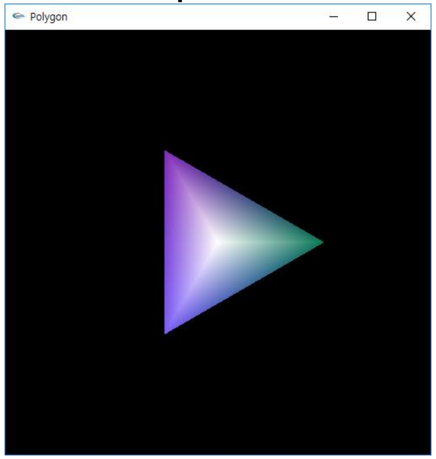
LAB I
Week 02

Seoul National University
Graphics & Media Lab
HyeonSeung Shin

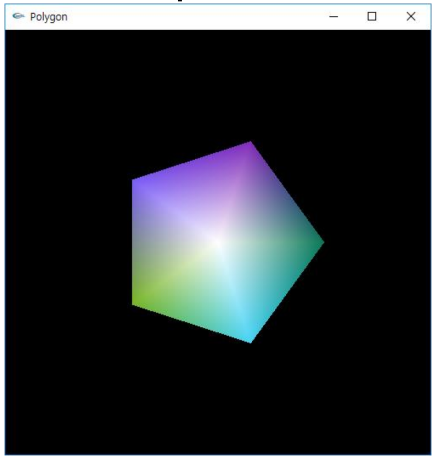
Today's assignment

- Draw square polygon using GL_POLYGON

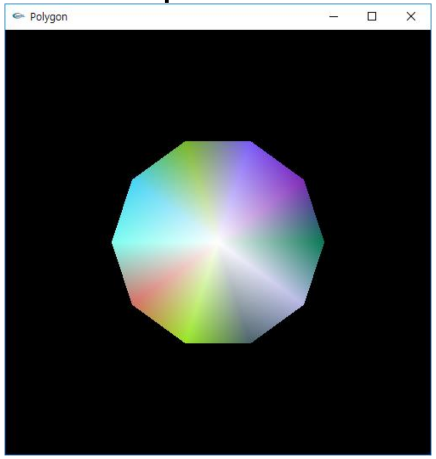
Input: 3



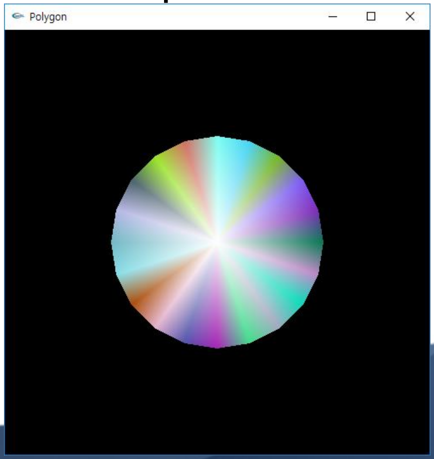
Input: 5



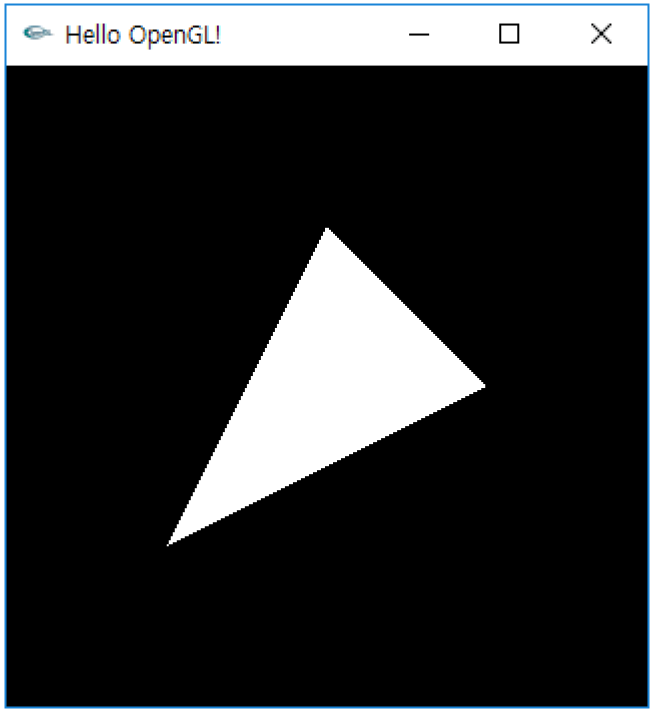
Input: 10



Input: 20



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

void drawTriangle() {

        glBegin(GL_TRIANGLES);
                glVertex3f(-0.5, -0.5, 0.0);
                glVertex3f(0.5, 0.0, 0.0);
                glVertex3f(0.0, 0.5, 0.0);
        glEnd();
}

void renderScene(void) {

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

        drawTriangle();

        glutSwapBuffers();
}

void main(int argc, char **argv) {

        // init GLUT and create Window
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
        glutInitWindowPosition(100, 100);
        glutInitWindowSize(320, 320);
        glutCreateWindow("Hello OpenGL!");

