



Durval Pereira César Neto  
Erivaldo Lourenço Mariano

Instituto de Computação - IC  
Universidade Federal de Alagoas - UFAL



# Motores de jogo (*Game Engines*)



- *Framework* desenhado p/ criação e desenvolvimento de jogos
- Usados para jogos de console, dispositivos móveis e PCs
- Economiza o processo de desenvolvimento de jogos
- Facilita o *port* de jogos para múltiplas plataformas

---

# Game Engines: Componentes



- Lógica do jogo
  - *Rendering engine* (renderização)
  - *Audio engine*
  - *Physics engine* (Física dos ambientes e personagens)
    - Emula las leis da físicas de forma realista na aplicação
  - Inteligência Artificial
  - Dentre outras
-

# Lista de *Game Engines*



# Lista de Game Engines

CRYENGINE<sup>2</sup>



CRYENGINE<sup>3</sup>



POWERED BY



UNREAL  
TECHNOLOGY



# Lista de *Game Engines*





# Lista de *Game Engines*





- Acrônimo de **O**bject-**O**riented **G**raphics **R**endering **E**ngine
  - Open Source
  - Multi-plataforma
    - Windows
    - Linux
    - Mac OSX
  - Abstração para bibliotecas
    - OpenGL
    - Direct3D
  - C++ (Java, Python, .NET)
-





# Torchlight



# Zombie Driver





A screenshot from a video game showing a character named Garshasp hanging from a horizontal wooden beam. The character is wearing a dark tunic and has a sword strapped to his back. The background features a misty, mountainous landscape with trees and a large, ornate structure in the upper left corner. The scene is set on a wooden platform or bridge.

# Garshasp





# Project AfterShock





# Victory: The Age of Racing



# Venetica



3 / 8

# Zero Gear



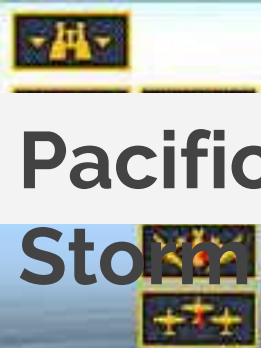


# Jack Keane



Help a former captain in need!





# Pacific Storm



Our unit destroyed: IL-2  
Our unit destroyed: IL-2  
Our unit destroyed: IL-2



1 193

Rank 10



**FAMILY FARM**

Work in Progress

# Family Farm

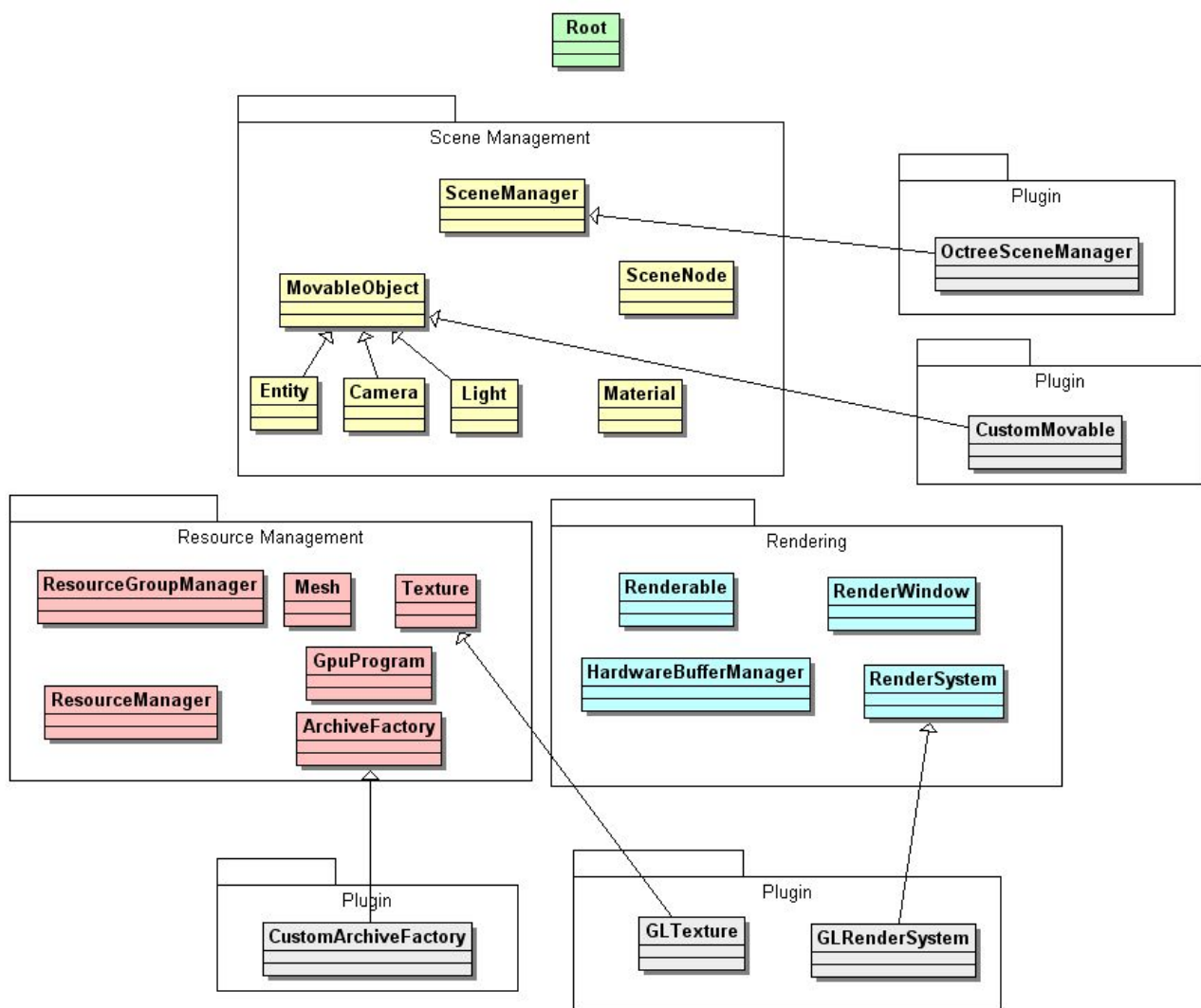
people are starving!



