

- ❖ Tutoriel de **développeur**
- ❖ Niveau **Hard**



Ajouter un système de scores dans un plateformer 2D

Rappels :

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

Objectifs du tutoriel :

- Créer des objets 2D collectables qui augmentent le score du joueur quand son personnage les touche dans le niveau.
- Afficher en fin de partie le score du joueur.

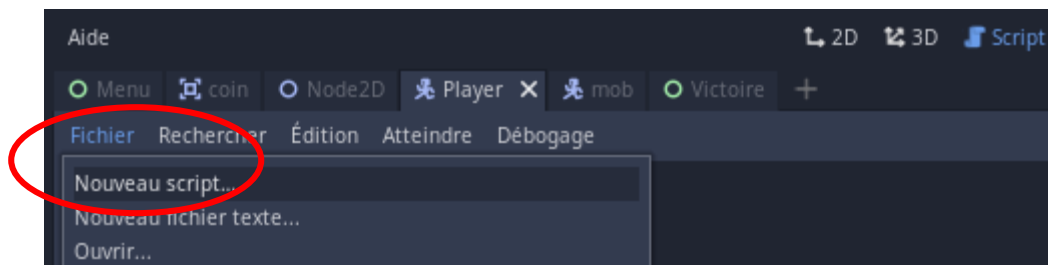
Prérequis obligatoire :

- 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).
- Avoir téléchargé ou créé un *sprite* pour l'objet collectable et un son qui sera activé en le récupérant.

A/ Création du script « Manager » :

Le script *Manager* que nous allons créer enregistrera le score et gèrera son évolution. Il ne sera pas lié à une scène en particulier mais commun à toutes les scènes du jeu.

1. Ouvrez votre projet Godot.
2. Dans GODOT, cliquez sur « Script » en haut au milieu de la page, puis sur « Fichier » et enfin « Nouveau script » (voir image ci-dessous).
3. Dans l'encart qui s'ouvre, vous pouvez changer son nom, choisissez « Manager », puis cliquez sur



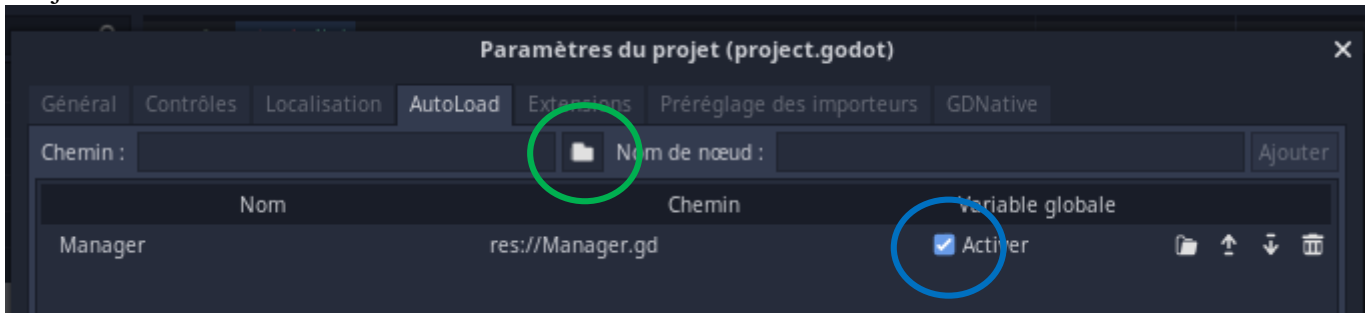
« Créer ».

4. Dans son code qui s'affiche, nous allons indiquer au jeu que le score à son lancement est de zéro. En dessous de la ligne « extends Node », écrivez le code suivant :

```
var score = 0
```

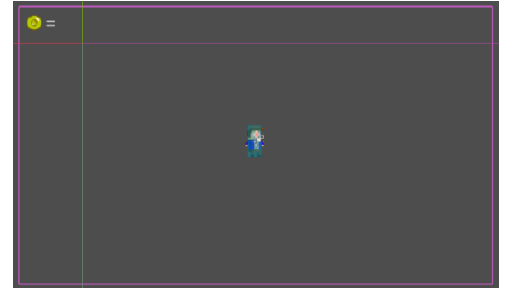
5. Cliquez ensuite sur « Projet » en haut à gauche de l'écran puis sur « Paramètres du projet ». Dans l'onglet « AutoLoad », cliquez sur l'icône de fichiers (cercle vert sur l'image ci-dessous) et sélectionnez parmi les fichiers le script *Manager.gd*.

- Une ligne « Manager » est apparue dans l'onglet « Autoload », cliquez sur « Activer » à sa droite (cercle bleu dans l'image ci-dessous). Grâce à cette manipulation, le script se lancera pour toutes les scènes du jeu.



B/ Affichez le score à l'écran pendant la partie :

Nous voulons maintenant que le score s'affiche durant le jeu dans un coin de l'écran du joueur, et évolue à chaque fois que le joueur ramasse un collectable.