        // register callbacks
        glutDisplayFunc(renderScene);

        // enter GLUT event processing cycle
        glutMainLoop();
}
```



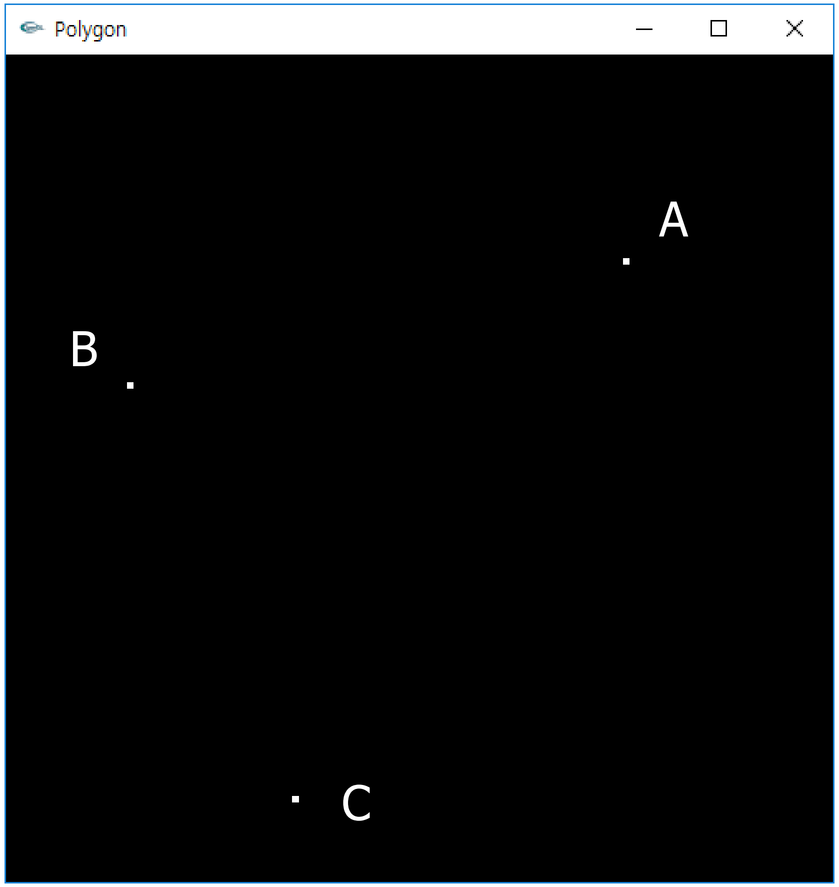
Displaying primitives

- glBegin(GLenum mode)
  - GL_POINTS
  - GL_LINES
  - GL_LINE_STRIP
  - GL_LINE_LOOP
  - GL_TRIANGLES
  - GL_TRIANGLE_STRIP
  - GL_TRIANGLE_FAN
  - GL_QUADS
  - GL_POLYGON
- glEnd()

Point

- glPointSize(GLfloat size)

