# Oriented Object Programming

B-400-OOP

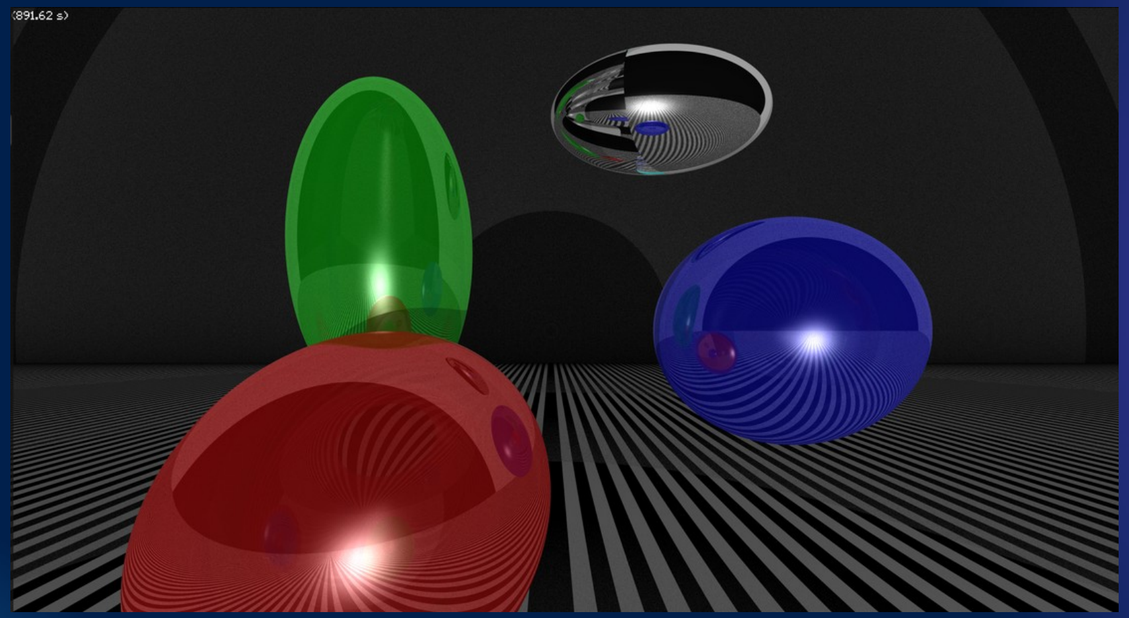
ALEXIS
THÉODORE
EDGAR
THOMAS

# Projets



# RayTracer

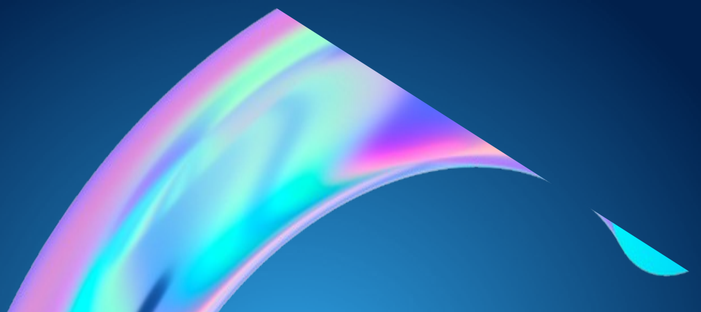
Wen your CPU goes brrrrr.



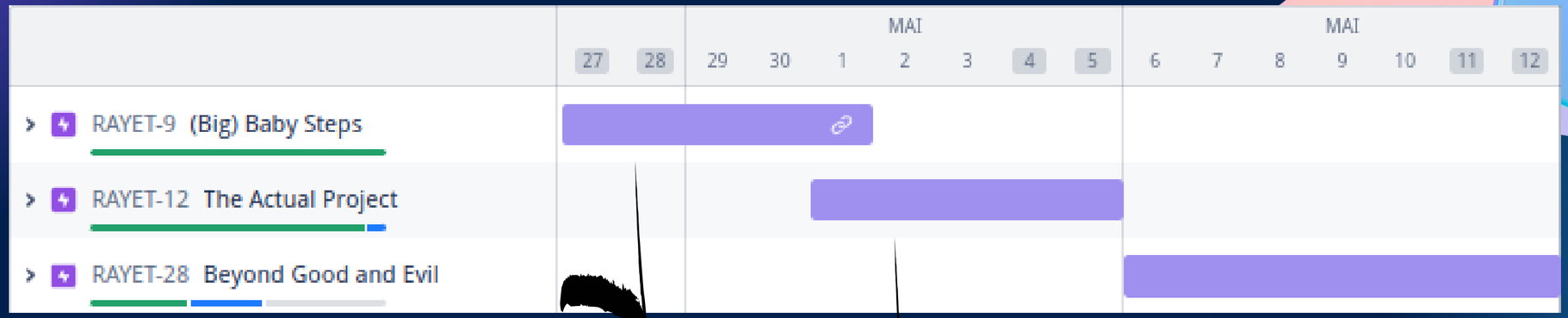
# Démonstration







# Timeline




## Baby steps

Architecture du code base des plugins

## Actual Project

Fin des Plugins
Début des displays
Surface
Primitive
Light
Camera

## Beyond Good and Evil

Materiaux
Fin des displays
Optimisation
Anti aliasing
Filtre
Documentation

# Techniques Utilisées



## Primitives

Le pouvoir des mathématiques




## Anti-aliasing

Algorithme de multisampling (MSAA)



