基本图形的绘制

basic structure of OpenGL programs

- ***`main()`**:*
    - 定义回调函数
    - 创建窗口
    - enters event loop
- ***`init()`**: sets the state variables*
    - Viewing
    - Attributes
- *callbacks*
    - Display function
    - Input and window functions

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

int main(int argc, char** argv)
{
  glutInit(&argc,argv);
  glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
  glutInitWindowSize(500,500);
  glutInitWindowPosition(0,0);
  glutCreateWindow("simple");
  glutDisplayFunc(mydisplay);

  init();

  glutMainLoop();
}
```

includes `gl.h`

define window properties

display callback

set OpenGL state

enter event loop

glut

- **`glutInit`** *allows application to get command line arguments and initializes system*
- **`gluInitDisplayMode`** *requests properties for the window (the rendering context)*
  - RGB color
  - Single buffering
  - Properties logically ORed together
- **`glutWindowSize`** *in pixels*
- **`glutWindowPosition`** *from top-left corner of display*
- **`glutCreateWindow`** *create window with title "simple"*
- **`glutDisplayFunc`** *display callback*
- **`glutMainLoop`** *enter infinite event loop*

init.c

black clear color

opaque window

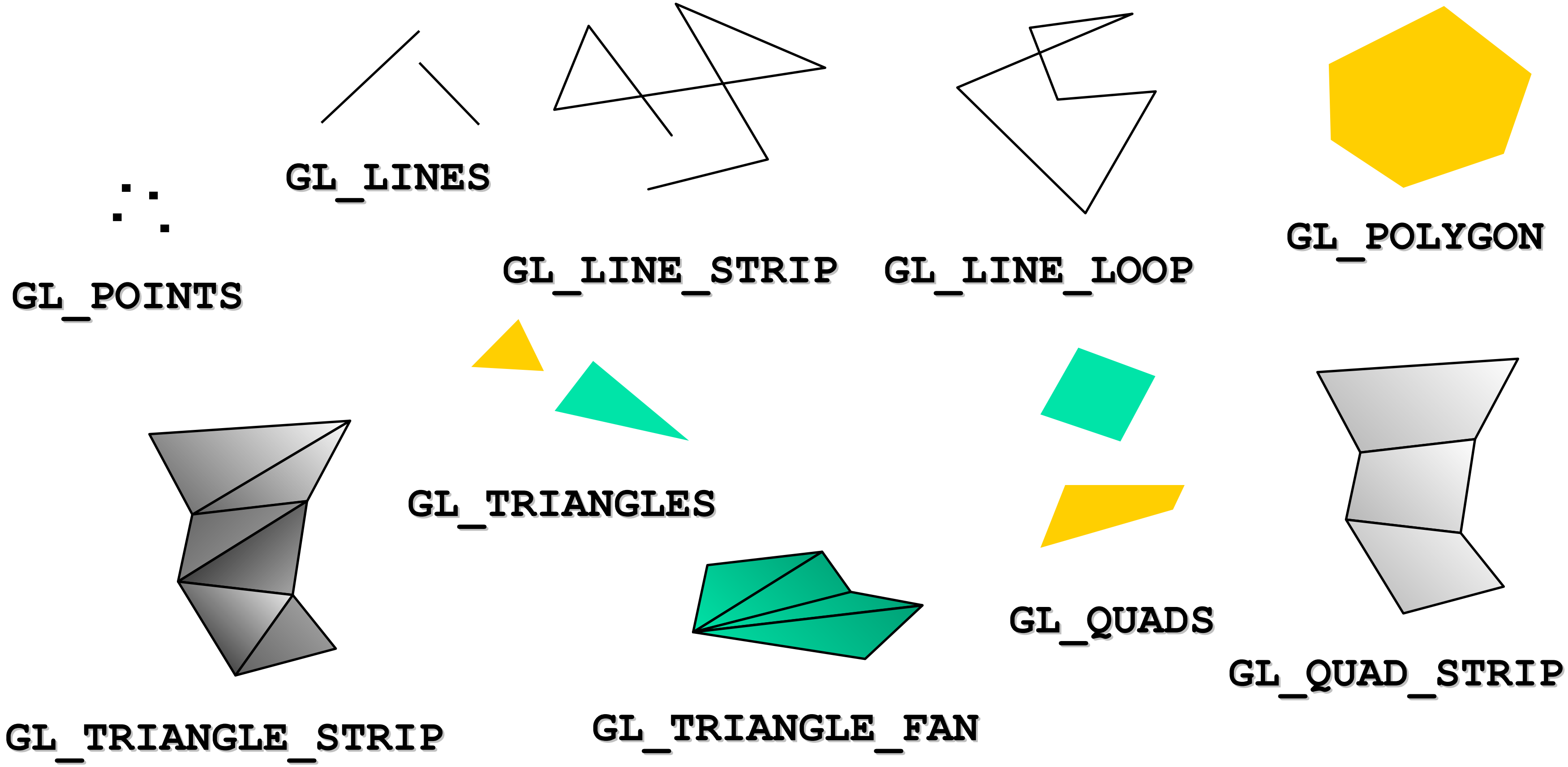
```
void init()
{
  glClearColor (0.0, 0.0, 0.0, 1.0);

