



## Game Engines

### Technical Game Development II

Professor Charles Rich  
Computer Science Department  
rich@wpi.edu

IMGD 4000 (D 11)

1

## Pedagogical Goal

- Your technical skills should not be tied to any particular game engine
- Just like your programming skills should not be tied to any particular programming language
- Use the best tools for each job
- ... or the tools you were given 😊



IMGD 4000 (D 11)

2

## Definition

---

### Game Engine

A series of modules and interfaces that allows a development team to focus on product *game-play content*, rather than *technical content*.

[Julian Gold, OO Game Dev.]

- *But this class is about “the technical content” ! 😊*

## Buy versus Build

---

- Depends on your needs, resources and constraints
  - technical needs (e.g., “pushing the envelope” ?)
  - financial resources (e.g., venture capital ?)
  - time constraints (e.g., 1 mo. or 2 yr. ?)
  - platform constraints (e.g., Flash ?)
  - other factors (e.g., sequel ?)
- Most games commonly built today with some sort of “engine layer”

## Choices: "It's a Jungle Out There"

- 319 3D engines reviewed at DevMaster.net

Most Reviewed Commercial Engines	Most Reviewed Open Source Engines	Latest Engines XML
<ol style="list-style-type: none"> <li>C4 Engine</li> <li>Torque Game Engine</li> <li>3DGameStudio</li> <li>TV3D SDK 6.5</li> <li>Leadwerks Engine 2</li> <li>Unity</li> <li>ShiVa Engine</li> <li>Esenthel Engine</li> <li>NeoAxis Engine</li> <li>DX Studio</li> </ol>	<ol style="list-style-type: none"> <li>OGRE</li> <li>Irrlicht</li> <li>Panda3D</li> <li>Crystal Space</li> <li>jME</li> <li>Blender Game Engine</li> <li>Reality Factory</li> <li>The Nebula Device 2</li> <li>RealmForge</li> <li>Cafu Engine</li> </ol>	<ul style="list-style-type: none"> <li>Graphite</li> <li>ioquake3</li> <li>Abyssal Engine</li> <li>PixelLight</li> <li>Infernal Engine</li> <li>nGENE Tech</li> <li>Linderdaum Engine 0.5.90</li> <li>OpenSpace3D</li> <li>G3D Innovative engine</li> <li>Spring Engine</li> </ul>

- We are *not* going to try to review them all here

## Many Evaluation Dimensions/Features

[ DevMaster.net ]

General Info	
<b>Graphics API</b> OpenGL   DirectX   Glide   Software   Other	<b>Status</b> Alpha   Beta   Productive/Stable   Inactive
<b>Operating Systems</b> Windows   Linux   MacOS   Solaris   SunOS   HP/UX   FreeBSD   Irix   OS/2   Amiga   DOS   Xbox   Playstation   GameCube   GBA   PSP   N-Gage   BeOS   Xbox360   PS3   PS3   Nintendo.Wii   Nintendo_DS	<b>Misc</b> Documentation
<b>Programming Language</b> C/C++   Java   C#   D   Delphi   Pascal   BASIC   Ada   Fortran   Lisp   Perl   Python   Visual_Basic_6   VB.NET	<b>General Features</b> Object-Oriented Design   Plug-In Architecture   Save/Load_System   Other
Game Features	
<b>Networking System</b> Client-Server   Peer-to-Peer   Master_Server	<b>Physics</b> Basic_Physics   Collision_Detection   Rigid_Body   Vehicle_Physics
<b>Tools &amp; Editors</b> Scripting   Built-in Editors	<b>Artificial Intelligence</b> Pathfinding   Decision Making   Finite_State_Machines   Scripted   Neural_Networks
<b>Sound &amp; Video</b> 2D_Sound   3D_Sound   Streaming_Sound	
Graphics Features	
<b>Lighting</b> Per-vertex   Per-pixel   Volumetric   Lightmapping   Radiosity   Gloss_maps   Anisotropic   BRDF	<b>Animation</b> Inverse_Kinematics   Forward_Kinematics   Keyframe_Animation   Skeletal_Animation   Morphing   Facial_Animation   Animation_Blending
<b>Shadows</b> Shadow Mapping   Projected planar   Shadow Volume	<b>Meshes</b> Mesh Loading   Skinning   Progressive   Tessellation   Deformation
<b>Texturing</b> Basic   Multi-texturing   Bumpmapping   Mipmapping   Volumetric   Projected   Procedural	<b>Surfaces &amp; Curves</b> Splines   Patches
<b>Shaders</b> Vertex   Pixel   High Level	<b>Special Effects</b> Environment Mapping   Lens Flares   Billboarding   Particle_System   Depth_of_Field   Motion_Blur   Sky   Water   Fire   Explosion   Decals   Fog   Weather   Mirror
<b>Rendering</b> Fixed-Function   Stereo_Rendering   Raytracing   Raycasting   Deferred_Shading   Render-to-Texture   Voxel   Fonts   GUI	<b>Terrain</b> Rendering   CLOD   Splatting
<b>Scene Management</b> General   BSP   Portals   Octrees   Occlusion_Culling   PVS   LOD	

