



Illusion **Engine**

Introduction

- C++ 17
- OpenGL 4.5
- GLM 0.9.8.6 -> Maths
- GLAD 0.1.34 -> Loader OpenGL
- GLFW 3.3.2 -> Système de fenêtre et d'inputs
- Open Asset Import Library (Assimp) 5.0.1 -> Load mesh et animations
- Dear ImGui 1.79 -> Interface de l'éditeur
- IrrKlang: 1.6.0 -> Son
- Premake 0.5 -> Setup le projet

Table of contents

01

ECS

02

Rendering

03

Animation

04

Physique

05

Conclusion



Démonstration

Live

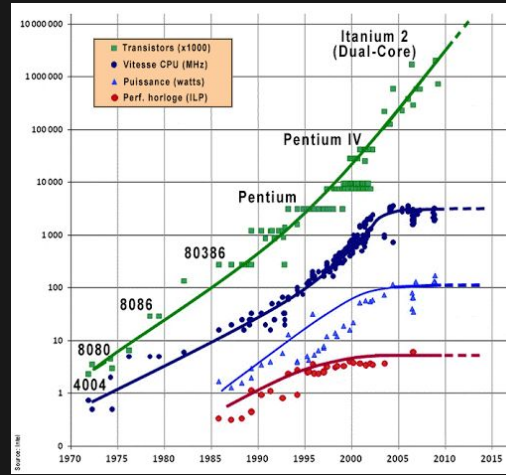


01

ECS

Généralités

- CPU -> Beaucoup de puissance de calcul car évolution exponentiel

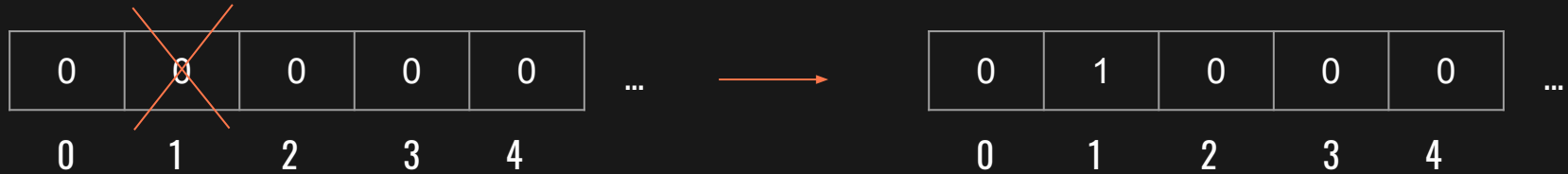


Généralités

- CPU -> Beaucoup de puissance de calcul car évolution exponentiel
- Accès mémoire -> Coûteux car évolution linéaire
- Solution -> Entity Component System pour compenser

Entity

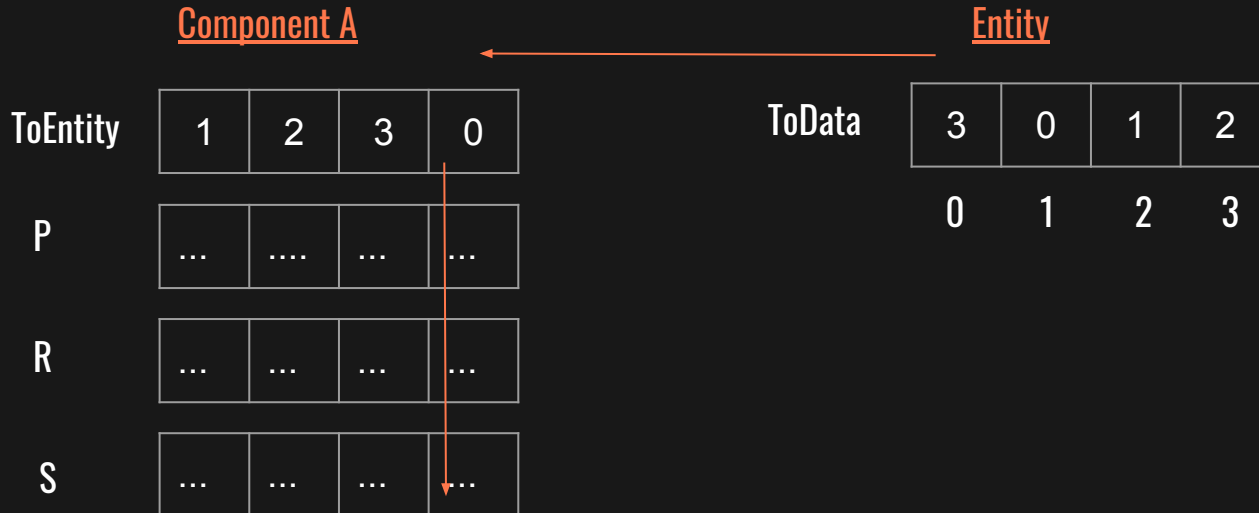
- Tableau d'ID sur 32 Bits
 - 22 bits pour l'index -> position dans le tableau
 - 10 bits pour la génération -> la valeur dans le tableau
 - Génération -> Réutilisabilité des index du tableau si suppression, évite les conflits



