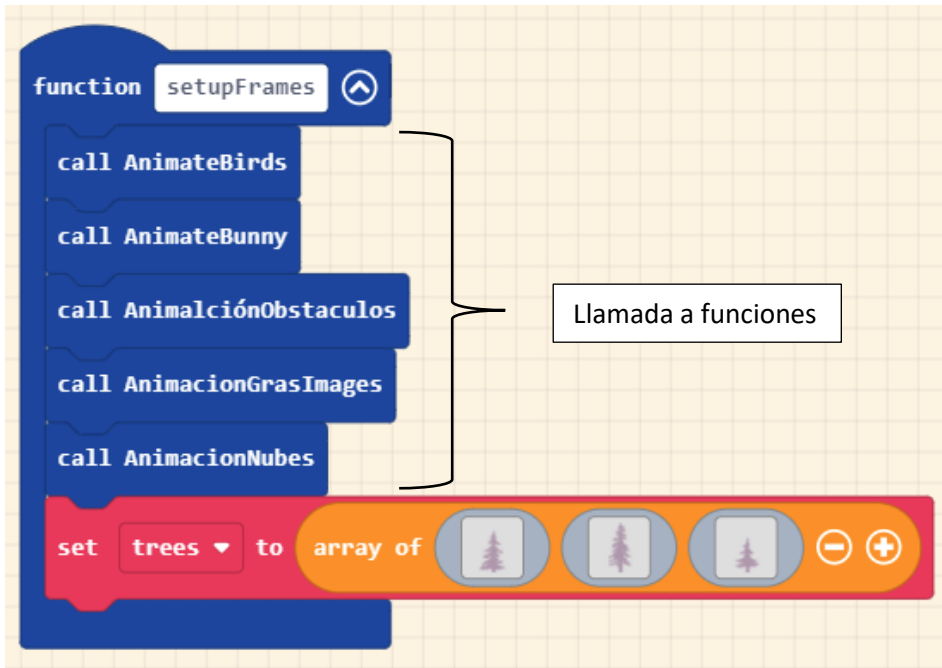
En la coordenada x lo colocamos en la posición 20.

Definimos otro Sprite llamado bird, sin imagen de tipo Projectile.

Sus dimensiones 16 x 16.

El Sprite bird lo situamos en las coordenadas x: 160, y:20.

Llamamos a la función setupFrames.