  glColor3f(1.0, 1.0, 1.0);

  glMatrixMode (GL_PROJECTION);
  glLoadIdentity ();
  glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
}
```

fill/draw with white

viewing volume

基本图形

- *All geometric primitives are specified by vertices*



定义基本图形

- *Primitives are specified using*

```
        glBegin( primType );
        glEnd();
```

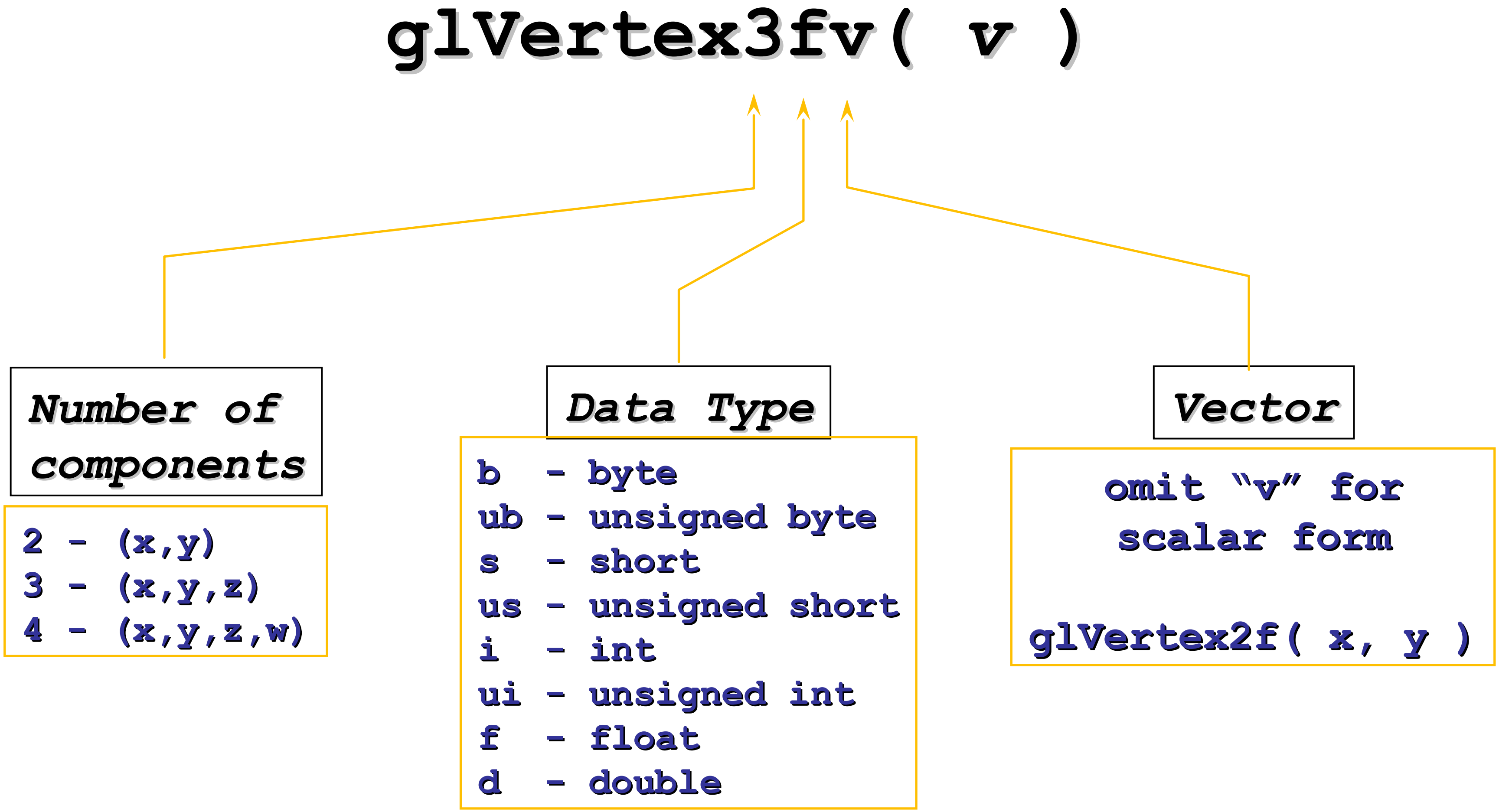
  - *primType* determines how vertices are combined

Simple Example

