

LOG725 - Ingénierie et conception de jeux vidéo

# Labo 5 - Rendering

Gabriel C. Ullmann

École de Technologie Supérieure, Hiver 2024



Le génie pour l'industrie

# Objectifs d'apprentissage

- Comprendre rendering de manière générale
- Comprendre les nœuds Godot:
  - AnimatedSprite
  - Camera
  - Viewport
- Réfléchir sur quelques problèmes liées au rendu
- Comprendre comment bien utiliser ces nœuds

# Activités



Revision:  
scène de base



Ajouter un obstacle  
et collision



Ajouter animation  
et camera (+challenge 1)



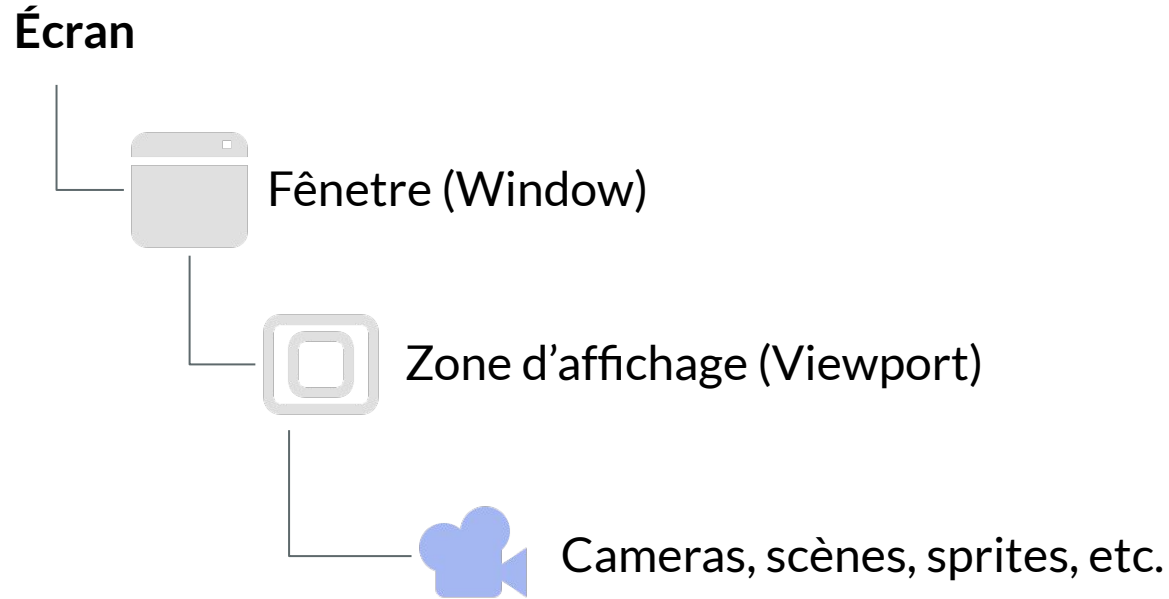
Ajouter des viewports  
(+challenge 2)

# Rendering

- Produire une image 2D/3D en utilisant des algorithmes (rendu).
- Nous pouvons prendre en compte les vrais phénomènes physiques (ou non).
- Métaphores courantes : fenêtre, caméra, rayons (éclairage).



# Métaphores dans les moteurs de jeu



# Une scène de base



Personnage (CharacterBody2D)