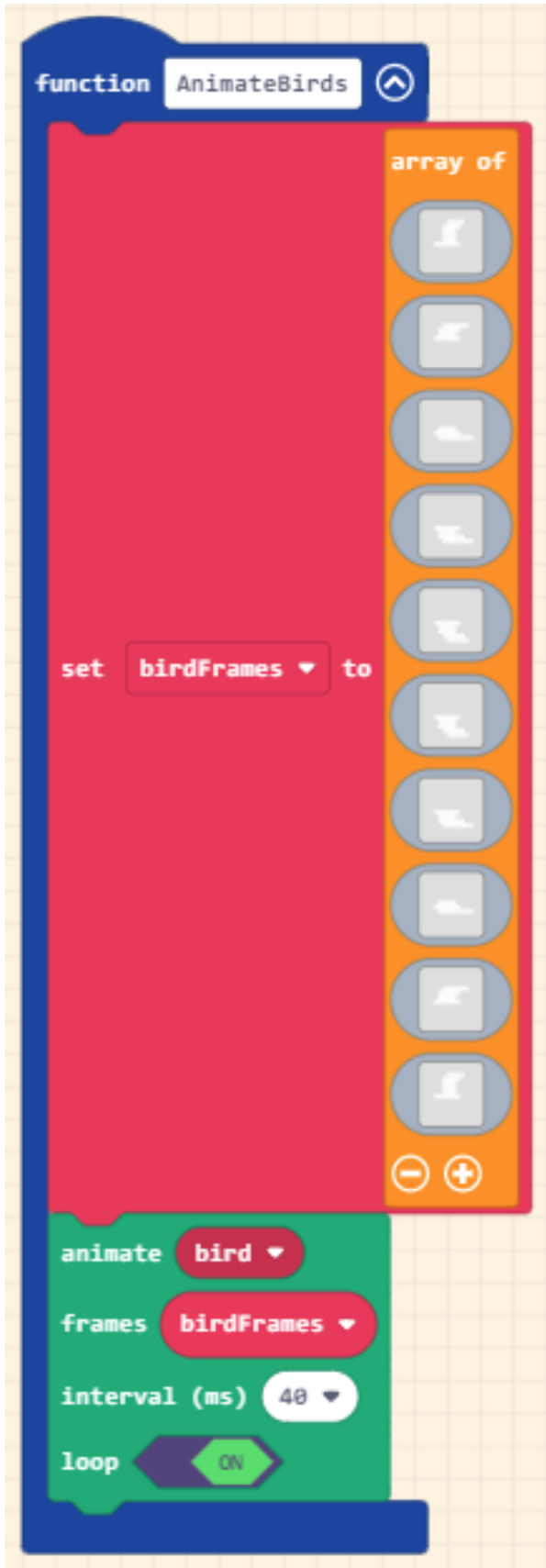


Esta a su vez llama a otras funciones, que detallaremos a continuación,

Por último se crear una variable que contiene tres imágenes.



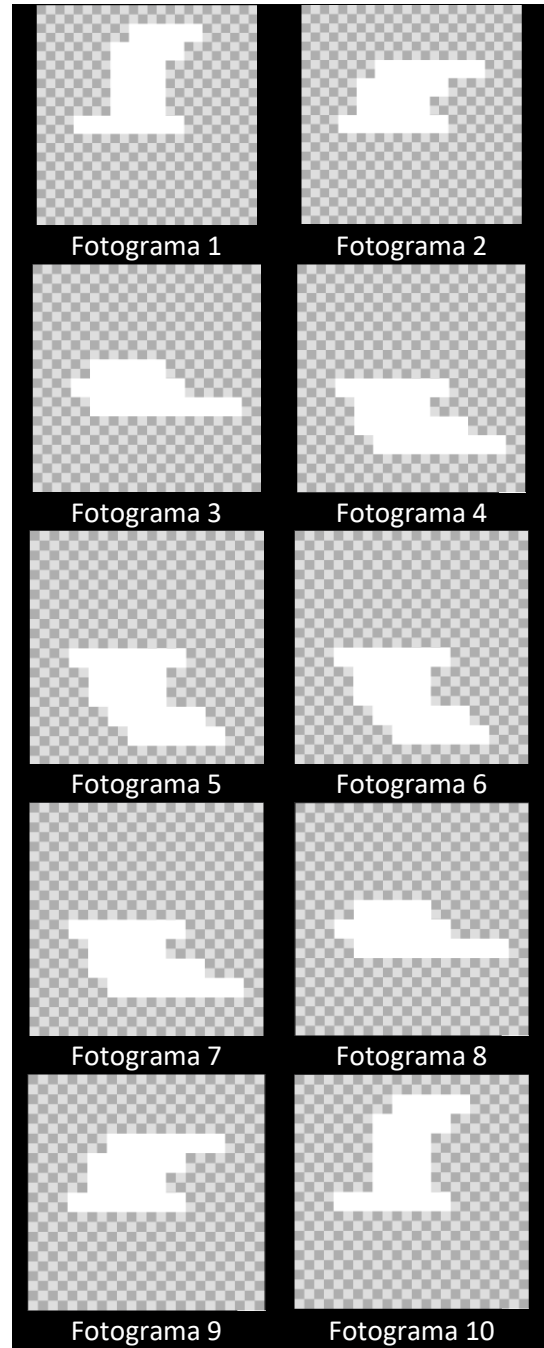
Las dimensiones son 36 x 48.



Consta de 10 fotogramas.