- Allez dans la scène de votre personnage-joueur (ici je l'appelle « Player »).
- Ajoutez à la scène un nœud enfant *Sprite* et faites glisser une image du collectable qui ajoute du score depuis la liste du « système de fichiers » (en bas à gauche) vers la case « Vide » à côté de « Texture » dans l'inspecteur (en haut à droite). Moi j'ai par exemple utilisé une image de pièce d'or.
- Dans l'écran 2D central, placez ce *Sprite* dans un coin de l'écran du joueur (cadre rose). Il apparaîtra ainsi sur l'écran du joueur.
- Ajoutez à la scène un nœud enfant *Label*, dans l'écran 2D, placez son cadre juste à la droite de l'image du collectable que nous avons placé à l'étape 3.
- Dans l'inspecteur de ce nœud *Label*, écrivez comme texte le signe « égal » (=).
- Cliquez sur le nœud *Label* puis sur l'icône de parchemin pour lui ajouter un *script*.
- Dans le *script* du nœud *Label*, écrivez sous la ligne « extends Label » le code suivant :

```
func _process(delta):  
    self.text = str(Manager.score)
```

Nous venons de créer une fonction (func) qui est active en permanence (_process(delta)). Elle modifie en direct le contenu de notre Label selon le score du joueur en puisant ses informations dans le script « Manager » que nous avons créé précédemment.

- Nous allons ajouter un élément à la scène Player qui sera utile plus tard : nous allons indiquer que le personnage-joueur fait partie d'une classe nommée « Player », l'objectif est d'aider GODOT à faire interagir le personnage avec les autres scènes (comme les collectables). Dans le script principal de la scène Player, sous la ligne « extends KinematicBody2D », ajoutez la ligne suivante :

```
class_name Player
```

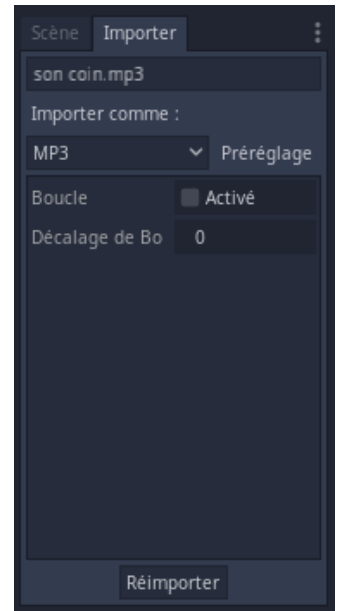
C/ Jouer un son quand le personnage ramasse un collectable :

- Importez dans les fichiers du jeu un son qui devra être joué quand le personnage ramasse un collectable (moi j'ai mis un son de clochette téléchargé depuis le site internet <https://lasonotheque.org/>).

2. Dans GODOT, retrouvez votre son dans la liste du « Système de fichiers », cliquez dessus, puis cliquez sur « importer » en haut à gauche de l'écran et désactivez la « boucle » (afin d'éviter que le son se joue sans arrêt !), puis enfin cliquez sur « Réimporter » (voir image de droite).
3. Ajoutez à la scène « Player » un nœud enfant *AudioStreamPlayer*.
4. Faites un double-clic sur ce nœud dans l'arbre de scène pour modifier son nom, appelons-le « AudioCoin».
5. Dans le script principal de la scène Player, ajoutez la fonction suivante :


```
func coinsound():
    $AudioCoin.play()
```

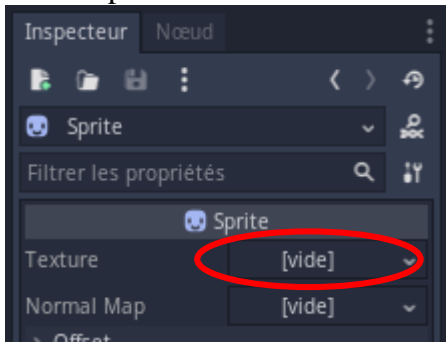
Nous venons de créer une fonction (que nous avons appelé "coinsound") qui jouera le son choisi quand elle sera active par le script du collectable (voir étape suivante).



D/ Créer la scène du collectable :


Nous allons enfin créer une scène dédiée au collectable qui augmentent le score quand ils sont ramassés par le personnage (ici des pièces d'or).

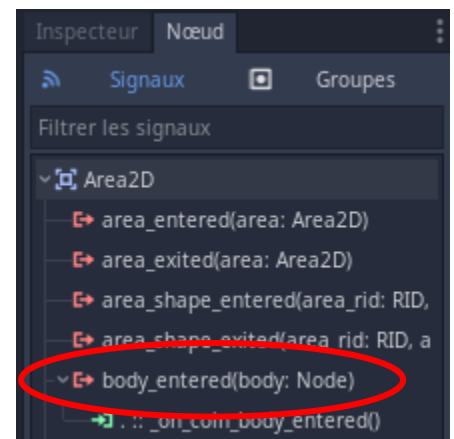
1. Dans votre scène de niveau de jeu, créez un nœud enfant « Area2D ».
2. Faites un double-clic sur ce nœud dans l'arbre de scène pour modifier son nom, appelons-le « Coin» (« pièce de monnaie » en anglais).
3. Faites un clic droit sur ce nœud et choisissez « sauvegarder la branche comme scène ».
4. Cliquez sur l'icône « clap de cinéma »  apparue à côté du nœud afin d'ouvrir cette scène indépendamment.



