Erik Ogenvik

Worldforge - www.worldforge.org
Ember, 3d client
Agenda

What is OGRE3D?
Why use it?
Core concepts.
Getting started.
Questions.
There will be code!
General purpose 3d engine.
Cross platform: *NIX, Windows, iOS, Android
Cross graphics layer: OpenGL (ES1.1), DirectX
MIT licensed
Mature
C++
Runs on OpenGL 1.2.1 hardware and up
www.ogre3d.org
Languages

Written in C++

Bindings:
  Python
  Java (ogre4j)
  .NET (Mogre)
  Lua (Lugre)
What it's not

Game engine
Physics
UI
Sound
Game rules
Server
Used in

Torchlight
Venetica
de Blob
Pacific Storm
Motom4x
Worldforge
Rigs of Rods
and many more...
General

General purpose engine
Scene graph based
Plugin architecture
Scalable with hardware
Mainly forward rendering
Probably what you need
Hard things in 3d (incomplete list)

Resource handling
Level of Detail
Culling
Hardware idiosyncracies
Shader management
Pipeline optimization

Why you probably want to use OGRE3D.
Scene graph

Common technique
Good simulation of real world
General solution
Node hierarchy
Children are altered with parents
Things attached to nodes
Time for car analogy!
Things?

Camera
Lights
Entity (Mesh)
Particles
Random geometry
Entity & Mesh

Mesh = geometry + materials (+ animations)
Entity = instance of Mesh
Blender3d for example
Material

Applied to surfaces
Textures + lightning + shaders
Shaders?
Programs run on the gfx card (GPU)
GLSL and HLSL (Cg runtime is not FOSS)
Decide target HW!
(Or use material fallbacks)
SceneManager

Manages a "scene"
One single root node
Specialized for scene layout (performance)
DefaultSceneManager, OctreeSceneManager
Ogre::Root root;
root.showConfigDialog();
Ogre::SceneManager* mgr =
    root.createSceneManager(Ogre::ST_GENERIC);
Ogre::Camera* camera = mgr->createCamera("MainCamera");
Ogre::SceneNode* node =
    mgr->getRootSceneNode()->createChildSceneNode();
node->translate(1.0f, 0.0f, 0.0f);
node->setAutoTracking(true, mgr->getRootSceneNode());
node->attachObject(camera);
Ogre::Entity* entity = mgr->createEntity("deer.mesh");
mgr->getRootSceneNode()->attachObject(entity);
root.startRendering();
mgr->setAmbientLight(Ogre::ColourValue(1.0f, 0.5f, 0.2f));
Ogre::Light* light = mgr->createLight();
light->setDiffuseColour(Ogre::ColourValue(1.0f, 0.5f, 0.2f));
node->attachObject(light);
Rendering frame

Prefer own render loop

Ogre::Root::startRendering()
Ogre::Root::renderOneFrame()
Ogre::FrameListener interface
Input

Not handled by Ogre
OIS (Object Oriented Input System)
SDL
Threading

Resource loading in separate thread
Render interaction on main thread
Application design

Game core != visualization
Server != client
Features != good visuals
Great assets > features
wombat.worldforge.org 5Gb+ GPL assets
3d and free software

OpenGL drivers suck
"suck" as in "freeze X, or crash"
Less now though
And not Nvidia's proprietary
Integrated GPUs > discrete GPUs = Intel
Learn more

www.ogre3d.org
Forum
Wiki
Source
Samples
Books: http://astore.amazon.com/ogre-20