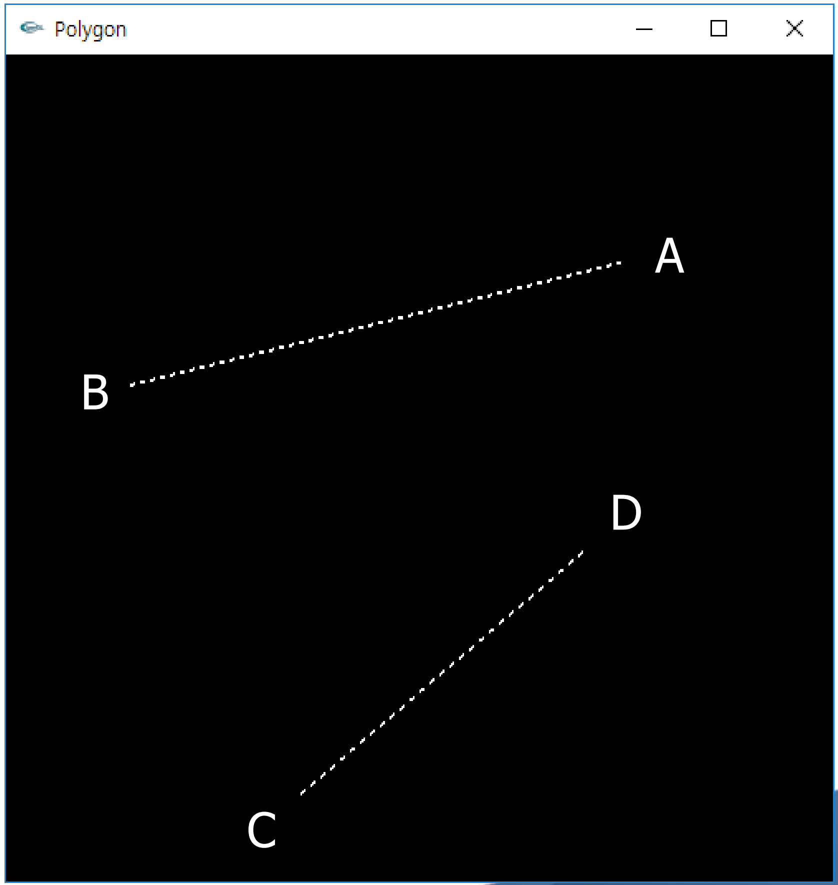
```
void drawPoint() {
        glColor3f(1, 1, 1);
        glPointSize(4.0f);
        glBegin(GL_POINTS);
                glVertex2f(0.5, 0.5);           // A
                glVertex2f(-0.7, 0.2);          // B
                glVertex2f(-0.3, -0.8);         // C
        glEnd();
}
```



Line

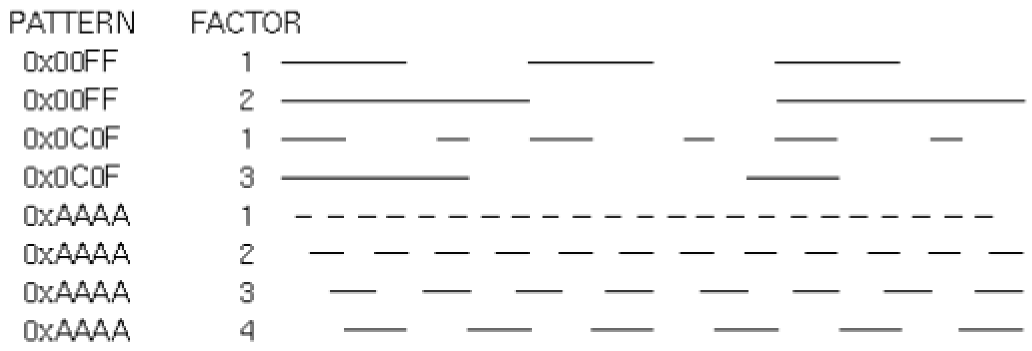
- glLineStipple(GLint factor, GLushort pattern)
- glLineWidth(GLfloat width)