*If there's a feature term here you don't know, you should look it up!*

## Types of Engine Architectures (Roughly)

- **Monolithic** (e.g., Unreal Engine)

- **Modular**

- **Extensible IDE** (e.g., Unity)



- **Open Class Library** (e.g., C4)



## Monolithic Engines (e.g., Unreal)

- “old style”--typically grew out of specific game
- tend to be genre-specific
- difficult to go beyond extensions/modifications  
not *anticipated* in (e.g., scripting) API
- proven, comprehensive capabilities

## Modular Engines (e.g., C4 and Unity)



- “modern”--often developed by game engine company
- use object-oriented techniques for greater modularity
- much easier to extend/replace components than monolithic engines

## Modular: Extensible IDE's (e.g., Unity)



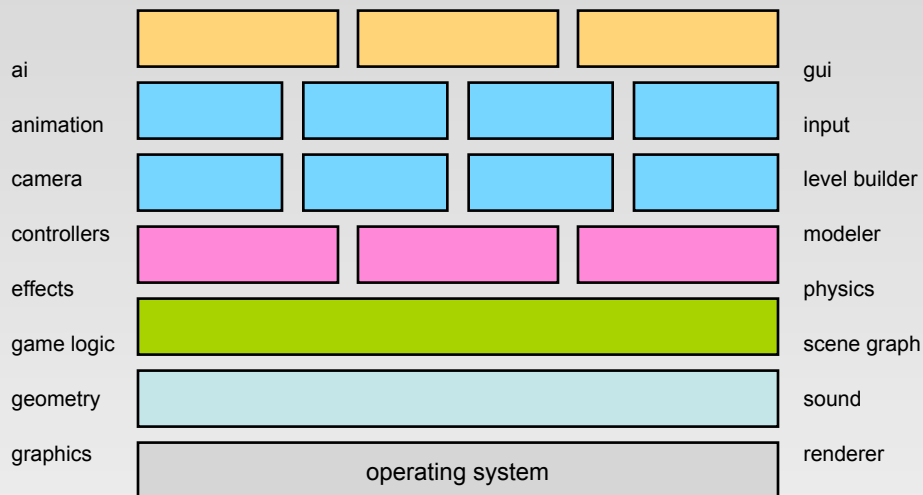
- GUI-oriented development process
  - more accessible for novice/casual programmers
  - more “art friendly”
- comprehensive asset management
- limited (controlled) exposure of internals
  - prevents abuse
  - prevents some extensions