## Lumières

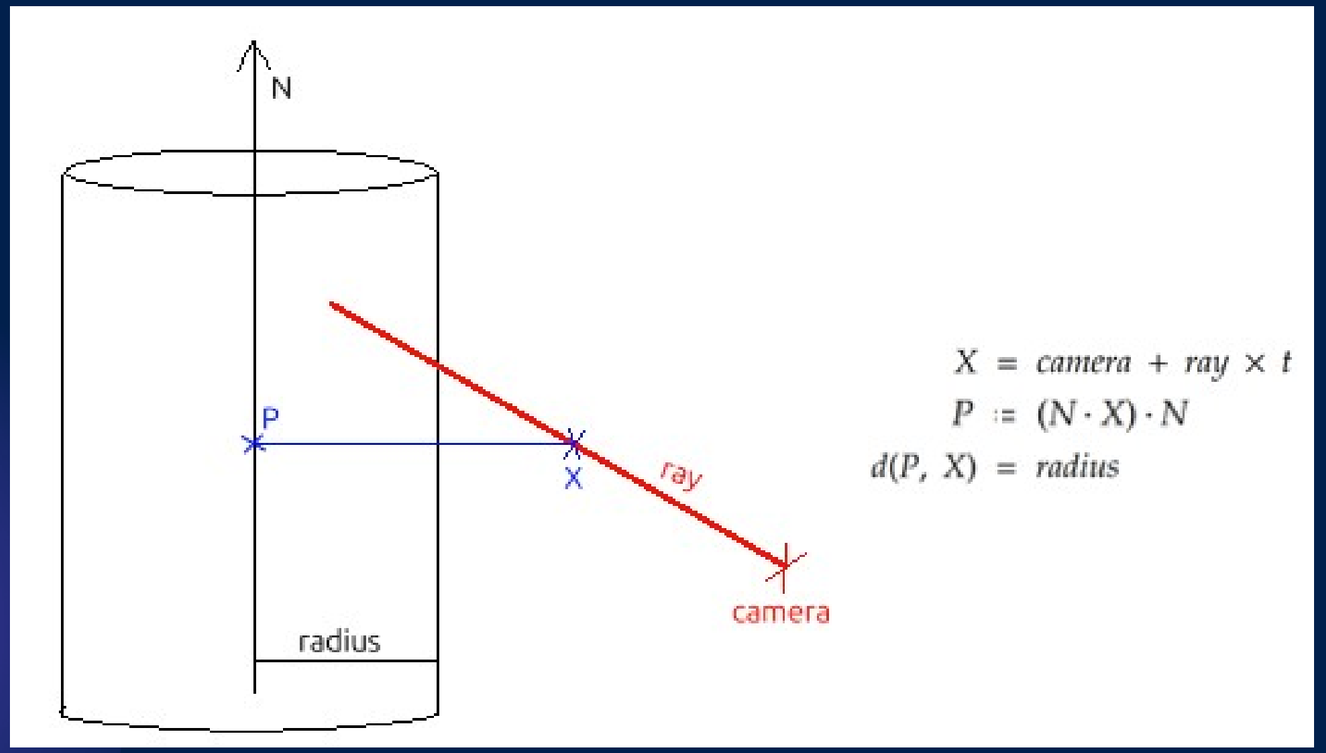
Waah ça brille !




