

- ❖ Tutoriel de **développeur** et **animateur**
- ❖ Niveau **hard**



## Créer des ennemis de *plateformer* 2D

### Rappels :

- Ce tutoriel vous permet de cocher une case d'expérience dans votre **carnet de progression** en tant que **développeur** et **animateur**.
- Le vocabulaire propre au moteur GODOT est écrit en *italique*.

### Objectif du tutoriel :

- Créer des ennemis pour un jeu de plateforme 2D qui se déplacent aléatoirement, peuvent tuer le Personnage Joueur en le touchant et être tués quand on saute directement sur eux.

### Prérequis obligatoires :

- Avoir déjà conçu un niveau 2D avec gestion des collisions.
- Avoir déjà conçu un personnage jouable en 2D (ayant donc son *Sprite*, sa *CollisionShape2D* et son *script*).

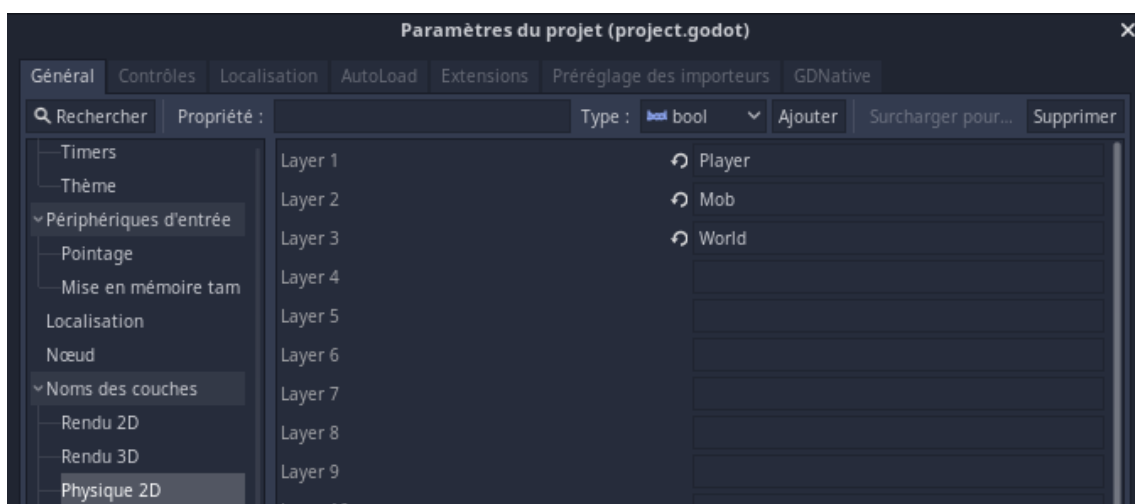
### Etapes à suivre :

#### **Etape 1 : organiser vos « layers »**

*(Ne pas refaire cette étape si vous l'avez déjà faite grâce au tutoriel « créer des chutes mortelles »)*

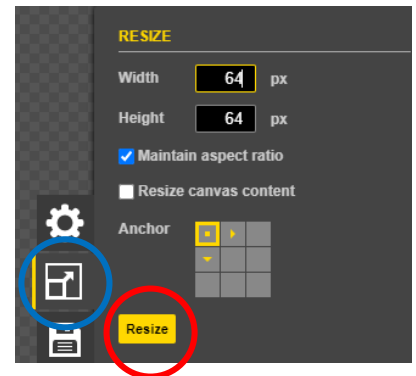
Les layers sont l'équivalent de « feuilles de calques » qui se superposent à l'écran pour donner une image d'ensemble au joueur : par exemple un calque pour le décor, un calque pour le personnage, un calque pour les ennemis... Dans Godot, l'usage des layers simplifie les interactions et collisions entre les différents éléments du jeu.

1. Ouvrez votre projet Godot
2. Cliquez sur l'onglet « Projet » en haut à gauche de la page, puis sur « paramètres du projet ».
3. Dans la page qui s'affiche, faites défiler la liste de gauche et cliquez sur « physique 2D »
4. Nous allons maintenant créer 3 layers : à côté de « Layer 1 » écrivez « Player » (ce sera la couche du personnage principal).
5. A côté de « Layer 2 » écrivez « Mob » (ce sera la couche des monstres, chutes mortelles et pièges).
6. A côté de « Layer 3 » écrivez « world » (ce sera la couche des décors). Vous devriez obtenir l'équivalent de l'image suivante.



