



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

INGENIERÍA EN COMPUTACIÓN

FUNDAMENTOS DE PROGRAMACIÓN

ACTIVIDAD ASÍNCRONICA 13

ALUMNO: Raudales Palma Leonardo de Jesús

27/10/2020



Estructuras de Control

Proyecto: Videojuego

Antes de empezar, tengo que aclarar que el lenguaje de programación utilizado para la realización del proyecto no es C, es GDScript que es el lenguaje por defecto del programa, aun así, este al igual que otros lenguajes tiene algunas similitudes con lenguaje c, uno de estos casos es el de la estructura de control de if - else. Uno de los usos más claros de esta estructura se da en 2 situaciones dentro del proyecto, la primera es para mover el jugador en caso de que se presione una tecla y especificar la tecla que activa cada dirección.

```
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()*S
```

Figura 1. Código del movimiento del jugador

Como se puede observar en el código, se usa la estructura de control if para poder declarar a partir de que sentencias ocurre el movimiento del jugador hacia una dirección dada, en este caso usamos Input que es una variable predeterminada del programa que nos permite relacionar el entorno con el teclado, luego usamos una función predeterminada del programa que indica que a la acción de presionar se haga tal cosa. Una vez codificado todo esto la orden completa sería que al presionar una tecla (Input) de alguna dirección, el

personaje cambie su posición aumentando valores en el vector donde está situado y con respecto al eje que relaciona a cada movimiento, movimientos horizontales serían eje x y verticales eje y.

