

Dodge the Creeps!

El joc es diu **Dodge the Creeps!**. El teu personatge ha de moure's i evitar els enemics tant de temps com sigui possible.

Continuació de la fitxa anterior!

36. Ara que el jugador pot moure's, necessitem canviar quina animació està reproduint l'**AnimatedSprite** en funció de la seva direcció.

37. Tenim l'animació **walk**, que mostra al jugador caminant cap a la dreta. Aquesta animació ha de ser voltejada horitzontalment fent servir la propietat **flip_h** per al moviment cap a l'esquerra.

També tenim l'animació **up**, que ha de ser voltejada verticalment amb **flip_v** per al moviment cap avall. Col·loquem el següent codi al final de la funció **_process()**:

```
if velocity.x != 0:
    $AnimatedSprite.animation = "walk"
    $AnimatedSprite.flip_v = false
    $AnimatedSprite.flip_h = velocity.x < 0
elif velocity.y != 0:
    $AnimatedSprite.animation = "up"
    $AnimatedSprite.flip_v = velocity.y > 0
```

Reprodueix l'escena de nou i revisa si les animacions són correctes en cadascuna de les direccions.

38. Agrega aquesta línia a **_ready()** perquè el jugador estigui ocult quan comenci el joc:

```
hide()
```

[COMMIT] Animacions jugador.

39. Volem detectar quan un enemic toca a **Player**, però no hem creat cap enemic encara. Això està bé, perquè utilitzarem la funcionalitat de senyals de Godot per a fer que funcioni.

40. Afegeix el següent en la part superior del script, després d'**extends Area2d**:

signal hit

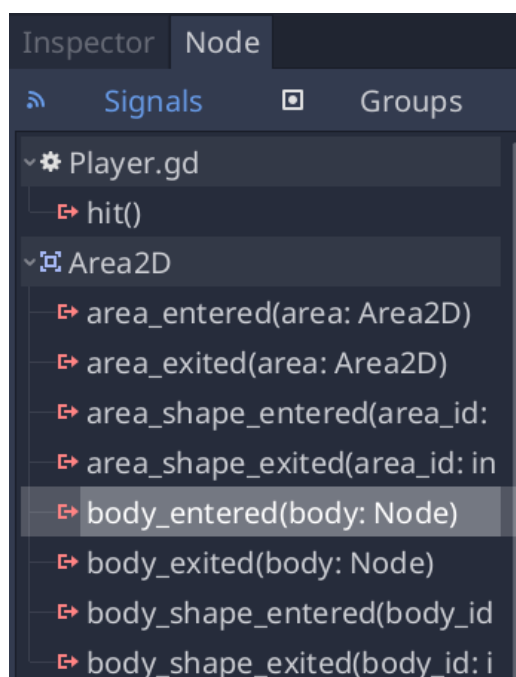
Això defineix un senyal personalitzat anomenat **hit** que el nostre **Player** emetrà en col·lisionar amb un enemic.

41. Farem servir **Area2D** per a detectar la col·lisió. Selecciona el node **Player** i fes clic en la pestanya **Nodes** al costat de la pestanya del menú per a veure la llista de senyals que pot emetre **Player**:

Observa que el nostre senyal personalitzat **hit** també apareix.

Com els nostres enemics seran nodes de tipus **RigidBody2D**, necessitem el senyal **body_entered**.

Aquest senyal s'emetrà quan un cos entri en contacte amb **Player**.



42. Fes clic a **Connectar** i després a **Connectar** de nou en la finestra **Connectar senyal**. No necessitem canviar cap configuració. Godot crearà automàticament una funció en el script de **Player**.

```
42 func _on_Player_body_entered(body):
43     pass # Replace with function body.
```

La icona verda indica que un senyal és troba connectat a aquesta funció.

43. Agrega aquest codi a la funció:

```
func _on_Player_body_entered(body):
    hide() # El jugador desapareix després de ser impactat.
    emit_signal("hit")
    # S'ha d'ajornar, ja que no podem canviar les propietats físiques en una crida de retorn de física.
    $CollisionShape2D.set_deferred("disabled", true)
```

Cada vegada que un enemic impacti al jugador, el senyal serà emès. Necessitem deshabilitar la col·lisió del jugador perquè no activem el senyal de **hit** més d'una vegada.

44. L'últim serà agregar una funció que cridarem per a reiniciar-ho quan comencem una nova partida.

```
func start(pos):
    position = pos
    show()
    $CollisionShape2D.disabled = false
```

[COMMIT] Preparem les col·lisions.

45. Ara és el moment de crear els enemics del joc. El seu comportament no serà molt complex: els enemics es generaran aleatòriament a les vores de la pantalla, triaran una direcció aleatòria i es mouran en línia recta.

Crearem una escena **Mob**, que després podrem instanciar per crear qualsevol conjunt d'enemics independents en el nostre joc.