Suppression de l'entité d'index 1, puis création d'une nouvelle entité qui prendra la place de l'ancienne en index 1

Component

- Listes d'éléments -> datas brutes
- Tableaux pour relier les entités à leurs données associées
- Tailles proportionnelles au nombre d'entité qui ont ce component



System

- Avoir un comportement -> modification des valeurs des components
- Pré-requis de 1 ou plusieurs component
- Liste des entités ayant les critères requis
- Vérification en cas d'ajout ou de suppression d'un component

System S

Requiert CA(3) et CB(3)
pour appliquer sur E(2)

Component A

...	2
-----	-----	-----	---

0 1 2 3

Component B

...	2
-----	-----	-----	---

0 1 2 3

ToEntityA

...	...	3	...
-----	-----	---	-----

0 1 2 3

ToEntityB

...	...	3	...
-----	-----	---	-----

0 1 2 3



02

Rendering

Mesh Rendering

RenderData

Shaders

Materials

Meshes

MeshInstance

Mesh

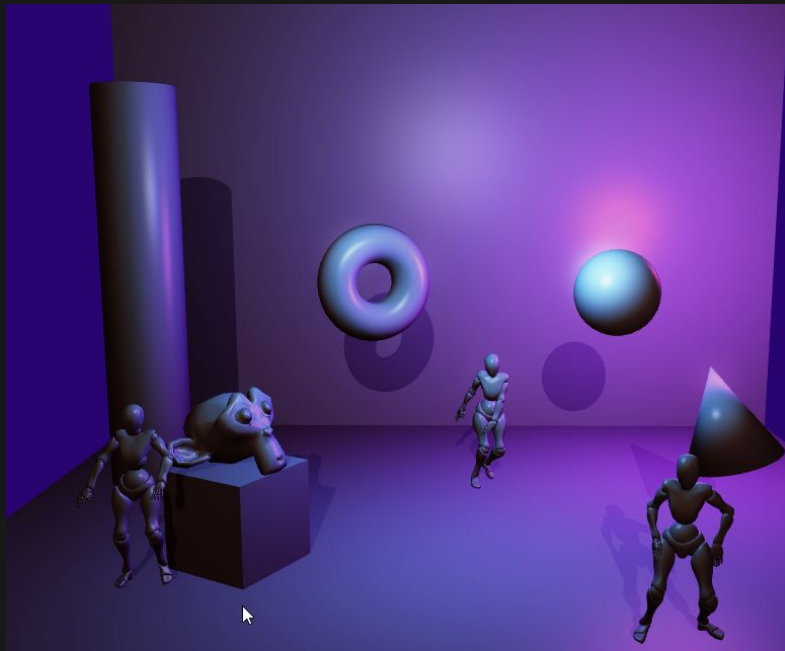
Material

Shader

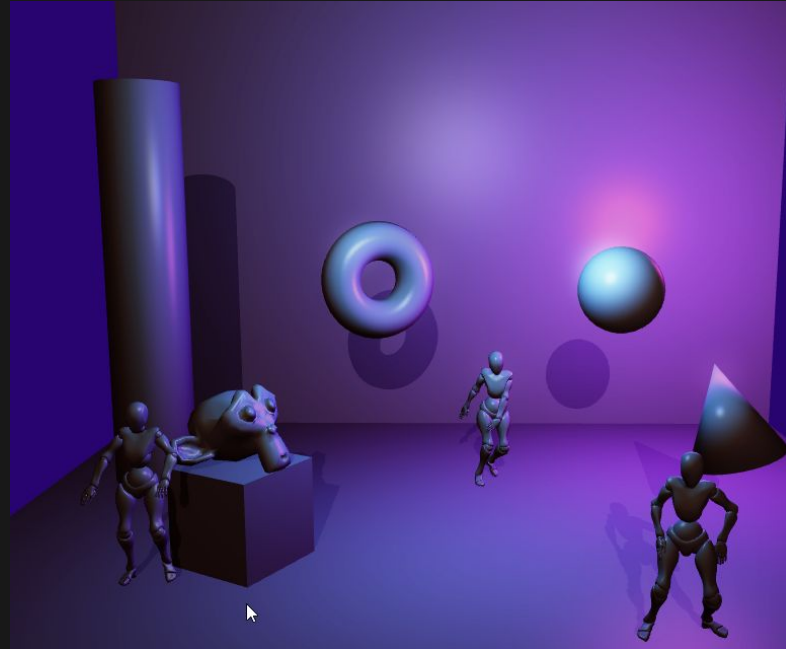
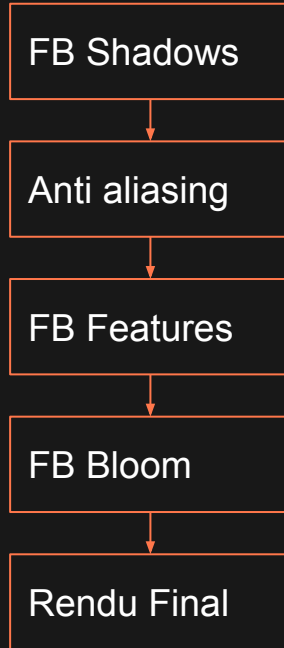
Renderer