## Plugins

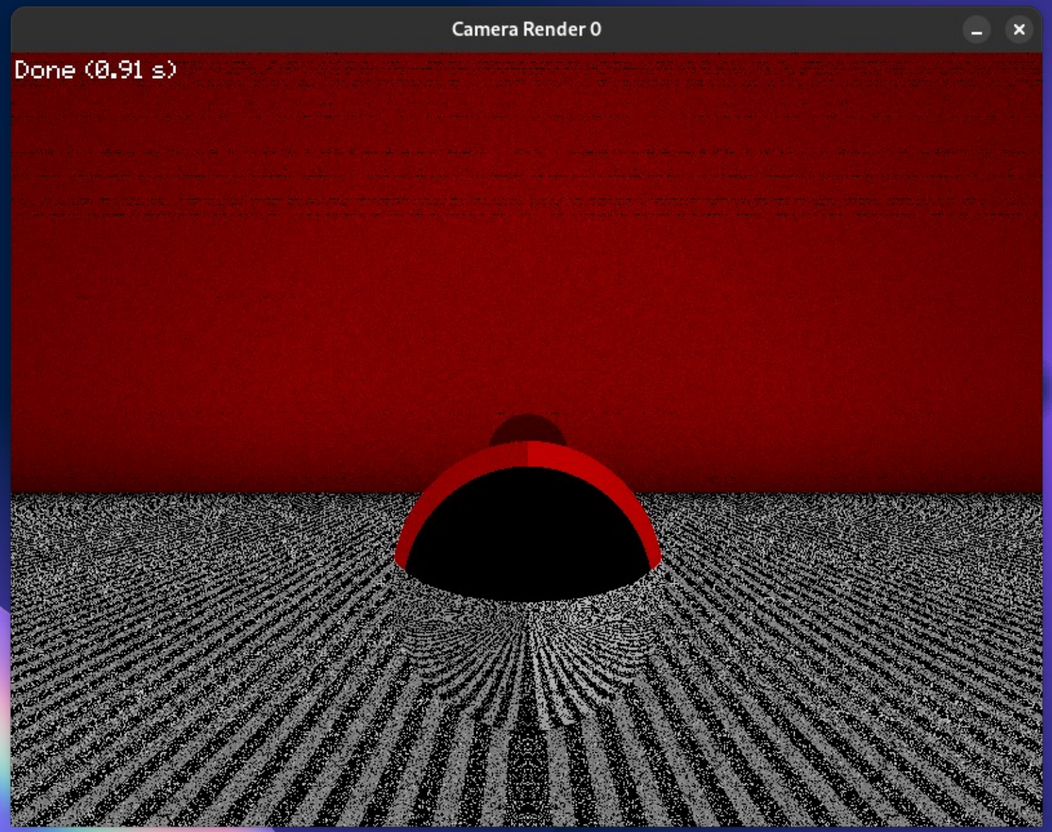
La modularité dans son état le plus pur !