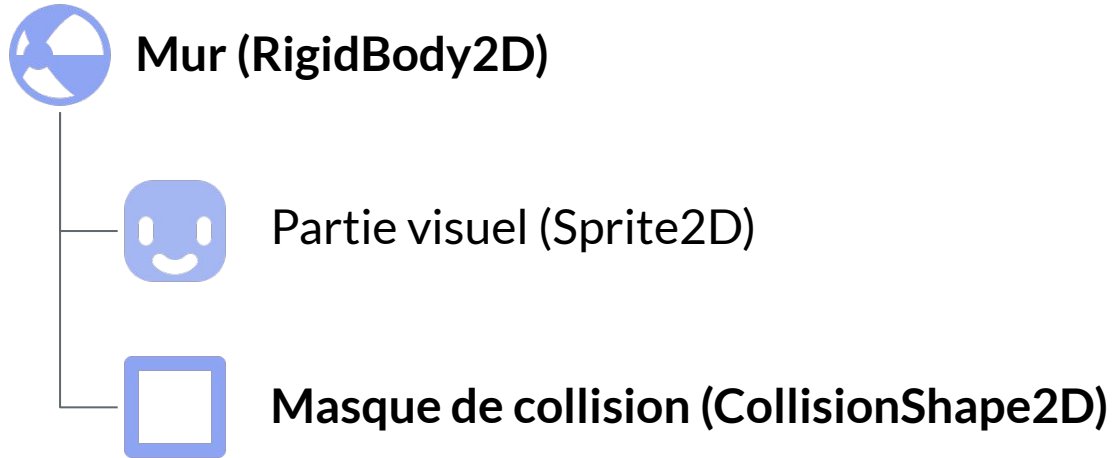


Partie visuel (Sprite2D)

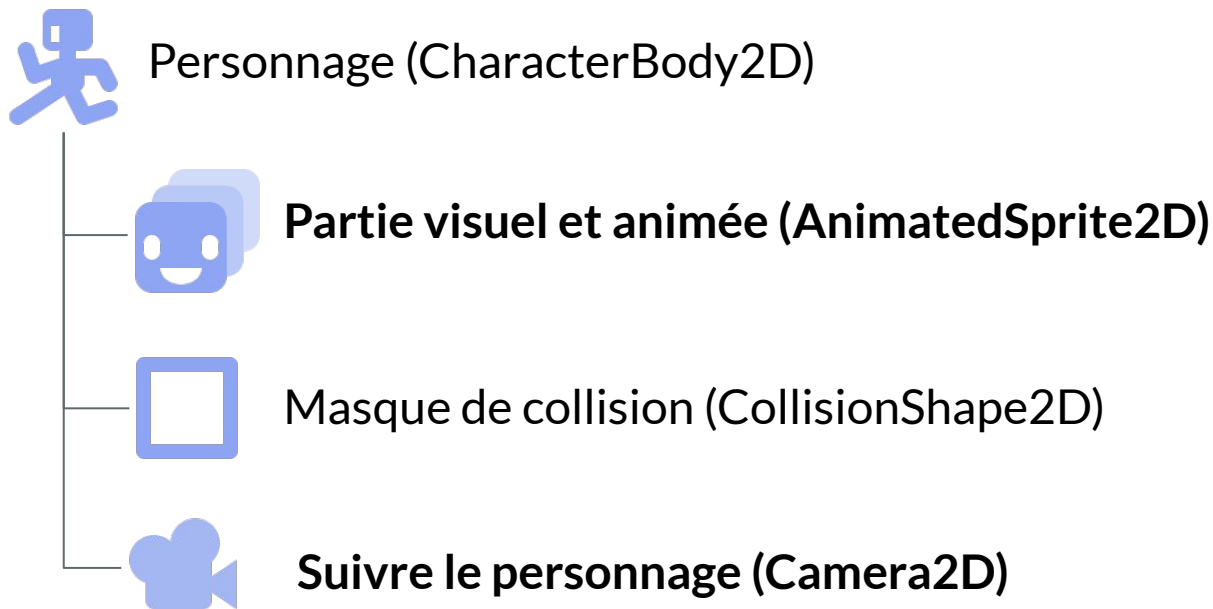


Masque de collision (CollisionShape2D)