## Etape 2 : dessiner le *SpriteSheet* de l'ennemi sur Piskel

7. Rendez-vous sur internet, sur le site [piskelapp.com](https://piskelapp.com) pour dessiner les images (*Sprites*) qui formeront les étapes des animations du monstre (*SpriteSheet*).
8. Il vous faut dessiner au moins deux animations pour votre monstre : une animation d'attente (*Idle*) et une de déplacement (*Walk*). Pensez à modifier avant de commencer la taille du *Sprite* sur Piskel (dans l'onglet « *Resize* », cerclé de bleu sur l'image de droite) pour qu'il soit format 64x64 pixels (enregistrez ce changement en cliquant ensuite sur « *resize* », cercle rouge sur l'image).



Je vous renvoie pour cette étape à notre tutoriel spécifique à Piskel :

<https://github.com/MajxClub/Tutoriels/blob/main/Graphiste/tuto%20graphiste%20facile%20-%20piskel.pdf>

Et au tutoriel pour y dessiner des animations en *SpriteSheet* :

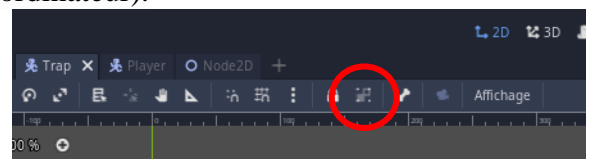
<https://github.com/MajxClub/Tutoriels/blob/main/Chara%20design/Tuto%20Chara%20design%20facile%20-%20SpriteSheet.pdf>



9. Vous devriez obtenir un *SpriteSheet* complet qui ressemble à celui ci-contre. Dans cet exemple de « Blob le monstre », en partant du premier *Sprite* en haut à gauche, les quatre premiers feront partie de l'animation *Idle*, les quatre suivants feront partie de l'animation *Walk* (ici orientée vers la gauche, comme le montrent les yeux du monstre).
10. Sauvegardez votre *SpriteSheet* en cliquant sur l'onglet « *Export* » et en choisissant le format PNG.
11. Depuis votre fichier de téléchargement Windows, déplacez l'image du *SpriteSheet* vers le dossier de votre projet GODOT. Pour l'instant nous allons laisser de côté cette image, elle sera utile pour l'étape 4.

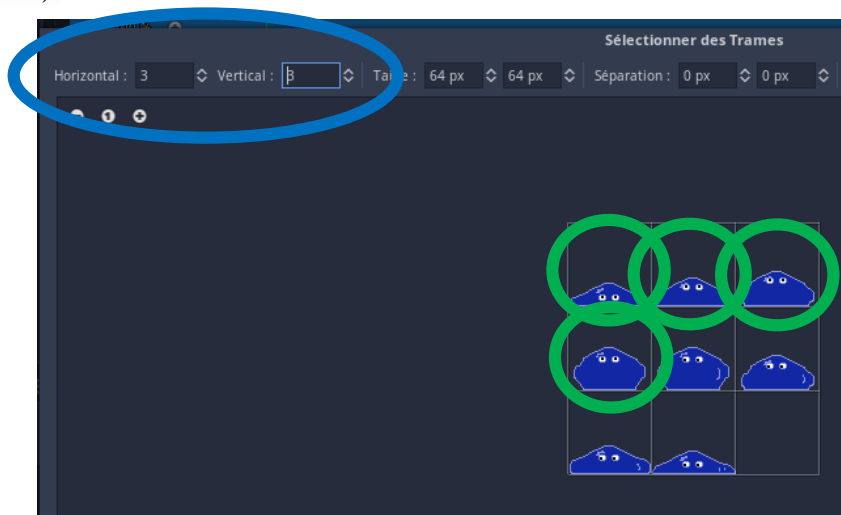
## Etape 3 : créer la scène de base de l'ennemi