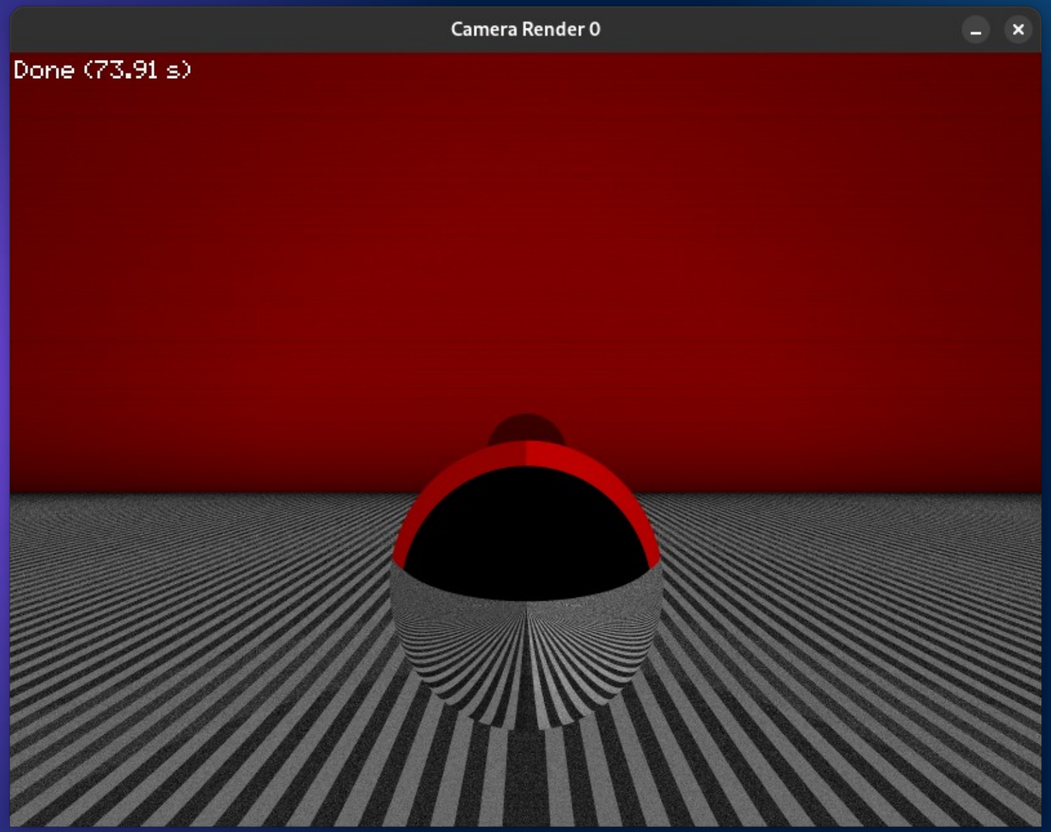
# Le cours de matématik




$$X = camera + ray \times t$$
$$P := (N \cdot X) \cdot N$$
$$d(P, X) = radius$$

# Oh M(S)AA God!



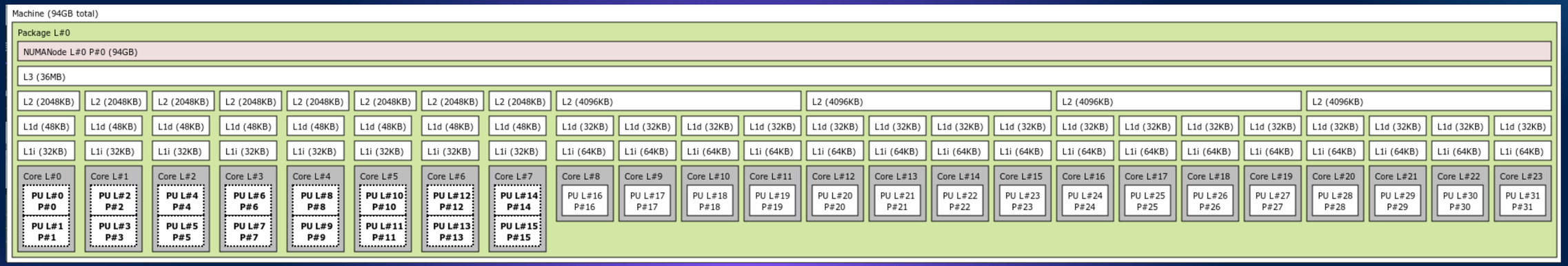



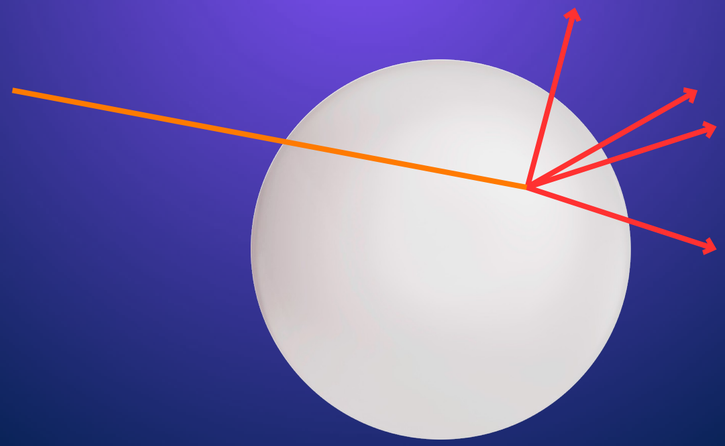
# Les lumières



# Problèmes de parallélisme

## ::rand()  std::mt19337



# La modularité

Custom
RayTracer plugin



# Bonus



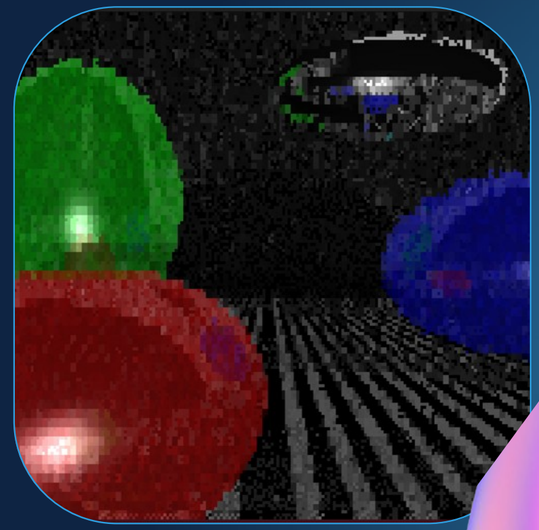
## Multiple caméras

Configuration de plusieurs caméras dans une seule scène




## Bibliothèques graphiques

Possibilité de choisir une bibliothèque graphique et de voir l'évolution du rendu en temps réel



## TDL

Rendu dans un terminal en utilisant des caractères spéciaux UTF-8 comme des sous-pixels

# Organisation



**Jira**
Outil d’organisation
Kanban



**Miro**
Conceptualiser
l’architecture



**Github**
Collaboration avec
des branches

# Mise en perspective

## Difficultés

- Mise en perspective et architecture du projet
- Capacité de travail effective divisé par deux (2 AER et 1 freelance)
- Temps de travail sur le projet réduit
- Mathématiques

## Opportunités

- Mathématiques appliquées
- Programmation en concurrence
- Optimisation et amélioration continue du code
- Application de « design patterns » pour répondre aux besoins techniques

# Conclusion



# Thank You!