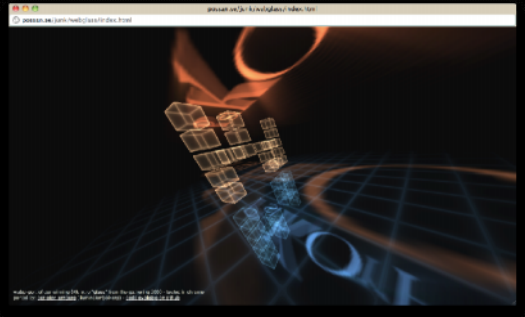
ykode

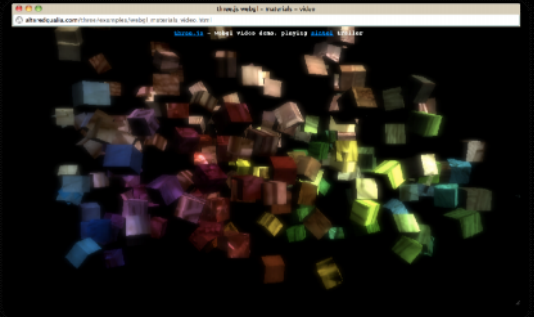
Sharing and code together ☺

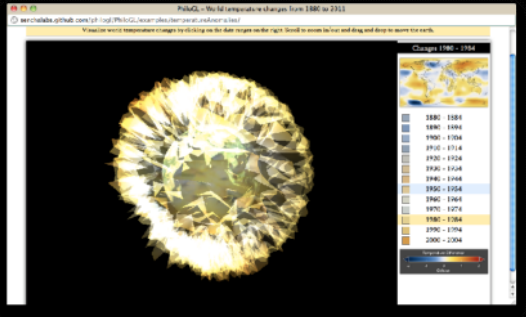
Bancakan 2.0

Ykode #2
AngkringQ, March 13 2011

Makes Web 3D Contents Possible

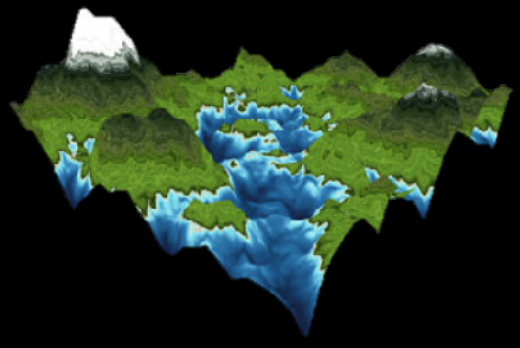


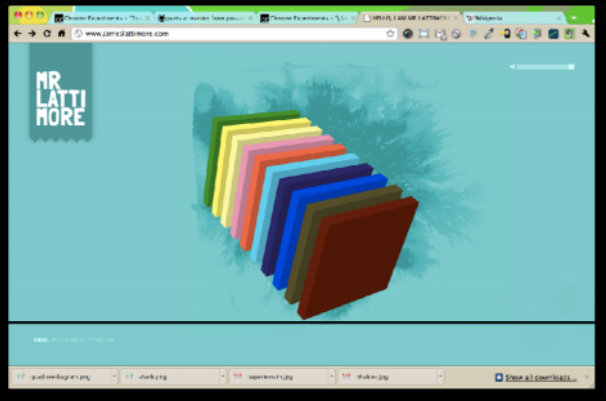




Before

Flash - PaperVision



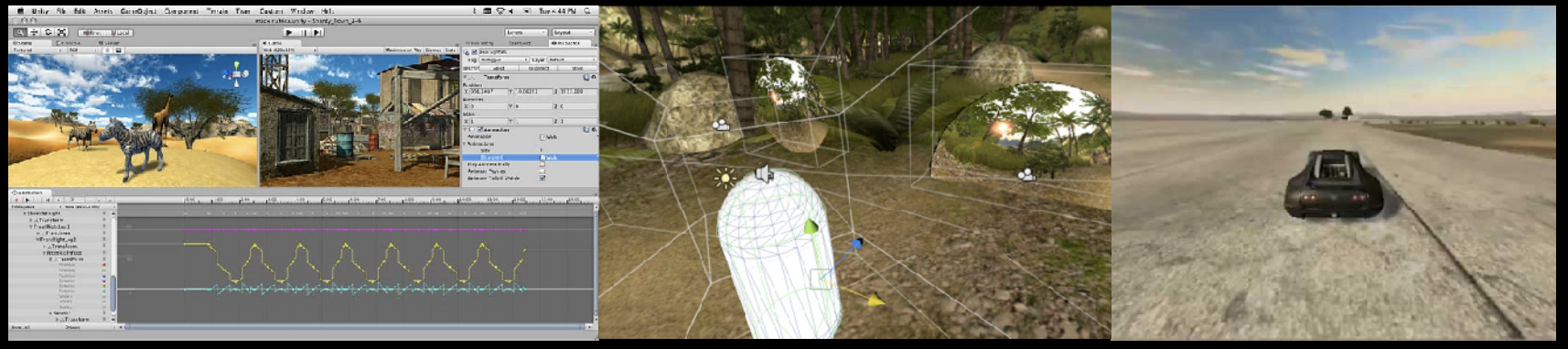


- Using Flash as the Rendering Engine
- 100% Software
- Not hardware accelerated
- Not programmable pipeline
- 3D Support coming Flash 11 – Molehill API

Silverlight

- No Known 3D Engine
- Support a little stereoscopic Perspective 2D Transformation (fake 3D)
- 3D support coming with Silverlight 5

Unity



- Well-known engine amongst Game Programmer
- Fully 3D accelerated.
- Programmable Shader Available
- Pro costs $1500 for each platform

Common Problems

- New Language to Learn, not web-centric.
  - An HTML/JS programmer suddenly needs to program with C# (unity/silverlight) or AS3 just to get 3D
- Use Plug-ins.
  - Install plugins or die
- Non-Standard.
  - It's adobe, microsoft, and unity property
- Some of them too costly
  - $1500 isn't sexy for Students in the 3rd world country

Introducing WebGL

- It’s standard (by Khronos Group)
  - Spec 1.0 released March 3, 2011 – fresh from the oven :D.
- Supported on Modern Browsers
  - Mostly beta/nightly build.
  - Except Internet Explorer (what do you expect from them :P)
- 100% Hardware Accelerated
- 100% Programmable Pipeline

WebGL Characteristics

- Compatible with OpenGL ES 2.0 API
  - The same OpenGL ES used by iPhone and Froyo
- Programmed using Javascript
- Shaders coded using ESSL
  - Shaders is a program runs on GPU during rendering pipeline.

Advantage of WebGL

- It’s fully hardware accelerated
- It based on OpenGL ES 2.0
  - Supports shaders
  - You can use the same paradigm for coding iPhone and Android ☺
- Scripted. No need of compile-debug cycle to get into.

