

Lab 01 - Open GL (Chapter 3)

I. Software Packages - Special Purpose (e.g. plotting programs) and General Purpose

- A. Direct X - Microsoft Proprietary. Largest use in industry. C++
- B. Open GL - Cross platform open source. Smaller use in industry. C
 - i. Chosen because it is easy to use
 - ii. Chosen because it is cross platform
 - iii. Will use it in C++ - Note: C++ is backwards compatible.

II. Graphics Functions Overview

- A. Graphics Primitives - Basic Building Blocks, e.g. points, lines, triangles (even spheres, cones, toroids)
- B. Attributes - Properties of primitives, e.g. color or line style
- C. Transformations - Scale, Translation, Rotation of objects
- D. Viewing - Camera + Projections
- E. ~~Input~~ Input Functions - Call backs. Event-driven programming!
- F. Control Functions - miscellaneous, e.g. clear screen.
- G. Note: resemblance to viewing pipeline + explain event driven programming.

III. Open GL introduction

- A. C library for only rendering. Will use GLU (OpenGL utility) for ease of viewing projection and Open GL Utility Toolkit (GLUT) for windowing/GUI/Interaction
 - i. Note can use any windowing if you would rather, but must be in C/C++.
 - ii. Use C++ 11, 14, 17.
- B. Syntax
 - i. All functions begin with gl - why? No namespaces in C e.g. glBegin()
 - ii. All constants/enums are GL_ then uppercase separated by _ e.g. GL_LINES
 - iii. Data type aliases are GLtype e.g. GLint or GLfloat Note: just typedefs of regular types.
 - iv. Function names commonly end with a number and/or type initials
 - why? No overloading in C. e.g. glVertex2i or glVertex3f

IV. Download prog05.zip + compile for students

A. Walk through code to explain:

- i. glutInit
- ii. glutCreateWindow
- iii. glutDisplayFunc
- iv. glutMainLoop
- v. glutWindowPosition()
- vi. glutInitWindowSize
- vii. glutInitDisplayMode()
- viii. glClearColor
- ix. glClear
- x. glColor3f
- xi. Projection/Camera/Lighting
- xii. glBegin()/glEnd()

B. Locking frame rate (rudimentary way). Locked at refresh rate of monitor e.g. 60fps/30fps

- i. Each frame record time
- ii. take difference from last frame (time to render)
- iii. $1/\text{fps} - \text{render-time} = \text{excess time}$. sleep for excess time

C. Swap-Draw-Show animation: glutSwapBuffers

D. Input with glut call backs: key presses, refer to Chapter 20 for more.

E. Can do Error Handling in GL. Important as graphics code is difficult to debug! see Ch. 3-5

V. Explain prog 01 assignment.