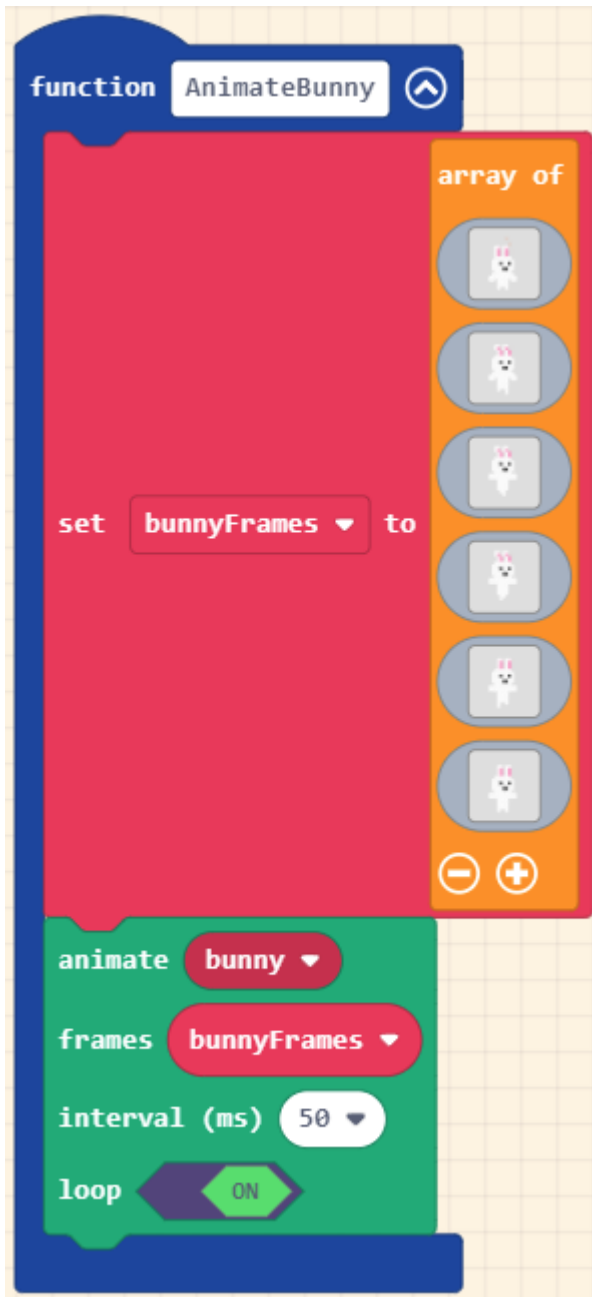
Disposición de los fotogramas.

Las dimensiones son 12 x 12.



Animamos con la variable bird los fotogramas que tenemos con la arrat birdFrames a un intervalo de 40 ms.

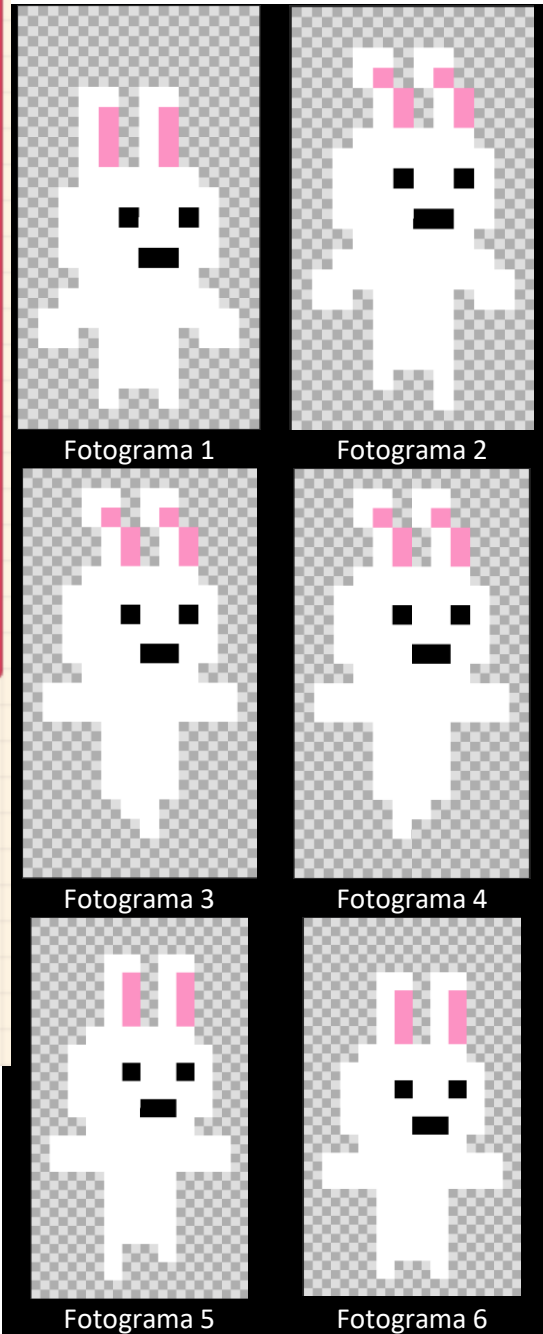
Activamos la opción bucle.



