CSE 3200 Micro-Computer Graphics
Graphics API

Presenter: Girendra Persaud
University of Guyana

Outline

- Declarative & Imperative Modeling
- API
- Some History
- Low Level APIs
- High Level APIs
- Direct 3D
- Java 2D & Java 3D
- Renderman Interface Specification
- RenderWare
- OpenGL
- Conclusion

Model & Modeling

- Model: A shape, object, form defined in a strict language –
  - the creation of a model
- Modeling is the creation, application, manipulation of a model
  - 3D modeling

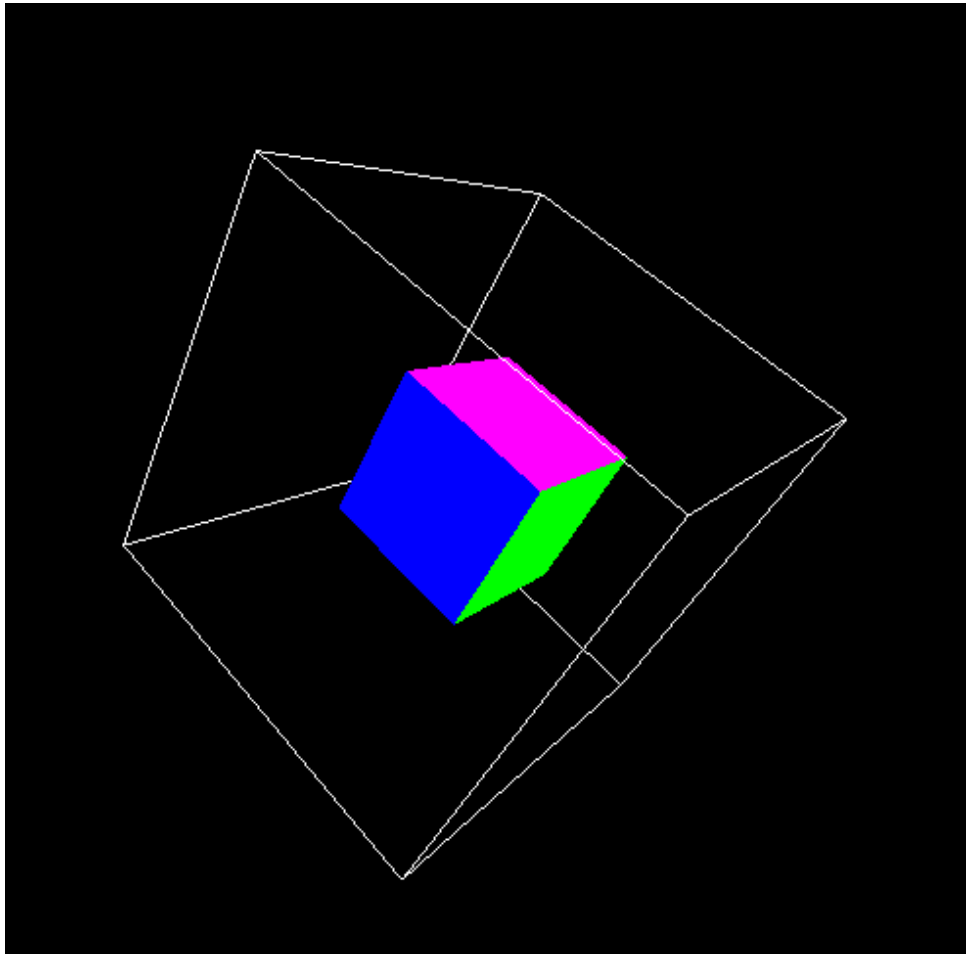
Model

- 3D Cube with differently colored sides

```
// vertex coords array
GLfloat vertices[] = {1,1,1,  -1,1,1,  -1,-1,1,  1,-1,1,        // v0-v1-v2-v3
                      1,1,1,  1,-1,1,  1,-1,-1,  1,1,-1,        // v0-v3-v4-v5
                      1,1,1,  1,1,-1,  -1,1,-1,  -1,1,1,        // v0-v5-v6-v1
                      -1,1,1,  -1,1,-1,  -1,-1,-1,  -1,-1,1,    // v1-v6-v7-v2
                      -1,-1,-1,  1,-1,-1,  1,-1,1,  -1,-1,1,    // v7-v4-v3-v2
                      1,-1,-1,  -1,-1,-1,  -1,1,-1,  1,1,-1};   // v4-v7-v6-v5

GLfloat colors[] = {1,1,1,  1,1,1,  1,1,1,  1,1,1,              // v0-v1-v2-v3
                    1,1,0,  1,1,0,  1,1,0,  1,1,0,              // v0-v3-v4-v5
                    1,0,1,  1,0,1,  1,0,1,  1,0,1,              // v0-v5-v6-v1
                    0,1,0,  0,1,0,  0,1,0,  0,1,0,              // v1-v6-v7-v2
                    0,1,1,  0,1,1,  0,1,1,  0,1,1,              // v7-v4-v3-v2
                    0,0,1,  0,0,1,  0,0,1,  0,0,1};             // v4-v7-v6-v5
```

Modeling



Declarative VS Imperative Modeling

- Declarative
  - The What from a pool of existing possibilities
  - High-level modeling environment
  - SDML: **Strictly Declarative Modelling Language**
- Imperative
  - The How (definition, implementation, control)
  - Low Level APIs

Game Engines

- reusable components that can be manipulated to bring a game/simulation to life
- A pool of defined possibilities
- High Level
- Some Examples:
  - **Irrlicht Engine** – http://irrlicht.sourceforge.net/
  - OGRE – http://www.ogre3d.org/
  - Panda3D – https://www.panda3d.org/

What is an API?

- Application Programming Interface
  - an interface that lets a program communicate with another program
  - In the case of Graphics API – it is an interface that lets programmers access the graphics hardware in an abstract/low level manner

Some History

- In 1947 the Association for Computing Machinery (ACM) was formed, from this group came SICGRAPH (Special Interest Committee in Graphics) was born.
- In 1977 the first 3D graphics framework called CORE was released by the group, which became the basis for many future developments.

Low Level APIs

- IrisGL
  - Integrated **R**aster **I**maging **S**ystem **G**raphics **L**ibrary
  - proprietary graphics API by Silicon Graphics
- OpenGL
  - the basic industry standard 3D API for direct hardware access, evolved from IrisGL
- MesaGL
  - Freeware OpenGL implementation, available on a wide range of platforms

High Level APIs

- **Open Inventor**
  - very flexible, extensible scene graph API for rapid prototyping, but not very fast; provides many interaction techniques
- **Performer**
  - monolithic scene graph API geared towards performance
- **OpenGL Optimizer** (http://www.sgi.com/Technology/OpenGL/optimizer/)
- **OpenGL++** (SGI, Intel, IBM)
  - should contain the best of Optimizer, Performer and Inventor
  - OpenGL++ has now been dropped in favor of Fahrenheit!

Direct 3D

- Part of Microsoft's DirectX API.
- Virtually all 3-D accelerator cards for PCs support Direct3D.
- Latest ver. 11 (Win7+)
  - Tessellation (tiling) is implemented on the GPU for smoother curves
  - Multi-threading support
  - Direct compute

JAVA 2D & JAVA 3D

- 2D or 3D scenes within java applets
- Scene graph based – (A **scene graph** is a general data structure commonly used by vector-based graphics editing applications and modern computer games).
- Java3D is wrapper around other APIs
- Promotes object-oriented concepts in graphics programming.