```
void drawLine() {
      glColor3f(1, 1, 1);
      glLineWidth(2.5f);
      glEnable(GL_LINE_STIPPLE);
      glLineStipple(3, 0xAAAA);
      glBegin(GL_LINES);
            glVertex2f(0.5, 0.5);          // A
            glVertex2f(-0.7, 0.2);         // B
            glVertex2f(-0.3, -0.8);        // C
            glVertex2f(0.4, -0.2);         // D
      glEnd();
}
```



Line

- glLineStipple(GLint factor, GLushort pattern)
  - factor: 1 ~256 (default = 1)
  - pattern



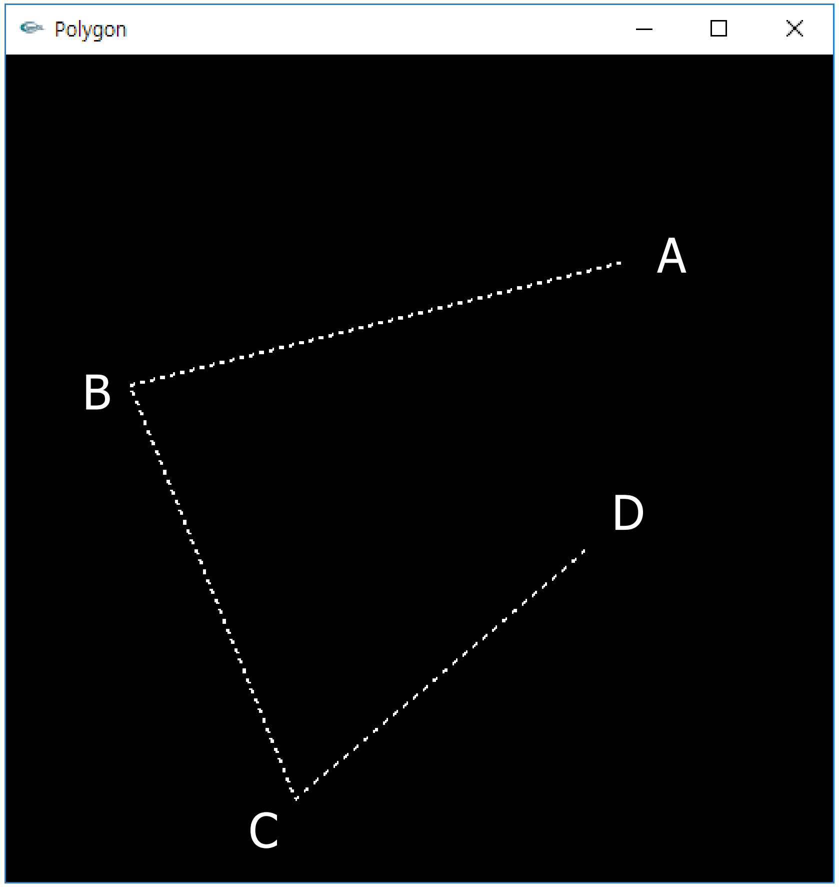
00FF

0000000011111111

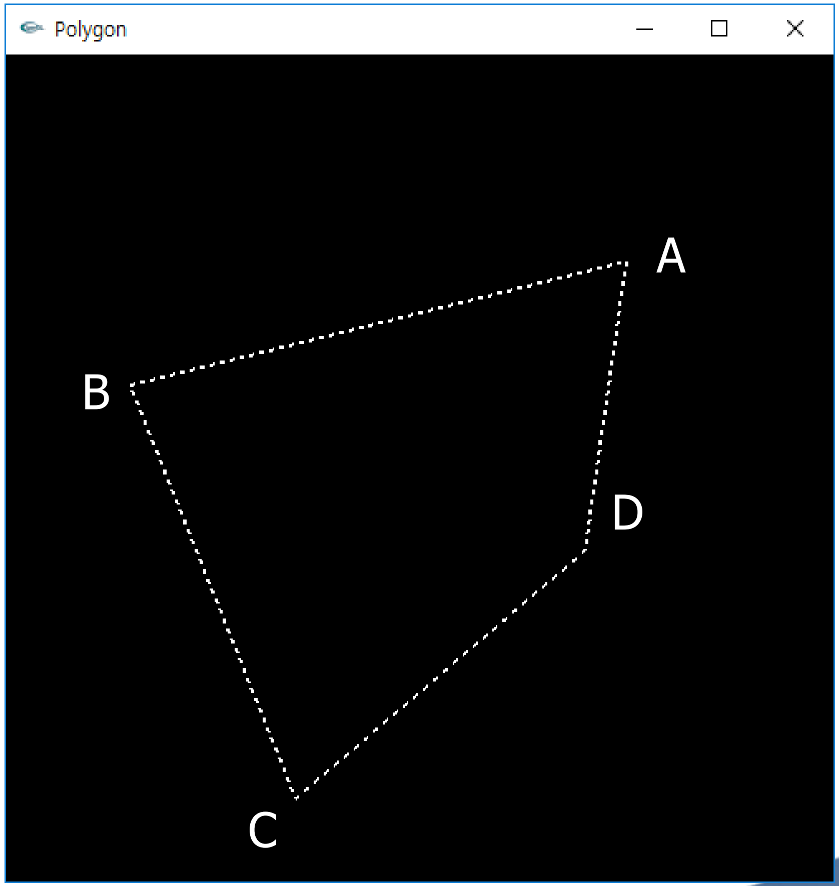
- glEnable(GL_LINE_STIPPLE)

Line

GL_LINE_STRIP

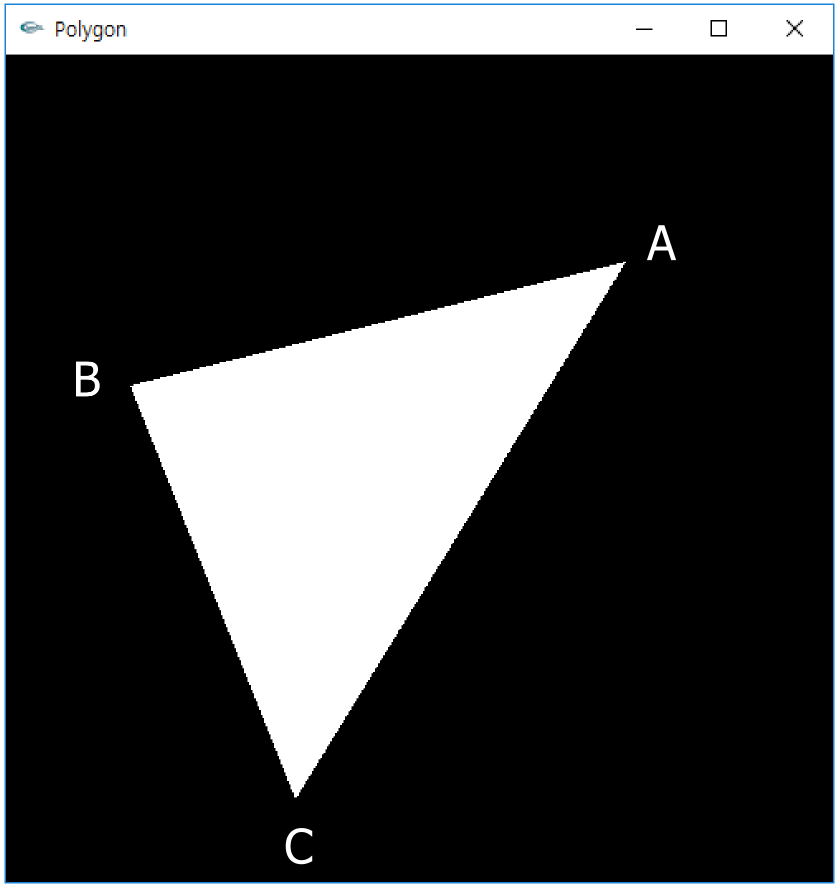


GL_LINE_LOOP



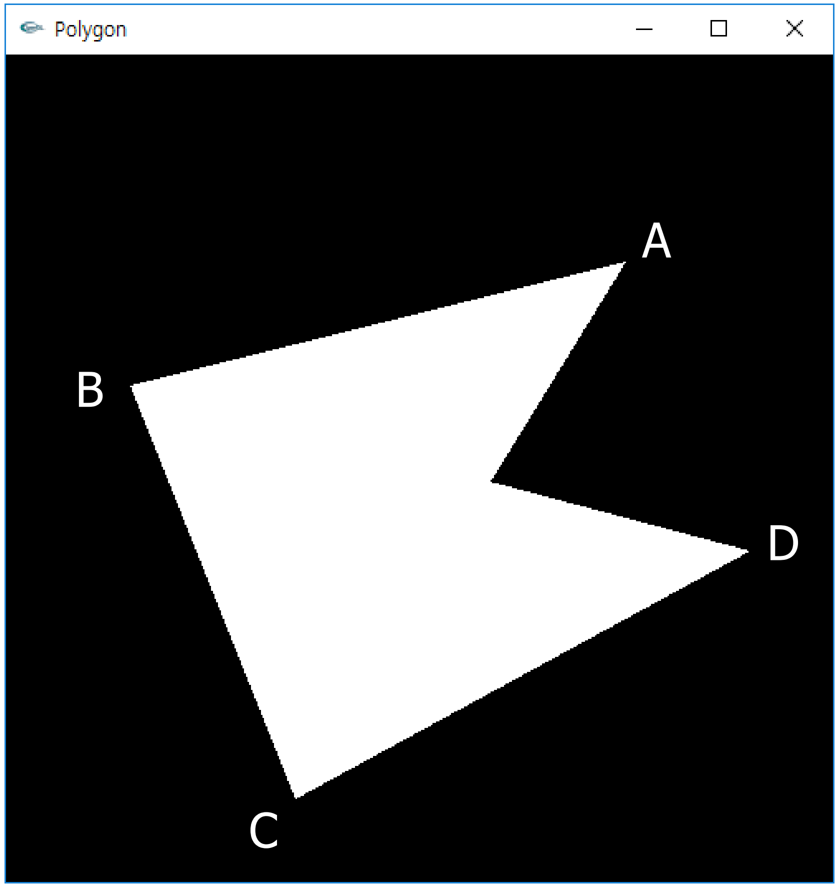
Polygon (Triangle)