12. Retourner dans votre projet GODOT, dans votre scène de niveau.
13. Ajoutez un nœud enfant à la scène principale, choisissez un *KinematicBody2D*. Cliquez ensuite deux fois (mais pas trop vite) sur ce dernier pour pouvoir changer son nom, nommez le par exemple « Monstre ».
14. Faites un clic droit sur « Monstre » et ajoutez à ce nœud un nœud enfant *AnimatedSprite*. Pour l'instant nous n'allons pas encore ajouter d'image à ce *Sprite* (nous le ferons à l'étape 4).
15. Faites un clic droit sur « Monstre » et ajoutez à ce nœud un 2<sup>e</sup> nœud enfant *CollisionShape2D*. Dans l'inspecteur à droite de l'écran, dans la propriété *Shape* de la *CollisionShape2D*, sélectionnez "Nouveau *RectangleShape2D*", nous redimensionnerons cette zone de collision plus tard, après avoir inclus les animations du monstre.
16. Faites un clic droit à nouveau sur le nœud « Monstre » et ajoutez à ce nœud un 3<sup>e</sup> nœud enfant *VisibilityEnabler2D*, grâce à ce nœud, le monstre ne s'activera que lorsque le personnage le verra à l'écran (cela permet d'économiser les ressources de votre ordinateur).
17. Dans l'interface 2D, cliquez sur le nœud « Monstre », puis cliquez sur l'icône « grouper les nœuds sélectionnés » (voir image de droite) ou le raccourci Control + G. Grâce à cela, tous ses nœuds enfants seront collés ensemble.
18. Dans l'interface 2D, faites un clic droit sur le nœud « Monstre » puis choisissez « Sauvegarder la branche comme scène ». Votre monstre est désormais une scène à lui tout seul. *Quand vous rouvrirez plus tard votre projet GODOT, Vous pourrez ouvrir cette scène soit en cliquant sur l'icône « clap de cinéma » à côté de son nœud, soit en double-cliquant sur « Monstre.tscn » dans la liste du « Système de fichiers »*



## Etape 4 : animer l'ennemi

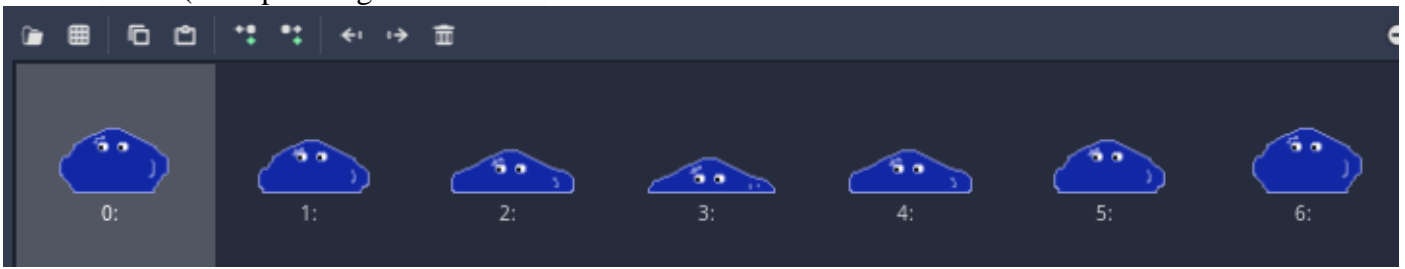
19. Ouvrez votre scène « Monstre ».
20. Dans l'inspecteur de l'*AnimatedSprite*, cliquez sur [vide] à côté de « Frames » et choisissez « Nouveau Spritesframes ».
21. Au même emplacement, cliquez sur « SpriteFrames », un nouvel onglet d'animation apparaît en bas de l'écran.
22. Dans cet onglet, cliquez sur l'icône du parchemin avec un « + » vert pour ajouter une nouvelle animation (voir image de droite).
23. Par défaut elle aura le nom « New Anim », double-cliquez dessus pour le modifier, appelons-le « Idle » (« inactif » en anglais).
24. Cliquez ensuite sur le symbole de grille en dessous de « Trames d'animation » (voir image de droite), ce qui permet de choisir dans le dossier votre image qui servira pour les animations.
25. Dans la page qui s'ouvre, choisissez votre image de *SpriteSheet* dans les ressources du dossier.
26. Votre image doit désormais apparaître dans une grille (voir image de gauche). Si vous avez dessiné vos *Sprites* de personnages en 64x64 pixels, la grille devrait être adaptée à chaque image. Si vous constatez un décalage, vous pouvez modifier les paramètres en haut de la page pour que chaque carré corresponde à une image (décalage de la grille, nombre de pixels par image, voir cercle bleu sur l'image ci-dessous).