Realtime vs Pre-render

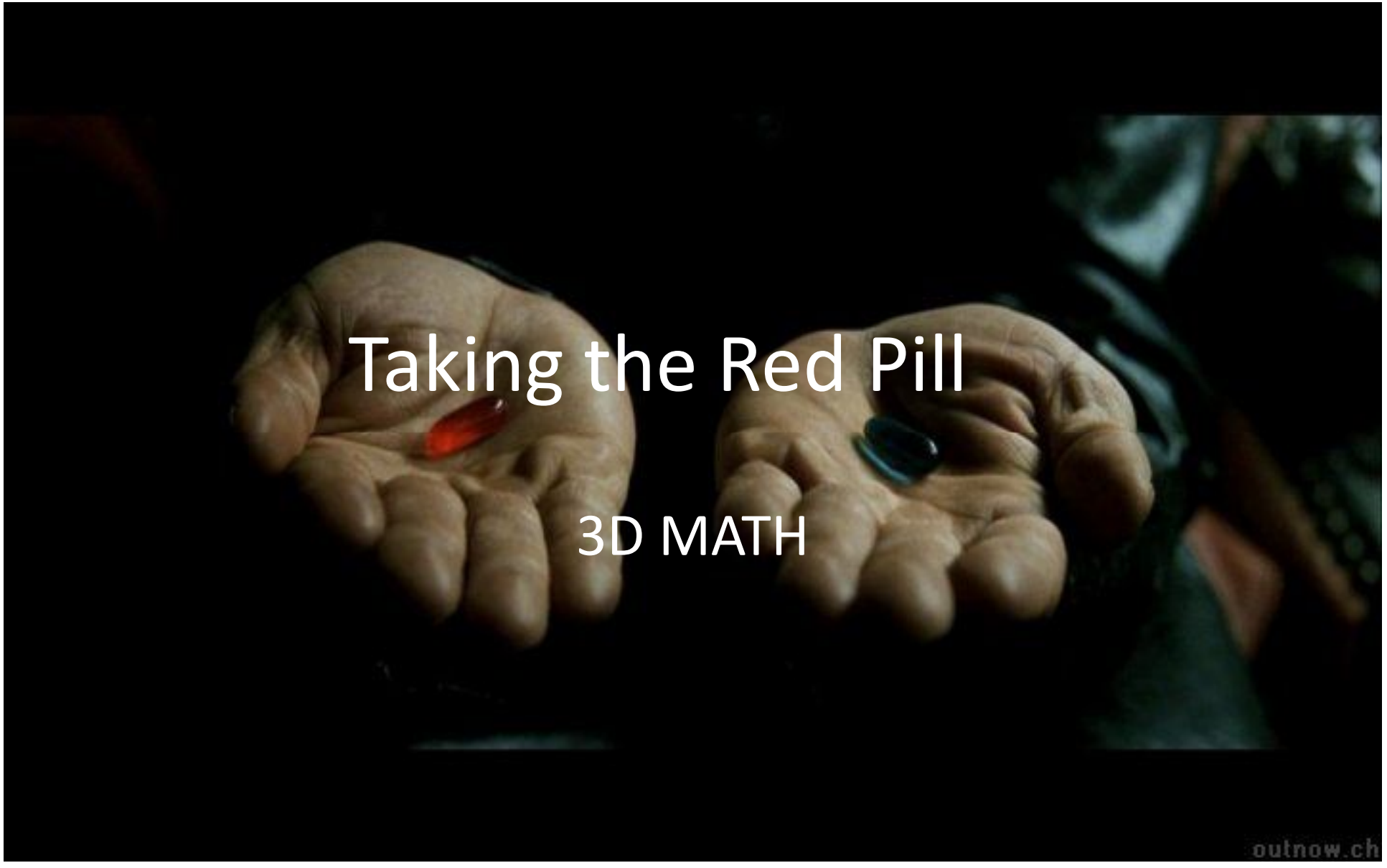
Realtime

- Example: Game
- Has deadline on howlong you produce a frame, usually 60fps -> 1 frame every 16.6 ms.
- Has budgets for polygon & texture.
- To produce, needs coding

Pre render

- Example: Movies
- Has no deadline, you can render each frame for minutes/hours.
- Has no budgets for polygon & texture.
- Just use DCC tools like 3D S Max, Cheetah, or Modo

Demo Time


Taking the Red Pill
3D MATH
outnow.ch

It’s just high-school math

- Geometry
  - E.g. understand how line and points formed
- Coordinate
  - Everybody know cartesian
- Vector & Matrix
  - Add, subtract, cross, dot
- Simple Linear Algebra
  - Mostly Interpolation
- Trigonometry
  - Sine, cosine, tangent
- Some Numerical Method (e.g. newton)
  - Mostly for optimization

What’s not

- Calculus
  - Left it to scientist
- Statistics

That’s why you were taught
those things on School and College

Don’t worry, you’ll be OK

Because they teach you how to get better income ;)

PHP Programmer - $56000
3D Programmer - $96000

IN USA

And they teach you the way you can create these things

Or these things....

Yes, this things make money ☺

Made in Magelang
Fandrey, Lucidrine
Got $6K for himself

Made in Jogja
$xxxxxx
Player 3M

Made in Jakarta
Undisclosed amount
Toge Production

Made in Bandung
Menara Games
Undisclosed amount
Paid Game

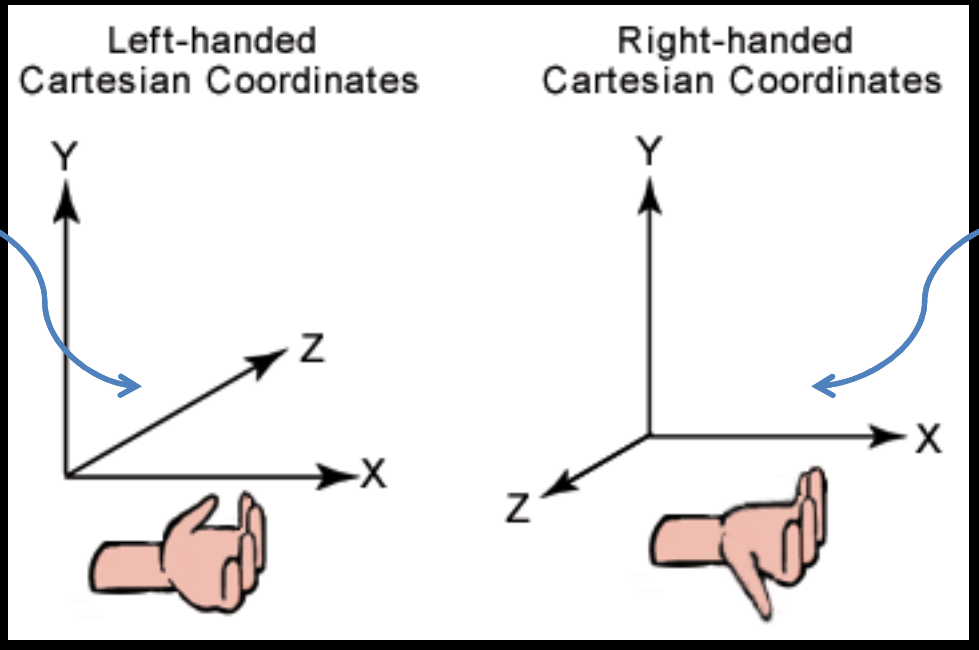
So what I have done?