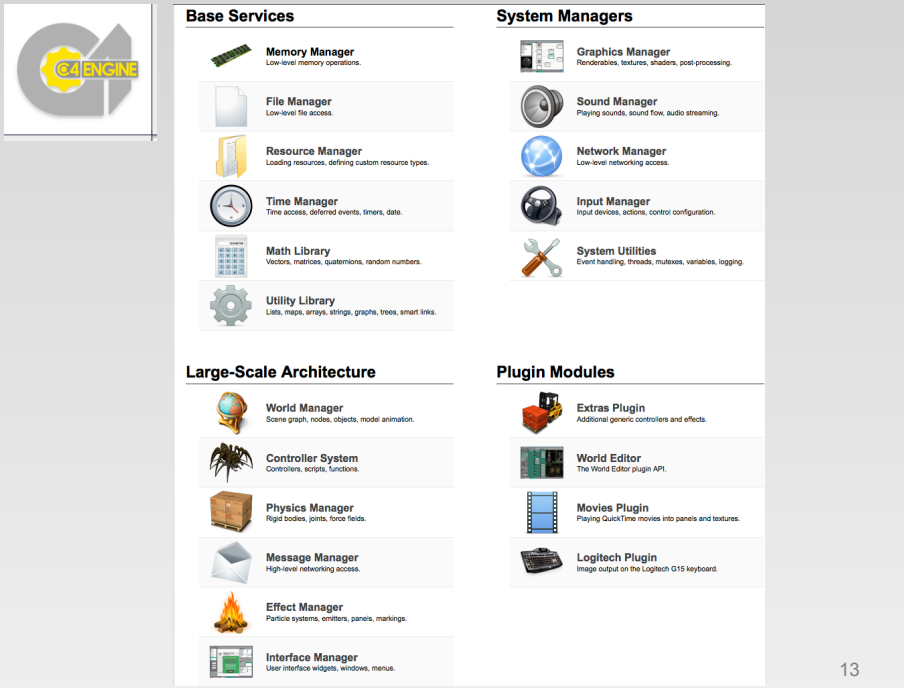
## Modular: Open Class Library (e.g., C4)



- code-oriented development
- very carefully layered
- allows maximum modifiability
- often open source (e.g., jME)
- not as accessible for novices and “casual” programmers

## Basic Game Engine Architecture Blocks





**G4ENGINE**

**Base Services**

- Memory Manager**: Low-level memory operations.
- File Manager**: Low-level file access.
- Resource Manager**: Loading resources, defining custom resource types.
- Time Manager**: Time access, deferred events, timers, date.
- Math Library**: Vectors, matrices, quaternions, random numbers.
- Utility Library**: Lists, maps, arrays, strings, graphs, trees, smart links.

**System Managers**

- Graphics Manager**: Renderables, textures, shaders, post-processing.
- Sound Manager**: Playing sounds, sound flow, audio streaming.
- Network Manager**: Low-level networking access.
- Input Manager**: Input devices, actions, control configuration.
- System Utilities**: Event handling, threads, mutexes, variables, logging.

**Large-Scale Architecture**

- World Manager**: Scene graph, nodes, objects, model animation.
- Controller System**: Controllers, scripts, functions.
- Physics Manager**: Rigid bodies, joints, force fields.
- Message Manager**: High-level networking access.
- Effect Manager**: Particle systems, emitters, panels, markings.
- Interface Manager**: User interface widgets, windows, menus.

**Plugin Modules**

- Extras Plugin**: Additional generic controllers and effects.
- World Editor**: The World Editor plugin API.
- Movies Plugin**: Playing QuickTime movies into panels and textures.
- Logitech Plugin**: Image output on the Logitech G15 keyboard.

13

## Best Choice is Relative to Situation

- Similar issues of needs, resources and constraints (as in buy vs. build)
  - platform, programming language constraints
  - cost constraints (commercial run \$ to \$\$\$)
  - specific technical features required (e.g., MMO)
  - previous experience of staff
  - support from developers, user community (e.g., forums)
  - pedagogical goals (e.g., this course)

## Choice of C4 and Unity for IMGD 3000/4000

### ■ C4 Engine



<http://www.terathon.com/c4engine>

- modular
- C++ language (industry standard)
- reasonable cost
- technically sophisticated
- good support community (forum)



IMGD 4000 (D 11)

15

## Choice of C4 and Unity for IMGD 3000/4000

### ■ Unity 3



<http://www.unity3d.com>

- any second engine better than none ☺
- much better for artists (trivial importing)
- programming in C# (good for structure and robustness)
- debugger support with Monodevelop
- getting very popular



IMGD 4000 (D 11)

16



## Detailed Feature Comparisons



- From DevMaster.net
- Caveats:
  - Info is not audited
  - Let's not get bogged down in the details---the idea is to get overall sense of emphasis

