Integrating Advanced GLSL Shading and XML Agents into a Learning-Oriented 3D Engine

Edgar Velázquez-Armendáriz, Erik Millán
ITESM-CEM
February 28th 2006.

Introduction

- A lot of Computer Science students chose their major because of their interest on Video Games.
- Highly capable commodity GPUs available today.
- Development moves towards custom shaders able to render special effects.

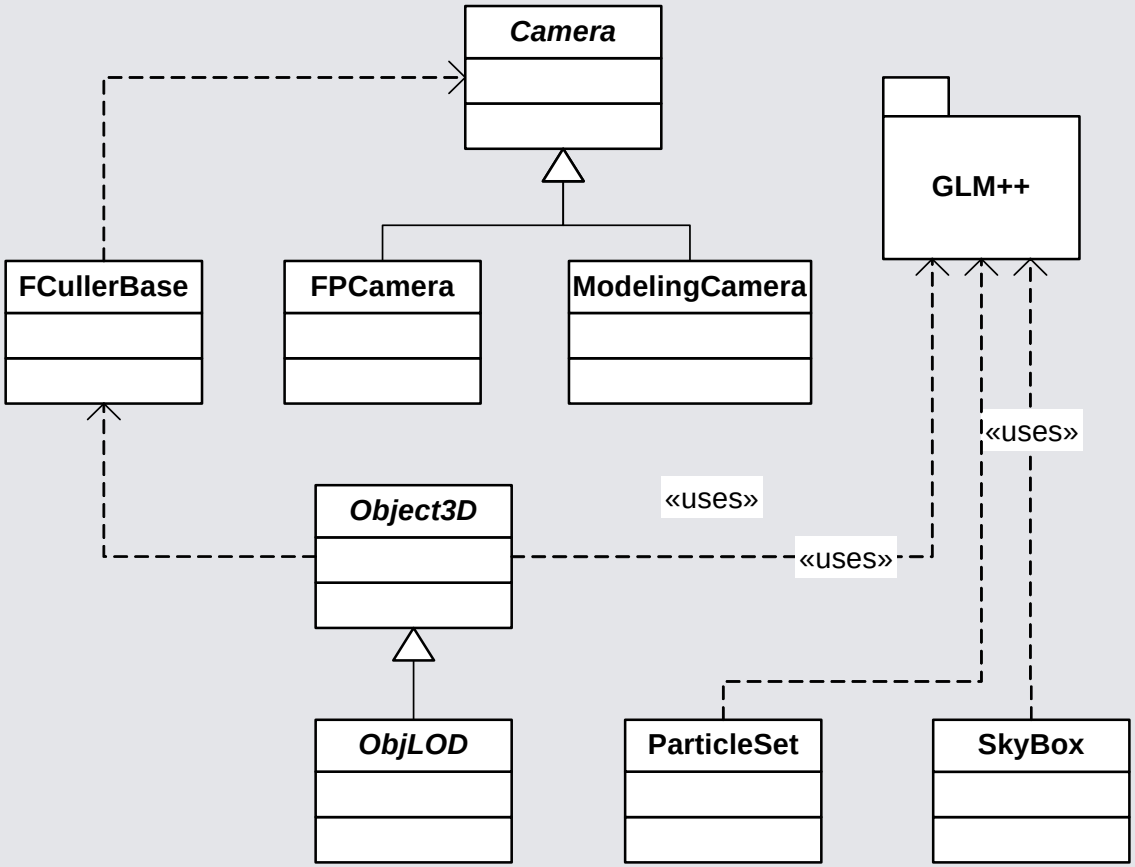
Building a 3D Graphics project

- Choices for building a serious project:
  - Existing 3D Engine (OGRE, Irrlicht).
    - Extremely complex.
  - Write their own engine.
    - Difficult, very time consuming.
    - Would not incorporate advanced features.
- How to add AI support for the characters?
  - Must be implemented on top of the provided API.

