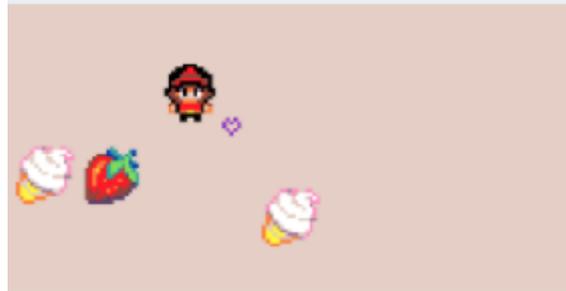


Eat de fruit



Cuando empezamos

Definimos el fondo del escenario.
Creamos un Sprite de tipo Player con un dibujo que hemos seleccionado desde la galería.

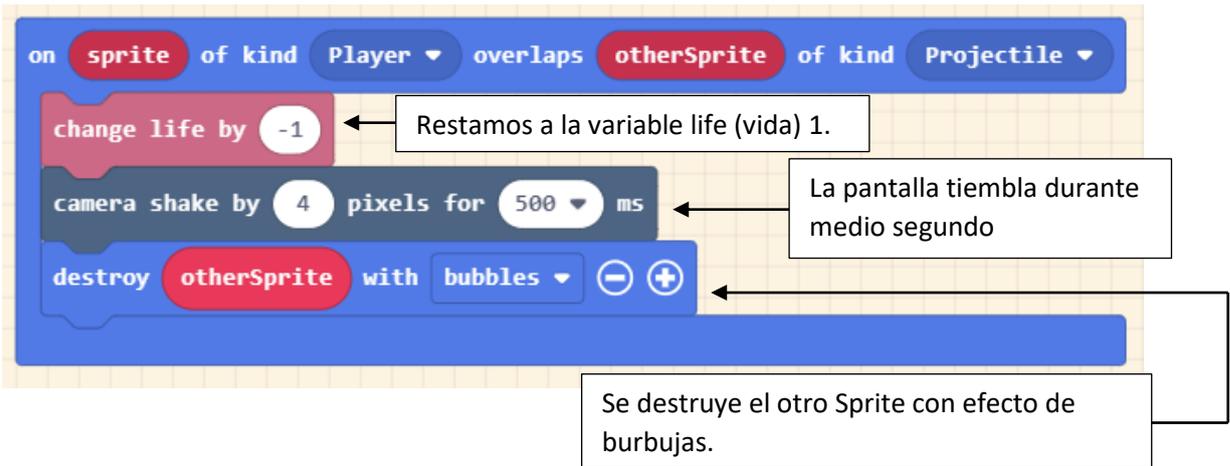
Moveremos dicho Sprite con las flechas de dirección a una velocidad tanto horizontal como vertical de 100.
El Sprite no puede salir de pantalla.
Empezar el juego teniendo 3 vidas.

Cuando Sprite2 de tipo Player toca a otro Sprite2 de tipo Food.

A la variable score (puntuación) le sumamos 1.
Destruye el otro Sprite2.
Al Sprite2 le damos un efecto hearts durante 100 ms.

Se reproduce un sonido.

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



on sprite of kind Player overlaps otherSprite of kind Projectile

- change life by -1 (Restamos a la variable life (vida) 1.)
- camera shake by 4 pixels for 500 ms (La pantalla tiembla durante medio segundo)
- destroy otherSprite with bubbles (Se destruye el otro Sprite con efecto de burbujas.)

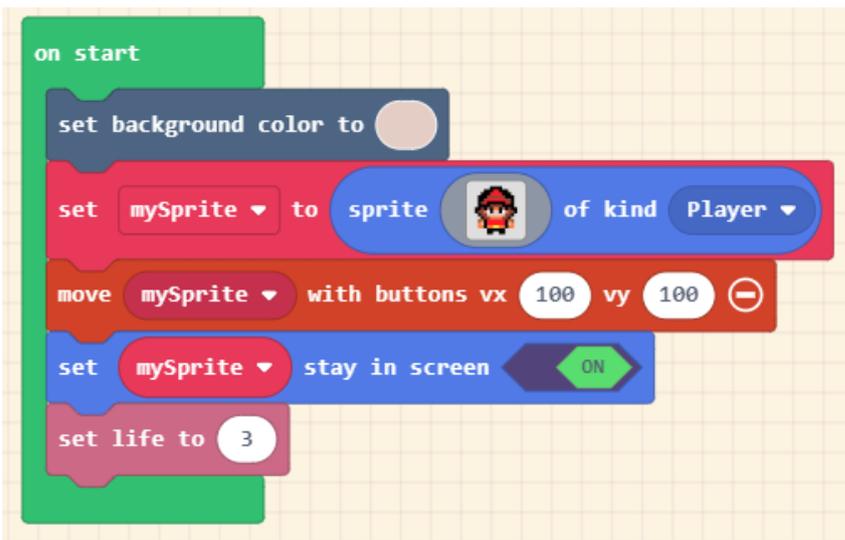
Durante el juego cada 500 milisegundos.