Consta de 6 fotogramas.

Disposición de los fotogramas:

Las dimensiones son de 12 x 21.



Llamaremos a la animación bunny con los fotogramas bunnyFrames, a intervalos de 50 ms. Y activamos la opción bucle.


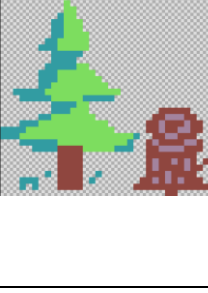


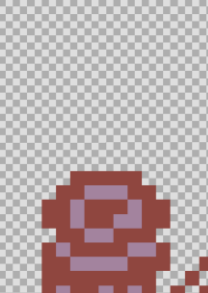
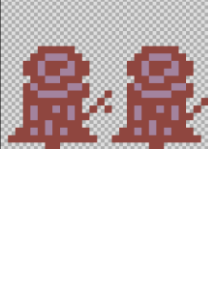
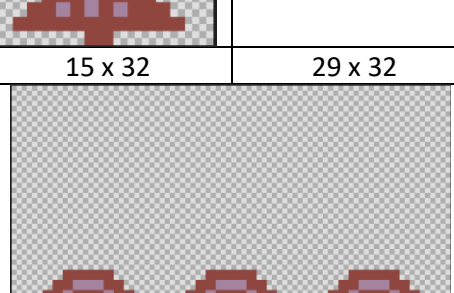
function AnimalciónObstaculos

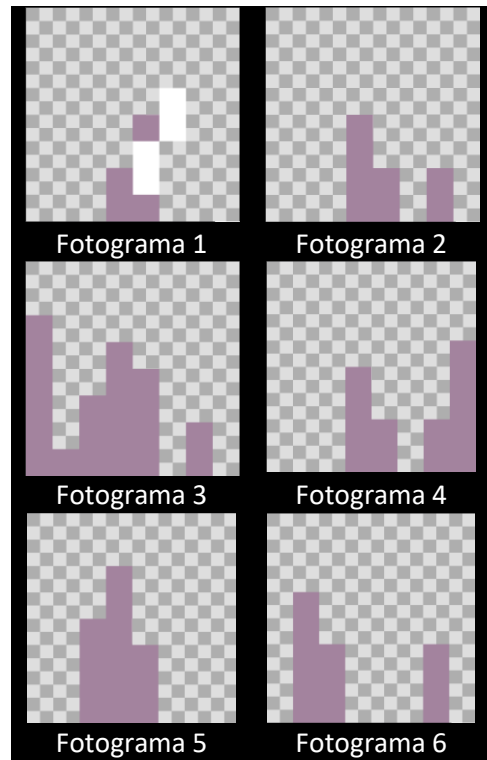
set obstacles to

array of

- [Tree]
- [Tree + Monkey]
- [Monkey + Tree]
- [Tree + Monkey]
- [Monkey]
- [Monkey + Monkey]
- [Monkey + Monkey + Monkey]

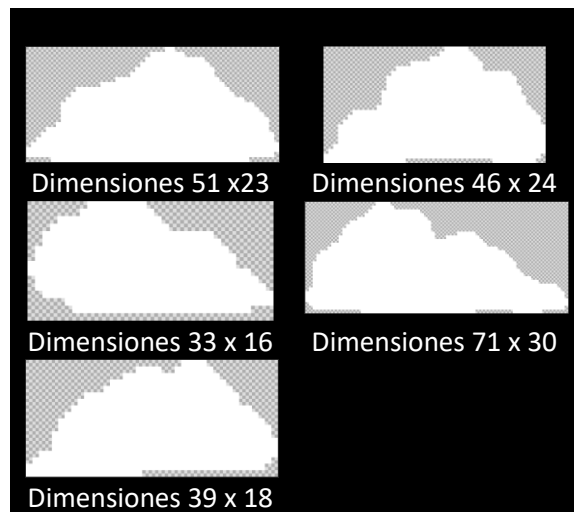
[-] [+]