# Example 1: ajouter les obstacles collisionnables



## Exemple 2: créer le comportement “side scroller”



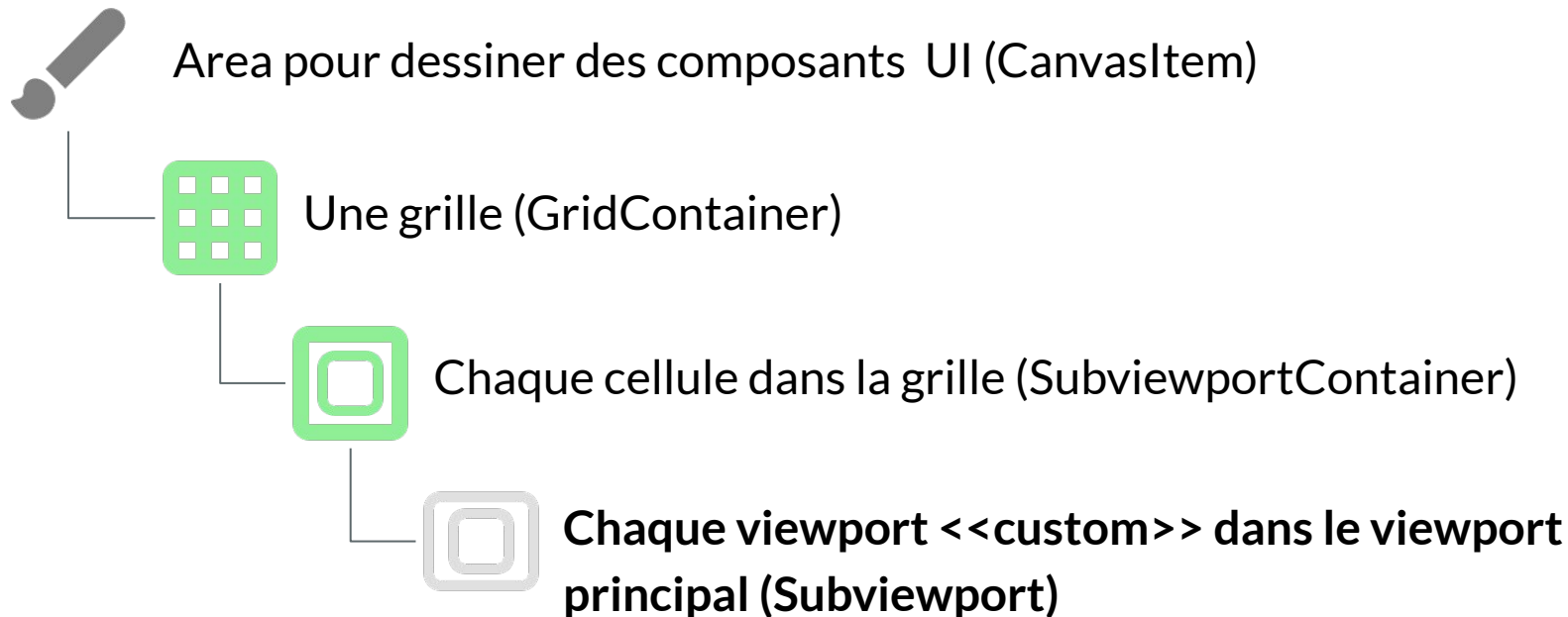


