

Cat jumper (Gato saltando)



Cuando comienza

```
on start
  set background color to #00bfff
  set background image to [background image]
  set current_level to 0
  set Hops_and_Paw to sprite [cat sprite] of kind Player
  move Hops_and_Paw with buttons vx: 80 vy: 0
  call startLevel
```

Podremos mover con los botones de derecha e izquierda nuestro Sprite a una velocidad de vx: 80, la coordenada vy: 0 no se moverá verticalmente desde dichos botones.



animation

Advanced state based animations for sprites

Hemos que tenido que importar esta extensión

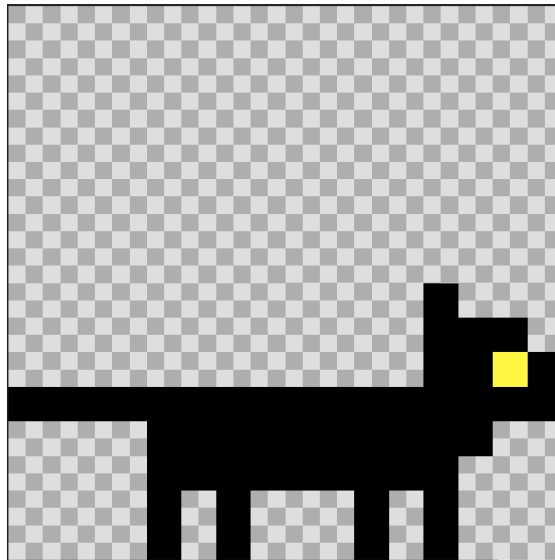
Creamos un fondo en el escenario de color azul.

Agregamos una imagen de fondo, dimensiones 160 x 120.

Definimos a la variable current_level a 0.

Definimos un Sprite de tipo Player con el nombre de Hops_and_Paw con su respectiva imagen.





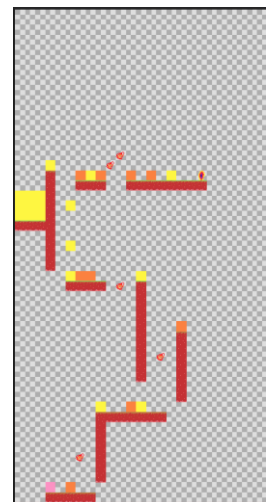
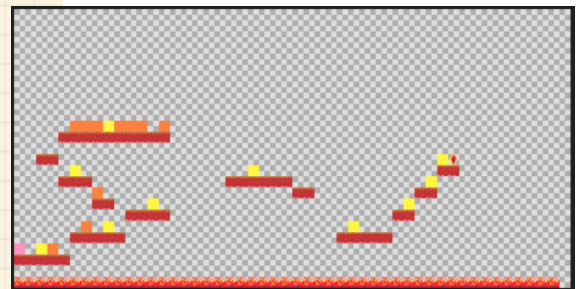
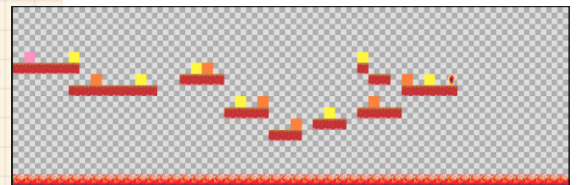
Dimensiones 16 x 16.

Función starLevel.

Según el valor de la variable current_lebes tendremos el siguiente fondo para nuestro escenario.

```

function startLevel
  if current_level = 0 then
    set tilemap to [tilemap_0]
  else if current_level = 1 then
    set tilemap to [tilemap_1]
  else if current_level = 2 then
    set tilemap to [tilemap_2]
  else
    game over WIN
  place Hops_and_Paw on top of random [square]
  for element value of array of all [square] locations
    do set [tilemap] at value
  
```

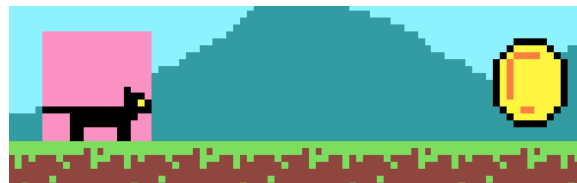


Si la variable `current_level` es mayor de 2 se termina la partida con éxito.

En cada escenario encontramos una baldosa de color rosa, esta es la referencia de donde tiene que aparecer el gato cuando empieza a pasar de nivel.