27. Comme nous voulons ici faire une animation de marche, sélectionnez en cliquant dessus les images qui font partie de cette animation de marche (dans le bon ordre) voire cercles verts sur l'image au-dessus.
28. Cliquez ensuite sur « Ajouter ... Trame(s) »
29. Les images que vous avez choisies apparaissent dans l'onglet en bas. Si elles sont dans le désordre, vous pouvez le modifier avec les flèches situées sous « Trames d'animation : ».
30. Vérifiez une nécessité : la première et la dernière image de l'animation doivent être la même image du personnage statique. En utilisant les boutons « copier » et « coller » (cercle jaune ci-dessous) et les boutons « déplacer » (cercle rouge ci-dessous), vous pouvez également déplacer les images en faisant des glisser-déposer (maintenir le clic gauche et le relâcher à l'endroit que vous souhaitez). Votre animation « Idle » devrait ressembler à l'image ci-dessous.



31. Ajoutez ensuite une autre animation « Walk » (marche) en reprenant depuis l'étape 22, en sélectionnant les 4 autres images et en les modifiant comme à l'étape 30 pour obtenir une animation fluide (exemple image ci-dessous :



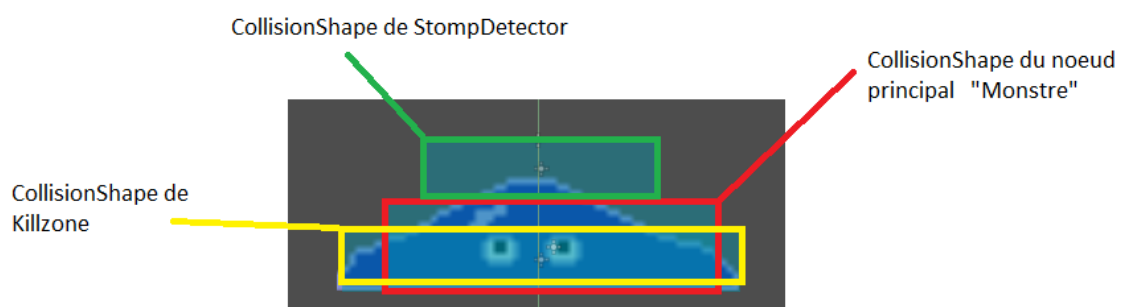
32. Quand vous avez ajouté toutes vos animations, cliquez sur le nœud *AnimatedSprite*, vous pourrez ensuite dans l'*Inspecteur* choisir l'image par défaut de votre monstre à côté de « Animation » (choisissez « Idle »). Vous pouvez également voir les animations être jouées si vous activez la case « Playing » de l'*Inspecteur* (mais désactivez-la avant de tester le jeu).

