



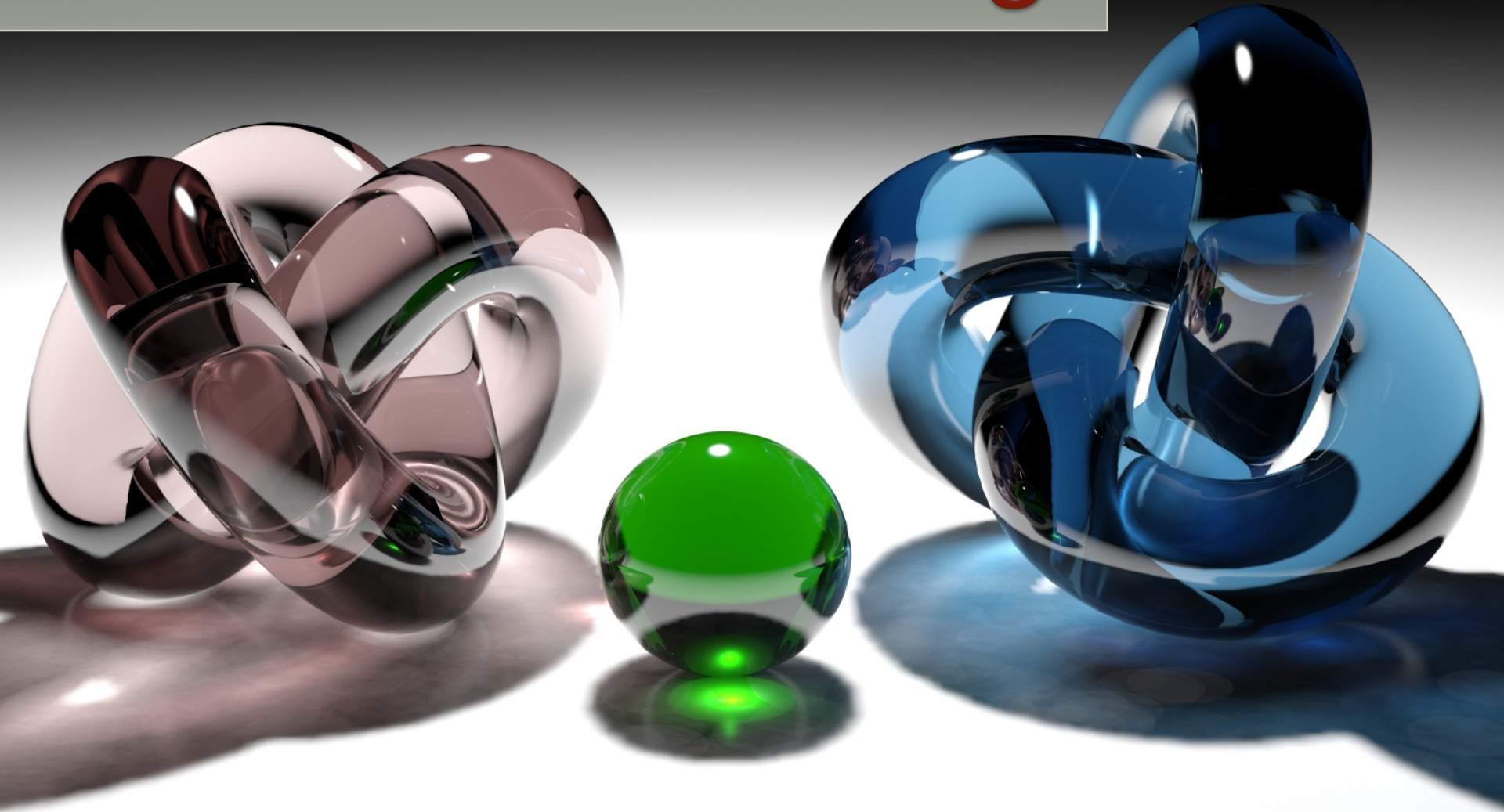
Computer Graphics I

OpenGL Introduction

Introduction

- ❖ Two ways of 3D computer graphics
 - ❖ **1. Pre-rendering**
 - ❖ Ray tracing, ray casting, radiosity
 - ❖ Precise, even physically correct
 - ❖ Slow
 - ❖ **2. Real-time rendering**
 - ❖ DirectX, OpenGL
 - ❖ To fake or not to fake
 - ❖ Hardware accelerated
 - ❖ Fast

Pre-rendering



Real-time Rendering



OpenGL

- ❖ Yet another way to render 3D content
- ❖ We can create graphics and special effects that will appear nearly identical on any operating system and any hardware that supports OpenGL.
- ❖ **Graphics library**
- ❖ Specify a set of commands
- ❖ Each command executes a drawing action or creates a special effect
- ❖ Texture mapping, transparency, antialiasing, fog, and lighting effects
- ❖ Infinite amount of possibilities

