

1 Definición de una partícula

Basta con copiar el código inferior. Puedes sustituir «MiParticle» por el nombre de tu partícula, es importante que esté entre comillas. Puedes cambiar los números de r, g, b y a para cambiar el color de la partícula. Por último, puedes añadir el comportamiento de la partícula entre las llaves. Para ello, es necesario programar en Lua. Se proporcionarán una serie de comandos para simplificar esta tarea.

```
1 addParticle(  
2     "MyParticle",                -- Particle Name  
3     { r = 255, g = 255, b = 255, a = 255 }, -- Color  
4     function(api)                -- Behaviour  
5  
6     end  
7 )
```

Lua

2 Programar en Lua

A continuación se detallan las bases de programar en Lua mediante un ejemplo de código que será explicado por el entrevistador tras la finalización de la lectura de este documento.

```
1 local myVariable = 0 -- Variable definition  
2 local myBool = true -- Boolean definition  
3 local randomInterval = math.random(0, 10) -- Random number between 0 and 10,  
    both included  
4 local particleType = ParticleType.SAND -- Id of the sand particle, error if  
    there is no particle with that name. Aunque el nombre de la partícula tenga  
    minúsculas, el id siempre se escribe en mayúsculas. Por ejemplo, si mi partícula  
    se llama MyParticle el id será ParticleType.MY_PARTICLE  
5 -- Siempre existe una partícula que representa la ausencia de partícula, esta se  
    llama EMPTY  
6  
7 -- Array  
8 local myArray = {1, 2, 3, 4, 5}  
9  
10 -- If statement  
11 if myVariable == 0 then  
12     myVariable = 1  
13 end  
14  
15 -- If else statement  
16 if myVariable == 0 then  
17     myVariable = 1  
18 elseif myVariable == 1 then  
19     myVariable = 2  
20 else  
21     myVariable = 0  
22 end  
23  
24 -- Logical operators  
25 -----  
26 --- == => Equal  
27 --- ~= => Not equal  
28 --- < => Less than  
29 --- > => Greater than  
30 --- <= => Less than or equal  
31
```

Lua

```

32 -- Loops
33 while myBool do
34     -- Do something
35 end
36
37 -- For loop
38 for i = 0, 10 do
39     -- Do something
40 end
41
42 -- Iterating over an array
43 -- The following code prints the index and value of each element in the array
44 for index, value in ipairs(myArray) do
45     print("Index: " .. index .. " Value: " .. value)
46 end

```

3 Funciones para programar partículas

Estas funciones se usan de la forma `api:funcion()`. A continuación se detallan las funciones que se pueden usar para programar partículas.

```

1  get_neighbours() -- Devuelve un array de direcciones. Las direcciones son un Lua
   objeto con componente X e Y, se usa así.
2
3  -- Se usa para recorrer los vecinos de una partícula
4  for _, direction in ipairs(api:get_neighbours()) do
5      if api:getParticleType(direction.x, direction.y) == ParticleType.TYPE then
6          -- Haz algo
7      end
8  end
9
10 -- Cambia la partícula en la posición x, y por la partícula con el id id. La
   posición x e y son relativas a la partícula actual, por lo que si se pasa x = 0
   y y = 0, se cambiará la partícula actual, o por ejemplo si se pasa x = 1 y y =
   0, se cambiará la partícula a la derecha de la actual. Esto aplica para todas
   las funciones que requieran x e y. Posición x positiva es derecha, posición y
   positiva es arriba. Posición x negativa es izquierda, posición y negativa es
   abajo.
11 setNewParticleById(x, y, id)
12
13 -- Ejemplo de uso
14 setNewParticleById(0, 0, ParticleType.WATER) -- Cambia la partícula actual por
   agua (asumimos que existe una partícula que se llama WATER)
15
16 swap(x, y) -- Intercambia la partícula actual con la partícula en la posición x,
   y
17 isEmpty(x, y) -- Devuelve true si la posición x, y está vacía
18 getParticleType(x, y) -- Devuelve el id de la partícula en la posición x, y
19 check_neighbour_multiple(x, y, id_array) -- Devuelve true si la posición x, y
   tiene una partícula con el id id

```

4 Ejemplo de Partícula Completa

Esta partícula va hacia abajo si tiene una partícula vacía o de aire debajo.

```

1  addParticle(
2      "GoDown",
   -- Text id

```

```

3     { r = 255, g = 0, b = 255, a = 255 },      -- Color
4     function(api)
5         local dirY = -1
6         local id_array = { ParticleType.EMPTY, ParticleType.AIR }
7
8         if api:check_neighbour_multi(0, dirY, id_array) then
9             api:swap(0, dirY)
10        end
11    end
12 )

```