- Originally written using Visual C++ 7.1 and Borland C++ 6
- Licensed MMORPG game with source code from Gravity Korea
- I rewrite most network and database engine to make it runs on linux
- Rewrite portion of graphic engine and tools to accommodate Windows Vista, Windows 7.
- Implementing anti-cheat technology (in-house)
- These things costs $80K to license.

Before, 3D is the dominated by Native Apps
Now you can run it using Javascript on Web Browser
(except Internet Explorer ☺)

Cartesian Coordinate

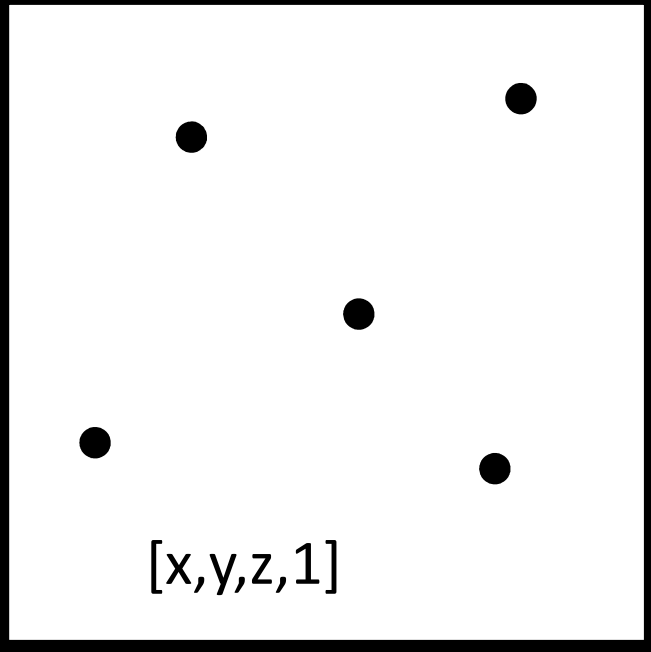
Default in Direct3D/XNA

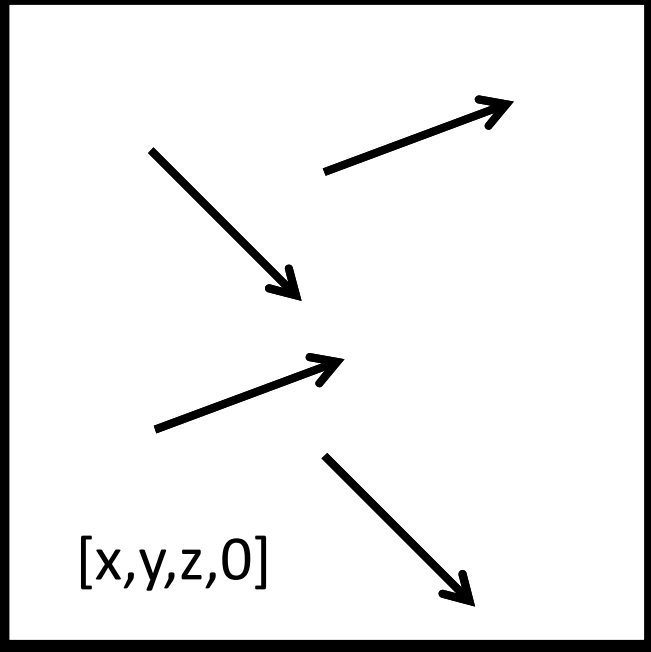


Default in OpenGL

WebGL uses Right-Handed Cartesian coordinate

Point vs Vector





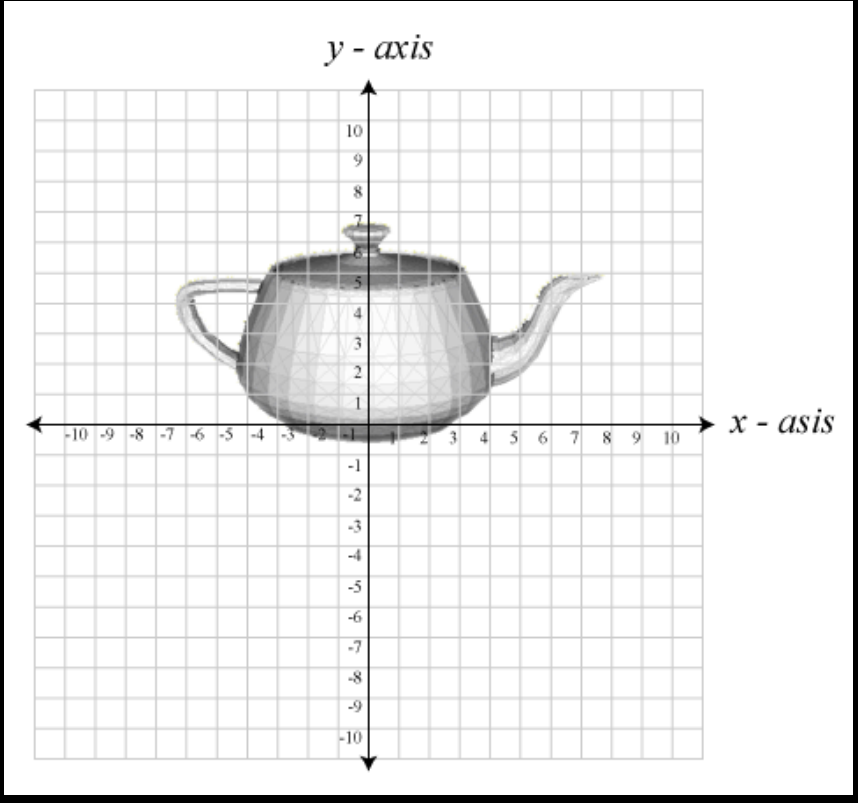
Vectors in OpenGL are 3-Dimensional.