	
22 x 32	35 x 32
	
38 x 32	52 x 32
	
15 x 32	29 x 32
	
43 x 32	

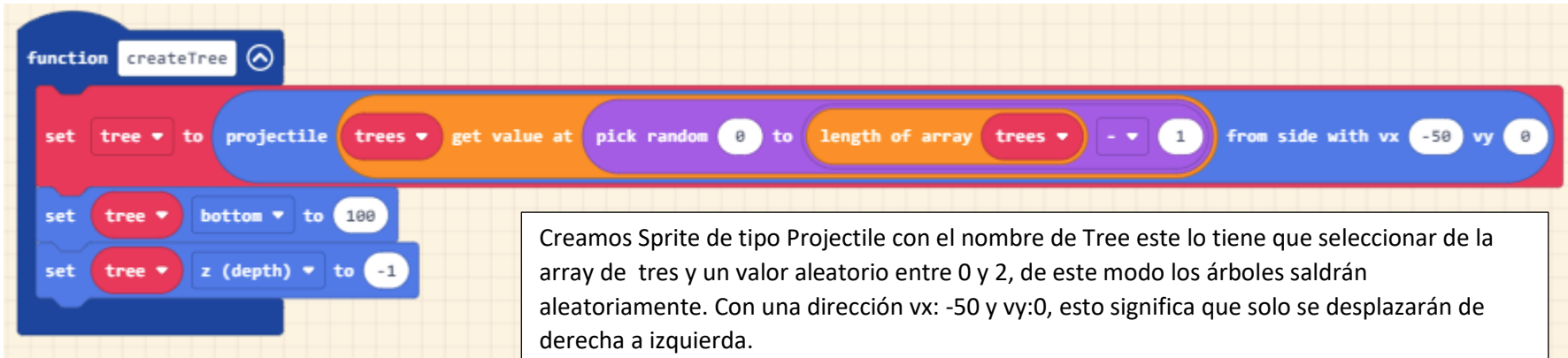


Dimensiones 8 x 8.

Son como cactus que se ven den el fondo.



La función createTree (Crea los árboles)



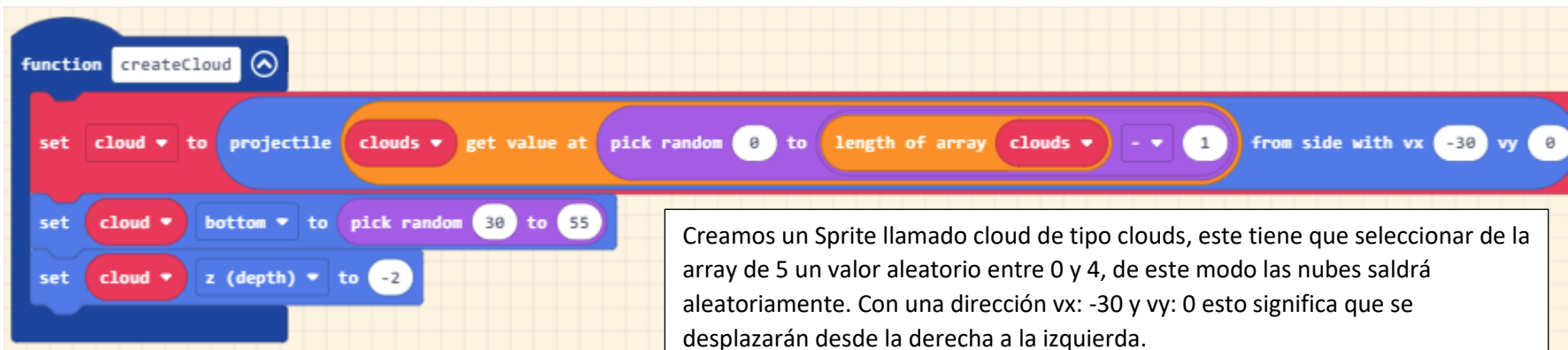
```
function createTree
  set tree to projectile trees get value at pick random 0 to length of array trees - 1 from side with vx -50 vy 0
  set tree bottom to 100
  set tree z (depth) to -1
```