### **Etape 5 : Améliorer la scène du monstre**

33. Retournez dans la scène du monstre, maintenant que l'image du monstre est visible, adaptez la *CollisionShape2D* à ses dimensions (voir image « Les CollisionShape » plus bas)
34. Faites un clic droit sur son Nœud principal « Monstre » et ajoutez-lui un nœud enfant *Timer*.
35. Cliquez sur le nœud *Timer*, dans l'inspecteur à droite de l'écran, cliquez sur « Activé » à côté de « Autostart ».
36. Faites un clic droit sur le Nœud principal « Monstre » et ajoutez-lui un nœud enfant *Area2D*.
37. En cliquant doucement deux fois sur le nœud *Area2D*, changez son nom pour « *StompDetector* »
38. Faites un clic droit sur « *StompDetector* », ajoutez-lui un nœud enfant *CollisionShape2D*.
39. Cliquez sur son nœud *CollisionShape2D*, dans l'inspecteur, à côté de « Shape », cliquez sur « [vide] » et choisissez « Nouveau RectangleShape2D ».
40. Dans l'interface 2D (au centre de l'écran), adaptez cette nouvelle *CollisionShape2D* afin qu'elle recouvre la tête du monstre (voir image « Les CollisionShape » plus bas), c'est dans cette zone que le joueur sautera pour le tuer.
41. Faites un clic droit sur le Nœud principal « Monstre » et ajoutez-lui un autre nœud enfant *Area2D*.
42. En cliquant doucement deux fois sur le nœud *Area2D*, changez son nom pour « *Killzone* »
43. Faites un clic droit sur « *Killzone* », ajoutez-lui un nœud enfant *CollisionShape2D*.
44. Cliquez sur son nœud *CollisionShape2D*, dans l'inspecteur, à côté de « Shape », cliquez sur « [vide] » et choisissez « Nouveau RectangleShape2D ».
45. Dans l'interface 2D (au centre de l'écran), adaptez cette nouvelle *CollisionShape2D* afin qu'elle recouvre le corps du monstre, en étant plus bas que la zone de collision de *StompDetector* et plus large que le zone de collision du nœud principal (voir image « Les CollisionShape » plus bas), c'est dans cette zone que le joueur sera tué si il entre à son contact).

*Astuce : si vous devez gérer trop de zones de collision et qu'elles se ressemblent toutes, vous pouvez « cacher » une zone en cliquant sur l'icône de l'œil à côté de son nœud, pensez à bien la rendre visible à nouveau quand vous avez terminé.*

### **Les CollisionShape**



## Etape 6 : créer un script et des signaux

46. Cliquez sur le nœud principal « Monstre », puis cliquez sur l'icône du parchemin avec un + vert située au-dessus (voir image de droite) afin de lui ajouter un script.



47. Sur l'écran qui apparaît, cliquez sur « créer ».



48. Cliquez sur le nœud « StompDetector », puis à droite de l'écran, à côté de « Inspecteur », cliquez sur « Nœud » (cercle rouge sur l'image de gauche), vous avez ouvert le panneau des signaux.

49. Double-cliquez sur la ligne « body-entered(body : Node » (cercle vert sur l'image).

50. Sur la page qui s'ouvre, cliquez sur « Connecter », vous venez en créant un signal d'ajouter une fonction au script du monstre. Nous la remplirons plus tard dans le code.

51. Cliquez sur le nœud « Killzone », recréer un signal similaire comme pour le *StompDetector* (à côté de « Inspecteur », cliquez sur « Nœud » puis double-cliquez sur la ligne « body-entered(body : Node », cercle vert sur l'image).

52. Sur la page qui s'ouvre, cliquez sur « Connecter ».

53. Cliquez sur le nœud « Timer », puis créez lui également un signal comme précédemment, double-cliquez sur la ligne « timeout » et dans la page qui s'ouvre cliquez sur « Connection ».

## Etape 7 : finaliser les Layers de collisions

54. Cliquez sur le nœud principal « Monstre », puis dans « l'inspecteur » à droite de l'écran, cliquez sur la ligne « Collision » (celle placée sous la ligne « CollisionObject2D »).

55. Un tableau apparaît, vous pourrez y choisir le « Layer » d'appartenance du monstre ainsi que son « Mask » (ce « Mask » correspond aux *Layers* avec lequel devra interagir le nœud). Dans « Layer » activez le « 2 » (qui doit apparaître en surbrillance) et désactivez le « 1 ». Dans « Mask » activez les « 1 » et « 3 ».

56. Cliquez sur le nœud « StompDetector », puis refaisons une démarche similaire : dans « l'inspecteur » à droite de l'écran, cliquez sur la ligne « Collision » (celle placée sous la ligne « CollisionObject2D »). Activez ici le *Layer* « 2 », désactivez le *Layer* « 1 », puis dans « Mask » laissez le *Layer* « 1 » activé.