```
void drawRhombus( GLfloat color[] )
{
 glBegin( GL_QUADS );
          glColor3fv( color );
          glVertex2f( 0.0, 0.0 );
          glVertex2f( 1.0, 0.0 );
          glVertex2f( 1.0, 1.0);
          glVertex2f( 0.5, 1.0 );
 glEnd();
}
```

See Demo.

OpenGL Command Formats



```
glVertex3fv( v )
```

Number of components

2 -	(x,y)
3 -	(x,y,z)
4 -	(x,y,z,w)

Data Type

b	- byte
ub	- unsigned byte
s	- short
us	- unsigned short
i	- int
ui	- unsigned int
f	- float
d	- double

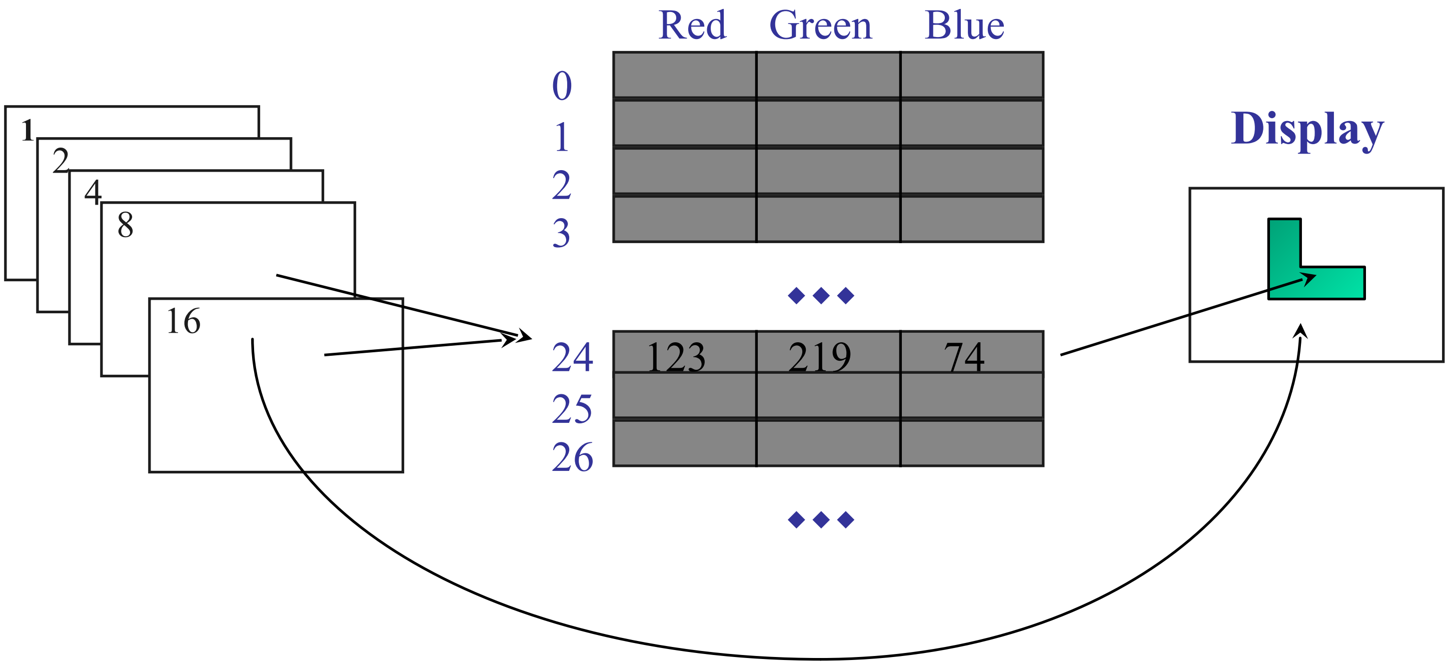
Vector

omit "v" for scalar form