Matrix

$$\begin{bmatrix} -3 & 2 & 2 & -3 \\ -8 & 7 & 7 & 9 \\ 5 & 3 & 2 & 4 \\ 9 & 7 & 8 & 2 \end{bmatrix}$$

We usually deals with 4x4 SQUARE matrices

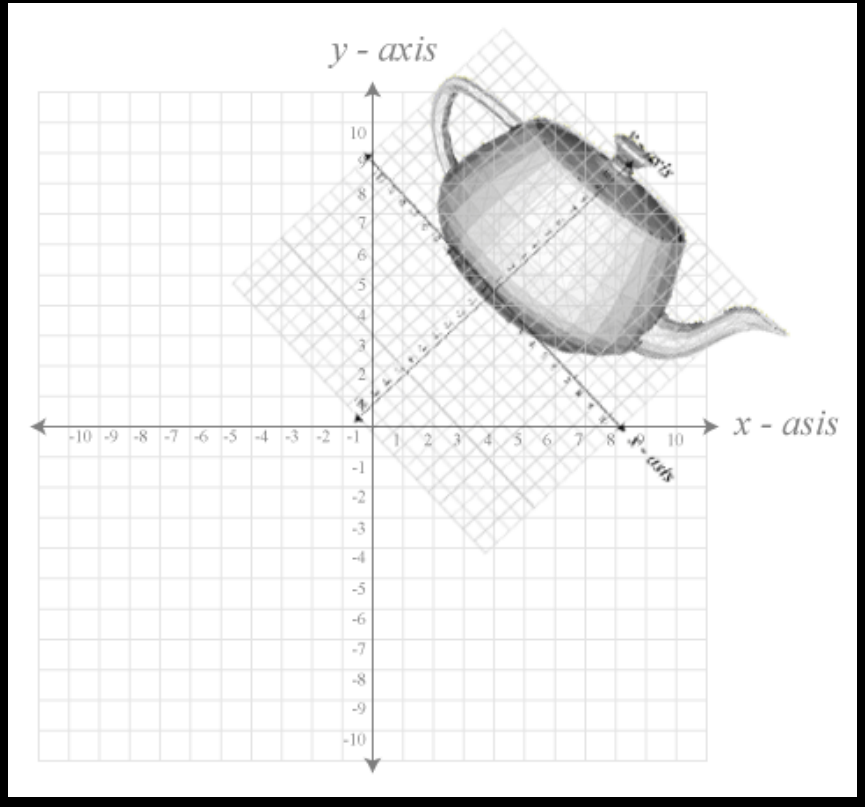
Local Space



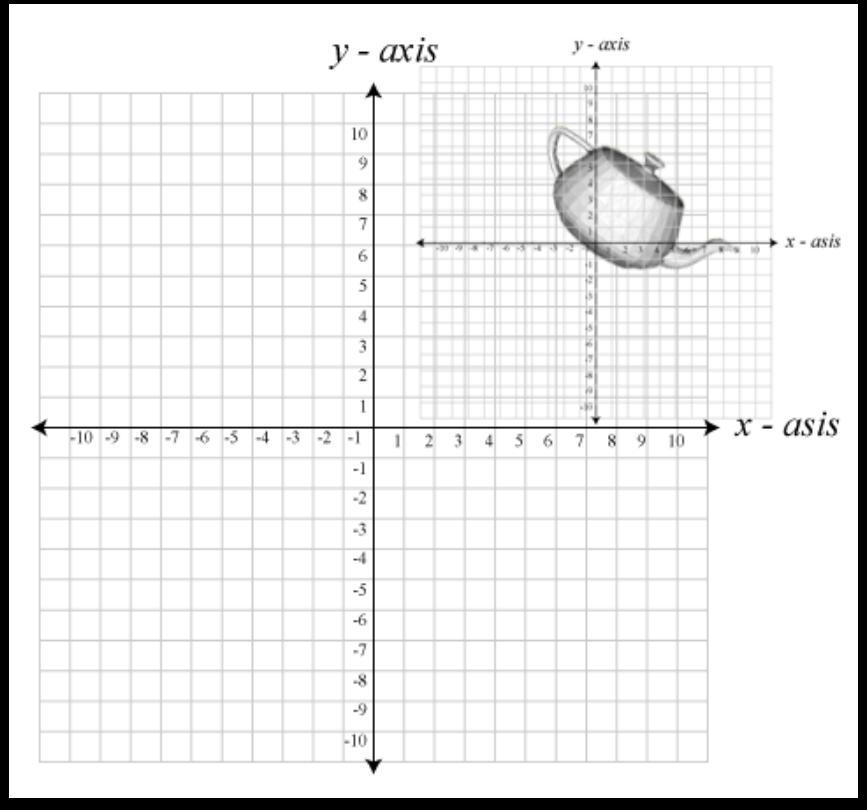
- Object Vertices expressed relative to the object.
- The way artist/modeller to be displayed without moving from origin.

World Space

- The location of objects in the world.
- Global coordinate space for which all other objects described.