57. Plus haut dans « l'inspecteur » du *StompDetector*, désactivez la ligne « Monitorable ».

58. Cliquez sur le nœud « Killzone », puis refaisons encore une démarche similaire : dans « l'inspecteur » à droite de l'écran, cliquez sur la ligne « Collision » (celle placée sous la ligne « CollisionObject2D »). Activez ici le *Layer* « 2 », désactivez le *Layer* « 1 », puis dans « Mask » laissez le *Layer* « 1 » activé.

## Etape 8 : Ecrire le script de l'ennemi

59. Cliquez sur l'icône parchemin à côté du nœud « Monstre » pour ouvrir son script.

60. Nous allons entièrement réécrire le script, y compris les fonctions déjà présentes, recopiez le code suivant (ou copiez/collez depuis la page indiquée en dessous, mais ce n'est pas le meilleur moyen d'apprendre...). Vous trouverez plus bas une explication en détails de ces lignes de code.

[https://github.com/MajxClub/Codes-valides/blob/main/code\\_monstre\\_complet](https://github.com/MajxClub/Codes-valides/blob/main/code_monstre_complet)

*(attention, les flèches « → » sont des indentations obtenus en appuyant sur la touche TAB du clavier)*

```
extends KinematicBody2D

const GRAVITY = 1000
const UP = Vector2(0, -1)

var vel = Vector2()
var speed = 200
var dirx = 0
onready var animated_sprite = get_node("AnimatedSprite")

func _on_Timer_timeout():
→var m = int(rand_range(0, 10))
→if m < 5 :
→→dirx = -1
→elif m > 5 :
→→dirx = 1
→else:
→→dirx = 0

func _physics_process(delta):
→vel.x = 0
→vel.y += GRAVITY * delta
→movement_loop()
→move_and_slide(vel, UP)

func movement_loop():
→var right = (dirx == 1)
→var left = (dirx == -1)
→if dirx == -1 :
→→vel.x -= speed
→elif dirx == 1 :
→→vel.x += speed
→elif is_on_floor():
→→vel.y = -300

→if is_on_wall():
→→dirx += -1
→if right:
→→animated_sprite.play("Walk")
→if left:
→→animated_sprite.play("Walk")

func _on_Killzone_body_entered(_body : KinematicBody2D) -> void:
→get_tree().change_scene("res://GameOver.tscn")

func _on_StompDetector_body_entered(_body : KinematicBody2D) -> void:
→emit_signal("touch")
→_body.bounce()
→queue_free()
```



Explications de ce code :

```
extends KinematicBody2D

const GRAVITY = 1000 #
const UP = Vector2(0, -1)

var vel = Vector2()
var speed = 200
var dirx = 0
onready var animated_sprite = get_node("AnimatedSprite")

func _on_Timer_timeout():
    var m = int(rand_range(0, 10))
    if m < 5 :
        dirx = -1
    elif m > 5 :
        dirx = 1
    else:
        dirx = 0

func _physics_process(delta):
    vel.x = 0
    vel.y += GRAVITY * delta
    movement_loop()
    move_and_slide(vel, UP)

func movement_loop():
    var right = (dirx == 1)
    var left = (dirx == -1)
    if dirx == -1 :
        vel.x -= speed
    elif dirx == 1 :
        vel.x += speed
    elif is_on_floor():
        vel.y = -300

    if is_on_wall():
        dirx *= -1
    if right:
        animated_sprite.play("Walk")
    if left:
        animated_sprite.play("Walk")

func _on_Killzone_body_entered(_body : KinematicBody2D) -> void:
    get_tree().change_scene("res://GameOver.tscn")

func _on_StompDetector_body_entered(_body : KinematicBody2D) -> void:
    emit_signal("touch")
    _body.bounce()
    queue_free()
```

Indique que nous modifions un noeud KinematicBody2D

Cette constante participe à la création d'une gravité (les corps sont attirés vers le bas)

Indique la direction du "haut"

Indique que les déplacements de ce KinematicBody se feront dans l'environnement 2D de l'écran

Définit la vitesse de déplacement en pixels par seconde

Dits au code d'activer le noeud "AnimatedSprite"

Cette fonction est liée au noeud "Timer", elle indique ce qui se passe quand le chrono arrive à sa fin

Crée une variable "m" à partir d'un nombre aléatoire

Modifie la direction horizontale du monstre selon le résultat du nombre aléatoire

Participe à la gestion des déplacements (delta = à chaque frame, par exemple la deuxième ligne applique la gravité sur l'axe "y", donc vertical, à chaque frame).