Creamos Sprite de tipo Projectile con el nombre de Tree este lo tiene que seleccionar de la array de tres y un valor aleatorio entre 0 y 2, de este modo los árboles saldrán aleatoriamente. Con una dirección vx: -50 y vy:0, esto significa que solo se desplazarán de derecha a izquierda.

Por la parte superior se ubicará en la coordenada 100.

Con la coordenada z, dirección en profundidad -1.

La función createCloud (Crear nubes)



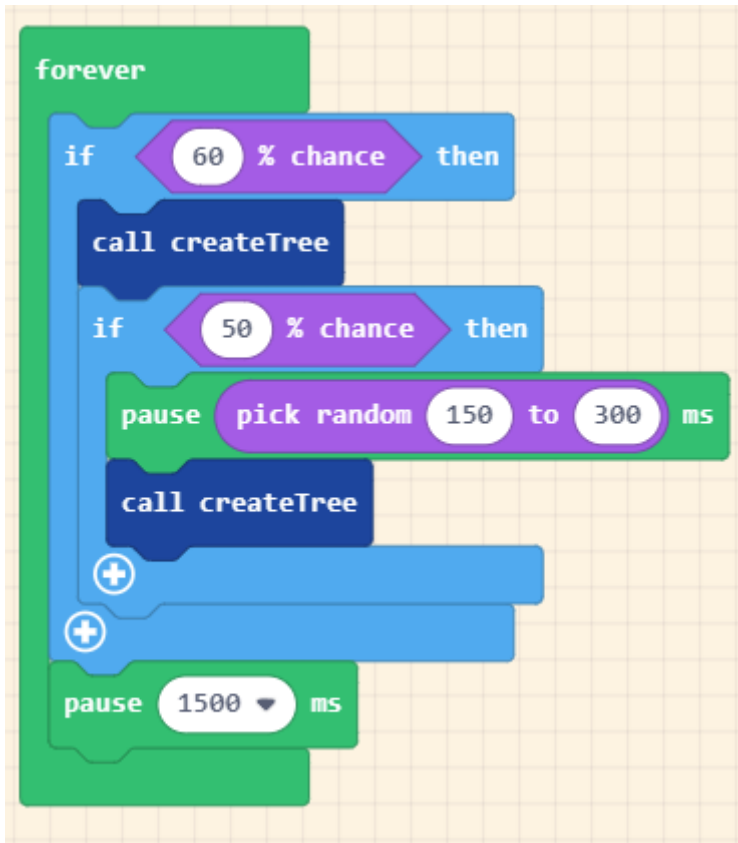
```
function createCloud
  set cloud to projectile clouds get value at pick random 0 to length of array clouds - 1 from side with vx -30 vy 0
  set cloud bottom to pick random 30 to 55
  set cloud z (depth) to -2
```

Creamos un Sprite llamado cloud de tipo clouds, este tiene que seleccionar de la array de 5 un valor aleatorio entre 0 y 4, de este modo las nubes saldrá aleatoriamente. Con una dirección vx: -30 y vy: 0 esto significa que se desplazarán desde la derecha a la izquierda.

Las nubes saldrán en su parte superior un valor aleatorio desde 30 hasta 55.

Con la coordenada z, dirección en profundidad -2.

Para siempre



Si con un porcentaje del 60% esto significa que si salen valores aleatorios hay un 60% de probabilidades de que salga True.

Llama función createTree.