# Arquitetura do Ogre3D

Segue padrões de projetos, dentre eles:

- *Factories*
- *Singletons*
- *Observers*



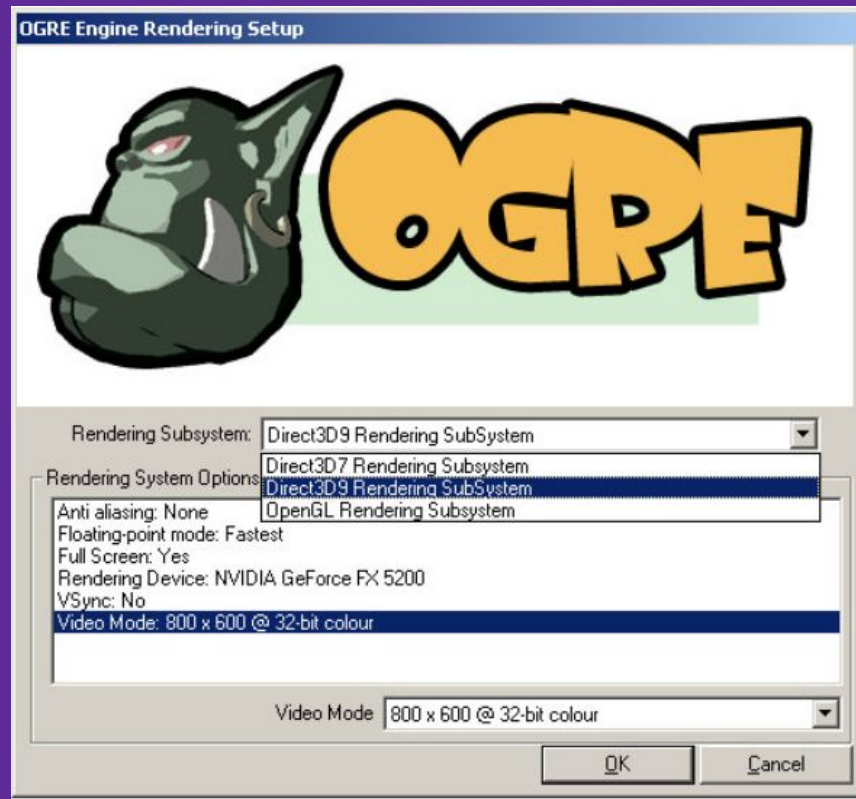
# Principais conceitos

- *Root*
- *RenderSystem*
- *SceneManager*
- *Mesh*
- *Entity*
- *SceneNode*
- *Material*

---

# O Objeto *Root*

- Ponto de partida do OGRE
- Determina a *base config*
- Obtém ponteiros para *core objects*
- Inicia a renderização



# O Objeto *RenderSystem*

- Classe abstrata
- Normalmente não é manipulado
- Define a interface entre o OGRE e a API gráfica utilizada
- *SceneManager* assume o manuseio das chamadas do *RenderSystem*