## General Features



- Object-Oriented Design, Plug-in Architecture, Save/Load System:
- Extremely clean class hierarchy for scene graph nodes, including geometries, cameras, lights, sounds, zones, portals, triggers, markers, and special effects
  - General state serialization support for saving worlds
  - Quick save and quick load capabilities
  - Separation between per-instance and shared data
  - External scene graph referencing from within another scene graph
  - Support for pack files and a virtual directory hierarchy
  - Skinable GUI's



- Object-Oriented Design, Plug-in Architecture, Save/Load System:
- Professional FPS controller ready to drop in (and tune)
  - Streamed loading for the Unity Web Player
  - Unity asset server / asset source code version control
  - Cross-platform web player content, the Unity Web Player is available for both Mac OS X and Windows users and works with all browsers
  - Standalone executables for both Mac OS X and Windows
  - Mac OS X Dashboard Widgets
  - iPhone Publishing is available as add-on product
  - Streaming Asset Bundles: the ability to stream in any asset (terrain, mesh, etc) into the game

## Scripting

---



- Graphical script editor
- Scripts are edited graphically for easy artist/designer access
- Games can easily define custom script components, and these automatically appear in the editor
- Controllers can advertise custom function calls that can be accessed from scripts
- Scripts support variables, looping, and conditional execution, all shown in a concise graphical manner



- Uses the Mono and supports JavaScript, C# and Boo, interoperable (to a certain extent) and JIT'ted to native code
- Complete scripting documentation
- Source-level debugging



## Builtin-Editors

---



- Full-featured integrated cross-platform world editor
- Interface panel editor
- Complete built-in windowing system
- Powerful and intuitive interface design
- Advanced surface attribute manipulation and material management



- Editor provides zero-cost asset pipeline: save a file and it updates automatically
- Editor Extensibility: Create completely custom editor windows, and entirely new tools and workflows.
- Asset Server that provides version control capabilities for Unity projects
- Optimized for use with large projects New Server view integrated into the Unity user interface
- Updates, commits, and graphical version comparisons are all performed inside the Unity editor.



## Physics

---



Basic Physics, Collision Detection, Rigid Body:

- Built-in character controller.
- Built-in projectile controller.
- Real-time fluid surface simulation.
- Real-time cloth simulation.



Basic Physics, Collision Detection, Rigid Body, Vehicle Physics:

- Powered by the PhysX Engine, which also supports particle physics
- Cloth simulation



IMGD 4000 (D 11)

21

## Lighting

---



Per-vertex, Per-pixel, Lightmapping, Radiosity, Gloss maps, Anisotropic:

- Support for fully dynamic infinite, point, and spot lights
- Gloss-mapped specular reflections
- Ambient radiosity
- Projected cube and spot textures
- Cook-Torrance microfacet shading



Per-vertex, Per-pixel, Lightmapping:

- Beast-Lightmapping



IMGD 4000 (D 11)

22

## Shadows

---



Shadow Mapping, Projected planar, Shadow Volume:

- All shadows are rendered in real time at global scale
- Three types of shadows are seamlessly combined in one world
- True penumbral soft shadows for area light sources



Projected planar:

- Blob shadows
- Realtime Dynamic Soft Shadows
- Shadows are dynamic, optimized, and allow self-shadowing



IMGD 4000 (D 11)

23

## Texturing

---



Basic, Multi-texturing, Bumpmapping, Mipmapping, Projected:

- Comprehensive bump mapping capabilities
- Enhanced parallax mapping
- Ambient occlusion channels
- Emission/glow maps
- Horizon mapping
- Realistic water shading



Basic, Bumpmapping, Procedural:



IMGD 4000 (D 11)

24

## Shaders

---



Vertex, Pixel, High Level:

- Extensive support for vertex programs and pixel shaders



Vertex, Pixel, High Level:

- Unity comes with an extensive library of 40 shaders including Vertex Lit, Diffuse, Glossy, Bumped, Bumped Specular, Reflective, Self-illuminating, a Toon (Cell) shader, and 9 different particle shaders.
- Everything falls back gracefully on low-end GFX cards.
- Parallax shaders
- Support for GLSL (*cf. lecture later in term*)



IMGD 4000 (D 11)

25

## Scene Management

---



General, Portals, Occlusion Culling, LOD:

- Efficient large-scale visibility determination
- Advanced inter-zone lighting analysis at runtime
- Special support for mirrors and remote portals
- Object instancing and external scene referencing
- Scene data can be imported from Collada format



Occlusion Culling:

- Umbra occlusion culling



IMGD 4000 (D 11)

26

## Animation

---



Skeletal Animation, Animation Blending:

- Full skeletal hierarchy support for deformable meshes
- Powerful hierarchical animation blending system
- Forward kinematics
- Inverse kinematics



Keyframe Animation:

- Skinned character animation
- Procedural Characters and Animation: the ability to stitch multiple body parts into one character, and reassign bones to different characters. The entire skinned animation system is now scriptable.



