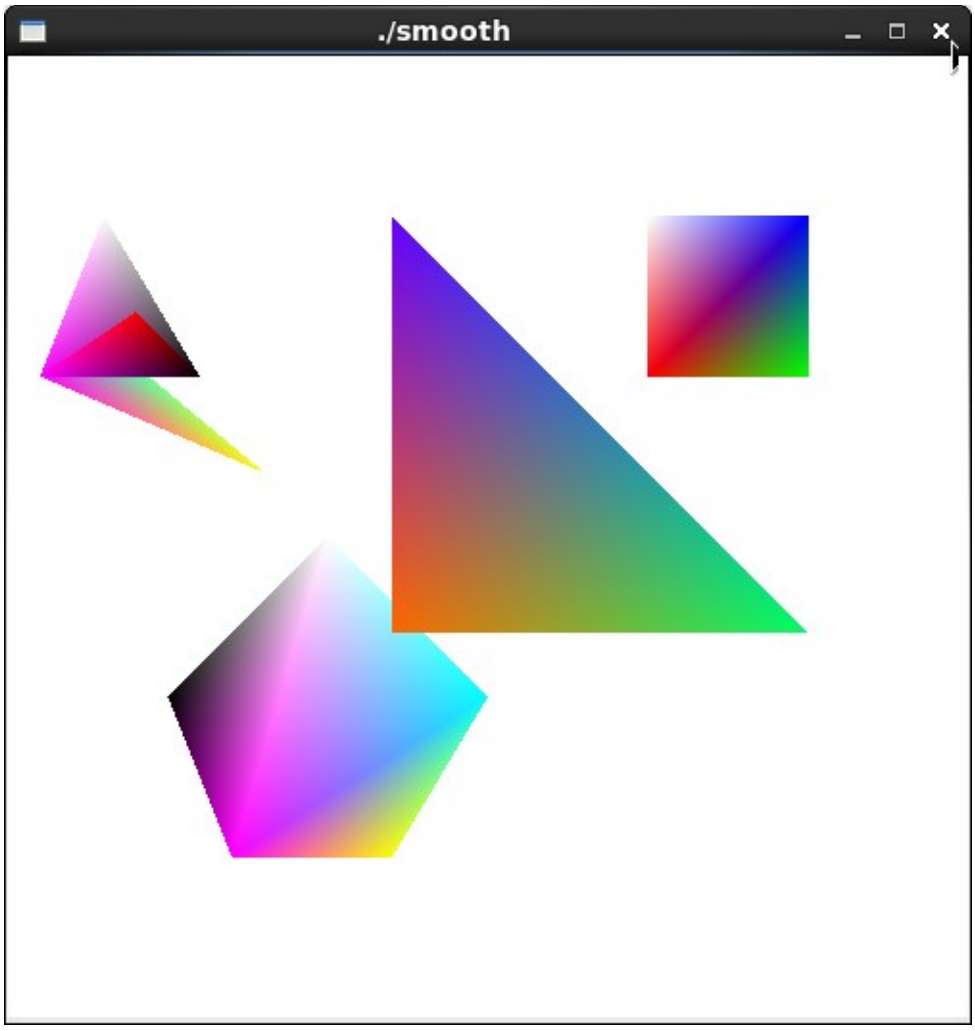
CSE420
Samuel Marrujo
Professor Yu
Lab 15

Coloring

In this part of the lab, it was a modification of the original smooth.cpp program. After some modifications to the program, and adding in the functions asked for, the result was the following reproduction of the figure. The change of the glShadeModel() function ended up producing a flat color across the triangle. I added a square to the program to show a different view of coloring. Because of this change to the functions and the addition of the figure, I believe I was able to accomplish this task successfully, since there were no errors and the display window was changed. Here are my results for this program:



```
/*
 * smooth.cpp
 * This program demonstrates smooth shading.
 * A smooth shaded polygon is drawn in a 2-D projection.
 */
#include <GL/glut.h>
#include <stdlib.h>

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

void triangle(void)
{
   glBegin (GL_TRIANGLES);
   glColor3f (1.0, 0.4, 0.0);
   glVertex2f (12.0, 12.0);
   glColor3f (0.0, 1.0, 0.4);
   glVertex2f (25.0, 12.0);
   glColor3f (0.4, 0.0, 1.0);
   glVertex2f (12.0, 25.0);
   glEnd();
}

void square(void) {
    glBegin(GL_POLYGON);
        glColor3f(1, 0, 0);
        glVertex2f(20, 20);
        glColor3f(0, 1, 0);
        glVertex2f(25, 20);
        glColor3f(0, 0, 1);
        glVertex2f(25, 25);
        glColor3f(1, 1, 1);
        glVertex2f(20, 25);
    glEnd();
}

void pentagon(void) {
    glBegin(GL_POLYGON);
        glColor3f(1, 0, 1);
        glVertex2f(7, 5);
        glColor3f(1, 1, 0);
        glVertex2f(12, 5);
        glColor3f(0, 1, 1);
        glVertex2f(15, 10);
        glColor3f(1, 1, 1);
        glVertex2f(10, 15);
```

```
        glColor3f(0, 0, 0);
        glVertex2f(5, 10);
    glEnd();
}

void arbitrary(void) {
    glBegin(GL_POLYGON);
        glColor3f(1, 0, 1);
        glVertex2f(1, 20);
        glColor3f(1, 1, 0);
        glVertex2f(8, 17);
        glColor3f(0, 1, 1);
        glVertex2f(2, 22);
        glColor3f(1, 1, 1);
        glVertex2f(3, 25);
        glColor3f(0, 0, 0);
        glVertex2f(6, 20);
        glColor3f(1, 0, 0);
        glVertex2f(4, 22);
    glEnd();
}

void display(void)
{
   glClear (GL_COLOR_BUFFER_BIT);
   pentagon();
   square();
   triangle();
   arbitrary();
   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;
}
```