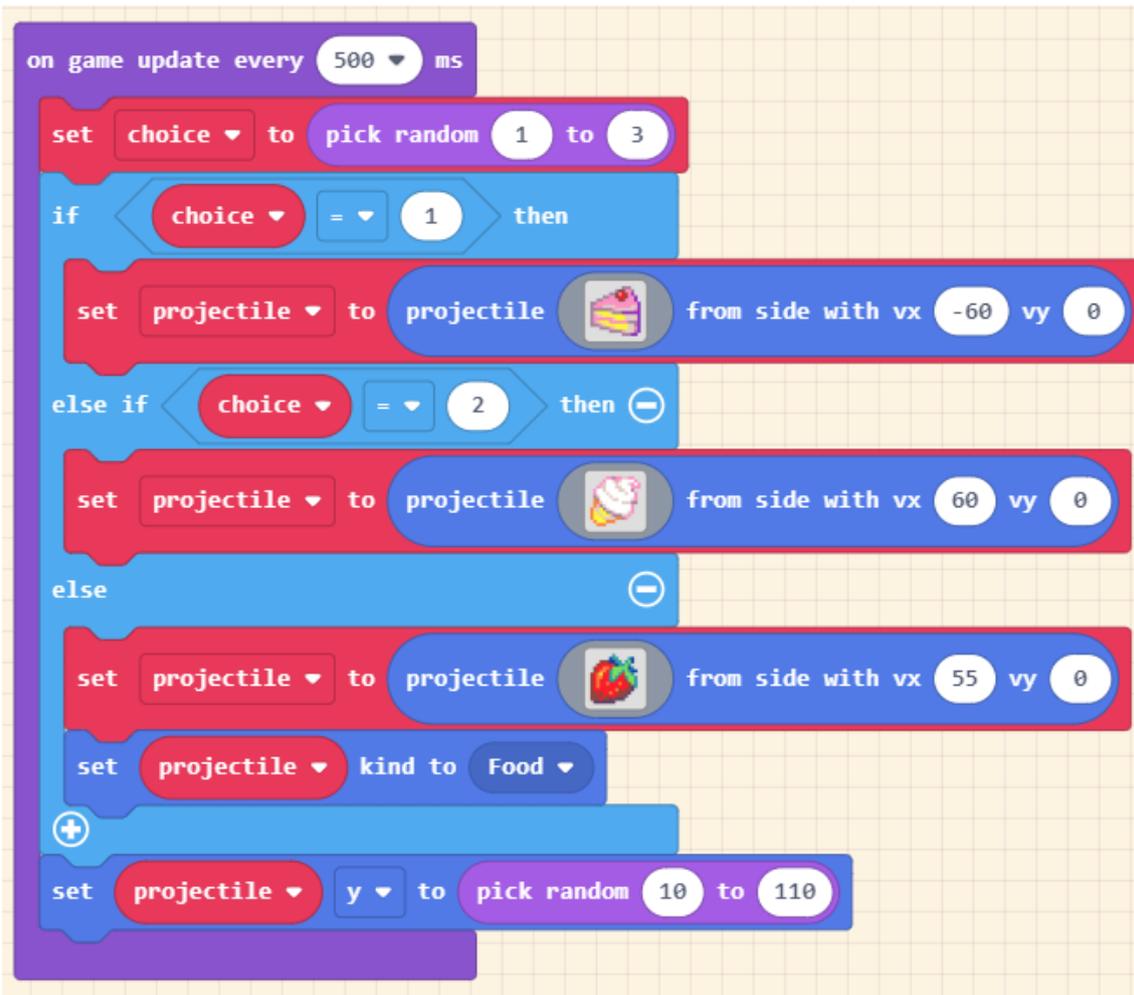
on game update every 500 ms

- set choice to pick random 1 to 3 (A la variable choice se le asigna un valor aleatorio desde 1 hasta el 3.)
- if choice = 1 then
 - set projectile to projectile (cake icon) from side with vx -60 vy 0 (Según su valor se creará un Sprite de nombre projectile con una imagen de pastel, helado o fresa.)
- else if choice = 2 then
 - set projectile to projectile (ice cream icon) from side with vx 60 vy 0
- else
 - set projectile to projectile (strawberry icon) from side with vx 55 vy 0
 - set projectile kind to Food (Este Sprite será de tipo Food.)
- set projectile y to pick random 10 to 110 (Se posicionará en la coordenada y un valor aleatorio desde 10 hasta 110.)

Para familiarizarnos con el código Python vamos a realizar unas comparaciones.



```
14 projectile: Sprite = None
15 choice = 0
16 scene.set_background_color(13)
17 mySprite = sprites.create(img("""
18     . . . . . f f f f . . . . .
19     . . . . f f f 2 2 f f f . . . .
20     . . . f f f 2 2 2 2 f f f . . .
21     . . f f f e e e e e e f f f . .
22     . . f f e 2 2 2 2 2 e e f . .
23     . . f e 2 f f f f f f 2 e f . .
24     . . f f f f e e e e f f f f . .
25     . f f e f b f 4 4 f b f e f f .