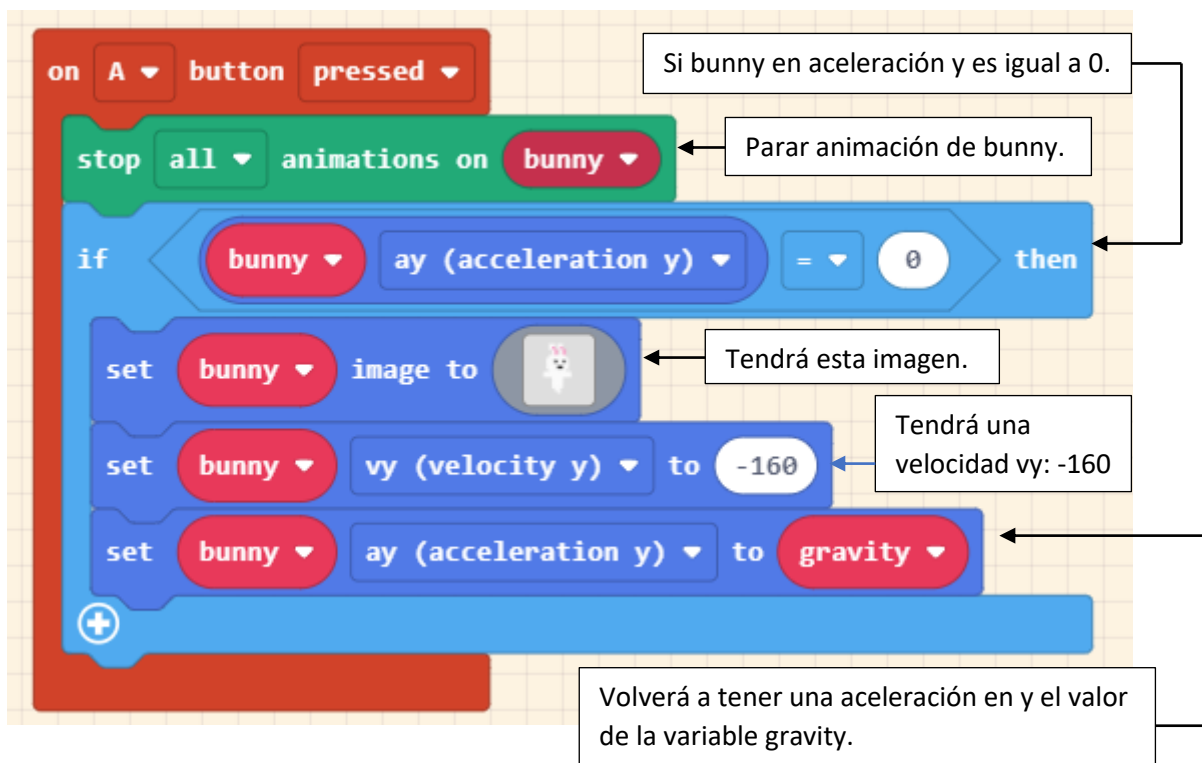
Si con un porcentaje del 50%

Una pausa con una valor aleatorio desde 150 a 300 ms.

Llama función createTree.

Pausa de 1500 ms.

Cuando se presiona el botón A.





Durante el juego cada 2000 milisegundos.

The code block consists of the following steps:

- on game update every 2000 ms**
- set obstacleImage to obstacles get value at pick random 0 to length of array obstacles - 1**
- set projectile to projectile obstacleImage from side with vx -100 vy 0**
- set projectile bottom to 105**
- set projectile kind to Obstacle**

Crea un Sprite de tipo obstacle de la array obstacle con un valor aleatorio entre 0 y 6 ya que estos son los elementos que tiene la array y cuyo nombre es obstacleImage. Este valor se lo asignamos al Sprite projectile de tipo projectile de obstacleImage con una velocidad de -100 en vx, esto significa que se desplazará desde derecha a izquierda. Con referencia a la partes superior del Sprite una posición de 105.

Este Sprite será de tipo Obstacle.

Durante el juego cada 200 milisegundos.

The code block consists of the following steps:

- on game update every 200 ms**
- if 40 % chance then**
- set grass to projectile grassImages get value at pick random 0 to length of array grassImages - 1 from side with vx -50 vy 0**
- set grass bottom to 100**
- set grass z (depth) to -1**

Si 40% de oportunidad entonces.

Definimos un Sprite de nombre Grass que le asignamos el valor que tiene Sprite grassImages con un valor de 0 a 5, ya que dispone de 6 elementos, con una velocidad en vx de -50 y vy de 0, se desplazará horizontalmente de derecha a izquierda.

