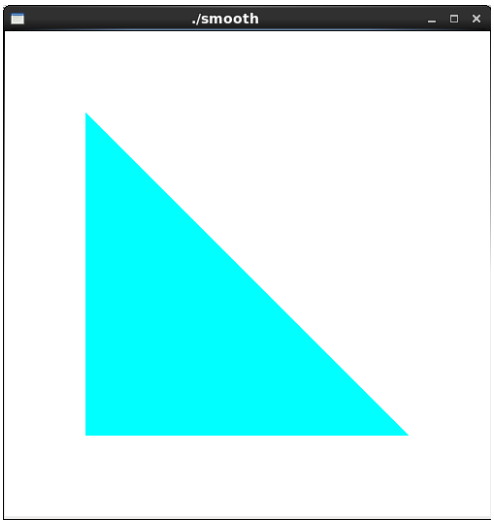
Yazhuo Liu
Lab 15

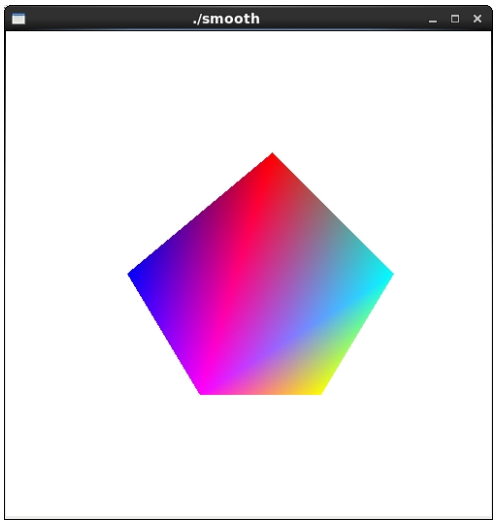
Copy the program smooth.cpp from the lecture notes. Compile and execute it.

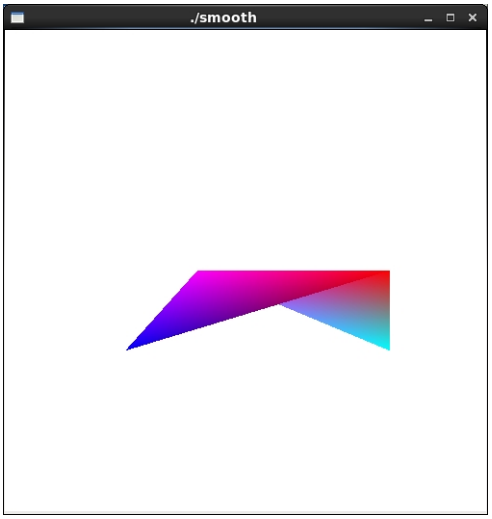
Modify the program with different vertex colors to see the color effects.

What happens if you use GL_FLAT in glShadeModel()? Try different colors with GL_FLAT mode.



Repeat the experiment with rectangles, pentagons and an arbitrary polygon.






```
//smooth.cpp
#include <GL/glut.h>
#include <stdlib.h>

void init(void)
{
   glClearColor (1.0, 1.0, 1.0, 0.0);
   glShadeModel (GL_SMOOTH);
   //glShadeModel (GL_FLAT);
}

void triangle(void)
{
/*   glBegin (GL_TRIANGLES);
   glColor3f (1.0, 0.0, 1.0);
   glVertex2f (5.0, 5.0);
   glColor3f (1.0, 1.0, 0.0);
   glVertex2f (25.0, 5.0);
```

```
   glColor3f (0.0, 1.0, 1.0);
   glVertex2f (5.0, 25.0);
   glEnd();

   glBegin (GL_POLYGON);
   glColor3f (1.0, 0.0, 1.0);
   glVertex2f (5.0, 8.0);
   glColor3f (1.0, 1.0, 0.0);
   glVertex2f (25.0, 8.0);
   glColor3f (0.0, 1.0, 1.0);
   glVertex2f (25.0, 22.0);
   glColor3f (1.0, 0.0, 0.0);
   glVertex2f (5.0, 22.0);
   glEnd();

   glBegin (GL_POLYGON);
   glColor3f (1.0, 0.0, 1.0);
   glVertex2f (8.0, 5.0);
   glColor3f (1.0, 1.0, 0.0);
   glVertex2f (13.0, 5.0);
   glColor3f (0.0, 1.0, 1.0);
   glVertex2f (16.0, 10.0);
   glColor3f (1.0, 0.0, 0.0);
   glVertex2f (11.0, 15.0);
   glColor3f (0.0, 0.0, 1.0);
   glVertex2f (5.0, 10.0);
   glEnd();
*/
   glBegin (GL_POLYGON);
   glColor3f (1.0, 0.0, 1.0);
   glVertex2f (8.0, 15.0);
   glColor3f (1.0, 1.0, 0.0);
   glVertex2f (13.0, 15.0);
   glColor3f (0.0, 1.0, 1.0);
   glVertex2f (16.0, 10.0);
   glColor3f (1.0, 0.0, 0.0);
   glVertex2f (16.0, 15.0);
   glColor3f (0.0, 0.0, 1.0);
   glVertex2f (5.0, 10.0);
   glEnd();
}

void display(void)
{
   glClear (GL_COLOR_BUFFER_BIT);
   glScalef (1.5, 1.0, 1.0);
   triangle ();
   glFlush ();
}
```

```
void reshape (int w, int h)
{
   glViewport (0, 0, (GLsizei) w, (GLsizei) h);
   glMatrixMode (GL_PROJECTION);
   glLoadIdentity ();
   if (w <= h)
      gluOrtho2D (0.0, 30.0, 0.0, 30.0 * (GLfloat) h/(GLfloat) w);
   else
      gluOrtho2D (0.0, 30.0 * (GLfloat) w/(GLfloat) h, 0.0, 30.0);
   glMatrixMode(GL_MODELVIEW);
}
int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
   glutInitWindowSize (500, 500);
   glutInitWindowPosition (100, 100);
   glutCreateWindow (argv[0]);
   init ();
   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
   glutMainLoop();
   return 0;
}
```

Report:

I successfully completed all parts of this lab.