26     . f e e 4 1 f d d f 1 4 e e f .
27     . . f e e d d d d d d e e f . .
28     . . . f e e 4 4 4 4 e e f . . .
29     . . e 4 f 2 2 2 2 2 2 f 4 e . .
30     . . 4 d f 2 2 2 2 2 2 f d 4 . .
31     . . 4 4 f 4 4 5 5 4 4 f 4 4 . .
32     . . . . . f f f f f f . . . . .
33     . . . . . f f . . f f . . . . .
34     """),
35     SpriteKind.player)
36 controller.move_sprite(mySprite, 100, 100)
37 mySprite.set_stay_in_screen(True)
38 info.set_life(3)
```



```

40 def on_update_interval():
41     global choice, projectile
42     choice = randint(1, 3)
43     if choice == 1:
44         projectile = sprites.create_projectile_from_side(img("...",
61             "..."),
62                 -60,
63                 0)
64     elif choice == 2:
65         projectile = sprites.create_projectile_from_side(img("...",
82             "..."),
83                 60,
84                 0)
85     else:
86         projectile = sprites.create_projectile_from_side(img("...",
103             "..."),
104                 55,
105                 0)
106         projectile.set_kind(SpriteKind.food)
107         projectile.y = randint(10, 110)
108     game.on_update_interval(500, on_update_interval)

```

Nos está indicando que hay una imagen la vamos a expandir.