Con el bloque `place`, mueve la posición de un objeto al centro de un mosaico aleatorio en la escena.

Hacemos un ciclo `for` para que la variable `value` almacene la posición de dicha baldosa y reemplace la baldosa rosa por una transparente.



```
camera follow sprite Hops_and_Paw
set life to 5
for element value2 of array of sprites of kind Enemy
do
destroy value2
for element value3 of array of sprites of kind Coin
do
destroy value3
for element value4 of array of sprites of kind Flower
do
destroy value4
for element value5 of array of all locations
do
set flower to sprite of kind Coin
animate flower
frames
interval (ms) 100
loop ON
```

La cámara que siga a nuestro gatito.
Definimos un inicio de 5 vidas.

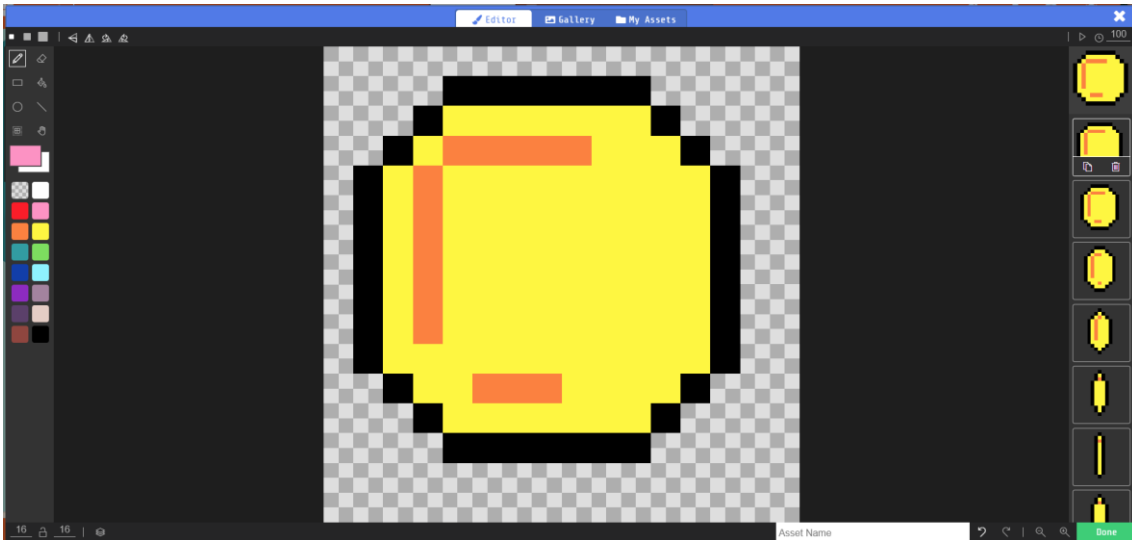
Con estos tres ciclos `for` utilizando las variables `value2`, `value3` y `value4` le estamos diciendo que borre (destruya) todos los Sprites de tipo `Enemy`, `Coin` y `Flower` que teníamos en el nivel anterior.

Ahora con este nuevo ciclo colocamos en la baldosa amarilla una moneda que rota sobre sí misma.

A Sprite `flower` le decimos sea de tipo `Coin` con la imagen de una moneda.

A la animación le asignamos es Sprite `flower`.

En frames (fotogramas) dibujaremos todas las monedas con sus respectivos movimientos.



```

place flower on top of value5
set [checkered tile] at value5

for element value6 of array of all [orange square] locations
do
  set flower to sprite [flower icon] of kind Flower
  place flower on top of value6
  set [checkered tile] at value6

for element value7 of array of all [fireball icon] locations
do
  set fireball to sprite [fireball icon] of kind Fireball
  place fireball on top of value7
  set [checkered tile] at value7
  animate fireball
  with "c 0 -100 0 100 0 0"
  for (ms) 2000
  loop ON
  
```

Reemplaza la baldosa amarilla por una baldosa transparente.

Hacemos lo mismo ponemos una flor en la posición que estás las baldosas de color naranja y reemplazamos la baldosa naranja por una baldosa transparente.

