OpenGL - Basic Draw

Document

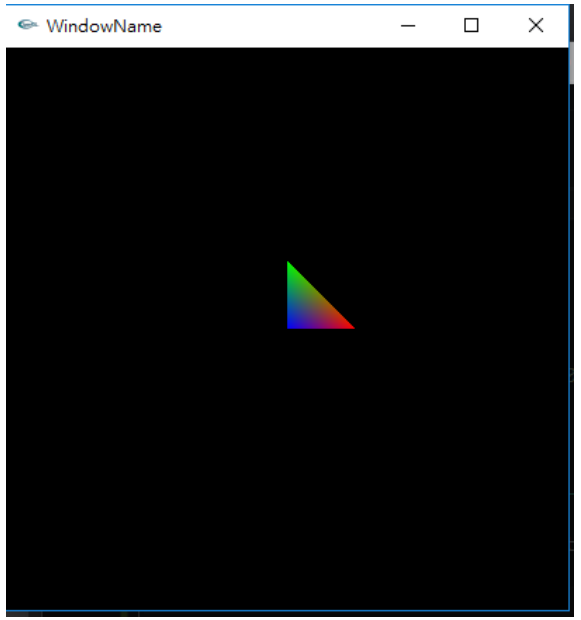
- https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/

Example

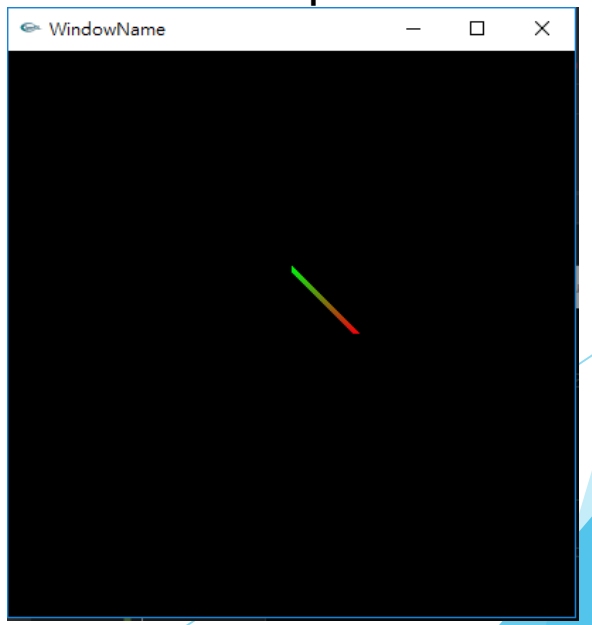
```
19      glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
```

```
37  void display()
38  {
39      //ModelView Matrix
40      glMatrixMode(GL_MODELVIEW);
41      glLoadIdentity();
42      gluLookAt(0.0f, 0.0f, 10.0f, 0.0f, 0.0f, 0.0f, 0.0f, 1.0f, 0.0f);
43      //Projection Matrix
44      glMatrixMode(GL_PROJECTION);
45      glLoadIdentity();
46      gluPerspective(45, width / (GLfloat)height, 0.1, 1000);
47      //Viewport Matrix
48      glViewport(0, 0, width, height);
49
50      //
51      glMatrixMode(GL_MODELVIEW);
52      glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
53      glClear(GL_COLOR_BUFFER_BIT);
54      glClearDepth(1.0);
55      glEnable(GL_DEPTH_TEST);
56      glDepthFunc(GL_LEQUAL);
57      glClear(GL_DEPTH_BUFFER_BIT);
58
59      glBegin(GL_TRIANGLES);
60
61      glColor3f(1.0f, 0.0f, 0.0f);
62      glVertex3f(1.0f, 0.0f, 0.0f);
63      glColor3f(0.0f, 1.0f, 0.0f);
64      glVertex3f(0.0f, 1.0f, 0.0f);
65      glColor3f(0.0f, 0.0f, 1.0f);
66      glVertex3f(0.0f, 0.0f, 0.0f);
67
68      glColor3f(0.0f, 0.0f, 0.0f);
69      glVertex3f(1.0f, 0.0f, -1.0f);
70      glColor3f(0.0f, 0.0f, 0.0f);
71      glVertex3f(0.0f, 1.0f, -1.0f);
72      glColor3f(0.0f, 0.0f, 0.0f);
73      glVertex3f(0.0f, 0.0f, -1.0f);
74
75      glEnd();
76
77      glutSwapBuffers();
```