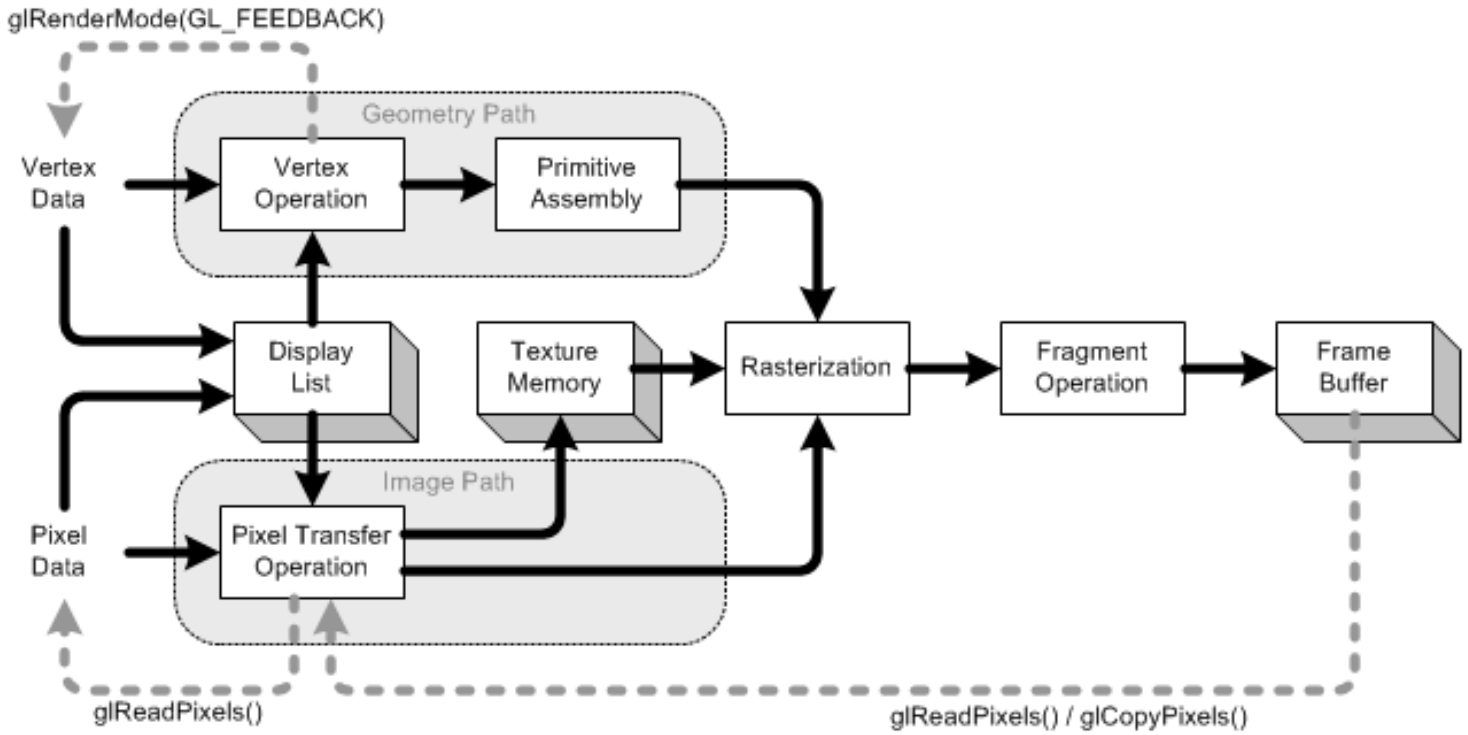
- ***Renderman Interface Specification***: Used by Pixar Corporation.
  - Published in 1988
  - Used for creating 3D movies and movie special effects.
  - Used with the Renderman Shading Language – proprietary-based Graphics Language.

- ***RenderWare***
  - Graphics rendering engine and 3D API
  - Used in Xbox, Wii, GameCube, Playstation etc.
  - Proprietary

OpenGL

- *OpenGL*: (Open Graphics Library) the industry standard for high performance graphics.
  - OpenGL ES (Embedded Systems – cell phones , PDA, video game consoles etc.),
- OpenGL is a software interface to graphics hardware.
- Introduced in 1992. Cross-platform, cross-language. Maintained by the Khronos group – a member funded consortium for open standard.
- This interface consists of about 150 distinct commands
- OpenGL is designed as a streamlined, hardware-independent interface to be implemented on many different hardware platforms
- With OpenGL, you must build up your desired model from a small set of *geometric primitives* – **points, lines, and polygons**
- The OpenGL Utility Library (GLU) provides many of the modeling features, such as quadric surfaces and NURBS (non-uniform rational B-Splines) curves and surfaces. GLU is a standard part of every OpenGL implementation.

OpenGL Pipeline



http://www.songho.ca/opengl/gl_pipeline.html

OpenGL

- Developer-Driven Advantages
  - **Stable.**
  - **Reliable and portable**
  - **Evolving**
  - **Scalable**
  - **Easy to use**
  - **Well-documented**

- **www.opengl.org**

Questions?

Review Questions

- What do you understand by the terms Declarative Modeling & Imperative Modeling?
- What is a low-level API?
- Describe a situation where it would be better to choose a low level Graphics API over a High Level API?
- What are some of the advantages of OpenGL?

Good List of Game Engines

- http://en.wikipedia.org/wiki/List_of_game_engines