Principes de design (beta)

# Composition

L’idée est de créer des librairies de scènes qui fonctionnent avec d’autres scènes quelconques.

## Exemple : oscillation d’un objet

Au lieu de faire un script qui oscille l’objet courant, il est pr

# Interface de signaux

On préfère créer nos objets et exposer des signaux pertinents. C’est préférable que de devoir appeler des méthodes directement sur les objets.

1. # enemy.gd

2. extends Area2D

3.

4. signal died # déclaration du signal

5.

6. func \_on\_hit():

7. emit\_signal("died", self) # émission du signal

8. queue\_free()

9.

1. # enemy\_manager.gd

2. extends Node2D

3.

4. func \_ready():

5. var enemy = preload("res://enemy.gd").new()

6. add\_child(enemy)

7.

8. # Connexion du signal `died` de l’ennemi à une méthode locale

9. enemy.died.connect(\_on\_enemy\_died)

10.

11. func \_on\_enemy\_died(enemy\_ref):

12. print("Un ennemi est mort :", enemy\_ref.name)

13.