```
void drawTriangle() {
	glColor3f(1, 1, 1);
	glBegin(GL_TRIANGLES);
		glVertex2f(0.5, 0.5);		// A
		glVertex2f(-0.7, 0.2);		// B
		glVertex2f(-0.3, -0.8);		// C
		glVertex2f(0.8, -0.2);		// D
	glEnd();
}
```

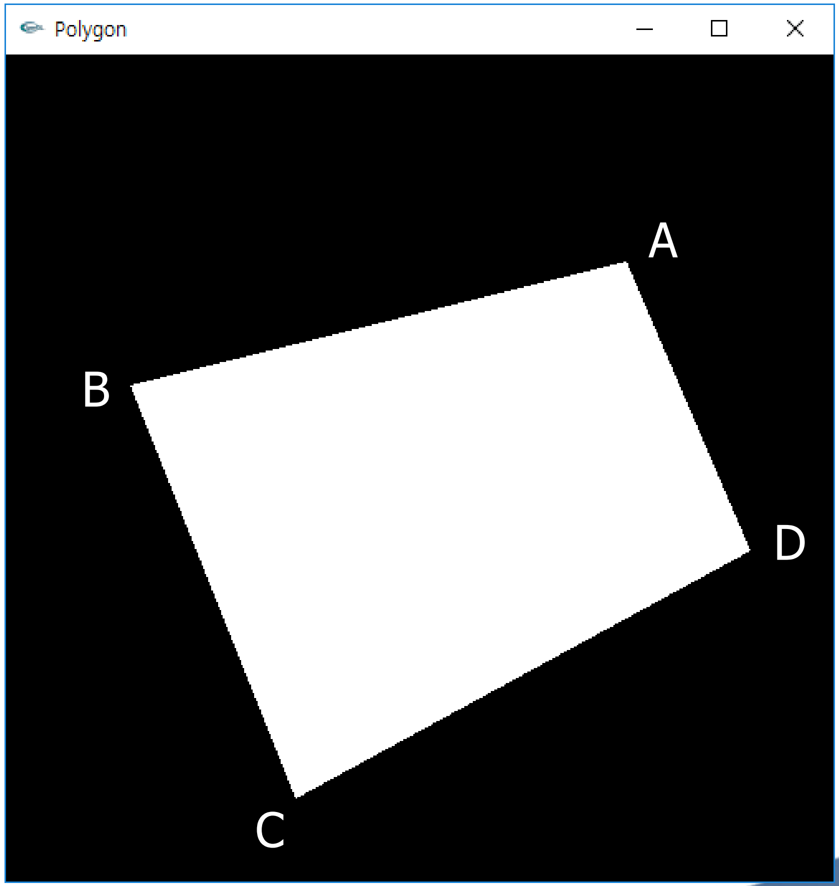


Polygon (Triangle)

GL_TRIANGLE_STRIP

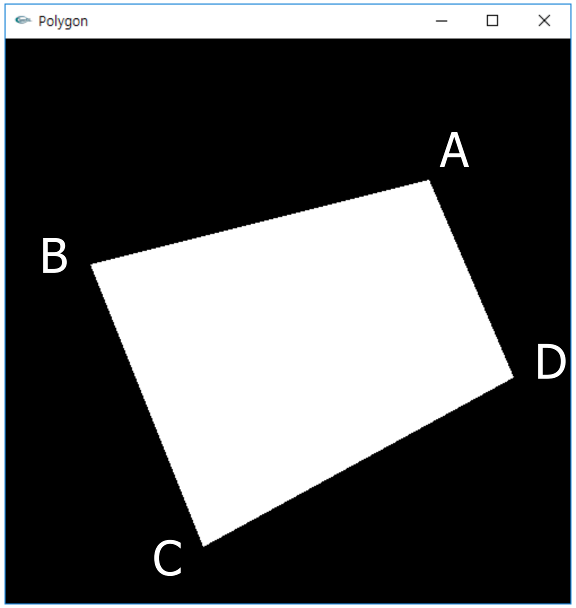


GL_TRIANGLE_FAN

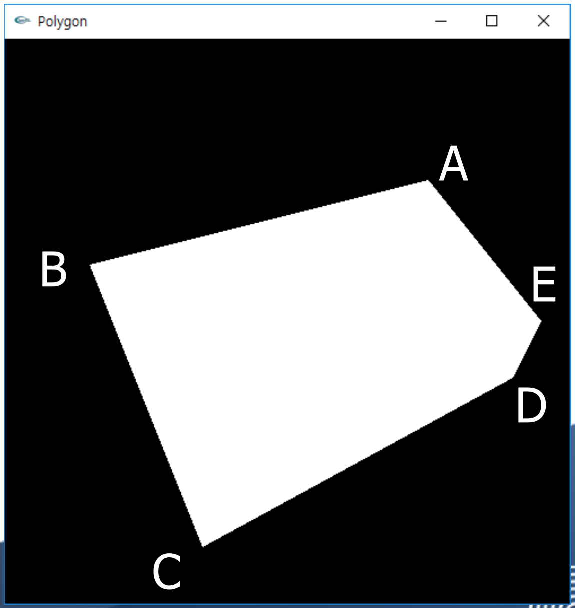


Polygon (Quadruple & Convex polygon)