Development

- ❖ Long way – 20 years
- ❖ **OpenGL 1.X** – Fixed Function
- ❖ **OpenGL 2.X** – Vertex and Fragment Shaders
- ❖ **OpenGL 3.X** – Geometry Shaders
- ❖ **OpenGL 4.X** – Tessellation and Compute
- ❖ Your version?
- ❖ <http://www.realtech-vr.com/glview/download.html>

OpenTK

- ❖ OpenGL support in C#
- ❖ <http://www.opentk.com/>
- ❖ **Support** – not a new library!
- ❖ Same functions
- ❖ Same principles
- ❖ Under development
- ❖ Advanced projects? – ship to C++ if necessary

Console vs. WinForms

- ❖ Console
 - ❖ Game.cs
 - ❖ Faster but without GUI
- ❖ WinForms
 - ❖ GUI
 - ❖ Slower, but fast enough for simple applications
- ❖ More?
 - ❖ **OpenTK documentation**

OpenTK vs. OpenGL

- ❖ Same parameters
- ❖ Proof

OpenGL

glLoadIdentity
glMatrixMode
glClear
glViewport
gluLookAt
gluPerspective
etc.

OpenTK

GL.LoadIdentity
GL.MatrixMode
GL.Clear
GL.Viewport
Matrix4.LookAt
Matrix4.
CreatePerspectiveFieldOfView
etc.

Exercise Plan

- ❖ **Not strict**
- ❖ **Basic OpenGL**
 - ❖ Console vs. WinForms
 - ❖ Where, what and when
- ❖ **Working environment**
 - ❖ Viewports, perspective and orthogonal projection
 - ❖ Text rendering
 - ❖ Camera
- ❖ **Rendering content**
 - ❖ Shaders