Purposed Work

- A 3D engine simple featuring:
  - GLSL shaders.
  - Shadows.
  - Particles and collisions.
- It also integrates previous work which allows the creation of virtual characters and crowds using images and XML files.

System’s architecture



Multi platform and Open Source libraries

- Computer Sciencie students use several OS.
- Built upon multi platform, open source libraries
  - Xerces – XML parsing
  - GLEW – OpenGL Extensions
  - FreeGLUT – windows management
  - Fmod – Sound support
- Source code compiles both in Visual Studio .NET 2003 and GCC 3.x

GLM++ library

- Based on *glm* library by Nate Robins.
- Provides useful yet laborious to implement features:
  - OBJ file loading.
    - Performs tangent space matrix calculation, required for per-pixel lighting.
    - Collision detection initialization.
  - Texture loading from PNG, BMP and PGM files.
  - GLSL Shaders abstraction.
    - Focus on shader logic, not setup details.

Collision Detection

Key Rendering Features

- Integrated Frustum Culling for all objects.
- Shadow maps.
- Per-pixel lighting using Blinn-Phong equations.
- Normal mapping and bump mapping.
- Wireframe and bounding volume drawing.
- Rendering mode may be changed at runtime.
  - GLSL or fixed pipeline rendering, shadows.
- Skybox support.

Normal and Bump mapping

Lighting Equations

$$v' = \left(\left[\vec{T} \quad \vec{B} \quad \vec{N}\right] \circ M^{-1}\right)(v)$$

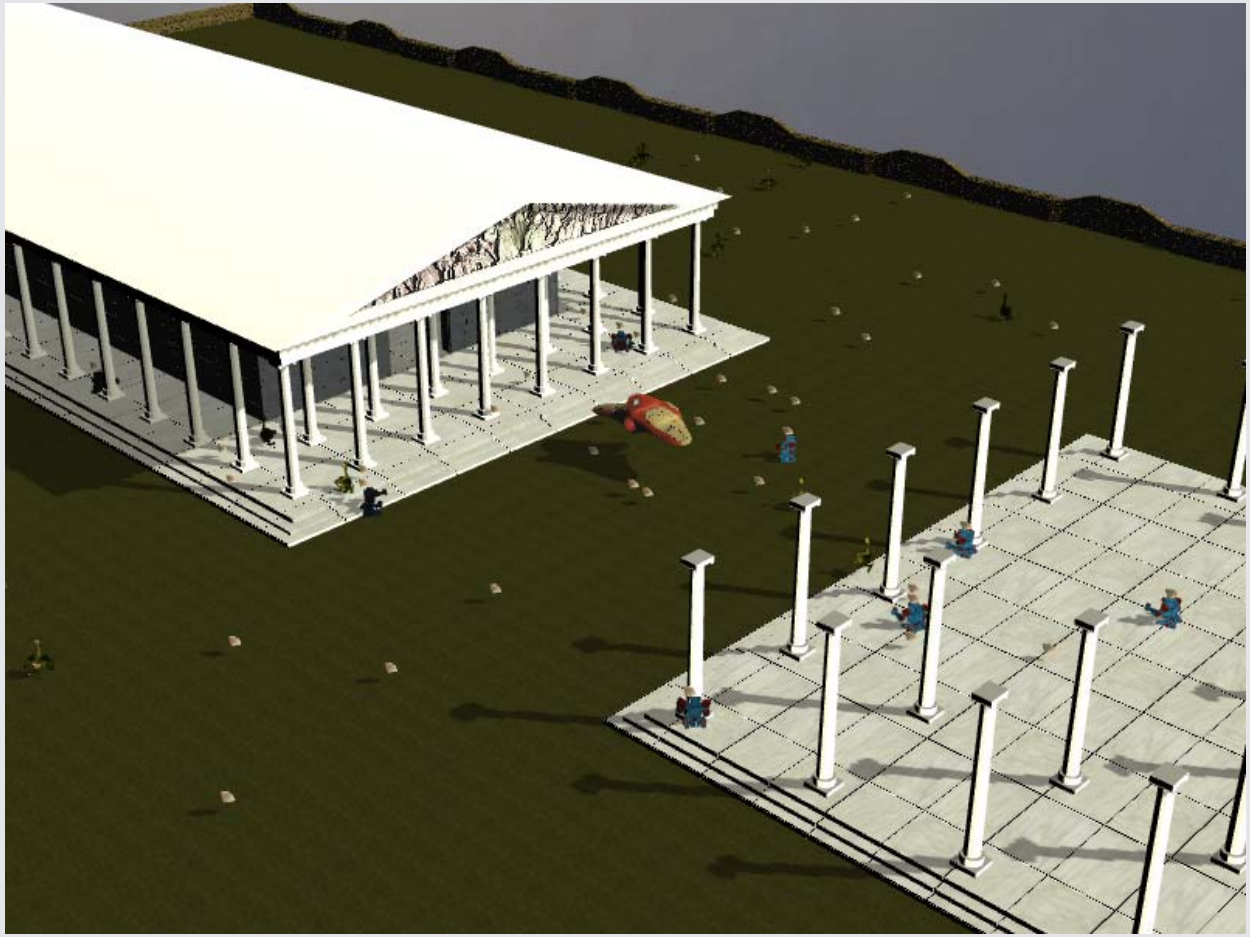
$$I_{out} = I_{light} k_d \max\left(0, \vec{N} \cdot \vec{L}\right) + I_{light} k_s \max\left(0, \vec{N} \cdot \vec{H}\right)^n$$