```
void drawQuad() {
	glColor3f(1, 1, 1);
	glBegin(GL_QUADS);
		glVertex2f(0.5, 0.5);		// A
		glVertex2f(-0.7, 0.2);		// B
		glVertex2f(-0.3, -0.8);		// C
		glVertex2f(0.8, -0.2);		// D
		glVertex2f(0.9, 0.0);		// E
	glEnd();
}
```



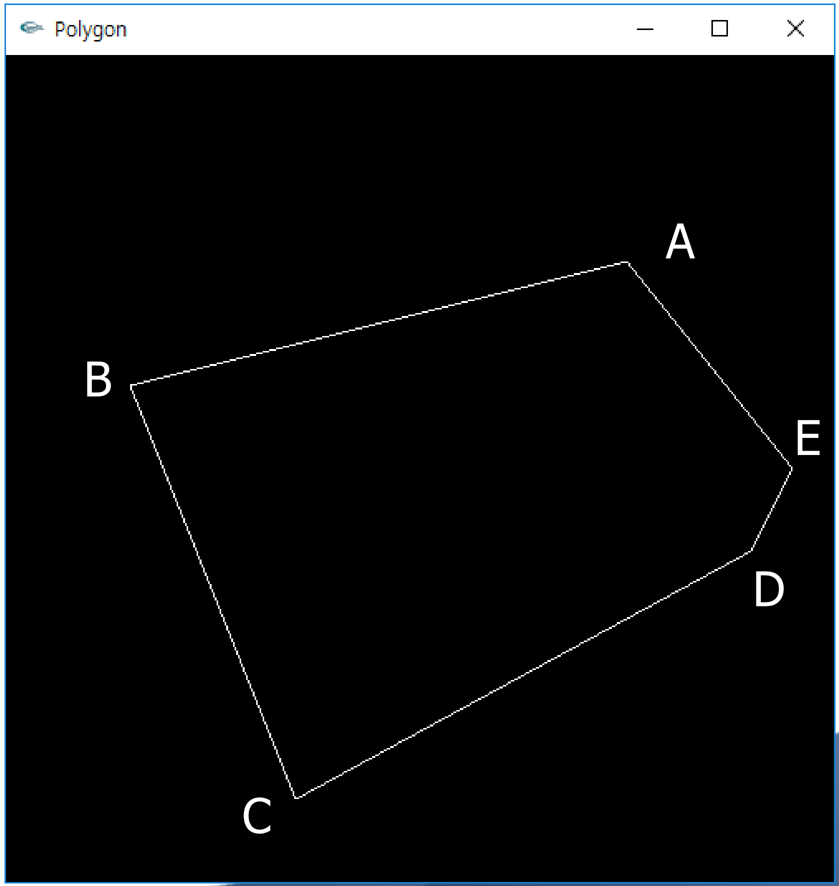
```
void drawPolygon() {
	glColor3f(1, 1, 1);
	glBegin(GL_POLYGON);
		glVertex2f(0.5, 0.5);		// A
		glVertex2f(-0.7, 0.2);		// B
		glVertex2f(-0.3, -0.8);		// C
		glVertex2f(0.8, -0.2);		// D
		glVertex2f(0.9, 0.0);		// E
	glEnd();
}
```



Polygon

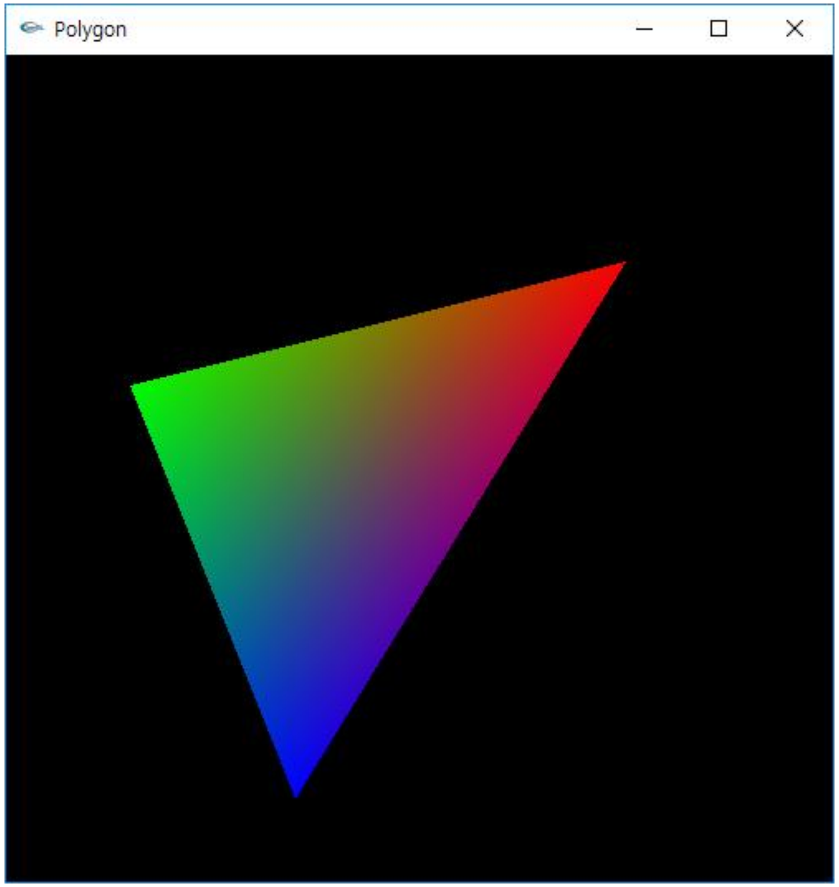
- glPolygonMode(GLenum face, GLenum mode)
  - face: GL_FRONT, GL_BACK, GL_FRONT_AND_BACK
  - mode: GL_POINT, GL_LINE, GL_FILL

