

# Star engine

Equipe :  
NITNINDO

Nathan Souvignet  
Xihao Wang



# Structure

Modulaire / Optimisé

## Rendu

Render	PBRrender	AnimatedPBRrender	InfosRender	LightRender
Transform	Transform	Transform	Transform	Transform
Drawable	Drawable	AnimatedDrawable	Drawable	Light
CustomProgram	Material	Material		

## Autre

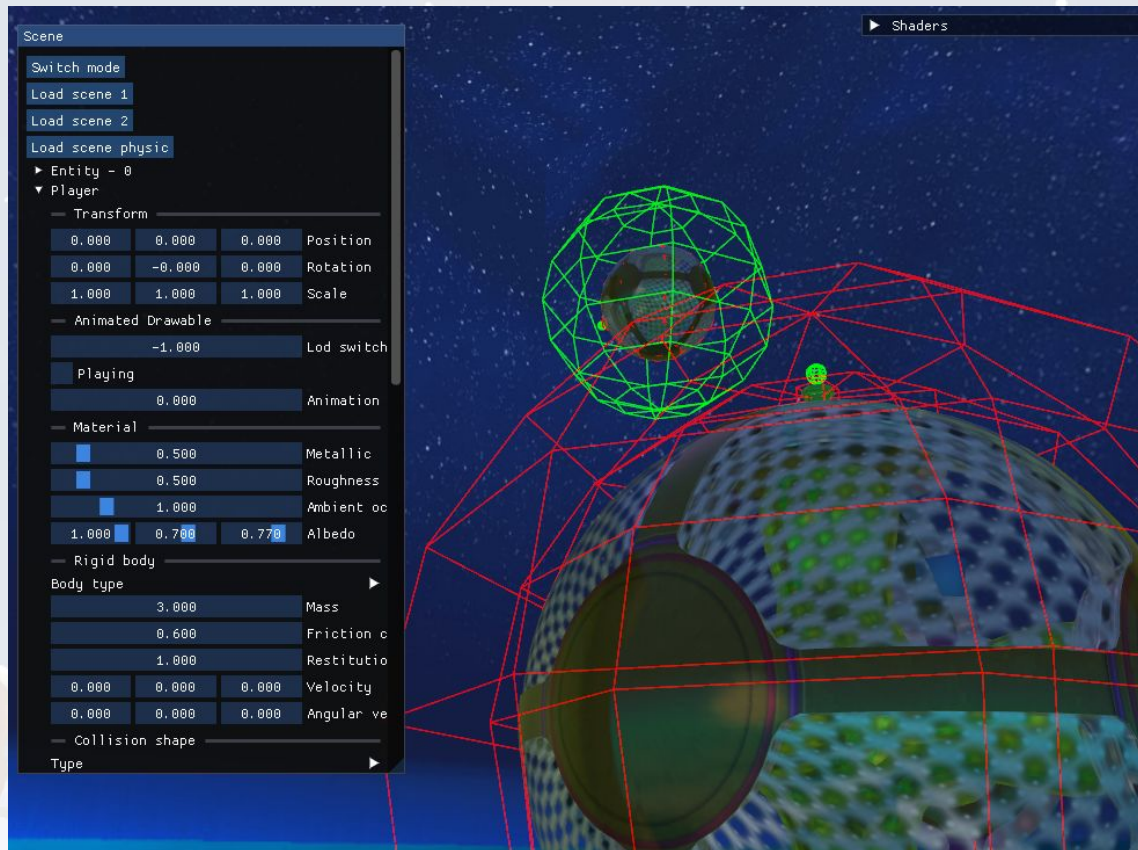
CameraSystem	CustomSystem
Transform	CustomBehavior
CameraComponent	CustomVar

## Physique

CollisionDetectionSystem	PhysicSystem	PhysicDebugSystem
Transform	Transform	Transform
CollisionShape	RigidBody	CollisionShape