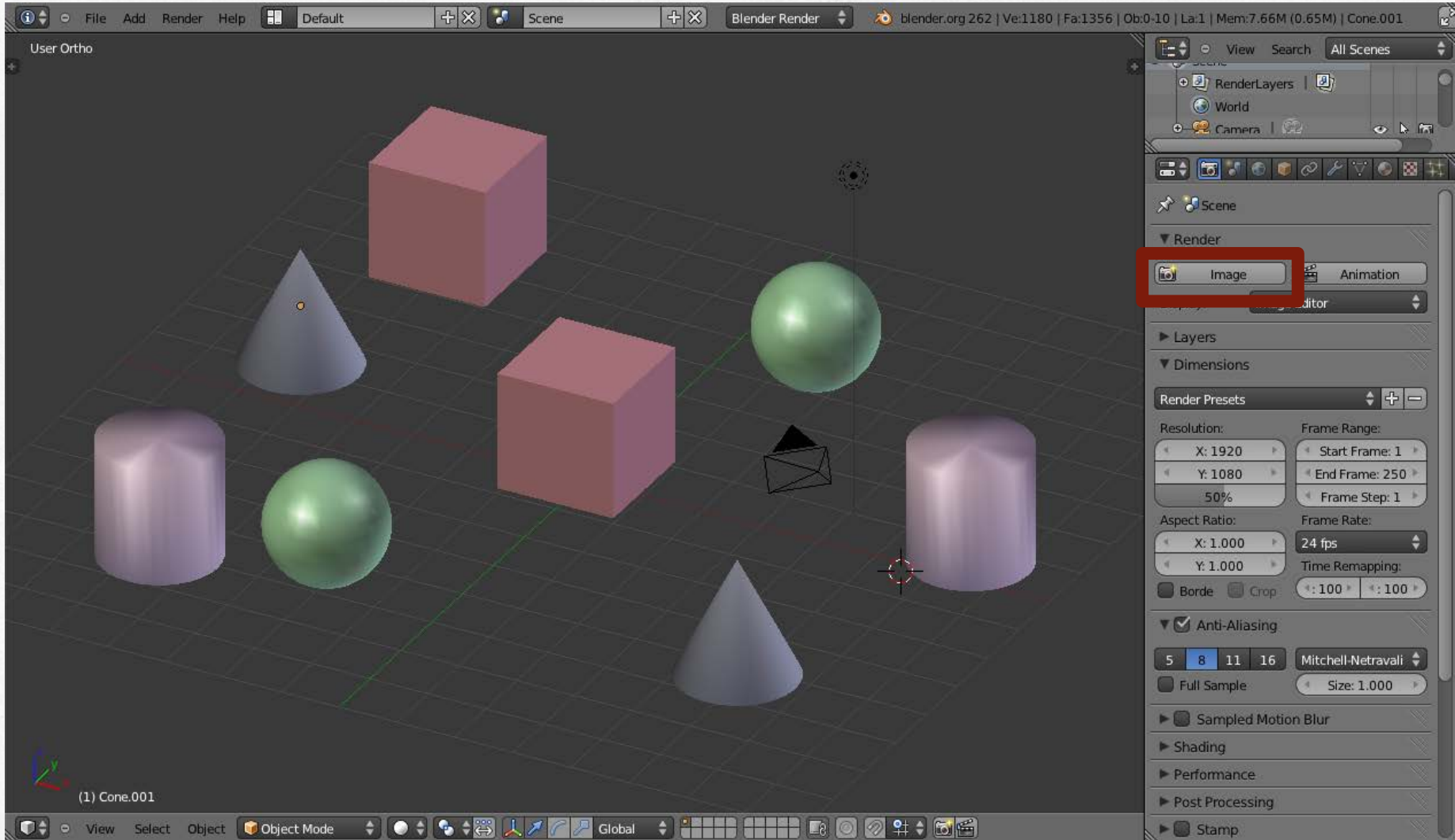
All in All

- ❖ We could divide OpenGL program
- ❖ Basic structure
 - ❖ Init, Reshap, On render frame, On update frame
- ❖ Environment
 - ❖ Projection, Camera, Viewports
- ❖ Rendering
 - ❖ Phong shader, etc.

All in All

- ❖ Now you know how 3D graphics program works
 - ❖ Blender, 3Ds Max, Maya, Zbrush
- ❖ Blender
 - ❖ Modeling environment and game engine in real-time graphics
 - ❖ But if you hit the render button
 - ❖ Pre-render
 - ❖ Compute geometry and start ray tracing

All in All



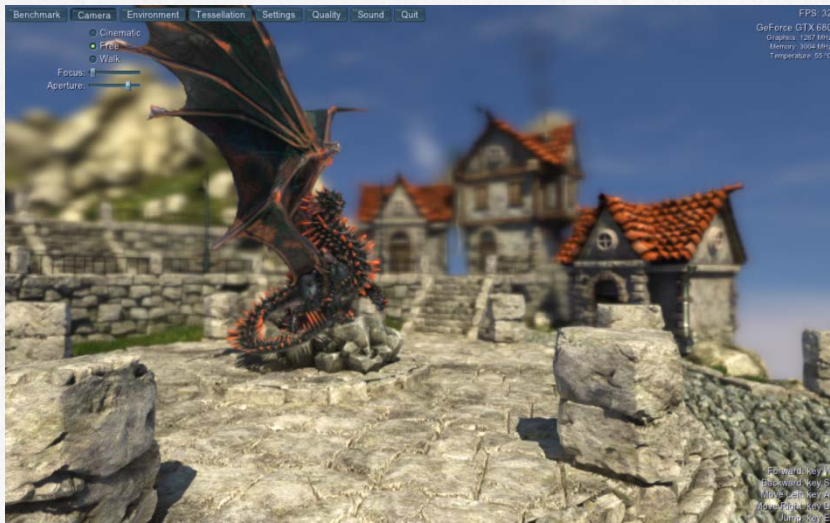
All in All

- ❖ 1. Take an older game
 - ❖ Quake III Arena
- ❖ 2. Take a new game
 - ❖ Bioshock Infinity
- ❖ 3. Compare



All in All

- ❖ See what your graphics card can do today
 - ❖ <http://www.youtube.com/watch?v=D4oRcpmLNgE>
 - ❖ http://www.youtube.com/watch?v=HO8MRT_Ymhk
 - ❖ <https://sketchfab.com/>



Tips

- ❖ Do not try to remember each GL function
 - ❖ Remember its **purpose instead**
- ❖ Do not try to remember each setup of OpenGL
 - ❖ OpenGL in C#, console, WinForms, C++ ...
 - ❖ “for any of these, the setup is kind of different and kind of same – so don’t worry, if you can learn one use of OpenGL, any other use of OpenGL you can learn very, very quickly”

OpenGL – Read This

❖ Base

❖ <http://www.opengl.org/>

❖ 2.X reference -

<http://www.opengl.org/sdk/docs/man/>

❖ Books – older editions

❖ <http://www.starstonesoftware.com/OpenGL/>

❖ Tutorials

❖ <http://www.arcsynthesis.org/gltut/>

❖ Google, GLSL, Shading, Shaders ...

OpenTK – Read This

- ❖ <http://www.opentk.com/doc>
- ❖ “must read” – ~120 pages
- ❖ From your first triangle to shaders in C#
- ❖ Documentation in your installed OpenTK
 - ❖ Examples
- ❖ OpenTK projects
 - ❖ <http://www.opentk.com/project/all>