# Challenge 1: play/stop

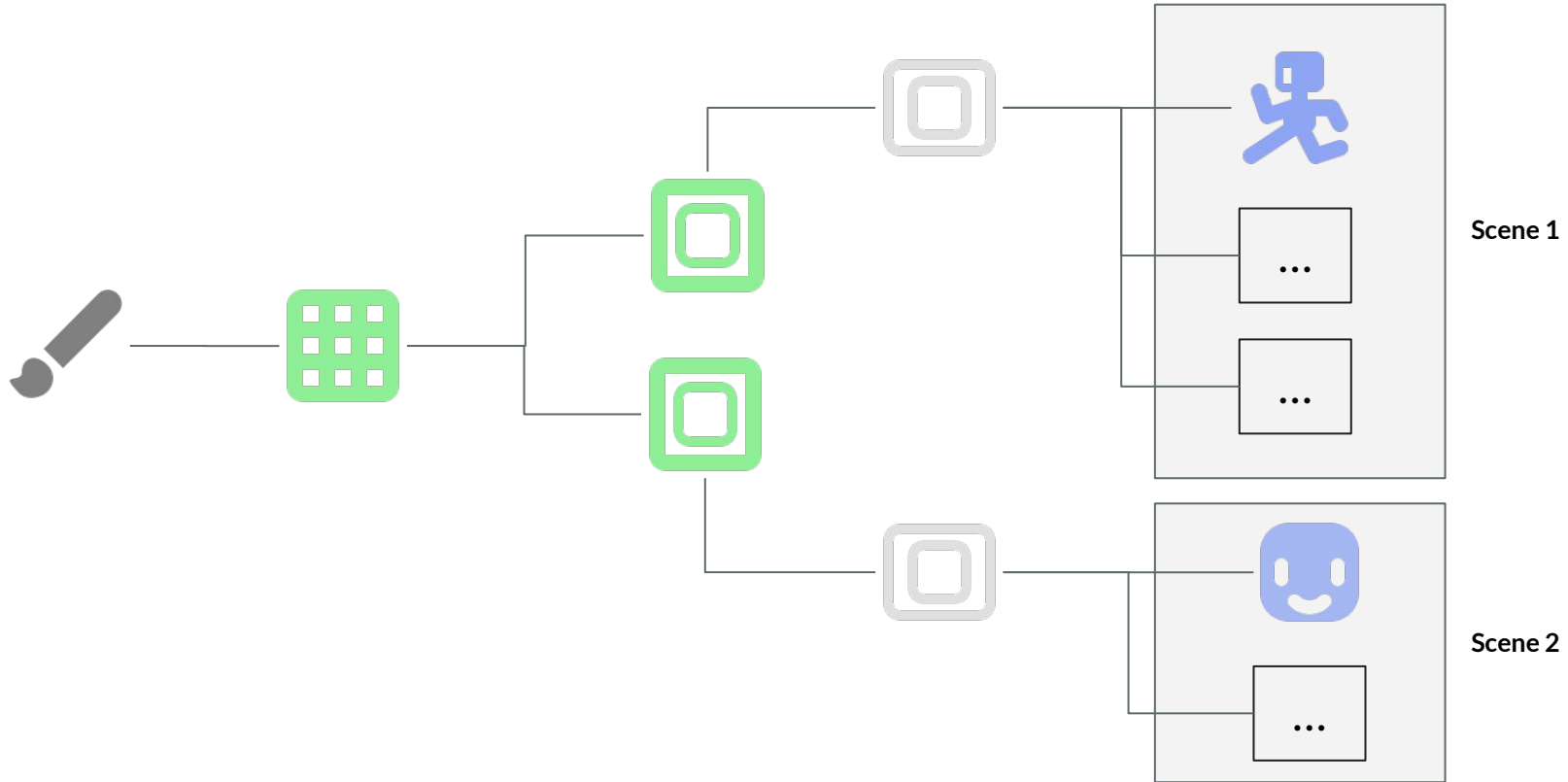


Comment pouvons-nous contrôler les changements d'état d'animation ?