# L'éditeur

- Edition des composants
- Changement de scène
- Prise de photo
- Debug

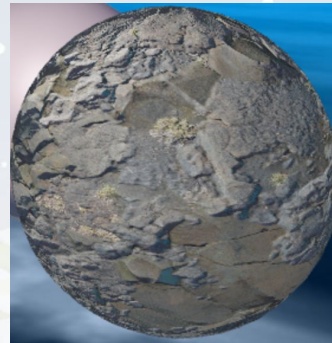
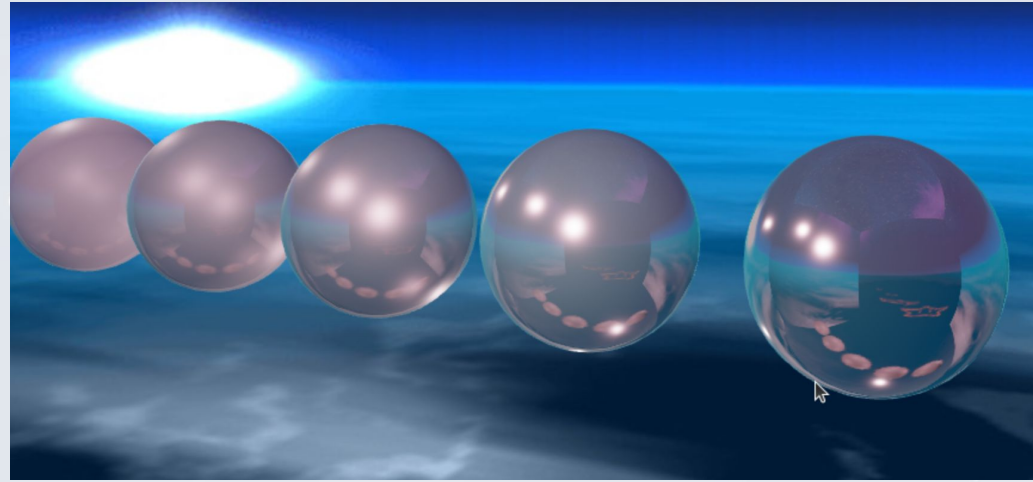




# PBR

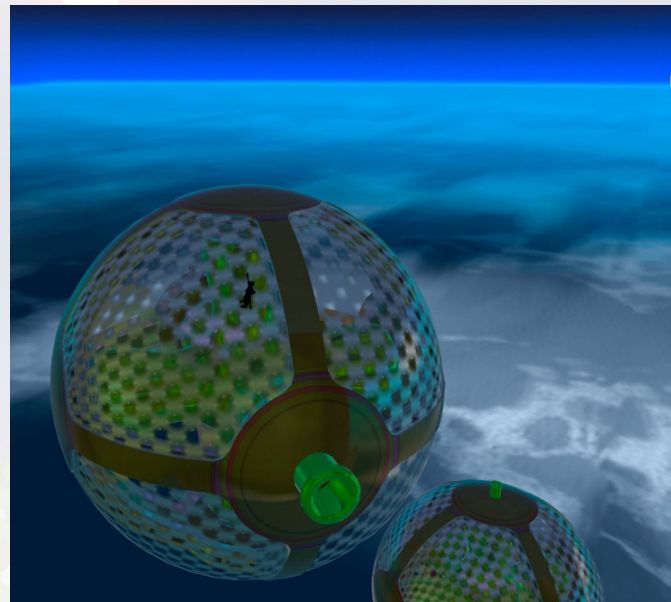
- Les base :