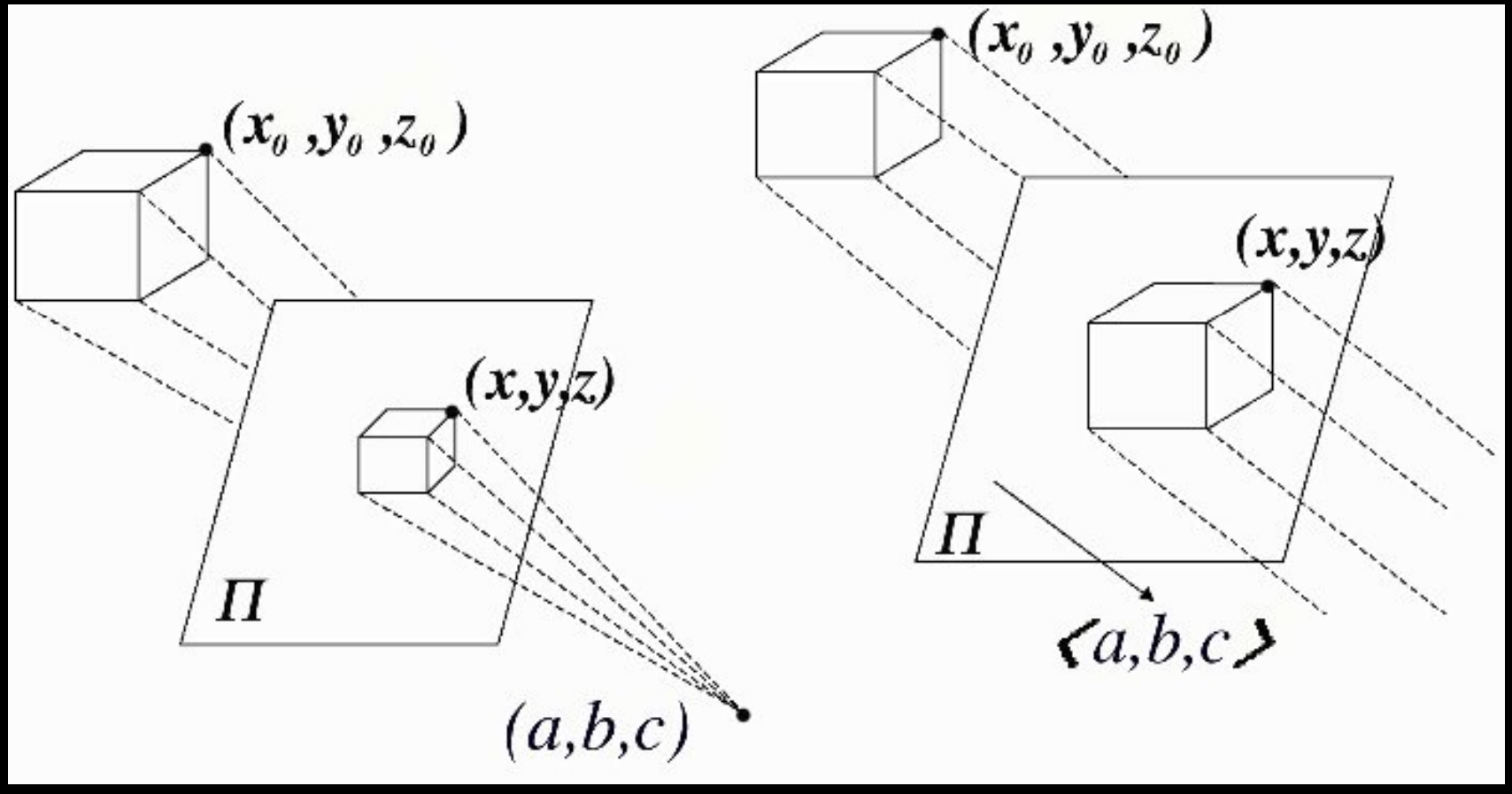


Inertial Space

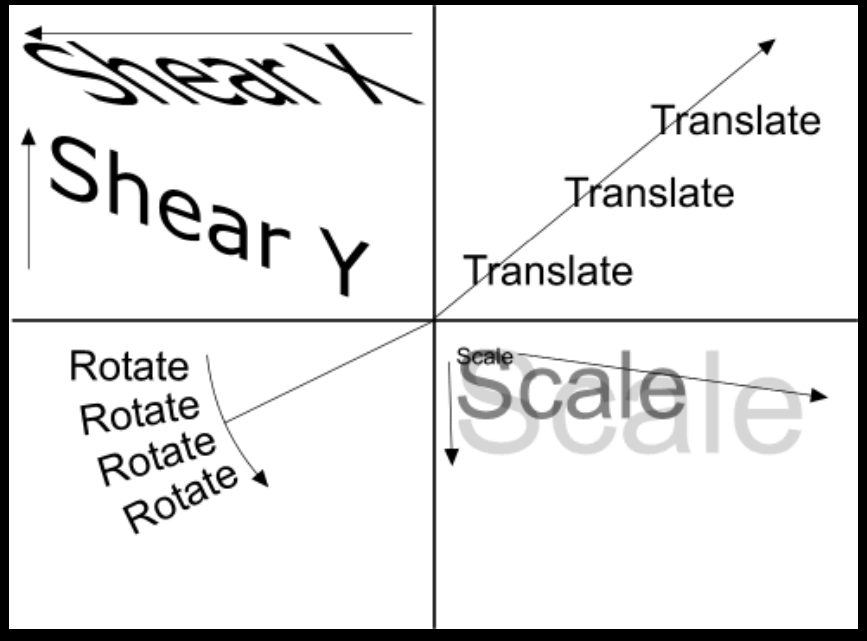


- “halfway” between world and local space
- World -> inertial by translation.
- Local -> inertial by rotation.

