

Real-Time Graphics: Game engines



Vortrag 1

~ Ogre ~

*Matthias Lang, 20725
SS09*

*Leitung:
Prof. Dr. Charles A. Wüthrich
Dipl.-Med.Sys.wiss. Bernhard Bittorf
Dipl.-Med.Sys.wiss. Thomas Wawrzinoszek*

*Bauhaus-Universität Weimar, Fakultät Medien
Weimar, 30. April 2009*

Ogre: Allgemein

- OGRE (Object-Oriented Graphics Rendering Engine)
(keine Game Engine im eigentlichen Sinne)
 - Open Source License (LGPL)
 - Sprache: C++, Objektorientiert
 - Plugin - Struktur
 - basiert auf Szenengraphen, unterstützte Manager (octree, BSP, Paging Landscape, portal(beta))
 - Unterstützt Vertex & Fragment-Shader, eigene programmierbar in: GLSL, HLSL, Cg, assembler
 - Progressive LOD, Material LOD
 - Animation: weighted multiple bone skinning

Ogre: Allgemein (fortsetzung)

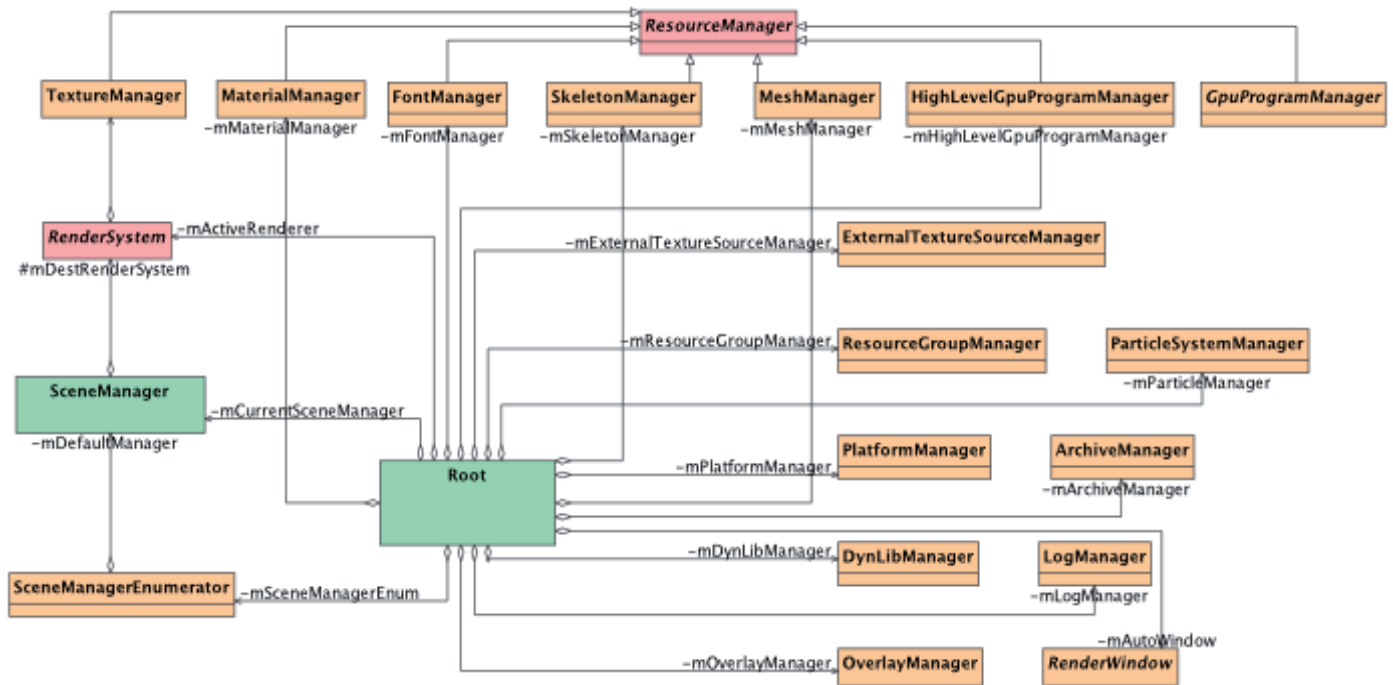
- Mesh-Import von Blender, Maya, 3DsMax etc,
Biquadric Bezier patches
- Effekte: u.a. Bump Mapping, Cube Mapping, particle System,
Fog, Skybox, Transparence, Post Processing

aktuell: OGRE v1.6.2 [Shoggoth] (released: April 11th, 2009)

Unterstützte Grafikschnittstellen:

- DirectX 7 und 9 (Windows)
- OpenGL (Windows, Linux und Mac OS X)

Ogre: Struktur



Ogre: Libraries: Physik

- Unterstützte Physikengines:
 - Ode (Open Dynamics Engine)
 - OgreODE or OPAL - Open Physics Abstraction Layer
 - NovodeX/PhysX -> NxOgre and Nogredex
 - Newton Game Dynamics -> OgreNewt Library
 - Bullet -> OgreBullet
 - weitere: True Axis Physics SDK; DynaMo; The Gangsta Wrapper
- Framework für Integration per „ReferenceAppLayer“
Verknüpfung von Geometrie, Physik und Kollision

Ogre: Libraries:

- AI:
OpenAI, FEAR, OpenSteer, A* Tactical Pathfinding, PathLib, Garfixia AI Repository, MicroPather, Boost Graph Library, FANN, OpenSkyNet

- Network:
 - OgreSocks
 - SDL-Net
 - lot more

Ogre: Libraries:

