

Set up OpenGL Programming Environment

**Professor Jim X. Chen
Department of Computer Science
George Mason University
Fairfax, VA 22030**

Setting Up Working Environment for JOGL

- <http://www.cs.gmu.edu/~jchen/graphics/setup.html>

- Install JDK

- uninstall Java first; download JDK and install Java JDK (32 or 64 bits, consistent with IDE and JOGL)

Windows x86	180.82 MB	jdk-8u60-windows-i586.exe
Windows x64	186.16 MB	jdk-8u60-windows-x64.exe

- Installing a Java IDE

- (**Eclipse**, jGRASP, Netbeans, or JBuilder)

- Installing JOGL libraries

Windows: Eclipse simplified JOGL Installation

- Download Eclipse for Java:
 - <http://www.eclipse.org/downloads/>
- Download the new JOGL libraries that matches your platform:
 - http://jogamp.org/wiki/index.php/Downloading_and_installing_JOGL#Downloading_the_latest_stable_version
- Download the following two files
 - jogamp-all-platforms.7z
 - jogl-javadoc.7z

Windows simplified instructions

- Start Eclipse using a working directory other than the sample programs’.
- Create a project corresponding to the directory: cs551 (or whatever you want to call)
 - [Unzip](#) jogamp-all-platforms.7z and jogl-javadoc.7z into this directory
 - In Eclipse under “Project->Properties-> (Java Build Path) -> Libraries”, you can use “Add External JARs” to add “gluegen-rt.jar” and “jogl-all.jar” under cs552/jogamp-all-platforms/jar/.
 - Click the triangle at the beginning of “gluegen-rt.jar” and “jogl-all.jar”; you have Native library location; add the directory of “cs552/jogamp-all-platforms/lib/windows-xxx/”, which allows access to the “*.dll” files.
 - Click the triangle at the beginning of “jogl-all.jar”; you have Javadoc location; add the directory of “cs552/jogamp-all-platforms/jogl/javadoc/”, which allows JOGL commands’ tool tips in Eclipse.
- Try a sample JOGL program
 - By default you may create a directory “src” under cs551, and put the first sample program from the new textbook here.
 - In eclipse you can click on the project cs551, and find src and under default package

For sample joglExamples2013

Eclipse

- Download the old examples and unzip them: joglExamples2013 ;
- Start Eclipse using a working directory other than the sample programs’.
- Create a project corresponding to the directory: joglExamples2013.
 - Under “Project->Properties-> (Java Build Path) -> Libraries”, you can use “Add External JARs” to add “gluegen-rt.jar.jar” and “jogl.jar” under “joglExamples2013/jogl-1.1.1-windows”.
 - Click the triangle at the beginning of “gluegen-rt.jar” and “jogl.jar”, you have Native Library Location, add the directory of your “*.jar” files, which allows access to the “*.dll” files.
 - Remove the two dead “*.jar” file names if you see them. They are created when I compile my program under my specific directory.

jGRASP

- In the project under “Settings->PATH/CLASSPATH->Workspace”, you can add the directory of the “*.dll” files to the system PATH window, and add “*.jar” files with full path to the CLASSPATH window.

Mac OS (Obsolete)

- <http://www.cs.gmu.edu/~jchen/graphics/jogl/notes/joglSetup.html>
 1. Download Eclipse;
 2. Download Mac OS X: *.jar and *.jnilib (which are packaged in the jar file: [jogl-1.1.1a-macosx-ppc.zip](#) or [jogl-1.1.1a-macosx-universal.zip](#) depending on your platform)
 3. Copy all downloaded files into directory /System/Library/Java/Extensions/
 4. Download the examples and unzip them: joglExamples2011
 5. Start Eclipse, create a project corresponding to the directory: joglExamples201
 6. In eclipse, you can add "*.jar" (the two file names) under "Project->Properties->Libraries". Remove the two dead "*.jar" file names if you see them. They are created when I compile my program under my specific directory.
 7. Try the examples. They should run from here.

<http://jogamp.org/>

- JOGL's new home
- Tutorial

http://jogamp.org/wiki/index.php/Jogl_Tutorial

- You can follow this page for the latest

OpenAL

- **OpenAL (Open Audio Library)** is a cross-platform audio [application programming interface](#) (API). It is designed for efficient rendering of multichannel three dimensional positional audio. Its API style and conventions deliberately resemble those of [OpenGL](#).

OpenCL

- **Open Computing Language (OpenCL)** is a framework for writing programs that execute across heterogeneous platforms consisting of central processing units (CPUs), graphics processing units (GPUs), digital signal processors (DSPs) and other processors.

Computer Graphics

Jim X. Chen, Ph.D.