Projection

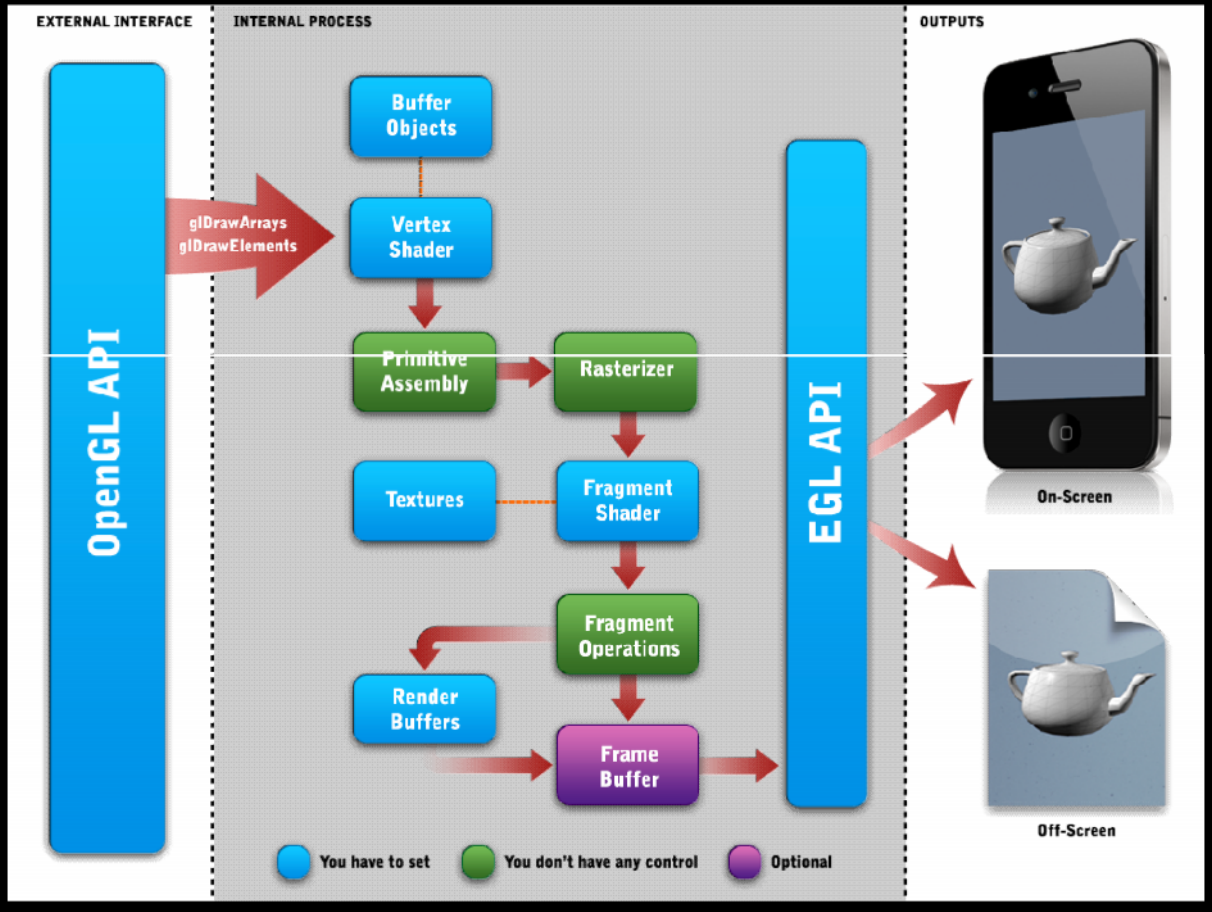


Transformation

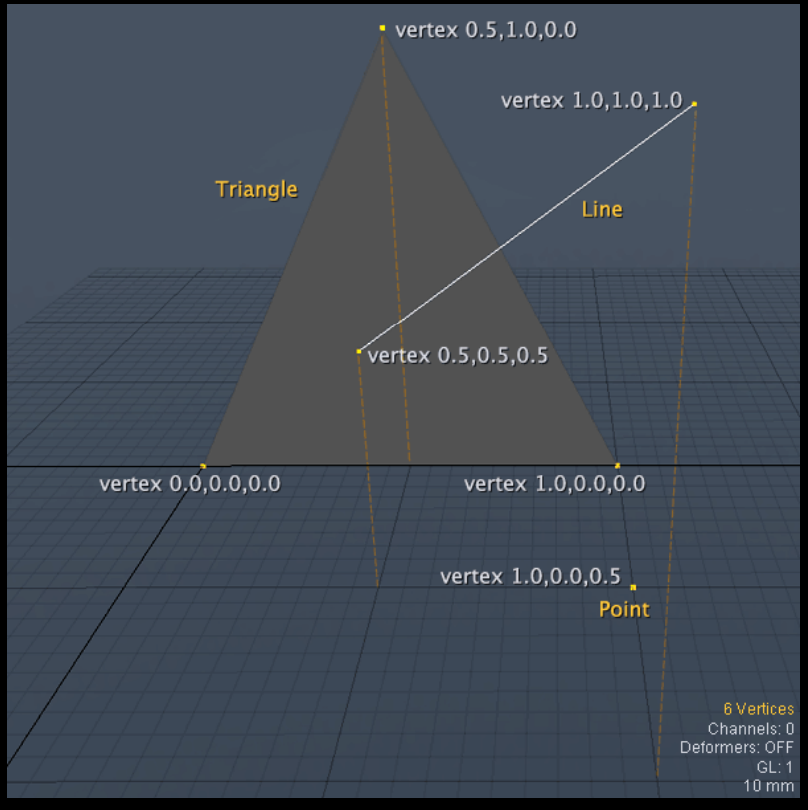


- Using Matrices
- Heavily used.
- Sequence matters
- We transform the COORDINATES not INDIVIDUAL VERTEX VALUES