5. Dans cette scène Coin, ajoutez un nœud enfant *Sprite* et choisissez comme texture dans l'inspecteur (voir cercle rouge à gauche) une image du collectable que vous avez préalablement inclus dans les fichiers du jeu, une pièce d'or, par exemple.

(Vous pouvez remplacer ce Sprite immobile par un nœud AnimatedSprite, en utilisant une image découpée de manière à former une animation (ex : images de pièce ci-contre qui tourne sur elle-même une fois animée), je vous renvoie dans ce cas vers les tutoriels dédiés à l'animation).

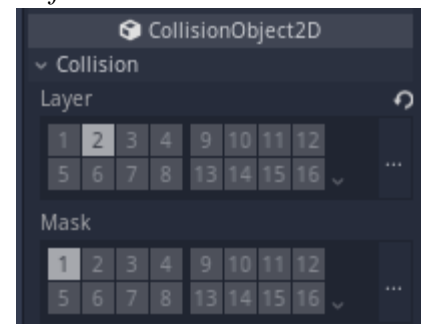
6. Ajoutez à la scène un autre nœud enfant « CollisionShape2D ».
7. Dans l'inspecteur, à côté de « Shape », cliquez sur « vide » pour définir la zone de collision de votre choix (*CircleShape2D* ou *RectangleShape2D*).
8. Cliquez sur le nœud CollisionShape2D (se zone de collision est apparue en bleu ciel sur l'écran central 2D) et à la souris faites en sorte qu'elle soit superposée au *Sprite* (ou l'*AnimatedSprite*).
9. Cliquez sur le nœud racine Coin, puis sur l'icône script  pour lui ajouter un script.
10. Cliquez sur le nœud racine Coin, puis dans « Nœud » à droite de l'inspecteur, cliquez sur « body_entered(body : Node) » (voir image de droite).
11. Dans le petit écran qui apparaît, cliquez sur « Connecter ».
12. Allez dans le script de la scène, sous la ligne « extends Area2D », supprimez la fonction déjà écrite et remplacez-la par le code suivant (attention aux *indentations*, obtenus avec la touche TAB du clavier) :



```
func _on_Coin_body_entered(_body : KinematicBody2D) -> void:
    if _body is Player :
        Manager.score += 1
        _body.coinsound()
    queue_free()
```

- Nous venons de créer une fonction qui s'active quand un KinematicBody2D touche la zone de collision de notre pièce d'or (1ère ligne)
- La 2^e ligne crée une condition : la fonction ne s'activera que si c'est le personnage « Player » qui touche la pièce (cela évite que les ennemis récupèrent nos collectables !).
- La 3^e ligne interagit avec le script « Manager » et ajoute 1 point au score lors du contact entre le personnage et la pièce.
- La 4^e ligne fera jouer le son que nous avons configuré dans la scène du personnage.
- La 5^e ligne fait disparaître le collectable une fois que le score a été modifié.

13. Ajustons maintenant les *layers de collision* : cliquez sur le nœud racine « Coin », puis dans l'inspecteur, cliquez sur « Collision ». Définissez le « Layer » du collectable en « 2 » (c'est le layer que j'utilise pour les ennemis) et celui du « Mask » sur 1 (c'est le layer de mon personnage, afin qu'ils puissent interagir) (voir image de droite).



E/ Remettre le score à zéro en cas de game over :

Ici, nous verrons comment faire retomber le score à zéro en cas de mort du personnage (game over).

Possibilité 1 : Si vous avez déjà une scène de game over qui s'affiche en cas de mort :

1. Allez dans le script de votre scène de game over.
2. Sous la ligne « extends Control », ajoutez la fonction suivante :

```
func _process(delta):
    Manager.score = 0
```

Possibilité 2 : vous n'avez pas de scène de game over :

1. Allez dans les scripts de toutes les scènes qui entraîne la mort (ennemis, pièges, chutes mortelles).
2. Ajoutez le code suivant :

```
func _process(delta):
    Manager.score = 0
```

(Si ces scènes ont déjà une fonction « func _process(delta) », ajoutez uniquement la 2^e ligne à l'intérieure de celle-ci (et indentée avec TAB)

F/ BONUS : faire apparaître le score dans la scène de fin de niveau :

Ces manipulations sont utiles si vous avez créé une scène de fin de niveau ou de fin de jeu. Nous allons y afficher le score du joueur.

1. Allez dans votre scène de fin de niveau.
2. Reproduisez les étapes 2 à 7 de notre précédente partie « **B/ Affichez le score à l'écran pendant la partie** ». Vous pouvez modifier l'emplacement du *Sprite* de collectable et du *Label* (par exemple pour les placer au centre de l'écran).