OpenGL

The Rosetta Stone of 3D Graphics

Slides :: Alexander Chia

Why OpenGL?

- Why not
  - Direct3D (Microsoft)
  - QuickDraw 3D (Apple)
  - My 1000 fps raytracer (?)

Why OpenGL?

- First standardized API to “talk” with the hardware
- Cross-platform
- Performance
- Professional Graphics (vs. Computer Games [Direct3D])
- Supports most extensions (OGL 3.0)
- The standard at academic institutions
    - SIGGRAPH

Overview

- Why OpenGL?
- History
- API Structure
- Command Syntax
- Hello World
- Project #1
- References

What is OpenGL?

- Open Graphics Library
- “Software interface to Graphics hardware” (http://www.opengl.org/registry/doc/glspec30.20080811.pdf)

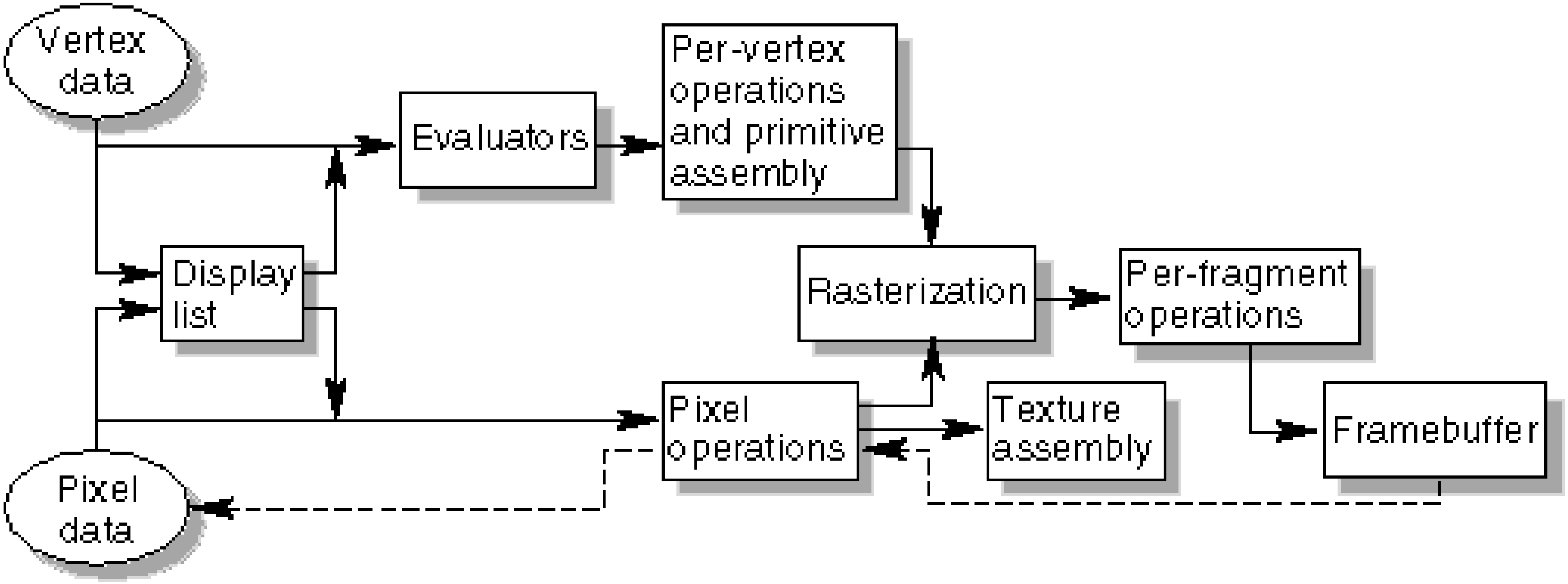
History

- Silicon Graphics Inc. (SGI) (1992)
  - Maintained by OpenGL ARB
- Khronos Group (2006)

Versions

- 1.1 (1992)
  - OpenGL State Machine
  - Fixed function pipeline
- 2.1 (2006)
  - Shaders (GLSL 1.2)
  - sRGB color space
- 3.0 (Aug 11 2008!)
  - GLSL 1.3 and more
  - Fully backward compatible
  - A good thing?
    - Mutable vs. Immutable

The OpenGL state machine



Taken from http://www.glprogramming.com/red/appendixa.html

API

- GL – OpenGL
- GLU – OpenGL Utility Library
- GLUT – OpenGL Utility Toolkit

```
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>
```

Command syntax

- Commands
  - glClear
  - gluPerspective

- Constants
  - GL_DEPTH_BUFFER_BIT
  - GLUT_RGB
- Params
  - glNormal3f
    - 3 arguments, Glfloats
    - void glNormal3f(GLfloat *nx*, GLfloat *ny*, GLfloat *nz*);

Hello World

- Creating the window
- Handling events
- Setting up the program
- Rendering the objects
- Primitives
- Transformations

Remember:
OpenGL 1.1 is a state machine

Creating the window

- glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);
    - buffers
- glutInitWindowSize(WINDOW_WIDTH, WINDOW_HEIGHT);
- glutInitWindowPosition(x, y);
- glutCreateWindow(“Hello World!");
- Play around!

Handling events

- Recall: Event–driven Programming
  - Callbacks
- glutDisplayFunc
- glutKeyboardFunc
- glutMouseFunc
- glutIdleFunc
- glutTimerFunc
- Many more..

Setting it up

- Menus
  - glutCreateMenu
  - glutAddMenuEntry
- Projection Matrix
  - View space to Projection space
  - glMatrixMode(GL_PROJECTION);
  - gluPerspective or gluOrtho
- Depth Testing
  - glEnable(GL_DEPTH_TEST)

Make it appear already!

- Clear buffers
  - glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
- Set Model Transforms
  - glScale, glRotate, glTranslate
- Draw Objects
  - glBegin
  - glColor/glNormal/glVertex
  - glEnd
- Swap front and back buffers
  - glutSwapBuffers

Primitives

- glBegin
  - GL_TRIANGLES
  - GL_TRIANGLE_STRIP
  - GL_TRIANGLE_FAN
- Data flowing btw. CPU and Hardware
- Indexed Primitives

Transformations

- glMatrixMode
  - GL_PROJECTION – View space to Projection space
  - GL_MODELVIEW – World space to View space
  - GL_TEXTURE – Pixel transforms
- Transform = PROJECTION * MODELVIEW * VIEWPORT
- OGL Camera at origin
  - Pointing at -z direction

Things to note

- Culling
  - CCW faces are the ‘front’ by default
  - glFrontFace
- glPushMatrix / glPopMatrix
- glLoadIdentity

Features and Optimizations

- Display lists
  - glNewList(…, GL_COMPILE)
  - glEndList
  - glCallList
- Transparency
  - RGBA (RGB + Alpha)
  - 8 bits/ channel
- Textures
  - glTexCoord
- Lights
- Animation / Interaction

For the advanced

- Text
  - glRasterPos
  - glutBitmapCharacter
- Fog
  - glFog
- Skybox
- Advanced User Interactions
- Dynamic Model Customization
- Shadows / Reflections
- Shaders?!

Project #1

- Start NOW
- Love OpenGL, not Hate it
- Embrace change with open arms..

Summary

- History of OpenGL
- OpenGL state machine
- GL, GLU, GLUT
- Hello World
- Extra Features
- Project #1

References

- Red book
  - http://www.glprogramming.com/red/ (OpenGL 1.1)
- OpenGL 2.1
  - http://www.opengl.org/sdk/docs/man/
- Questions?
  - cmu.cs.class.cs462