```
void drawPolygon() {
	glColor3f(1, 1, 1);
	glPolygonMode(GL_FRONT, GL_LINE);
	glBegin(GL_POLYGON);
		glVertex2f(0.5, 0.5);		// A
		glVertex2f(-0.7, 0.2);		// B
		glVertex2f(-0.3, -0.8);	// C
		glVertex2f(0.8, -0.2);		// D
		glVertex2f(0.9, 0.0);		// E
	glEnd();
}
```



Vertex Colors

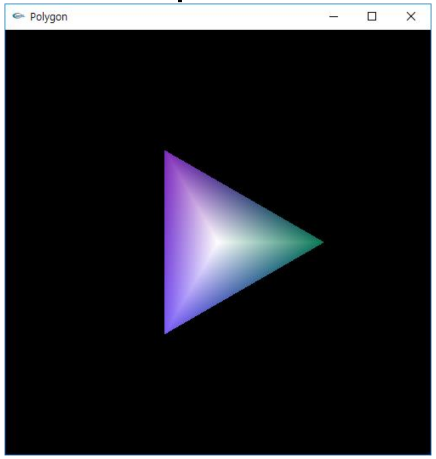
```
void drawTriangle() {
      glBegin(GL_TRIANGLES);
            glColor3f(1, 0, 0);
            glVertex2f(0.5, 0.5);          // A
            glColor3f(0, 1, 0);
            glVertex2f(-0.7, 0.2);         // B
            glColor3f(0, 0, 1);
            glVertex2f(-0.3, -0.8);        // C
      glEnd();
}
```



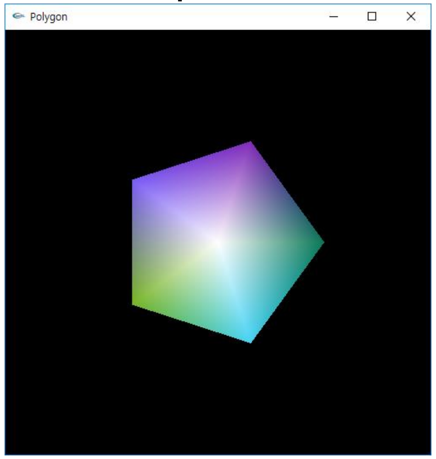
Today's assignment

- Draw square polygon using GL_POLYGON

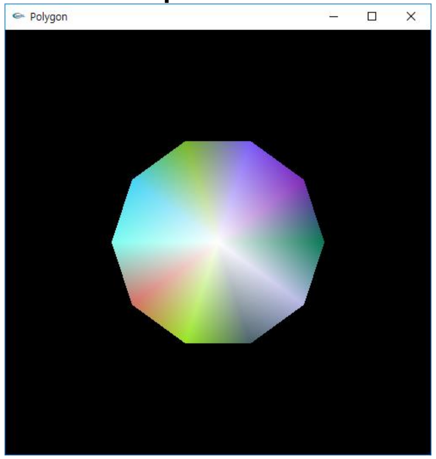
Input: 3



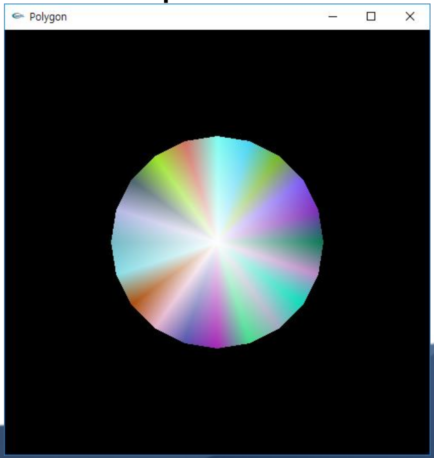
Input: 5



Input: 10



Input: 20

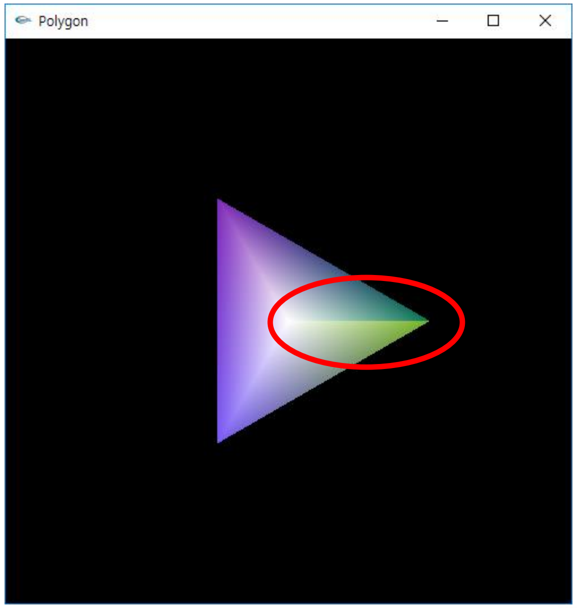


Today's assignment

- Given
  - Center (0, 0, 0)
  - Center color: white (1, 1, 1)
  - Distance between center and each vertex: 0.5
  - Circumference of circle (π)
  - Number of sides: Keyboard input

- Implementation
  - Draw polygon
    - Use loop statement
  - Calculate vertex color
    - Use random function: rand()
  - Calculate vertex position
    - Use sine, cosine function: sin(a), cos(a)

Caution!!



Assignment Submission

- Upload ETL
  - Attachment
    - One Source code file
  - Attachment Tile
    - Student ID + Name (2017-11111 OOO)
  - Due Date
    - Until Tuesday 11:59:59 pm