Cette fonction gère les déplacements de personnages (par exemple vel.x est le déplacement sur l'axe horizontal, vel.y l'axe vertical)

Grâce à ces deux lignes, le monstre se retourne quand il touche un mur

Ces 4 lignes gèrent les animations selon la direction que prend le monstre (ici très simplement, une seule animation est jouée)

Cette fonction est liée au signal de l'Area2D "Killzone", la première ligne demande ce qui se passe quand un KinematicBody y entre (comme le joueur), la 2e ligne en donne l'effet, le joueur est envoyé vers la scène de Game Over

Cette fonction est liée à l'Area2D "StompDetector", quand le KinematicBody du joueur y entre, un signal "touch" est émit (il n'apparaît pas en jeu)

La ligne "body.bounce" renvoi à une fonction dans le code du personnage joueur qui le fait rebondir

La dernière ligne fait disparaître le monstre (tué par le personnage)

## Etape 9 : améliorer le personnage joueur

61. Allez ensuite dans la scène du personnage jouable, nous allons lui ajouter une fonction « bounce » qui fonctionne avec l'avant dernière ligne du code du monstre.
62. Ouvrez le script du personnage et écrivez le code suivant tout en bas du script :

```
func bounce():
    velocity.y = jump_speed * 0.6
```

(attention, les flèches « → » sont des indentations obtenus en appuyant sur la touche TAB du clavier)

Grâce à cette fonction, votre personnage rebondira après avoir tué un monstre en sautant dessus.

63. Bonus : si vous voulez ajouter un son lorsqu'on saute sur la tête d'un monstre, suivez les lignes ci-dessous, si cela ne vous intéresse pas, passez directement à l'étape 10.
64. Commencer à télécharger un son (libre de droit, ou créez-le), format .mp3 ou .wav, et placez-le dans votre fichier du projet GODOT.
65. Dans la scène de votre personnage joueur, faites un clic droit sur son nœud principal et ajoutez un nœud enfant *AudioStreamPlayer*.
66. Cliquez deux fois sur ce nouveau nœud afin de changer son nœud, nommez-le « AudioSaut »
67. Ouvrez le script du personnage joueur et ajoutez la ligne suivante en bas de la fonction « bounce »

```
→$AudioSaut.play()
```

*(attention, les flèches « → » sont des indentations obtenus en appuyant sur la touche TAB du clavier)*

### **Etape 10 : Placer un ennemi dans son niveau**

68. Si la scène « Monstre » est déjà présente dans la liste des nœuds de votre scène de niveau, et que vous ne l'avez pas déplacé depuis le début, le monstre doit être située en haut à gauche de l'écran du joueur sur l'interface 2D, vous pouvez le déplacer ou bon vous semble dans votre niveau.
  69. Si vous avez construit la scène monstre à part, pour l'ajouter dans votre niveau procéder comme suit : ouvrez votre scène de niveau, puis faites un glissé-déposé (en maintenant le clic gauche) de votre scène *Monstre.tscn* présent dans le « Système de fichiers » vers le nœud principal de votre niveau, le monstre sera ajouté comme un nouveau nœud enfant du niveau.
  70. Si vous voulez créer plusieurs monstre, faites un clic droit sur le nœud « Monstre » dans la scène de niveau et sélectionnez « dupliquer ». Un second monstre a été créé (attention dans l'interface 2D il est superposé au premier). Vérifiez qu'il n'est pas un nœud enfant du premier monstre mais bien un nœud indépendant (car les 2 seraient tués à la mort du premier monstre).
- Bravo pour avoir suivi ce long tutoriel ! Vous gagnez un point comme **développeur** et un point comme **animateur** dans votre livret de compétence !