```
glVertex2f( x, y )
```

OpenGL Color Models

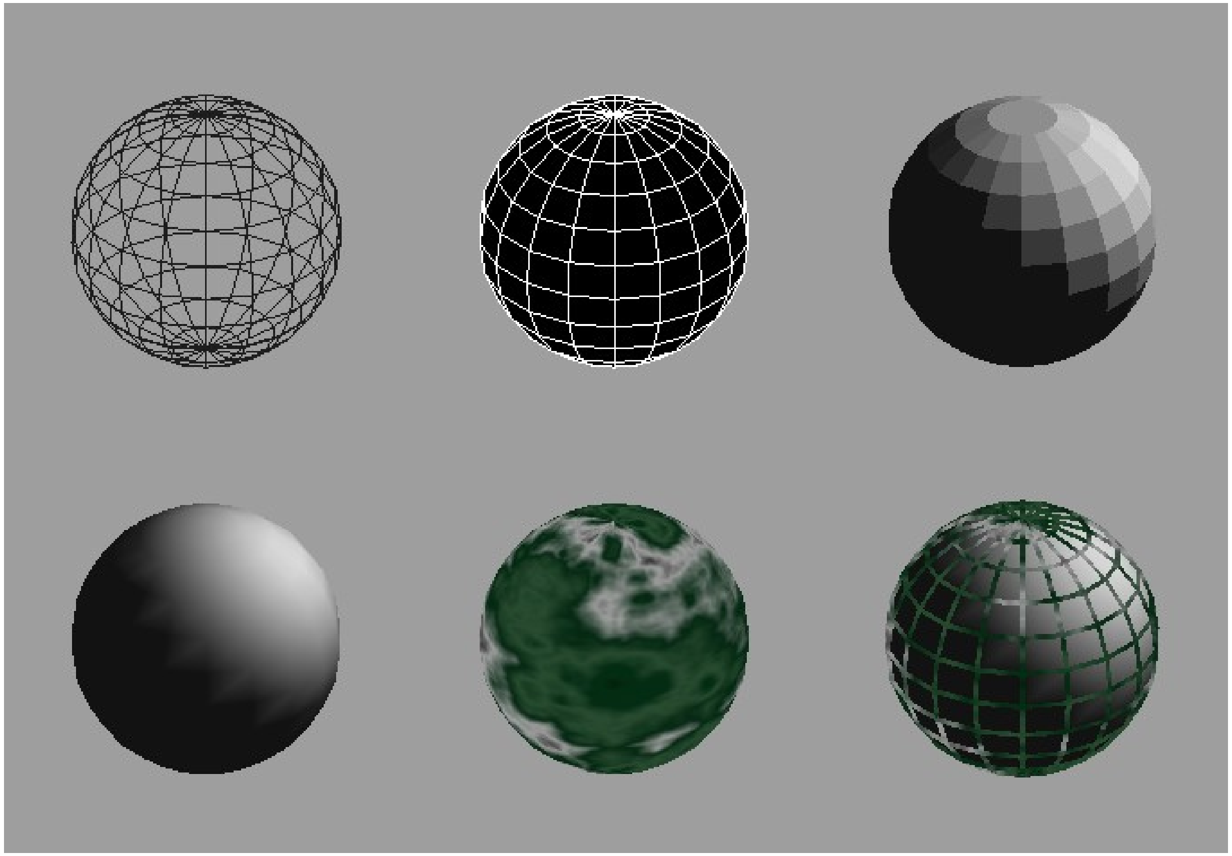
- *RGBA or Color Index*



	Red	Green	Blue
0			
1			
2			
3			
	◆◆◆		
24	123	219	74
25			
26			
	◆◆◆		

Controlling Rendering Appearance

- *From Wireframe to Texture Mapped*



OpenGL's State Machine

- *All rendering attributes are encapsulated in the OpenGL State*
  - rendering styles
  - shading
  - lighting
  - texture mapping

Manipulating OpenGL State

- *Appearance is controlled by current state*

```
for each ( primitive to render ) {
    update OpenGL state
    render primitive
}
```

- *Manipulating vertex attributes is most common way to manipulate state*

```
glColor*() / glIndex*()
glNormal*()
glTexCoord*()
```

Controlling current state

- *Setting State*

```
glPointSize( size );
glLineStipple( repeat, pattern );
glShadeModel( GL_SMOOTH );
```

- *Enabling Features*

```
glEnable( GL_LIGHTING );
glDisable( GL_TEXTURE_2D );
```

实验内容

- *绘制基本图形：线段，多边形等。*
- *修改图形的颜色。*
- *参考资料：*
  - 在线 OpenGL 手册 OpenGL Online Manual
  - https://www2.cs.sfu.ca/~haoz/teaching/openglman.html
  - https://www2.cs.sfu.ca/~haoz/teaching/htmlman/begin.html
  - https://www2.cs.sfu.ca/~haoz/teaching/openglman.html