Rendering Pipeline

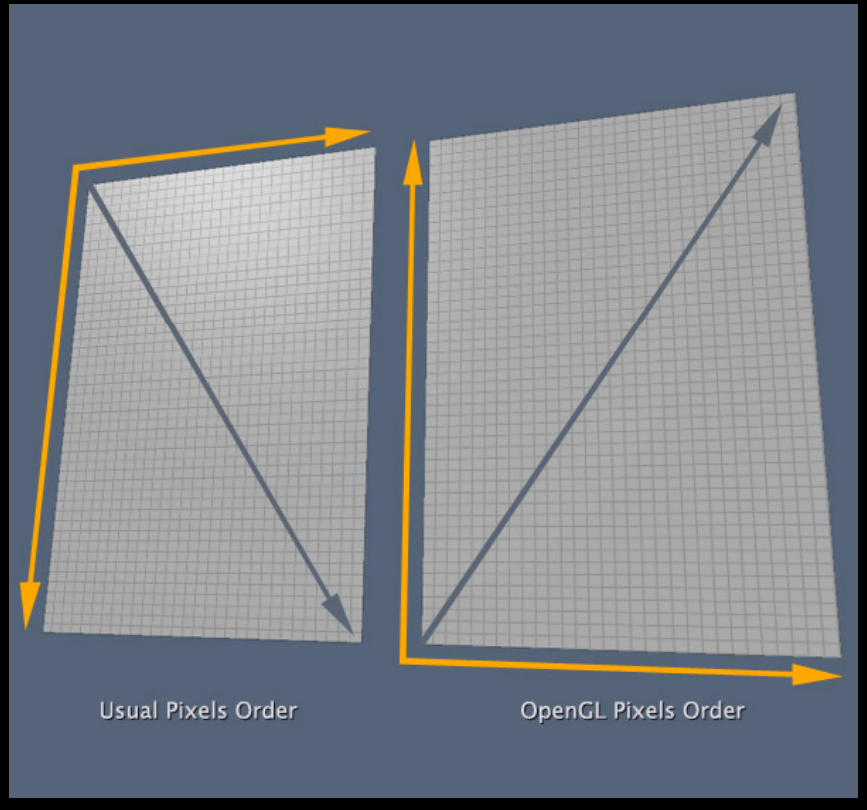


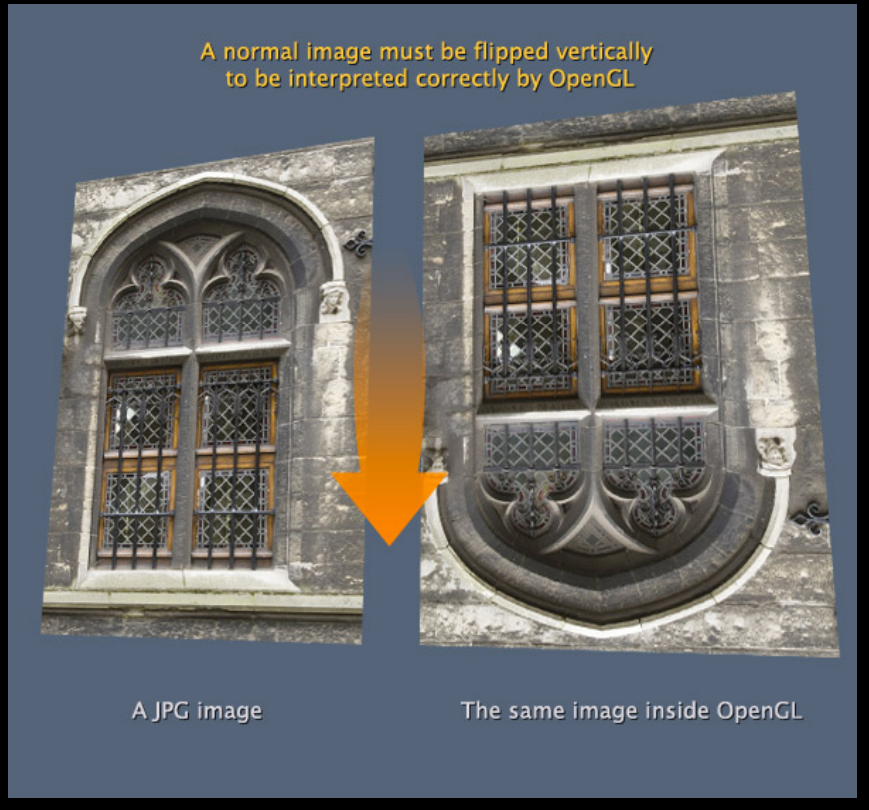
Primitives



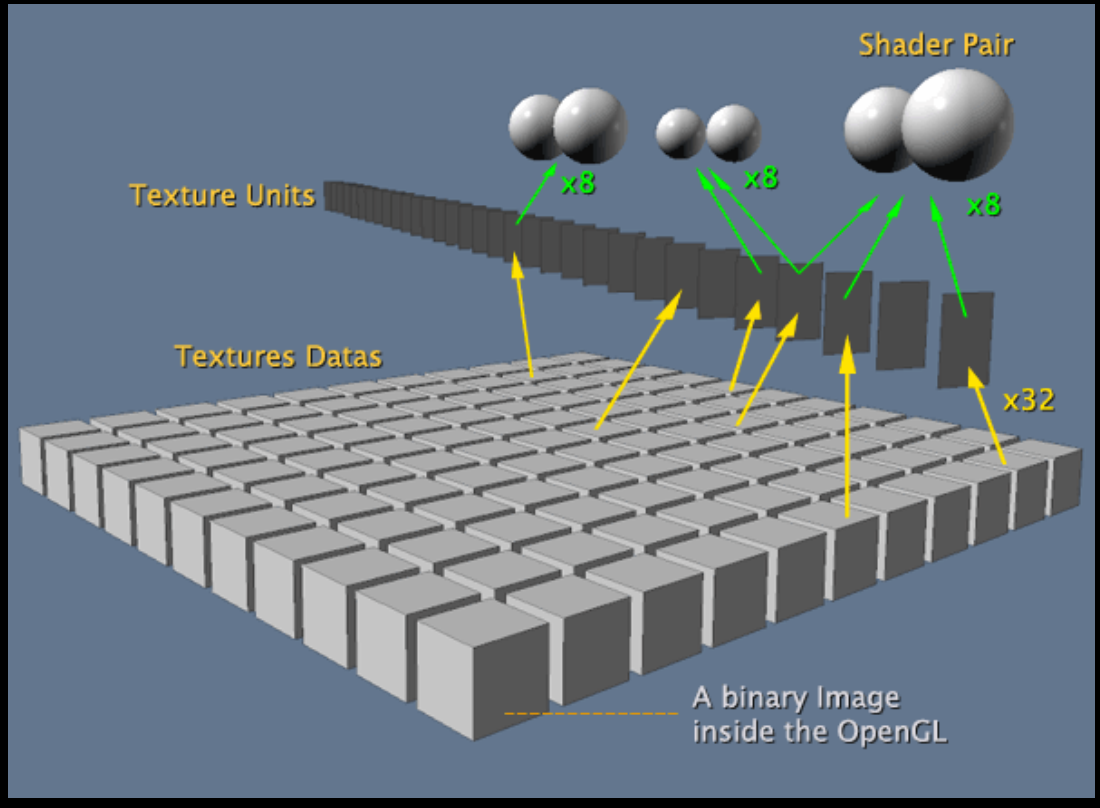
- Line
- Point
- Triangles
  - All kind of polygon can be formed using triangle!

Textures

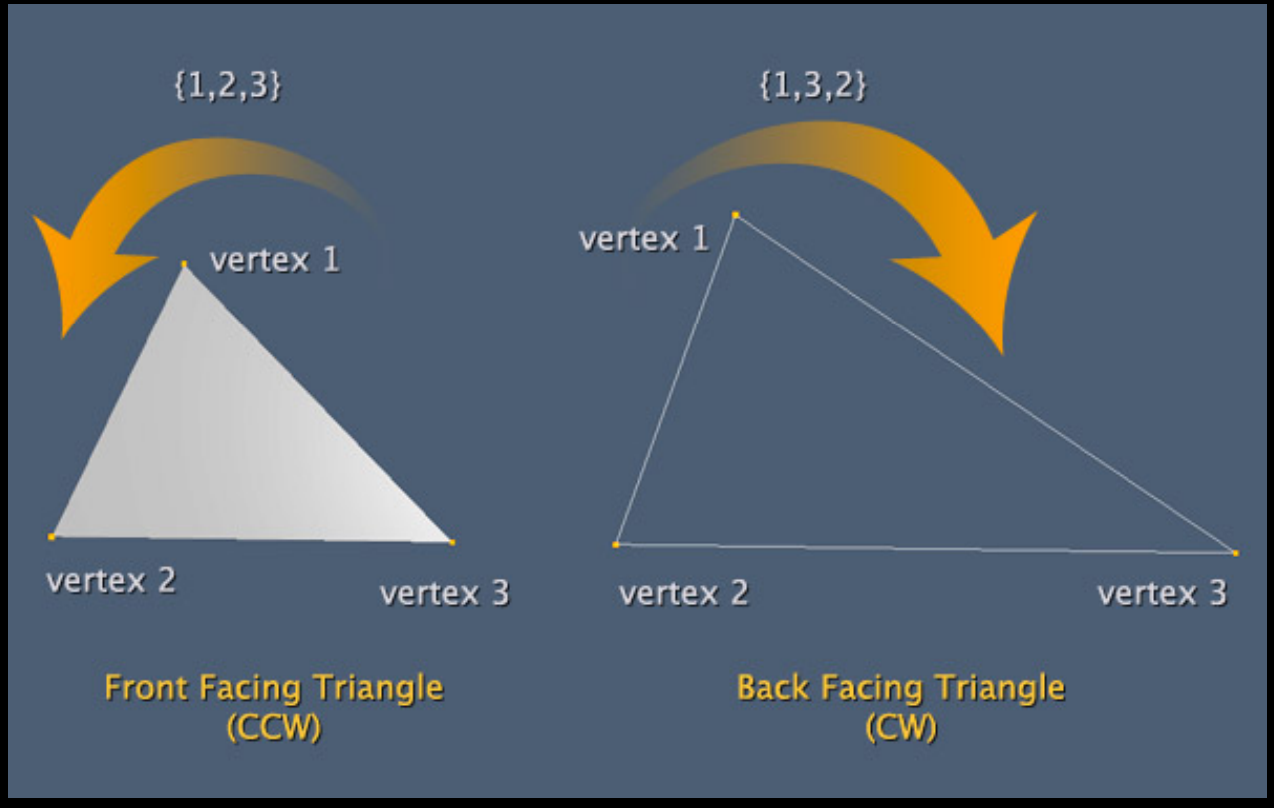

Usual Pixels Order
OpenGL Pixels Order


A normal image must be flipped vertically
to be interpreted correctly by OpenGL
A JPG image
The same image inside OpenGL

