

# Introduction à GDScript

Zel

February 2024

## 1 Le concept de nodes

Dans Godot, chaque objet est une node. Chaque Node a ses propres caractéristiques modifiables depuis l'inspecteur avec son usage spécifique. (ex: Polygon2D, Sprite2D, AnimationPlayer, etc...).

Toutes les nodes héritent de 4 nodes principales:

- Node2D: correspondant à toute la logique visuelle et physique sur un plan 2D
- Node3D: de même pour un espace 3D
- Control: Correspondant à tout ce qui touche à l'interface utilisateur
- Les nodes système: tout ce qui touche à la musique, les interactions réseau etc...

## 2 GDScript

GDScript est le langage de scripting principal de Godot. Il est possible d'utiliser C#, Rust et C++ mais de manière plus limitée avec moins d'intégration avec le reste de l'éditeur. La syntaxe est très proche de celle du python et la documentation intégrée dans le moteur rend le scripting très accessible.

Pour ajouter un script à une node, vous pouvez simplement faire clic droit => attacher un script. Il sera éditable depuis l'onglet script en haut de l'écran.

Chaque script a plusieurs fonctions de base à implémenter, voici les principales:

- `_ready()`: la fonction qui sera lancée à l'instanciation d'un objet
- `_process(delta)`: la fonction qui sera exécutée à chaque frame du jeu. `delta` correspond au temps écoulé depuis la dernière frame.
- `_physics_process(delta)`: la fonction exécutée chaque frame du jeu avant les opérations du moteur physique. Toute opération modifiant la physique d'un corps doit être exécutée/appelée dedans.

## 3 La Syntaxe

### 3.1 Les variables

Les variables sont définies à l'aide du mot var:

```
var n : int = 5
```

le type n'a pas à être nécessairement défini, mais il aide pour l'autocomplétion dans le cas où il n'est pas évident.

Il est possible d'utiliser deux mots clés avant la définition d'une variable:

- `@onready var x = ...` : garrantit que la variable sera initialisée à l'instanciation avant l'exécution de la fonction ready
- `@export var x = ...` : permet de modifier la variable depuis l'inspecteur. Pratique pour les valeurs qui vont être beaucoup modifiées pour le debug. (ex: vitesse d'un personnage, etc...). Il est possible d'en faire un slider, checkbox etc... La doc a toutes les informations à ce propos.

### 3.2 Les fonctions

Les fonctions sont défini avec le mot clé "func" tq:

```
func ma_fonction(PremierArgument, DeuxiemeArgument):  
<corp de la fonction>
```

## 4 Les Signaux

Les nodes communiquent entre elles en émettant et recevant des signaux. Un signal est émi à l'aide la méthode "emit()" du type signal: `monsignal.emit(mesarguments)`

On peut lié un signal à une node via l'éditeur, et la fonction associée sera exécutée lorsque la node émettra le signal.