Enable depth test

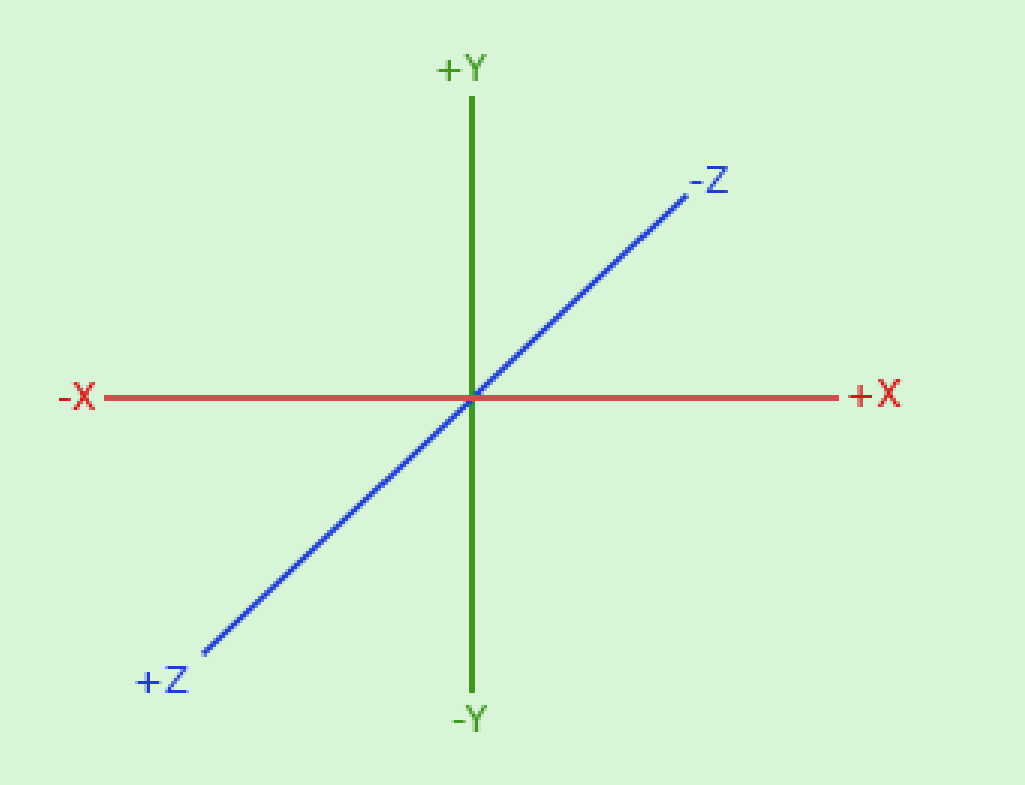


Disable depth test



OpenGL coordinate system

- Right-handed system



OpenGL datatype

Suffix	Data Type	Typical Corresponding C-Language Type	OpenGL Type Definition
b	8-bit integer	signed char	GLbyte
s	16-bit integer	short	GLshort
i	32-bit integer	int or long	GLint, GLsizei
f	32-bit floating-point	float	GLfloat, GLclampf
d	64-bit floating-point	double	GLdouble, GLclampd
ub	8-bit unsigned integer	unsigned char	GLubyte, GLboolean
us	16-bit unsigned integer	unsigned short	GLushort
ui	32-bit unsigned integer	unsigned int or unsigned long	GLuint, GLenum, GLbitfield