$$\vec{H} = \frac{\vec{L} + \vec{V}}{\left|\vec{L} + \vec{V}\right|}$$

$$I_{frag} = I_{amb} + \frac{1}{2}(1 + s) I_{out}$$

More Normal and Bump mapping examples

GLSL Shadows



GLSL vs. Fixed Pipeline Shadows

Fixed Pipeline **GLSL**

XML Crowds

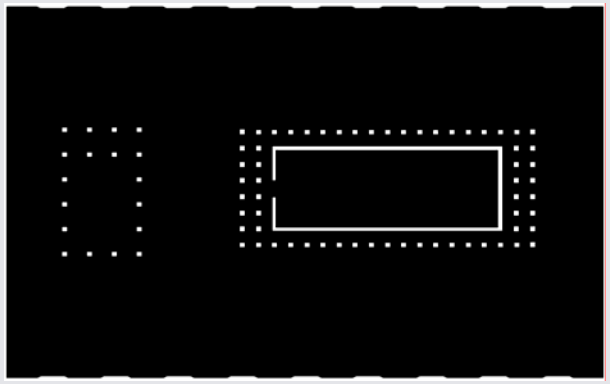
- Based on previous work at ITESM-CEM.
- Creates crowds of virtual characters through XML.
- These interactive agents can interact with an arbitrary environment using image based collision and height maps.
- The crowd's members can be shaded using custom GLSL programs, and they also cast and receive shadows.

Agent's XML Code example

```
<procedure name="wander">
    <state name="init" initial="true">
        <probset cumulative="true">
            <option prob="25%">
                <behavior type="turn" style="run" angle="50" time="0.5" />
            </option>
            <option prob="25%">
                <behavior type="turn" style="run" angle="-50" time="0.5" />
            </option>
            <default>
                <behavior type="go" style="run" dist="5" time="0.5" />
            </default>
        </probset>
        <return />
    </state>
</procedure>
```