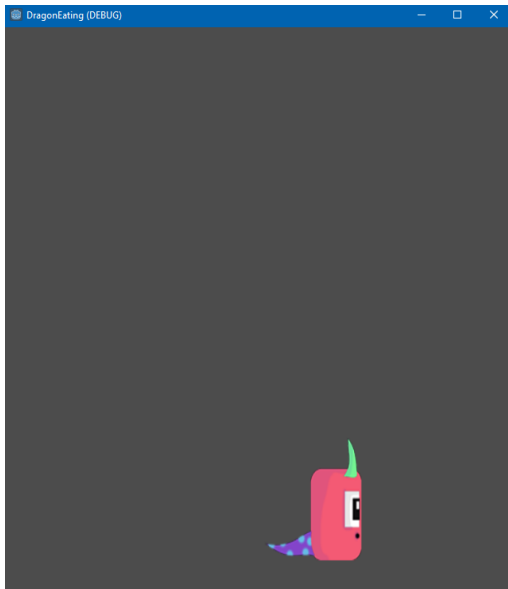


Figura 2. Player - Right

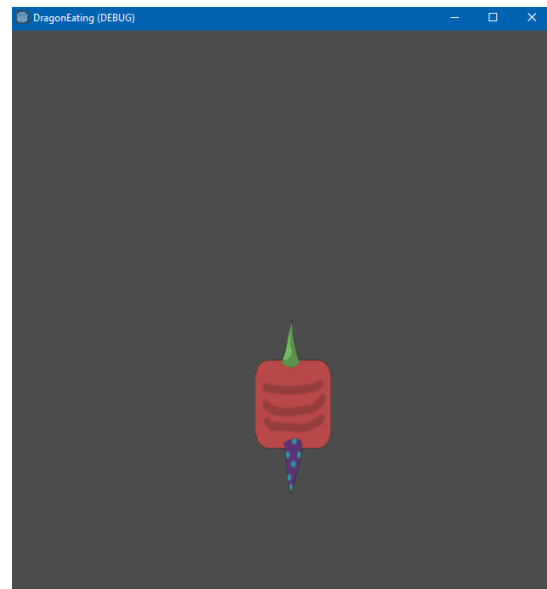


Figura 3. Player - Up

Como se puede apreciar, no solo podemos usar la estructura de control if-else para el movimiento del jugador, sino que también, el segundo módulo donde interviene es en la animación del personaje con respecto al movimiento. En las imágenes se ve como al ejecutar un movimiento la animación que se presenta en forma de un sprite cambia, esto es porque se le asigna condiciones en base al código anterior que era el del movimiento, por ejemplo, se usa if estableciendo que, si el movimiento del personaje es hacia arriba, entonces se muestra el sprite llamado "espalda" en pantalla. Lo mismo ocurre si el personaje va hacia algún lado, esto además se puede extender si en vez de crear un sprite para cada lado usas únicamente un sprite y sentencias con if que dependiendo si el movimiento es hacia la derecha o hacia la izquierda entonces la animación cambie o se refleje. Por último, con else simplemente pones la condición que si no se está realizando ninguna de las sentencias anteriores pues se muestre el sprite llamado "frente" en pantalla. Esto hace que cuando el personaje vaya hacia arriba se muestre la espalda del dragón, hacia los lados, se muestre el perfil de este mismo, y cuando se mueve hacia abajo o si no está haciendo ningún movimiento entonces muestre la animación del dragón de frente.

```
tar Ir A Depurar Documentación Online Buscar en la Ayuda < >

27 >I # arriba
28 >I if Input.is_action_pressed("ui_up"):
29 >I >I M.y -=1
30 # Verificar si se está moviendo correctamente y normalizar la velocidad.
31 >I if M.length() > 0:
32 >I >I M = M.normalized()*S
33 # Establecemos un cambio en los vectores del jugador
34 >I position += M*delta
35 >I position.x = clamp(position.x, 0, limite.x)
36 >I position.y = clamp(position.y, 0, limite.y)
37 >I
38 >I if M.x !=0:
39 >I >I $Sprite_player.animation = "lado"
40 >I >I $Sprite_player.flip_h = M.x < 0
41 >I elif M.y < 0:
42 >I >I $Sprite_player.animation = "Espalda"
43 >I elif M.y > 0:
44 >I >I $Sprite_player.animation = "frente"
45 >I else:
46 >I >I $Sprite_player.animation = "frente"
47
48 >I func _on_Player_body_entered(_body):
49 >I hide() #Personaje desaparece
50 >I emit_signal("golpe")
51 >I $Collision.disabled = true
52
53 >I func start(pos):
54 >I position = pos
55 >I show() #Personaje reaparece
56 >I $Collision.disabled = false
57 >I
```

Figura 4. Código animación del Player

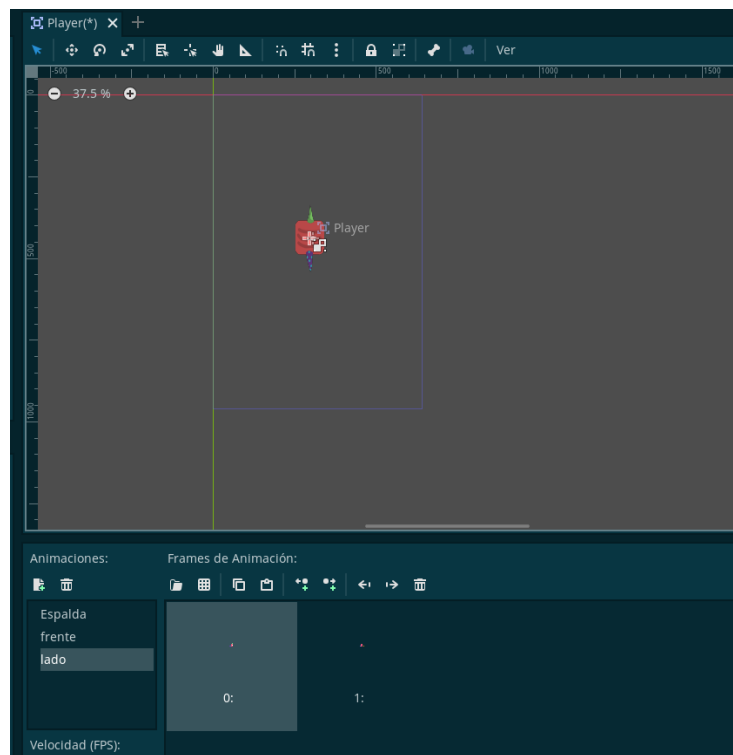


Figura 5. Sprites relacionados con el uso del if-else