# O Objeto *SceneManager*

- Mais importante objeto depois do *Root Object*
  - Reúne e organiza todo o conteúdo da cena
  - Maioria das chamadas são durante a configuração da cena
  - Aplica algoritmos para otimizar a exibição da cena
  - Cria e gerencia todas as câmeras e os objetos móveis
-

# O Objeto *ResourceGroupM anager*

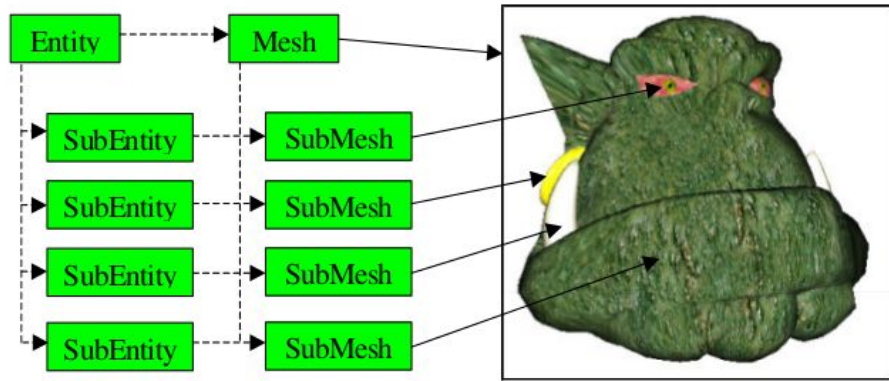
- Funciona como um *hub* de carga/descarga de recursos reutilizáveis
- Interage majoritariamente com o objeto *SceneManager*
- É um *Singleton*

# O Objeto *Mesh*



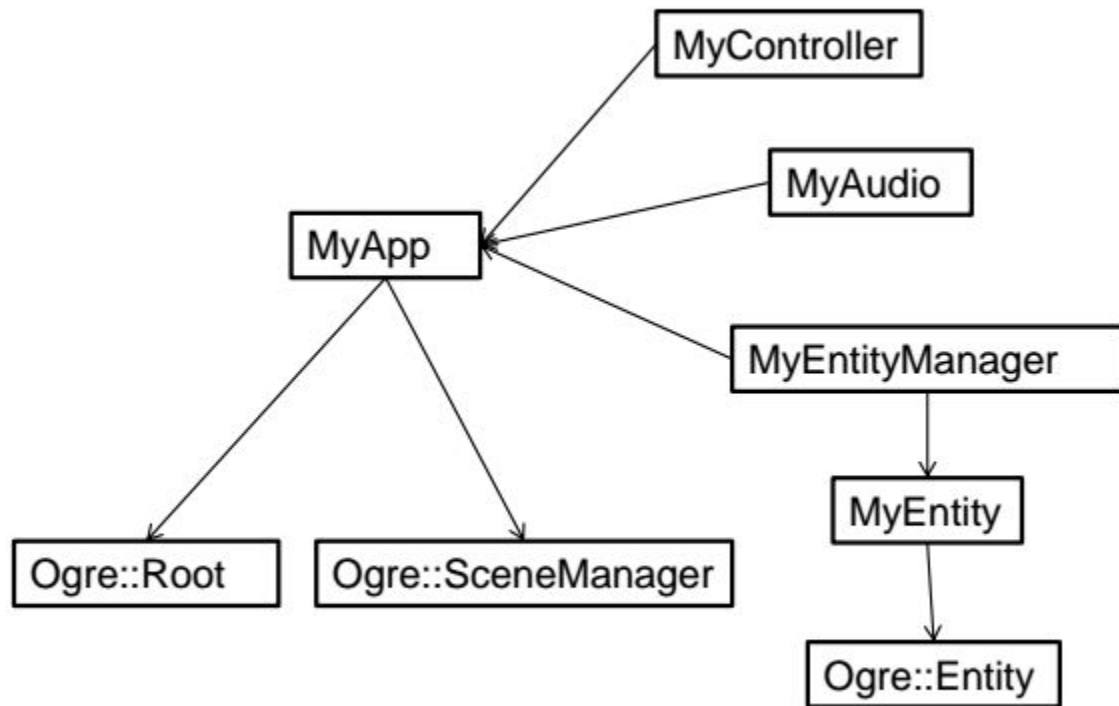
- Representa um objeto móvel, base para as *Entities*
  - Esse recurso é manipulado pelo objeto *MeshManager*
  - Carregamento de malhas através dos arquivos no formato **.mesh**
    - 3DS Max
    - Maya
    - Blender
  - Podem ser animados
-