MeshInstances Sorted By Mesh and By Shader

Lumières



Pipeline de Rendu

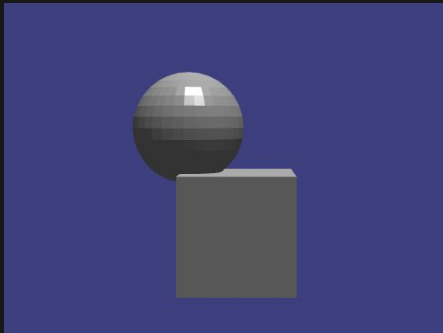




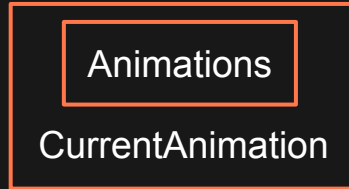
03

Animation

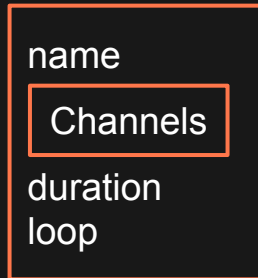
Animation



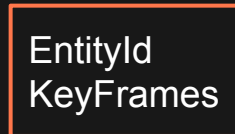
Animator : Component



Animation



Channel



Animation

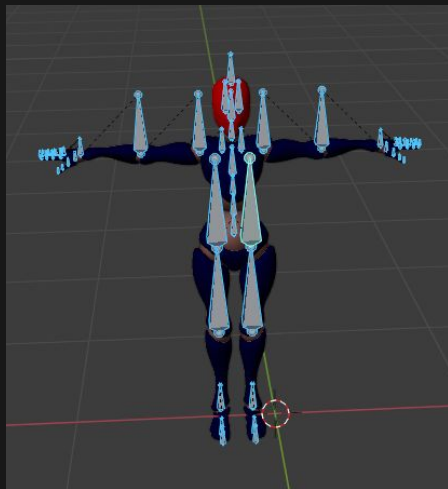
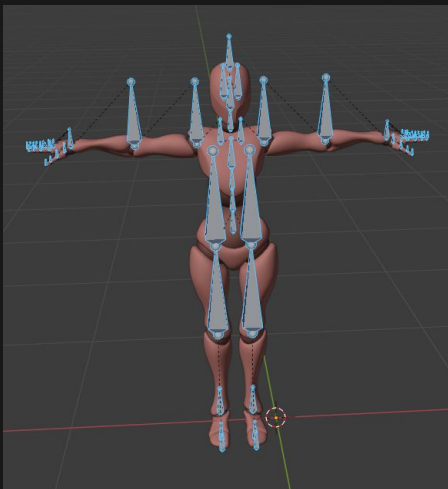
Animator : System

Pour Chaque Channel de l'animation Courante :