## Example 3: ajouter plusieurs viewports

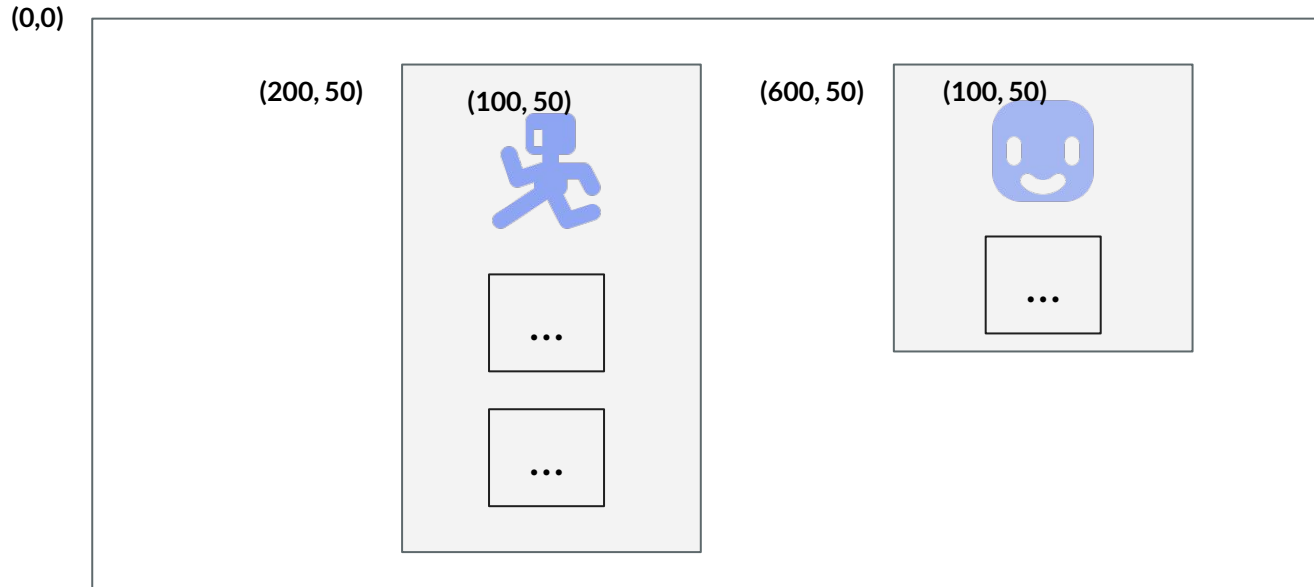


## Example 3: ajouter plusieurs viewports

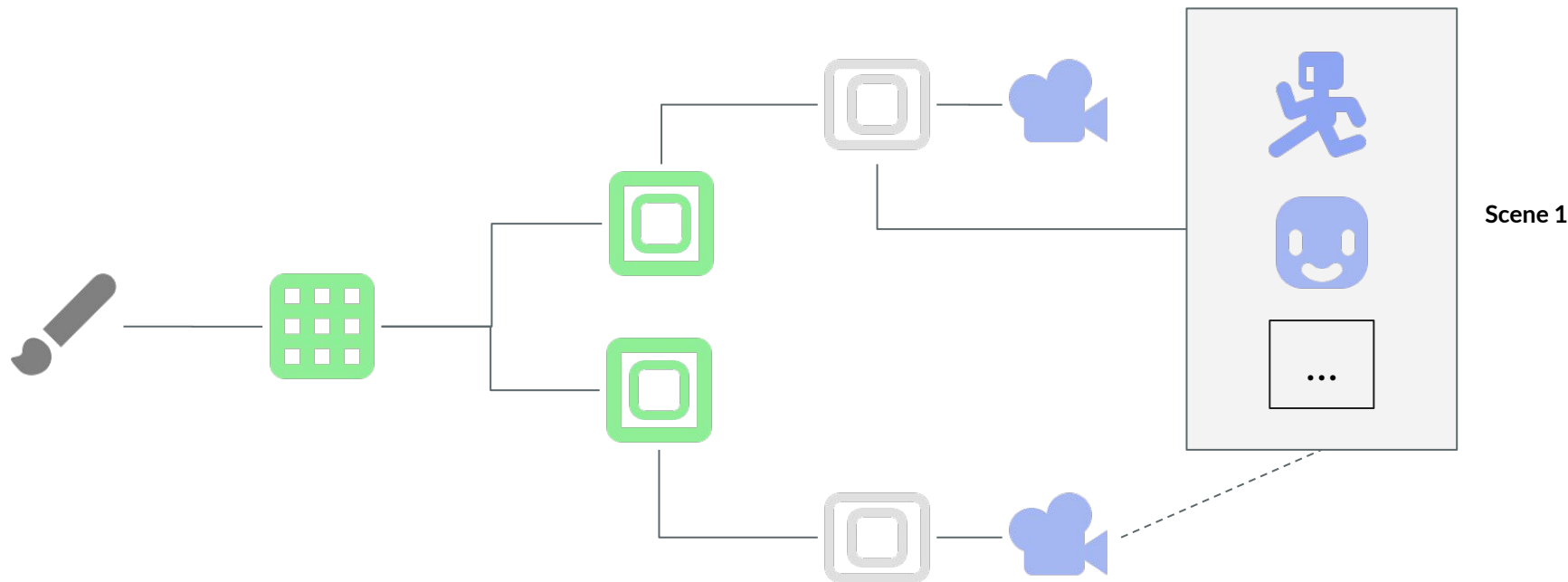


## Challenge 2: positions dans le viewport

Comment pouvons-nous dessiner les sprites par rapport à son viewport parent plutôt que par rapport au viewport principal ?



## Example 4: viewports sincronizés à la même scène



# Conclusion

- Les moteurs de jeu s'occupent de l'abstraction de plusieurs concepts (caméra, viewport, etc.).
- Les bibliothèques bas-niveau ne nous donnent pas ces abstractions.
- Faire attention aux différences de nomenclature (par exemple, entre moteurs de jeu).

LOG725 - Ingénierie et conception de jeux vidéo

# Labo 5 - Rendering

Gabriel C. Ullmann

École de Technologie Supérieure, Hiver 2024



Le génie pour l'industrie