- Lumières
- Métallité
- Rugosité
- AO
- Albédo
- Normale



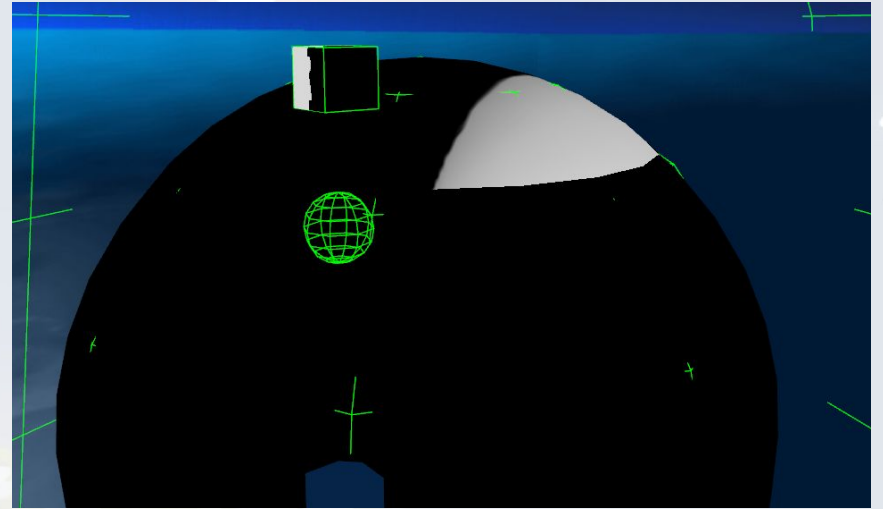
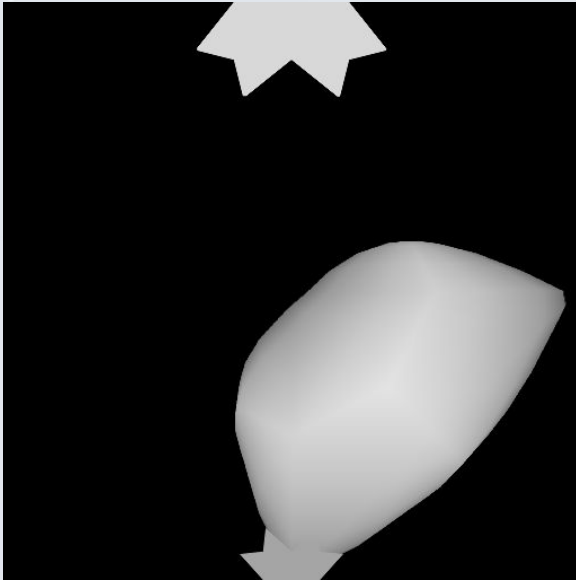
# PBR

- système flexible
  - Entité : *material* (5 textures)
  - Assimp -> pointeur
- Diffuse et spéculaire
  - L'intégration des lumières
  - L'ajoute et la lecture de la boîte de ciel
- Comment appliquer dans notre système
  - initialisation du programme PBR
  - Chargement des textures PBR
  - Génération de la cubemap d'environnement
  - Enregistrement dans l'ECS



# Shadow mapping

Projection octaédrique

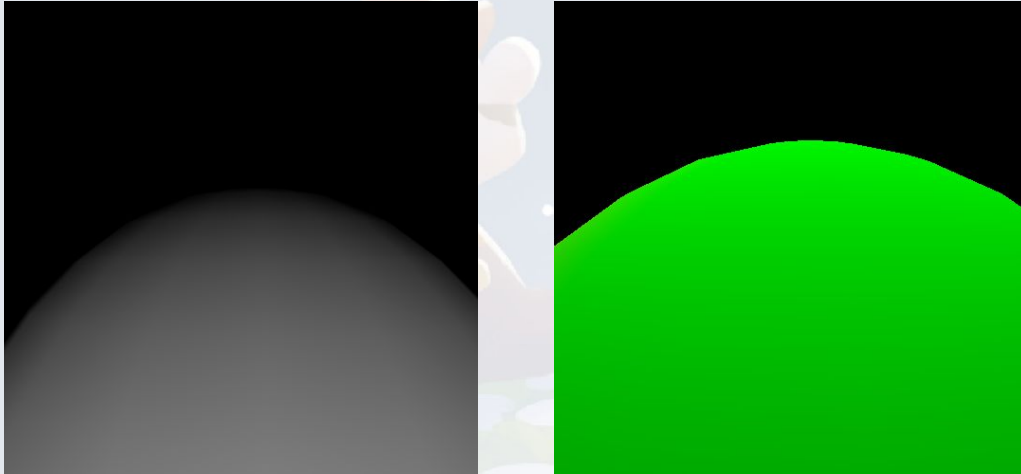


# Outils

#include dans shaders

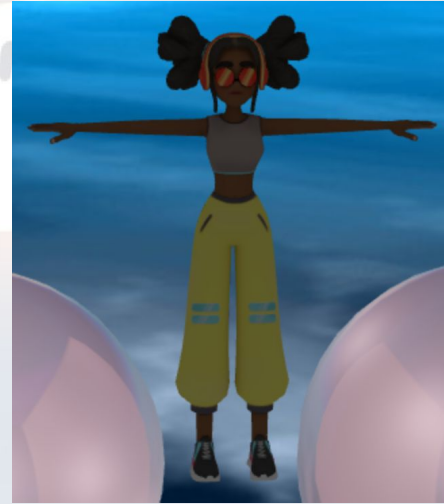
Rendu de la scène dans une cubemap / sur une texture