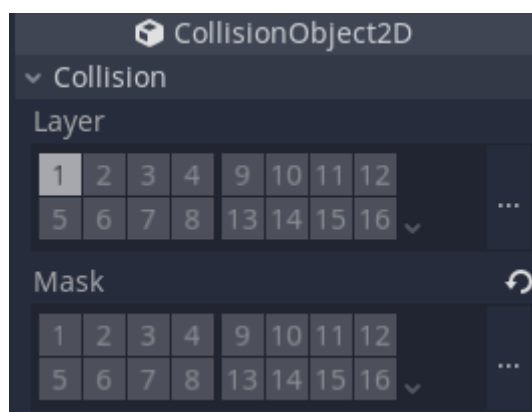
46. Feu clic a **Escena/Nova Escena** i afegiu els següents nodes:

- **RigidBody2D** (l'anomenarem **Mob**)
 - **AnimatedSprite**
 - **CollisionShape2D**
 - **VisibilityNotifier2D**

No oblidis configurar els fills perquè no puguin ser seleccionats, com vas fer amb l'escena del jugador.

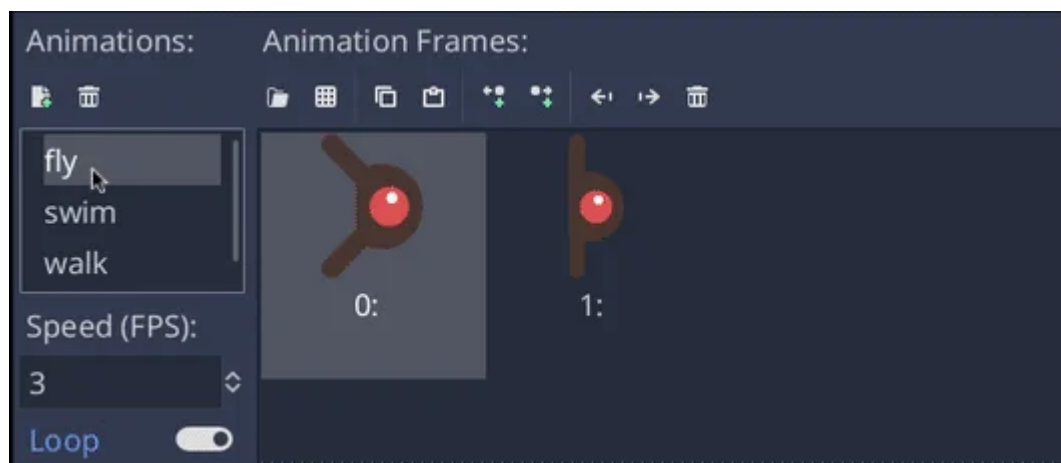
47. A les propietats **RigidBody2D**, establiu **Gravity Scale** a **0**, de manera que els **Mobs** no caiguin cap avall.

A més, sota la secció **ColisionObject2D**, fes clic a la propietat **Mask** i desmarca la primera casella. Això garantirà que els **Mobs** no xoquin entre si.



48. Configureu l'**AnimatedSprite** com ho vau fer per al jugador. Aquesta vegada, tenim 3 animacions: **fly**, **swim** i **walk**. Hi ha dues imatges per a cada animació a la carpeta d'art.

Ajusteu **Speed (FPS)** a **3** per a totes les animacions.



49. Estableix la propietat **Playing** del menú a **On**. Seleccionarem una d'aquestes animacions a l'atzar perquè els **Mobs** tinguin certa varietat.

50. Igual que amb les imatges del jugador, les imatges dels **Mobs** s'han d'escalar. Estableix la propietat **Scale** d'**AnimatedSprite** a **(0.75, 0.75)**.

51. Com a l'escena **Player**, afegiu una **CapsuleShape2D** per a la col·lisió. Per a alinear la forma amb la imatge, haureu d'establir la propietat **Rotation Degrees** a **90** (a l'opció **Transform** del menú).

Guardem l'escena i fem **[COMMIT] Creació del node dels enemies**.

52. Afegeix un script a **Mob** amb el següent contingut:

```
extends RigidBody2D
```

53. A la funció **_ready()** es reproduïx l'animació i es tria aleatòriament un dels tres tipus d'animació:

```
func _ready():
    $AnimatedSprite.playing = true
    var mob_types = $AnimatedSprite.frames.get_animation_names()
    $AnimatedSprite.animation = mob_types[randi() % mob_types.size()]
```

Primer, obtenim la llista amb els noms de les animacions de la propietat **frames** de **AnimatedSprite**.

Això retorna un array que conté els tres noms d'animació: **["walk", "swim", "fly"]**.

Lavors hem de triar un nombre aleatori entre **0** i **2** per seleccionar un d'aquests noms de la llista (els índexs d'array comencen a **0**).

randi() % n selecciona un enter aleatori entre **0** i **n-1**.

54. Ara volem que les **Mobs** s'eliminin quan abandonin la pantalla. Connecta el senyal screen **screen_exited()** del node **VisibilityNotifier2D** i afegeix aquest codi:

```
func _on_VisibilityNotifier2D_screen_exited():
    queue_free()
```

[COMMIT] Escena Mob acabada.