Color representation

- RGBA: red, green, blue, alpha
  - Each channel has intensity from 0.0~1.0
  - Values outside this interval will be clamp to 0.0 or 1.0
  - Alpha is used in blending and transparency
- Specify a color:
  - void glColor{34}{sifd}[v](…);
    - EX: glColor3f(1.0f, 0.0f, 0.0f);
    - EX: glColor4f(1.0f, 0.0f, 0.0f, 1.0f);
    - EX: glColor3fv(float*);

Color buffer

- void glClearColor(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha);
  - Set the current values for use in cleaning color buffers in RGBA mode
  - red, green, blue, alpha: Specify the red, green, blue, and alpha values used when the color buffers are cleared
- void glClear(GLbitfield mask);
  - Clear the specified buffers to their current clearing values
  - mask: GL_COLOR_BUFFER_BIT, GL_DEPTH_BUFFER_BIT, GL_ACCUM_BUFFER_BIT, GL_STENCIL_BUFFER_BIT
  - EX: glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

Depth buffer & depth test

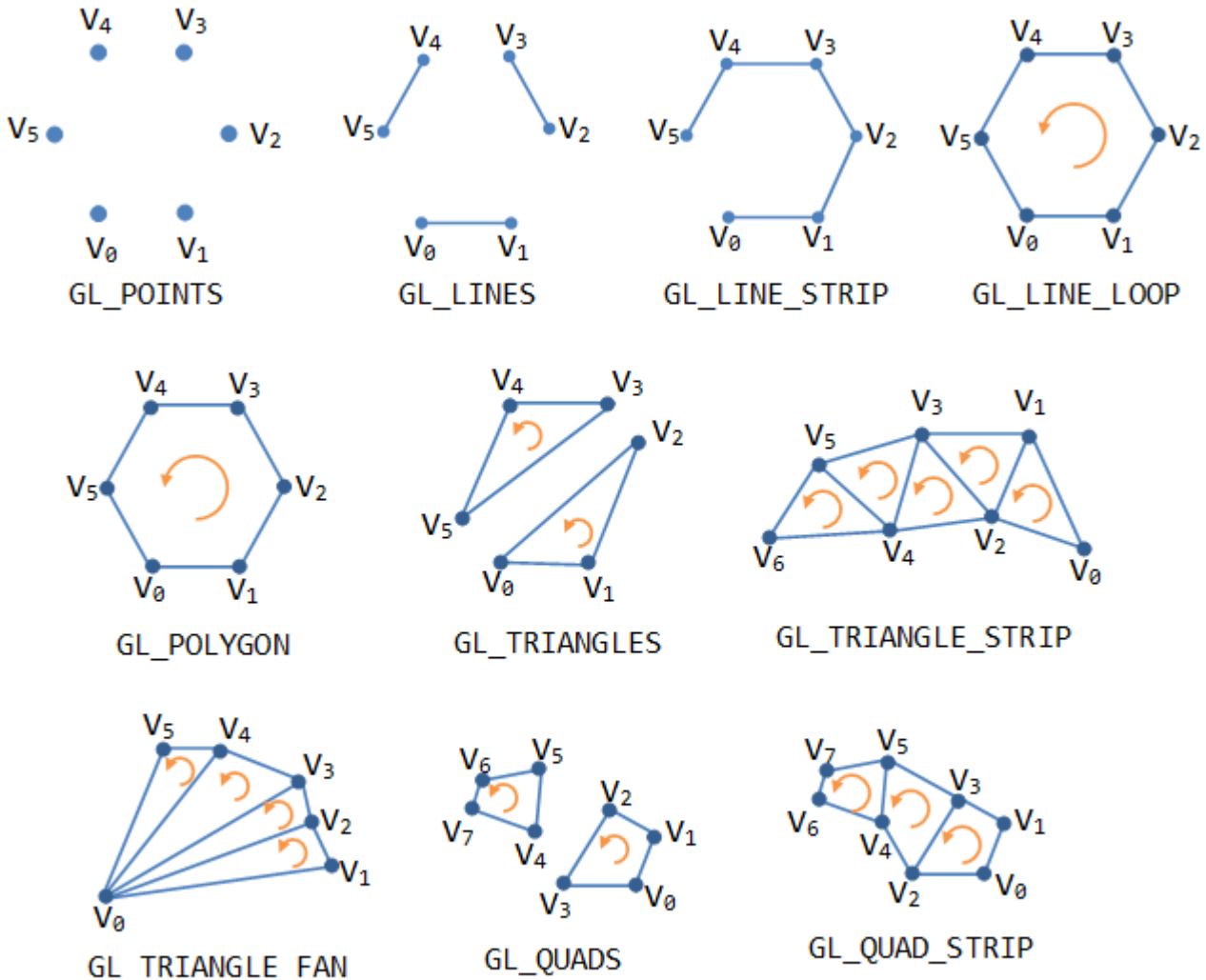
- void glClearDepth(GLdouble depth);
  - Set the current values for use in cleaning depth buffer
  - depth: Specifies the depth value used when the depth buffer is cleared
- void glEnable(GLenum cap);
  - enable or disable GL capabilities
  - Cap: GL_DEPTH_TEST, GL_STENCIL_TEST…
- void glDepthFunc(GLenum func);
  - Func: GL_NEVER, GL_LESS, GL_EQUAL, GL_LEQUAL, GL_GREATER, GL_NOTEQUAL, GL_GEQUAL, and GL_ALWAYS.
  - The initial value is GL_LESS.
- void glClear(GLbitfield mask);