Rendu d'informations



# Animations

- Deux jeux de données
  - extraits par Assimp
  - format .glb
  - la bind pose de chaque os
  - les keyframes d'animation
- Le calcul de la pose animée d'un os
  - ramener l'os dans l'espace de la racine
  - accumulation et interpolation des transformations
  - remettre les os dans leur espace local



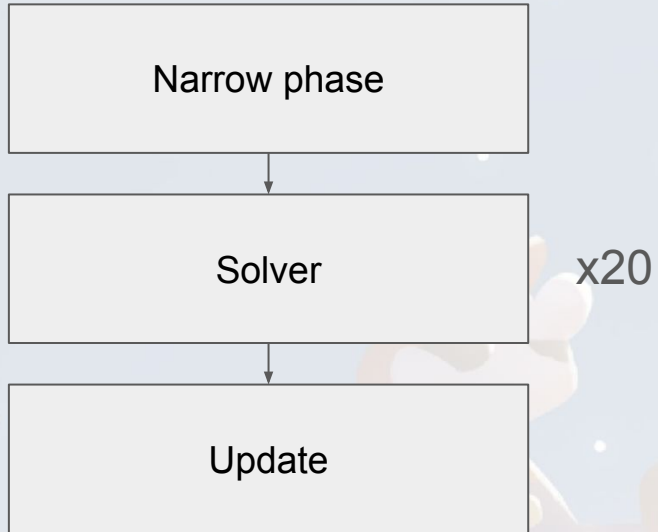


# Animations

- Système d'animation
  - keyframes
  - interpolation
  - reconstruction de la hiérarchie
  - transmission au GPU
- Côté vertex shader
  - Recevoir les indices et les poids des os
  - skinning

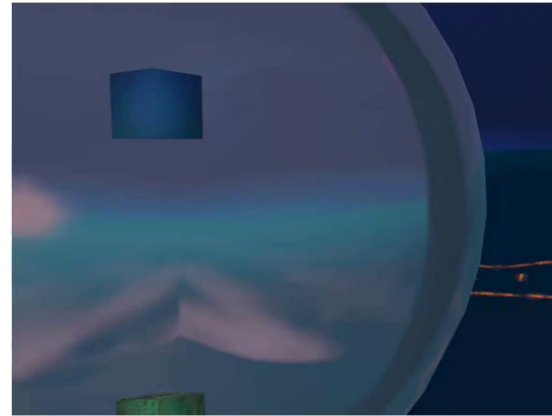


# Physique



## RigidBody

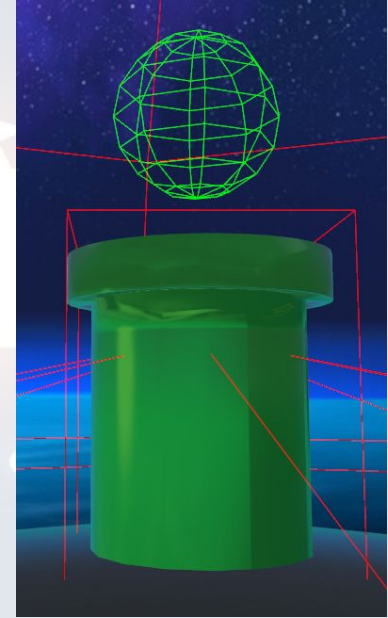
- ...
- Type
- Centre de gravité
- ...



# Formes de collisions

Pourquoi les séparer des objets physiques ?

- caméra en fonction de la position
- culling
- interactions
- ...



Layer = où l'objet existe  
Mask = couches que l'objet scan

# Conclusion

Un moteur généraliste

Une base solide

Trop de fonctionnalités prévues



**Merci pour votre attentions**