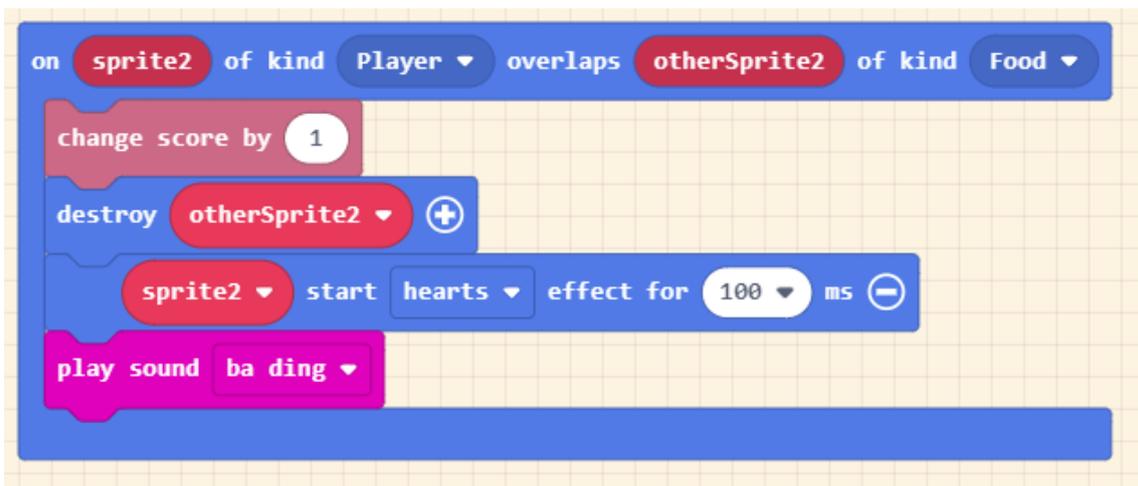
45			 b b b . . .
46			 b e e 3 3 b . .
47			 b b e 3 2 e 3 a . .
48			 b b 3 3 e 2 2 e 3 3 a .
49				. . b b 3 3 3 3 3 e e 3 3 3 a .
50				b b 3 3 3 3 3 3 3 3 3 3 3 a
51				b 3 3 3 d d d d 3 3 3 3 d d a
52				b b b b b b 3 d d d d d 3 a
53				b d 5 5 5 5 d b b b a a a a a
54				b 3 d d 5 5 5 5 5 5 d d d d a
55				b 3 3 3 3 3 d 5 5 5 d d d d a
56				b 3 d 5 5 5 3 3 3 3 3 b b b a
57				b b b 3 d 5 5 5 5 5 5 d d b a
58				. . . b b b 3 d 5 5 5 5 d d 3 a
59			 b b b b 3 d d d b a
60			 b b b a a .

66	∨		 3 3 b 3 3 d d 3 3 . .
67			 3 1 1 d 3 d 1 1 1 1 3 .
68				. . . 3 d 1 1 1 d 1 1 1 d 3 1 3
69				. . 3 d d 1 1 1 d d 1 1 1 3 3 3
70				. 3 1 1 d 1 1 1 1 d d 1 1 b . .
71				. 3 1 1 1 d 1 1 1 1 1 d 1 1 3 .
72				. b d 1 1 1 d 1 1 1 1 1 1 3 .
73				. 4 b 1 1 1 1 d d 1 1 1 1 d 3 .
74				. 4 4 d 1 1 1 1 1 1 d d d b b .
75				. 4 d b d 1 1 1 1 1 1 1 3 . .
76				4 d d 5 b d 1 1 1 1 1 1 3 . .
77				4 5 d 5 5 b b d 1 1 1 1 d 3 . .
78				4 5 5 d 5 5 d b b b d d 3 . . .
79				4 5 5 5 d d d d 4 4 b 3
80				. 4 5 5 5 4 4 4
81				. . 4 4 4
82			"""),	

```

87      . . . . . 6 . . . . .
88      . . . . . 8 6 6 . . . 6 8 . .
89      . . . e e e 8 8 6 6 . 6 7 8 . .
90      . . e 2 2 2 2 e 8 6 6 7 6 . . .
91      . e 2 2 4 4 2 7 7 7 7 7 8 6 . .
92      . e 2 4 4 2 6 7 7 7 6 7 6 8 8 .
93      e 2 4 5 2 2 6 7 7 6 2 7 7 6 . .
94      e 2 4 4 2 2 6 7 6 2 2 6 7 7 6 .
95      e 2 4 2 2 2 6 6 2 2 2 e 7 7 6 .
96      e 2 4 2 2 4 2 2 2 4 2 2 e 7 6 .
97      e 2 4 2 2 2 2 2 2 2 2 e c 6 .
98      e 2 2 2 2 2 2 2 4 e 2 e e c . .
99      e e 2 e 2 2 4 2 2 e e e c . . .
100     e e e e 2 e 2 2 e e e c . . . .
101     e e e 2 e e c e c c c . . . . .
102     . c c c c c c c . . . . .

```



```

7 def on_on_overlap2(sprite2, otherSprite2):
8     info.change_score_by(1)
9     otherSprite2.destroy()
10    sprite2.start_effect(effects.hearts, 100)
11    music.ba_ding.play()
12    sprites.on_overlap(SpriteKind.player, SpriteKind.food, on_on_overlap2)

```



```
1 def on_on_overlap(sprite, otherSprite):  
2     info.change_life_by(-1)  
3     scene.camera_shake(4, 500)  
4     otherSprite.destroy(effects.bubbles)  
5 sprites.on_overlap(SpriteKind.player, SpriteKind.projectile, on_on_overlap)
```