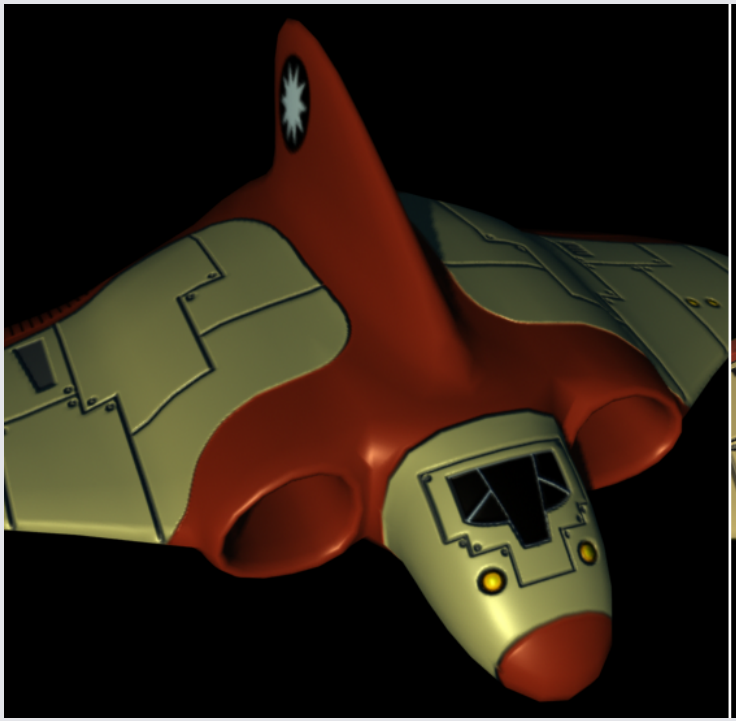
Height and Collision Maps

Height map

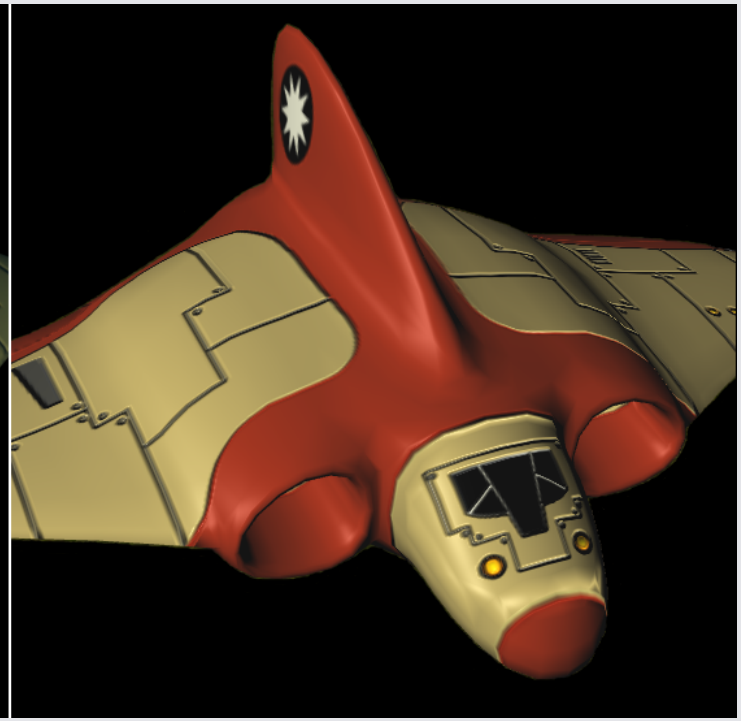


Collision Map

Results



Maya: 11 sec.



Engine: 0.05 sec.

Conclusions

- We have presented a 3D engine for students in computer graphics and artificial intelligence.
- Resulting visual quality encourages further exploration of shading programs and autonomous crowds programming.
- The portability of this platform allows the use of a variety of hardware platforms.

Future Work

- XML multiple level-of-detail mesh specification.
- Use of OpenGL Framebuffers for direct rendering.
- Add communication capabilities between characters.
- Communication between different environments using networks.

Integrating Advanced GLSL Shading and XML Agents into a Learning-Oriented 3D Engine

Edgar Velázquez-Armendáriz, Erik Millán
ITESM-CEM
February 28th 2006.