- Scripting:
Ogre-internes Scripting für Material, Compositor, Particle, Overlay und Font Definition Scripts (plain text)

Für alles andere u.a.:

Lua, Lua Plus (Lua C++ oriented fork), Java, Python, Ruby, Squirrel, AngelScript, GameMonkey

- Input:
 - SDL
 - LibGII
 - OpenInput
 - OIS

Ogre: Libraries:

- Audio:
 - OpenAL
 - FMOD, BASS, Audiere, irrKlang

Ogre: Libraries:

- OgreSpeedTree (Bäume), Hydrax (Wasser), Caelum (Atmosphäre), Particle Universe plugin, HDRlib, PagedGeometry Engine (Vegetation), OGRE Collada Plugin(xml)
- lot of GUIs
- Scene Manager (Terrain):
 - Octree Scene Manager
 - Terrain Scene Manager
 - Nature Scene Manager (ogreaddons)
 - Paging Scene Manager (ogreaddons)
 - BSP Scene Manager
 - DotSceneOctree SceneManager (ogreaddons)
 - Portal Connected Zone Scene Manager (PCZSM)
 - Myrddins Paging Landscape Plugin
 - Editable Terrain Manager

Ogre: Wrapper: (Zusammenfassung)

- Wrapper:
 - C#, Python, Java, LUA, NET
 - NxOgre, OgreODE, OgreNewt, OgreBullet, OgreTokamak
 - OgreOpCode, Minimal Ogre Collision (collision detection)
 - OIS, OpenInput
 - OgreAL, OgreOggSound

Ogre: Animation:

- Direkt in Ogre unterstützte Arten von Animation:
 - Skeletal Animation
 - Vertex Animation
 - (z.B. Facial Animation aus Poses von 3DS Max)
 - SceneNode Animation
 - Numeric Value Animation

Ogre: Anwendungen:

- Blender Game Engine = OGRE
- Kommerzielle Produkte:
 - The Book of Unwritten Tales
 - Pacific Storm
 - Ankh
 - Jack Keane
 - Anarchy Online
 - Pacific Storm
 - ...
- Private Projekte

Ogre: Demos

- Demo Show Room



OGRE Official Demos Distribution v1.4.9 (Windows)

-> http://downloads.sourceforge.net/ogre/OgreDemos_v1.4.9.msi

Gezeigt werden die aktualisierten und neu kompilierten Demos von Ogre 1.6.2

Ende

- *Danke für die Aufmerksamkeit*
- *Diskussion: Fragen, Anregungen?*

Quellen

Books:

Gregory Junker, Pro OGRE 3D Programming, Apress 2006

Web:

Allgemeine Informationen: <http://en.wikipedia.org/wiki/OGRE>

Ogre Wiki: http://www.ogre3d.org/wiki/index.php/Main_Page

Get Ogre: <http://www.ogre3d.org/download/sdk> (currently SDK 1.6.2)

Get Appwizard: http://sourceforge.net/project/showfiles.php?group_id=168190

Howto Appwizard: http://www.ogre3d.org/wiki/index.php/The_Complete_Blanks_Guide_To_Using_The_OGRE_SDK_AppWizard

Übersicht Scene Manager: <http://www.ogre3d.org/wiki/index.php/SceneManagersFAQ>

Übersicht tools, lib, wrapper: <http://www.ogre3d.org/wiki/index.php/AssemblingAToolset>

Ogre 3D API „It's so BIG!“: <http://www.ogre3d.org/docs/api/html/>

Forum: <http://www.ogre3d.org/forums/>

<http://www.ogre3d.org/addonforums>

Anhang

Tutorials:

Basic Tutorial:

- 1 Introduction to the basic Ogre constructs: SceneManager, SceneNode, and Entity
- 2 Cameras, Lights, and Shadows
- 3 Terrain, Sky, and Fog
- 4 Frame Listeners and Unbuffered Input
- 5 Buffered Input
- 6 The Ogre Startup Sequence
- 7 CEGUI and Ogre
- 8 Multiple and Dual SceneManagers

Intermediate Tutorials

- 1 Animation, Walking Between Points, and Basic Quaternions
- 2 RaySceneQueries and Basic Mouse Usage (Part 1 of 2)
- 3 Mouse Picking (3D Object Selection) and SceneQuery Masks (Part 2 of 2)
- 4 Volume Selection and Basic Manual Objects
- 5 Static Geometry
- 6 Projective Decals
- 7 Render to texture (RTT)

Advanced Tutorial 1: Resources and ResourceManagers

In Depth Tutorials:

- Example Framework Demystified: A line by line breakdown of the example framework.
- CEGUI In-Depth: A closer look at select CEGUI concepts.
- Manual Resource Loading: How to load your data without using Ogre's file format.

<-- Übersicht Tutorials
Schritt für Schritt

<-- Erklärung: „*ExampleApplication.h*“

Anhang: Anmerkungen zur Einrichtung (VC++ 9.0)

Umgebungsvariablen in VC++ setzen:

env:

\$(OGRE_HOME)\include

\$(OGRE_HOME)\samples\include

lib:

\$(OGRE_HOME)\lib

Bei Benutzung des Appwizards und VC 9.0 script anpassen! (Fehlerhaft, 8->9)

Debug & Release im Homeordner\bin erstellen oder alle Abhängigkeiten im Projektordner unterbringen sowie plugins.cfg und resources.cfg anpassen.

Interessant: (noch entfernen:)

<http://www.ogre3d.org/wiki/index.php/BulletMeshStrider>

→