



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

*INGENIERÍA EN COMPUTACIÓN*

## FUNDAMENTOS DE PROGRAMACIÓN

### ACTIVIDAD ASÍNCRONICA 14

**ALUMNO:** Raudales Palma Leonardo de Jesús

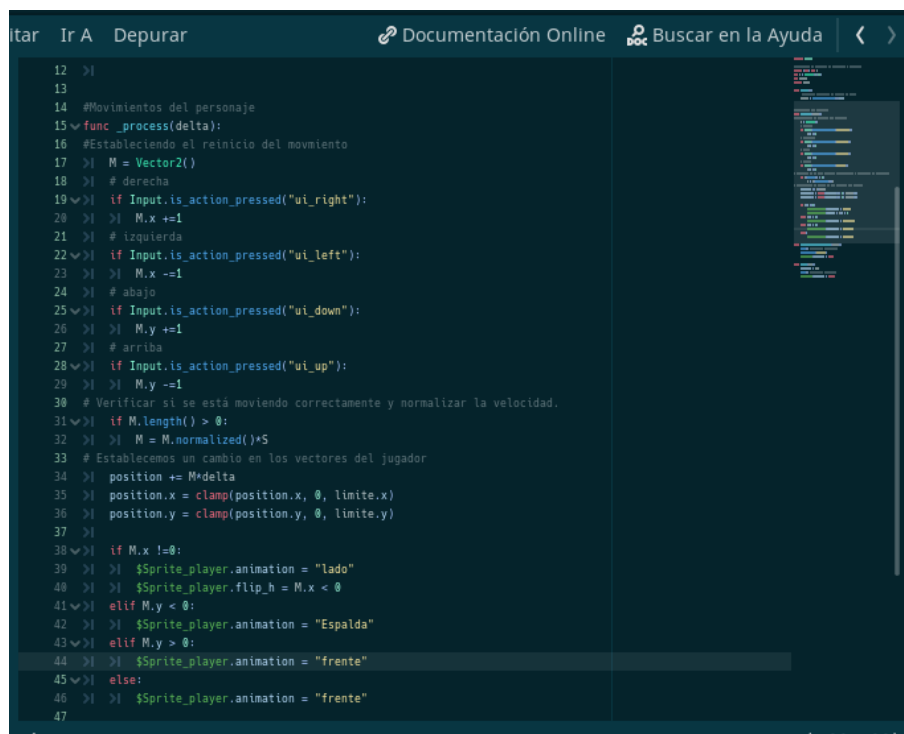
04/12/2020



# Estructuras de Repetición

## Proyecto: Videojuego

A diferencia del punto anterior relacionado al proyecto, que fue estructuras de control, en sí el desarrollo el videojuego con el programa Godot Engine no cuenta con el uso de estructuras de repetición de lenguaje c. Sin embargo, si se utilizan estas mismas con funciones predeterminadas del programa, que forman parte del lenguaje de programación utilizado en el juego, el cual es GDScript- En este caso la función que se utiliza para crear un proceso de repetición `_process` el cual nos indica una serie de sucesos que se repetirán continuamente a lo largo de gameplay. Un caso específico de esto es lo que vimos con anterioridad que fue el movimiento del jugador y la animación de este, si metemos todo esto dentro de la función `_process` entonces esto sucederá una y otra vez, es decir, cada que el elemento "jugador" entre en progreso o ejecución, la animación y desplazamiento del player dentro de la pantalla se podrá hacer un número indefinido de veces, si no se usara la función `process_` lo que ocurriría es que solo podrías moverte una vez cuando comience el juego.



```
12 >|
13
14 #Movimientos del personaje
15 func _process(delta):
16     #Estableciendo el reinicio del movimiento
17     >| M = Vector2()
18     >| # derecha
19 >| if Input.is_action_pressed("ui_right"):
20     >| >| M.x +=1
21     >| # izquierda
22 >| if Input.is_action_pressed("ui_left"):
23     >| >| M.x -=1
24     >| # abajo
25 >| if Input.is_action_pressed("ui_down"):
26     >| >| M.y +=1
27     >| # arriba
28 >| if Input.is_action_pressed("ui_up"):
29     >| >| M.y -=1
30     # Verificar si se está moviendo correctamente y normalizar la velocidad.
31 >| if M.length() > 0:
32     >| >| M = M.normalized()*5
33     # Establecemos un cambio en los vectores del jugador
34     >| position += M*delta
35     >| position.x = clamp(position.x, 0, limite.x)
36     >| position.y = clamp(position.y, 0, limite.y)
37     >|
38 >| if M.x !=0:
39     >| >| $Sprite_player.animation = "lado"
40     >| >| $Sprite_player.flip_h = M.x < 0
41 >| elif M.y < 0:
42     >| >| $Sprite_player.animation = "Espalda"
43 >| elif M.y > 0:
44     >| >| $Sprite_player.animation = "frente"
45 >| else:
46     >| >| $Sprite_player.animation = "frente"
47
```

Figura 1- `_process` en código GDScript

Un ejemplo aún más claro de esto es el evento del reinicio del movimiento del jugador, como se puede ver en el código colocamos al inicio dentro de `_process` la variable `M` referente al movimiento que es igual al `vector2`, que es el vector utilizado para asignar el desplazamiento del personaje con respecto a un cambio en el valor de la posición, misma que también asociamos a `vector2` dentro de la misma estructura de repetición. La razón de declarar la variable `M` que es el movimiento al inicio de la estructura de repetición es para indicar que cuando presionas una tecla y te desplazas a un valor dado que establecimos el movimiento debe reiniciarse a 0, es decir, si tú te mueves 1 hacia la izquierda entonces volverás a 0 nuevamente y tendrás la posibilidad de mover 1 otra vez, esto se repite continuamente durante toda la acción de `_process` o lo que es lo mismo, mientras el elemento `player` permanece ejecutándose. Se utiliza esta función para poder evitar que a la hora de mover el personaje este se detenga y no siga moviéndose hacia la dirección que tecleamos desde un principio, para ello es necesario establecer este proceso de repetición o reinicio para que una vez que se aumente el valor que establecimos si el personaje se mueve a tal dirección este comience de nuevo y el usuario sea capaz de entonces presionar otra tecla y obtener el mismo valor pero ahora en otra dirección, ya el conjunto de todas estas acciones crearía la percepción del movimiento fluido del personaje dentro del juego.



Figura 2. Movimiento sin estructura de repetición

Como se puede apreciar en la figura, el dragón se encuentra situado en la parte inferior de la pantalla debido a que no se implementó la estructura de repetición y entonces al presionar la tecla abajo el personaje suma el valor establecido hacia la posición y sin reiniciarse lo que ocasiona que no te permita dar otro valor en una dirección contraria.