Se colocará a una altura de 100 teniendo como referencia la parte superior del Sprite.

A demás a una profundidad de -1

Durante el juego.

```
on game update
  if bunny bottom > 105 then
    set bunny bottom to 105
    set bunny vy (velocity y) to 0
    set bunny ay (acceleration y) to 0
    animate bunny
      frames bunnyFrames
      interval (ms) 50
      loop ON
```

Si la parte superior de bunny es mayor de 105 entonces.

Que se quede en la posición 105.

Una velocidad en vy de 0.

Una aceleración ay en 0.

Que tenga activada la animación.

Durante el juego.

```
on game update
  if bird x < 0 then
    set bird position to x pick random 160 to 240 y pick random 20 to 60
```

Si bird de x es menor de 0 entonces.

Lo colocaremos en una posición con x valor aleatorio desde 160 hasta 240 e y un valor aleatorio desde 20 hasta 60.

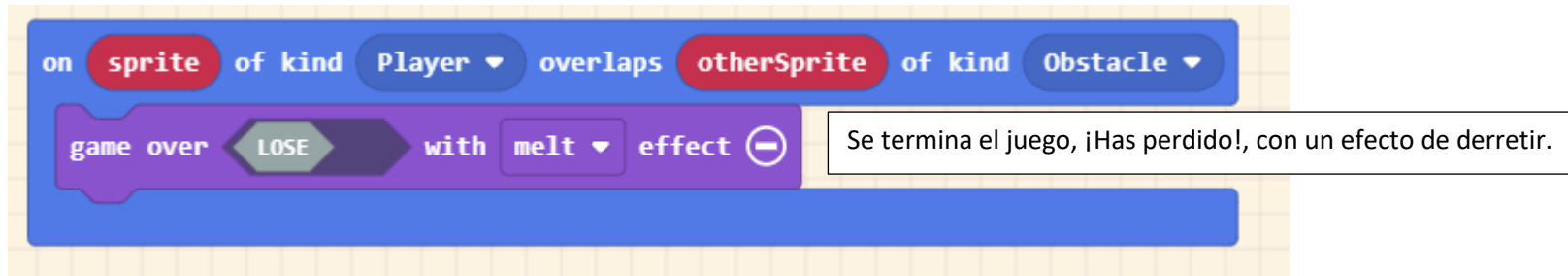
```
on game update every 1000 ms
  if 40 % chance then
    call createCloud
```

Durante el juego cada 1 segundo.

Si 40% de probabilidades que salga true.

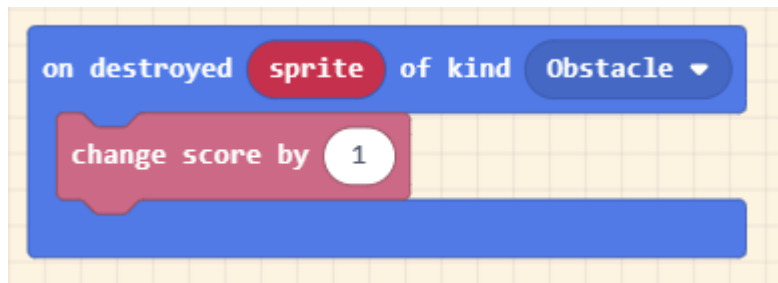
Llama a la función createCloud.

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



A Scratch script block with a blue background. The top part is a blue 'when green flag clicked' block containing the text 'on sprite of kind Player overlaps otherSprite of kind Obstacle'. Attached to its bottom is a purple 'say' block with the text 'game over' and a speech bubble containing 'LOSE'. To the right of the 'say' block is a 'with melt effect' block with a minus sign icon. A white text box with a black border is positioned to the right of the script, containing the text 'Se termina el juego, ¡Has perdido!, con un efecto de derretir.'

Cuando se destruye un Sprite de tipo Obstacle.



A Scratch script block with a blue background. The top part is a blue 'when green flag clicked' block containing the text 'on destroyed sprite of kind Obstacle'. Attached to its bottom is a red 'change score by' block with the value '1' in a white circle.