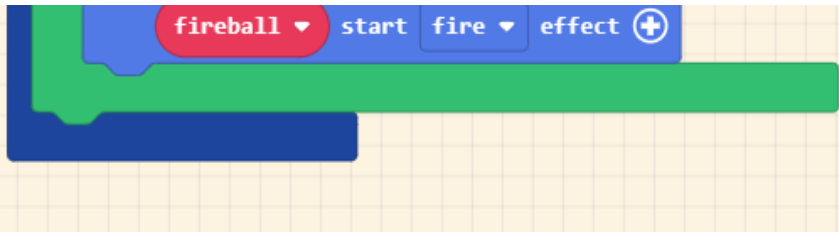
Al final de cada nivel encontramos un círculo de fuego, esto nos permitirá saltar de nivel.

Definimos un Sprite de tipo Fireball con el nombre de fireball.

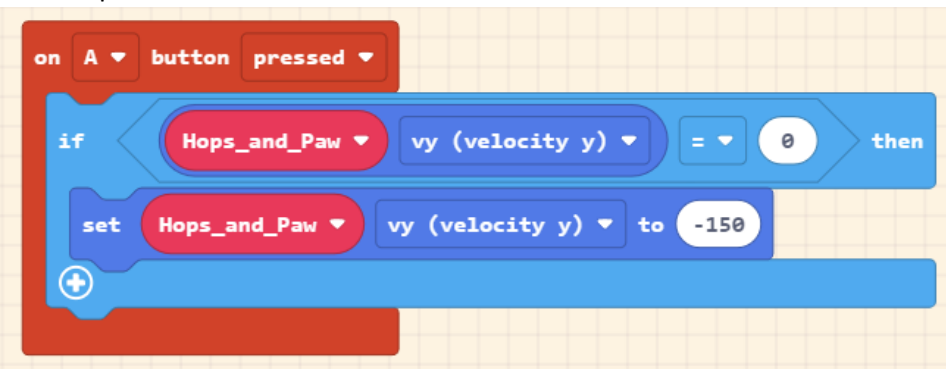
Coloca este Sprite donde esté la baldosa con el dibujo de fuego.

Cambia la baldosa que tiene el fuego por una baldosa transparente.

A la animación fireball con “c 0 -100 0 100 0 0” hace que dicho Sprite tenga unos movimientos de arriba hacia abajo y viceversa cada 2 segundos.



Además de un efecto de fuego. Cuando presionamos el botón A.



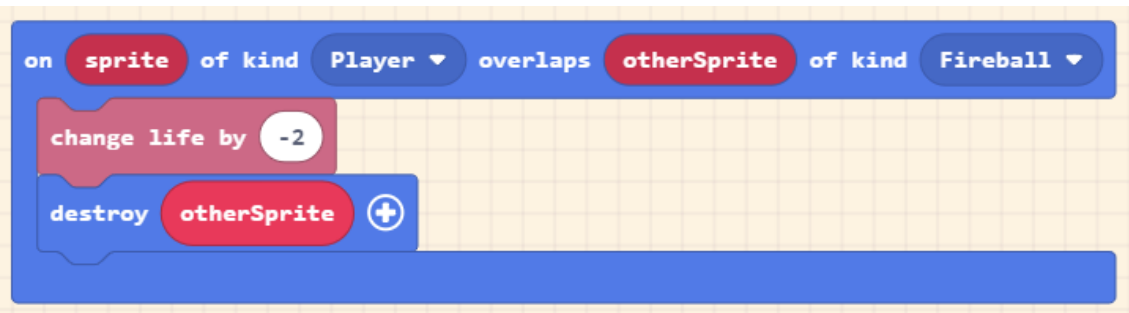
Nuestro gatito tendrá una velocidad en horizontal igual a 0 y una velocidad en y de -150 esto hará que realice un salto.

Cuando Sprite de tipo Player toca a otro Sprite de tipo Coin (Monedas)



La variable change (puntos) se incrementa en 1, la moneda se destruye.

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



La variable life (vidas) te queta dos y destruye el fuego.