# O Objeto *Entity*

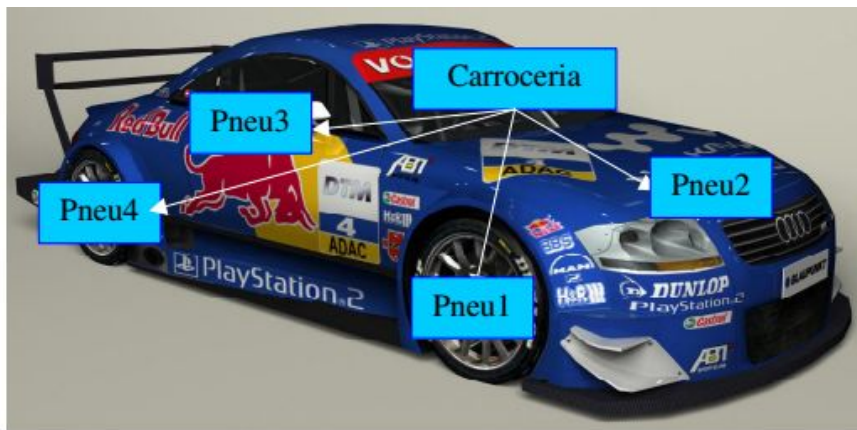


- Uma entidade (*entity*) é uma instância de um objeto móvel na cena
- Toda entidade possui uma malha (*mesh*) associada que descreve o objeto graficamente

# Arquitetura do projeto



# O Objeto SceneNode



- Utilizado para agrupar entidades, luzes, câmeras e objetos móveis
  - Armazena informações relativas a posição, orientação e escala desses elementos
  - Toda entidade deve estar associada a um *SceneNode*
  - Estrutura de nó raiz e nós-filhos, gerando uma hierarquia de *SceneNodes*
-

# O Objeto

## *Material*

- Controla como os objetos da cena serão renderizados
- Nele são especificados os componentes de iluminação
  - Ambiente
  - Difuso
  - Especular
  - Emissivo
- Fazem parte também dele as Texturas e suas propriedades

---



# Texturas



- Biblioteca DevIL
  - .bmp, .cut, .dcx, .dds, .ico, .gif, .jpg, .lbm, .lif, .mdl, .pcd, .pcx, .pic, .png, .pnm, .psd, .psp, .raw, .sgi, .tga, .tif, .wal, .act e .pal.

# Gerando Sombras



Figura 21. Geração de sombras: (a) *Stencil Additive*; (b) *Stencil Modulative*; (c) *Texture Modulative*; (d) sem sombra.

- Suporta três tipos de cálculos
  - a) *Stencil Additive* - renderiza cada luz como um passo aditivo separado na cena. Mais custosa, porém é a mais precisa.
  - b) *Stencil Modulative* - mais rápida com bons efeitos, renderiza os volumes das sombras após renderização dos objetos não transparentes
  - c) *Texture Modulative* - menos custosa, não gera efeitos tão bons. Utiliza a técnica *Render-To-Texture* (renderiza cenas em texturas que são utilizadas em materiais para torná-los reflexivos).



*Tools APIs*



Microsoft  
**DirectX**

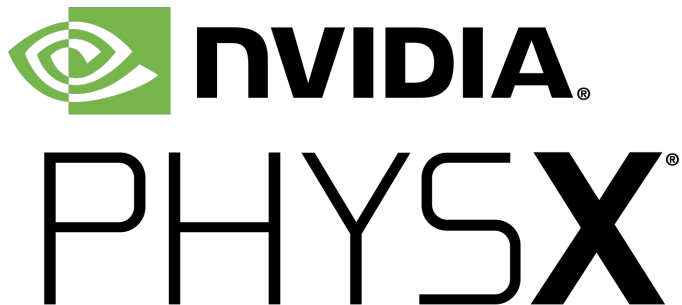
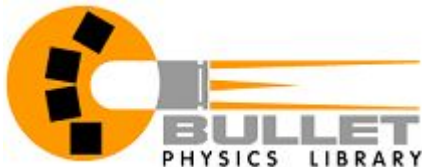
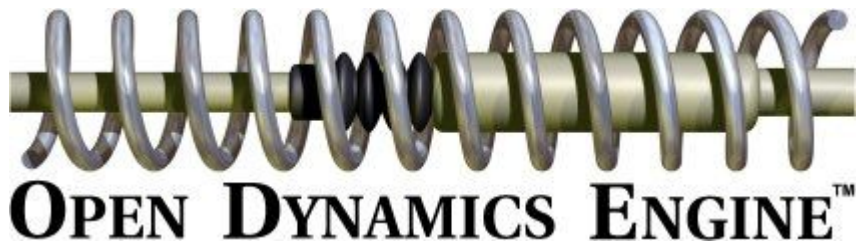
# APIs Gráficas

---



# APIs de Audio

---



# APIs de Física



# APIs de Rede

---



# Perguntas?

Obrigado