IMGD 4000 (D 11)

27

## Meshes

---



Mesh Loading, Progressive:

- Support for the Collada scene format, enabling models to be imported from 3D Studio MAX, Maya, XSI, Blender, and other content creation packages



Mesh Loading, Skinning:

- Native importing from Cinema 4D, Maya, Cheetah3D, Blender. Also support for Collada, FBX, 3DS, OBJ formats



IMGD 4000 (D 11)

28

## Special Effects

---



Environment Mapping, Lens Flares, Billboarding, Particle System, Motion Blur, Sky, Water, Fire, Decals, Fog, Mirror:

- Cube environment mapping
- Environment-mapped bump mapping
- Fully extensible particle systems
- Surface markings on arbitrary geometry
- Bump-mapped (fully lit) surface markings
- Real-time fire and electrical effects
- Transparent warping effects (heat haze, etc.)
- Bumpy reflection and refraction
- Postprocessed glow
- Fog volumes
- Full-scene cinematic motion blur
- Interactive in-game interface panels



Lens Flares, Particle System, Motion Blur, Sky, Water, Mirror:

- Color correcting filter, grayscale, sepia, and twirl
- Skinnable in-game GUIs



IMGD 4000 (D 11)

29

## Networking

---



Client-Server:

- Fast, reliable network implementation using UDP/IP
- Solid fault tolerance and hacker resistance
- Advanced security measures, including packet encryption
- Automatic message distribution to entity controllers
- Cross-platform internet voice chat



Client-Server:

- Build on Raknet
- Supports .NET library and asynchronous WWW API
- Multiplayer Networking (advanced NAT punch-through, delta compression, easy to set up)  
(*cf. guest lectures later in term*)



IMGD 4000 (D 11)

30

## Sound and Video

---



2D Sound, 3D Sound, Streaming Sound:

- Fully spatialized 3D sound effects
- Unlimited streaming music channels with seamless looping and concatenation
- Doppler shift and other frequency effects
- High-precision sound travel delay
- Atmospheric absorption effects
- Reverberation with multiple simultaneous environments
- Directional sounds with cone attenuation
- Obstruction attenuation applied to direct and reflected paths
- Frequency-dependent volume settings for all effects
- Permeation system determines how far sounds travel through interiors
- Apple's QuickTime technology can be used to play movies or soundtracks from numerous formats



2D Sound, 3D Sound, Streaming Sound:

- Streaming video and audio Based on FMOD, includes sound effects (Reverb Zones, Various Filters: Low Pass Filter, High Pass Filter, Echo Filter, Distortion Filter, Reverb Filter, Chorus)



IMGD 4000 (D 11)

31

## Rendering

---



Fixed-function, Render-to-Texture:

- Antialiasing (up to 8x)
- Bilinear and trilinear filtration
- Anisotropic filtration (up to 16x)
- Vertical Sync control



Deferred Shading, Render-to-Texture, Fonts:

- Extensible full-screen graphics effects (blur, motion blur, sepia, grayscale, more...)
- TrueType typography
- DirectX 9 and OpenGL rendering support
- Rendering with Shader Replacement: render the scene, replacing shaders of all objects.
- Immediate Mode Rendering: a single function call can render any mesh anywhere, with full per-pixel lighting and custom shader support.



IMGD 4000 (D 11)

32



## Summary Ratings (5 star scale)



Overall: **4.0** (127 votes)  
Features: 4.0  
Ease of Use: 4.0  
Stability: 4.5  
Support: 4.5

**?!?**



Overall: **4.0** (59 votes)  
Features: 4.0  
Ease of Use: 4.5  
Stability: 4.0  
Support: 4.0