Point, line, and polygon

- Describe points, lines, polygons
- void glBegin(GLenum mode);
  - Marks the beginning of a vertex-data list
  - Vertex-data include vertex's color, normal, position, etc.
  - mode:
- void glEnd();
  - Marks the end of a vertex-data list

Value	Meaning
GL_POINTS	individual points
GL_LINES	pairs of vertices interpreted as individual line segments
GL_LINE_STRIP	serious of connected line segments
GL_LINE_LOOP	same as above, with a segment added between last and first vertices
GL_TRIANGLES	triples of vertices interpreted as triangles
GL_TRIANGLE_STRIP	linked strip of triangles
GL_TRIANGLE_FAN	linked fan of triangles
GL_QUADS	quadruples of vertices interpreted as four-sided polygons
GL_QUAD_STRIP	linked strip of quadrilaterals
GL_POLYGON	boundary of a simple, convex polygon

Point, line, and polygon



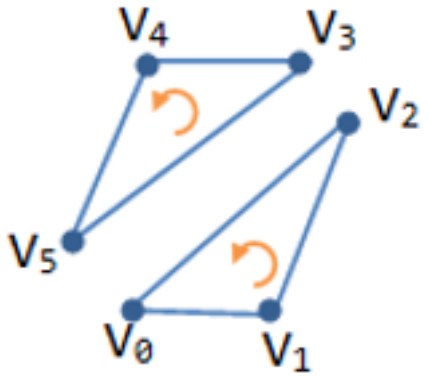
OpenGL Primitives

Point, line, and polygon

- void glColor{34}{sifd}[v](…);
- void glNormal3{bsifd}[v](…)
  - set the current normal vector
- void glVertex{234}{sifd}[v](…)
  - Specifies a vertex for use in describing a geometric object
  - Can only effective between a glBegin() and glEnd() pair
- You have to set vertex's attributes before glVertex
  - Ex: Use glColor and glNormal before glVertex to set the vertex's color

Face culling

- glEnable(GL_CULL_FACE);
- void glCullFace(GLenum mode);
  - specify whether front- or back-facing facets can be culled
  - Mode: GL_FRONT, GL_BACK, and GL_FRONT_AND_BACK. The initial value is GL_BACK.
- void glFrontFace(GLenum mode);
  - define front- and back-facing polygons
  - Mode: GL_CW, GL_CCW. The initial value is GL_CCW.



GL_TRIANGLES

Completion of Drawing

- void glFlush();
  - force execution of GL commands in finite time.
- void glFinish();
  - block until all GL execution is complete.
- void glutSwapBuffers();
  - Swap front and back buffers.