Durante el juego.

```
on game update
  if < Hops_and_Paw > vy (velocity y) < 0 > then
    set Hops_and_Paw image to [Image 1]
  else if < Hops_and_Paw > vy (velocity y) > 0 > then
    set Hops_and_Paw image to [Image 2]
  else if < remainder of Hops_and_Paw x + 2 = 0 > then
    set Hops_and_Paw image to [Image 3]
  else
    set Hops_and_Paw image to [Image 4]

  if < is Hops_and_Paw hitting wall left > or < is Hops_and_Paw hitting wall right > and < Hops_and_Paw > vy (velocity y) >= 0 > then
    set Hops_and_Paw vy (velocity y) to 0
    set Hops_and_Paw ay (acceleration y) to 0
    set Hops_and_Paw image to [Image 4]
  else
    set Hops_and_Paw ay (acceleration y) to 350

  if < Hops_and_Paw > vx (velocity x) < 0 > or < is Hops_and_Paw hitting wall left > then
    flip Hops_and_Paw image horizontally
    set Hops_and_Paw image to Hops_and_Paw image
```

Si Hops_and_Paw tiene una velocidad en vy de menor de 0 creamos un Sprite con el nombre Hops_and_Paw con la imagen 1.

Si no si Hops_and_Paw tiene una velocidad en vy de mayor de 0 creamos un Sprite con el nombre Hops_and_Paw con la imagen 2.

Si no si el resto de dividir la posición de x de nuestro gatito dividido entre dos es igual a 0, sabremos si la posición de x es un valor par o impar, si es así crearemos un Sprite con el nombre Hops_and_Paw con la imagen 3.

Si no le asignaremos la misma imagen.

Si toca las paredes con su parte derecha o izquierda y además su velocidad en vy en mayor / igual a 0, La velocidad en y será 0, la aceleración en y será 0 y la imagen para nuestro gatito será la 4.

La aceleración de nuestro gatito en y será 350.

Si la velocidad del gatito en vx es menor a 0 o estás tocando la pared con tu parte izquierda hacemos una rotación horizontal de la imagen 4.



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

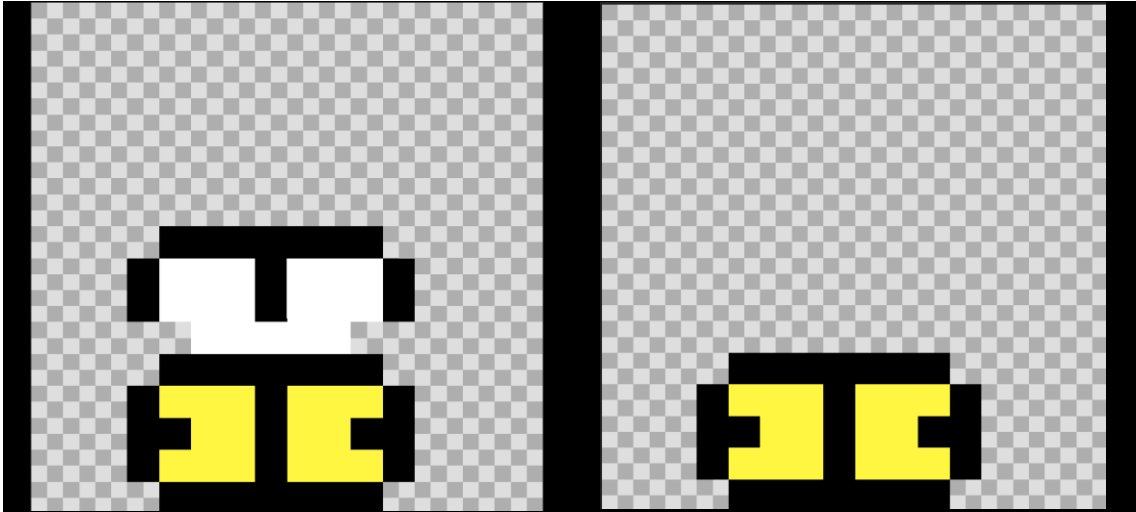


Cuando Sprite de tipo Player toca otro Sprite de tipo Flower.



Creamos un Sprite bee con una posición con respecto al gato en x de + 80 y una posición con respecto al gato en y de -80.

El Sprite bee (abeja) tiene que seguir al gatito con una velocidad de 50.



Cuando Sprite de tipo Playe toca imagen en escenario.

```
on sprite of kind Player overlaps [image] at location
  change current_level by 1
  call startlevel
```

La variable currremt_level se incrementa en una (Subimos de nivel)

Llamamos a la función starLevel.

Cuando Sprite de tipo Player toca imagen en escenario.

```
on sprite of kind Player overlaps [image] at location
  game over [LOSE] with melt effect
```

Se termina el juego y ¡Perdiste!.