Director of Computer Graphics Lab

George Mason University

Reference

- V. Scott Gordon and John L. Clevenger, [Computer Graphics Programming in OpenGL with Java](#), January 2017.
- Jim. X Chen and Chunyang Chen, *Foundations of 3D Graphics Programming*, Springer Verlag, Second Edition.
- Jim. X Chen, *Guide to 3D Graphics Tools*, Second Edition, Springer Verlag, companion in C with a survey of graphics tools.
- Mason Woo, Jackie Neider, and Tom Davis, *OpenGL Programming Guide*, Addison Wesley. Programming examples
- James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes, *Computer Graphics: Principles and Practice*, Addison Wesley. Detailed theory in depth.
- Donald Hearn and M. Pauline Baker, *Computer Graphics*, Printice-Hall. Easy to read textbook for undergraduates. Contains more than enough.

Objects and Models

- *Objectives*

- Introduce basic graphics concepts
 - object, model, image, graphics library, frame buffer, scan-conversion, clipping, and anti-aliasing
- Set up OpenGL programming environment
- Understand simple OpenGL programs

Computer Graphics



Points



Lights



Raytracing



nfaces

- CG displays or animates real or imaginary objects from their computer-based models;
 - Modeling creating and modifying a model
- image processing treats the inverse process: the analysis of images, pattern recognition, or the reconstruction of 2D or 3D objects from images.

Display

- A graphics *display* is a drawing area comprised of an array of fine points called **pixels** (picture elements).
- At the heart of a graphics system there is a **magic pen**,
 - move at lightning speed to a specific pixel
 - draw the pixel with a specific color — a red, green, and blue (RGB) vector value.
 - Computer graphics is about using this pen automatically through programming.



Object, Model, and Image

- A real or imaginary *object* is represented in a computer as a model, and is displayed as an image.
- A *model* is an abstract description of the object's shape (vertices) and attributes (colors),
 - which can be used to find all the points on the object corresponding to the pixels in the drawing area.
 - Given a model, the application program will control the pen through a graphics library to generate the corresponding image.
- An *image* is simply a 2D array of pixels.

Interactive

- User controls the creation, modification, and animation of the models through input devices (keyboard & mouse)



Primitive and Graphics Library



Primitive Lab

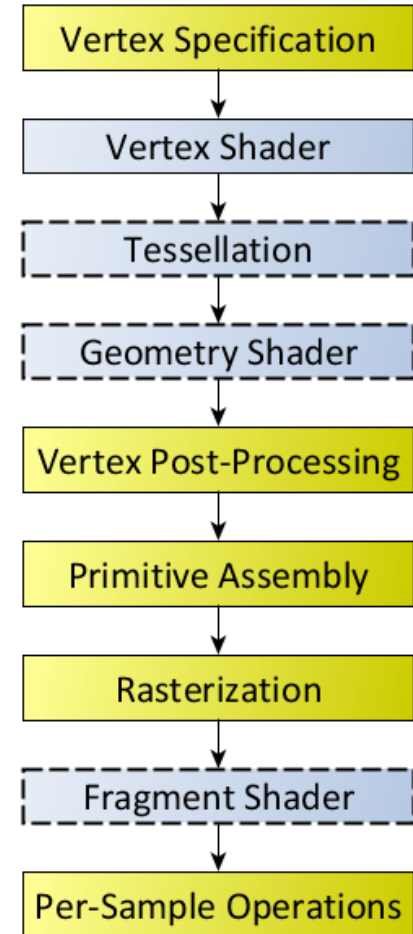
- A *graphics library* provides a set of graphics commands or functions.
 - bound in *C*, *Java*, or other programming languages on different platforms.
 - specify primitive 2D and 3D geometric models to be digitized and displayed.
- *Primitive models* or simply *primitives* stand for some simple shapes (such as points, lines, and polygons)
- *OpenGL* is a graphics library; *DirectX* includes a graphics library *Direct3D*

Old OpenGL Programming

- OpenGL is the most widely used graphics library (GL) or application programming interface (**API**)
- Compile and run Example *J1_0_Point.java*
 - To change the font size in Eclipse: Window->Preferences... General->Colors and Fonts->Basic->Text Font, then select.
- links to all the example programs, and setting up working environments on different platforms:
 - <http://www.cs.gmu.edu/~jchen/graphics/>

OpenGL 4.x Programming

- Set up working environments
- Compile and run Example:
 - *JOGL1_0_Frame.java*
- **For efficiency**, under Run->Run configurations, select application: *JOGL1_0_Frame*
 - Select Arguments folder button
 - Add under “VM arguments:”
 - -Djogamp.gluegen.UseTempJarCache=false
 - -Dsun.java2d.d3d=false



History of OpenGL

- https://www.khronos.org/opengl/wiki/History_of_OpenGL#Deprecation_Model
- OpenGL has evolved over the years dramatically, but the graphics theory remains.
- We will learn the theory and use the new OpenGL platform to exercise for implementations

Set up your programming environment (due before next class)

1. Set up your programming environment, and run the first sample program that draws a point (or randomly generated points).
2. If you failed, schedule a time to come to my office or bring your computer to the next class.
3. You don't have to let me know if you succeed in setting up your working environment.
4. You have to let me know if you cannot get the sample programs to run within a week. Please come to my office during my office hours or make an appointment with me or my TA.