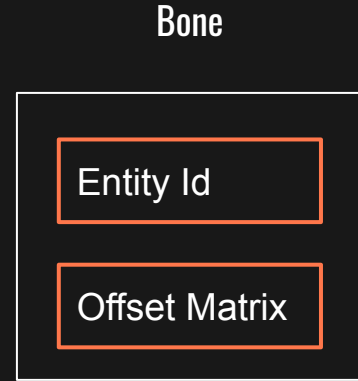
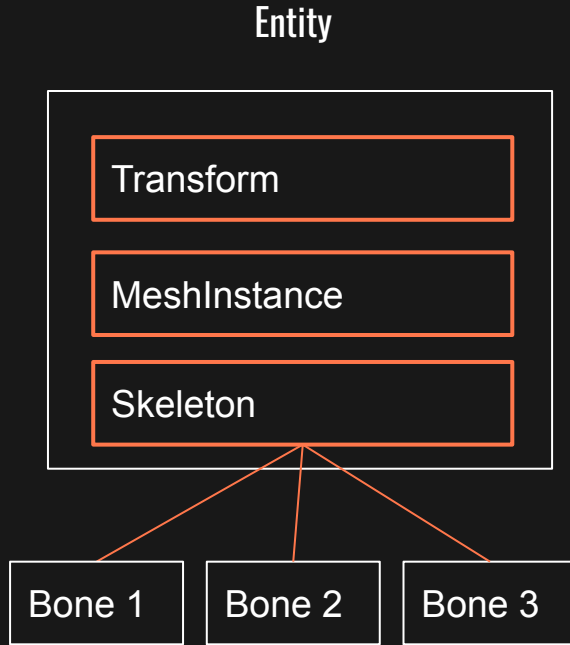
- Calcul l'interpolation entre les 2 frames les plus proches de t
- Update de la position, rotation et/ou scale de l'entity correspondante



Skeletal Animation



Skeletal Animation



Skeletal Animation

Mesh Structure

Positions

Normals

Uvs

Weights

Bonelds



Bones Matrices

Shader



04

Physique

Références & Inspirations

Game Physics Cookbook

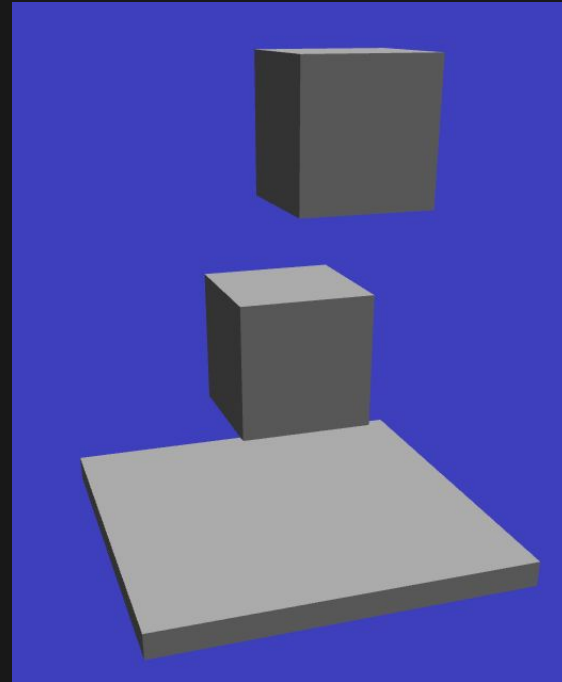
Discover over 100 easy-to-follow recipes
to help you implement efficient game physics
and collision detection in your games

Gabor Szauer

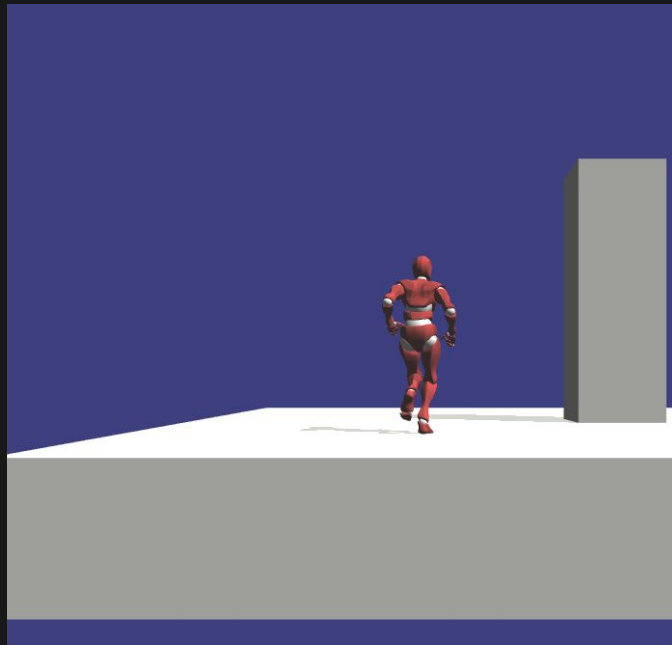
Packt>

BIRMINGHAM - MUMBAI

Collisions et Rigid Bodies



PlayerControler





05

Conclusion

Améliorations & Ajouts

- **Instanced Rendering**
Optimisation si plusieurs fois le même Mesh
- **Animation transitions**
Pouvoir changer d'animation
- **Animation Blending**
Transitions de l'animation soit plus fluide
- **Et bien d'autres.**

Merci de votre attention !

Avez-vous des questions ?



Illusion
Engine

© 2019 Illusion Engine. All rights reserved.