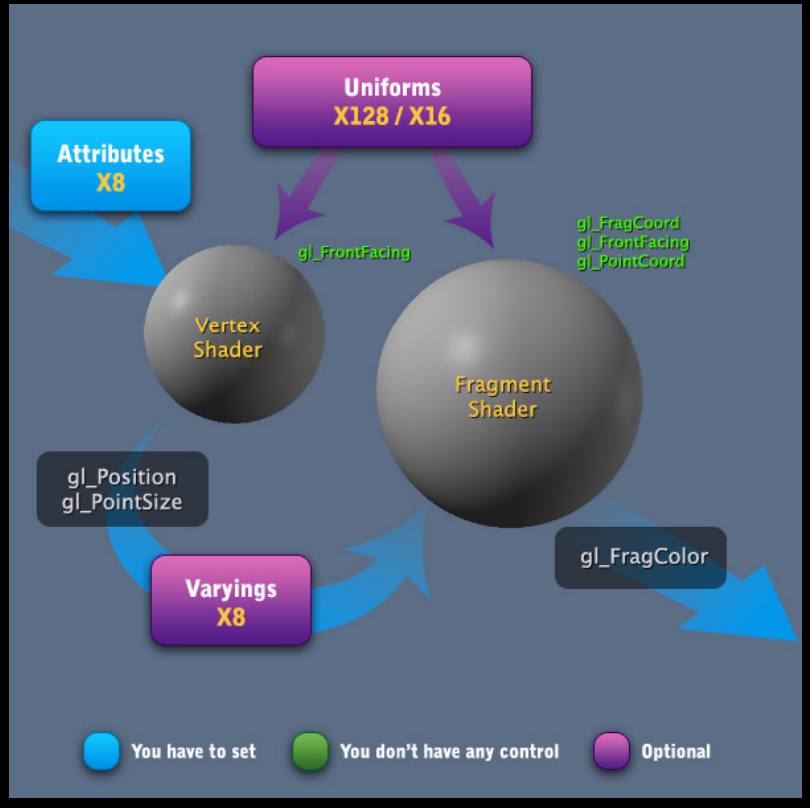
Texture Unit



Face Culling

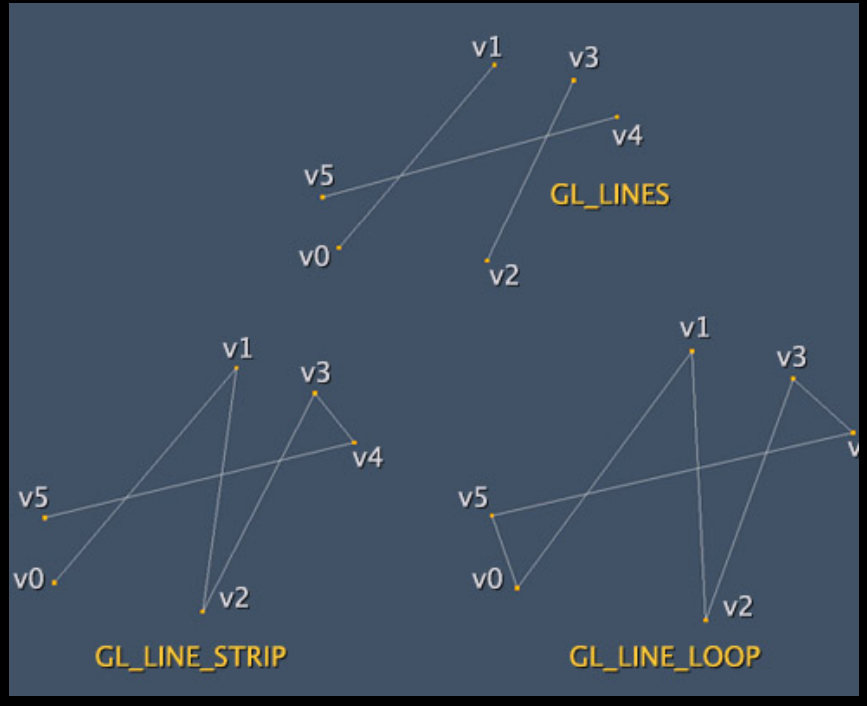


Shaders

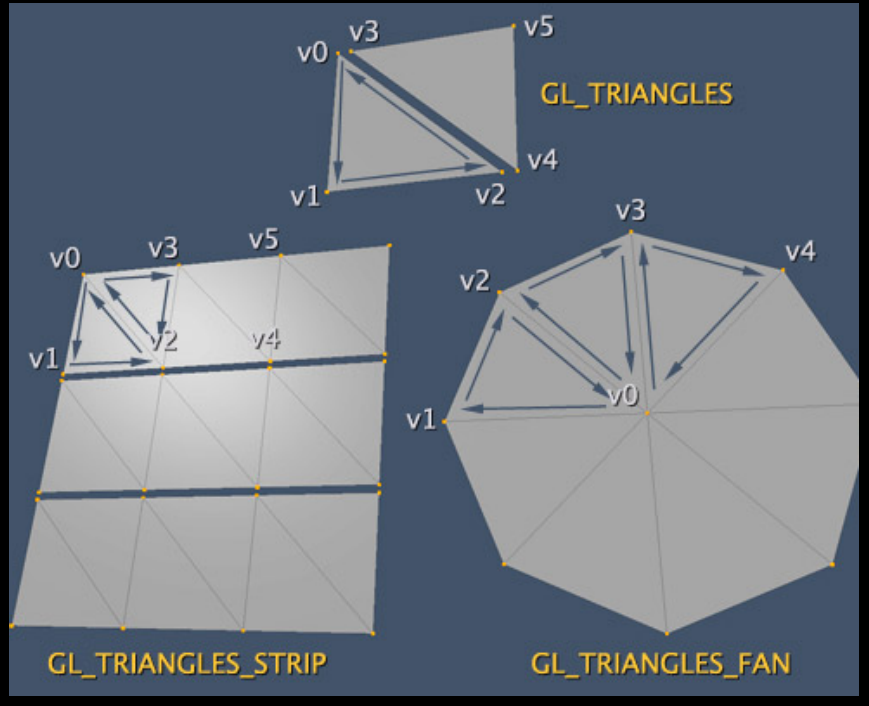

Uniforms
X128 / X16
Attributes
X8
gl_FrontFacing
gl_FragCoord
gl_FrontFacing
gl_PointCoord
Vertex
Shader
Fragment
Shader
gl_Position
gl_PointSize
Varyings
X8
gl_FragColor
You have to set
You don't have any control
Optional

Drawing

Lines



Polygon/Tris



Hands-on Lab

- Creating Context
- Drawing 2D
- Drawing 3D
- Texturing
- Lighting
- Book: Essential Math for 3D programming (a programmer guide)