

# Triangles

---

OpenGL

# Learning Outcomes

---

- On completion of this lesson you should:
  - Understand the TRIANGLES primitive
  - Understand the meaning of “Winding” and distinguish between clockwise and counterclockwise winding
  - Be able to use the glFrontFace and glPolygonMode functions in this context
  - Have used TRIANGLE\_FAN and TRIANGLE\_STRIP primitives

# Drawing Triangles

---

- To draw a solid surface, you need more than just points and lines; you need polygons.
- A polygon is a closed shape that may or may not be filled with the currently selected color, and it is the basis of all solid-object composition in OpenGL.
- The simplest polygon possible is the triangle, with only three sides.
- The `GL_TRIANGLES` primitive draws triangles by connecting three vertices together

# Triangles

---

```
void renderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);

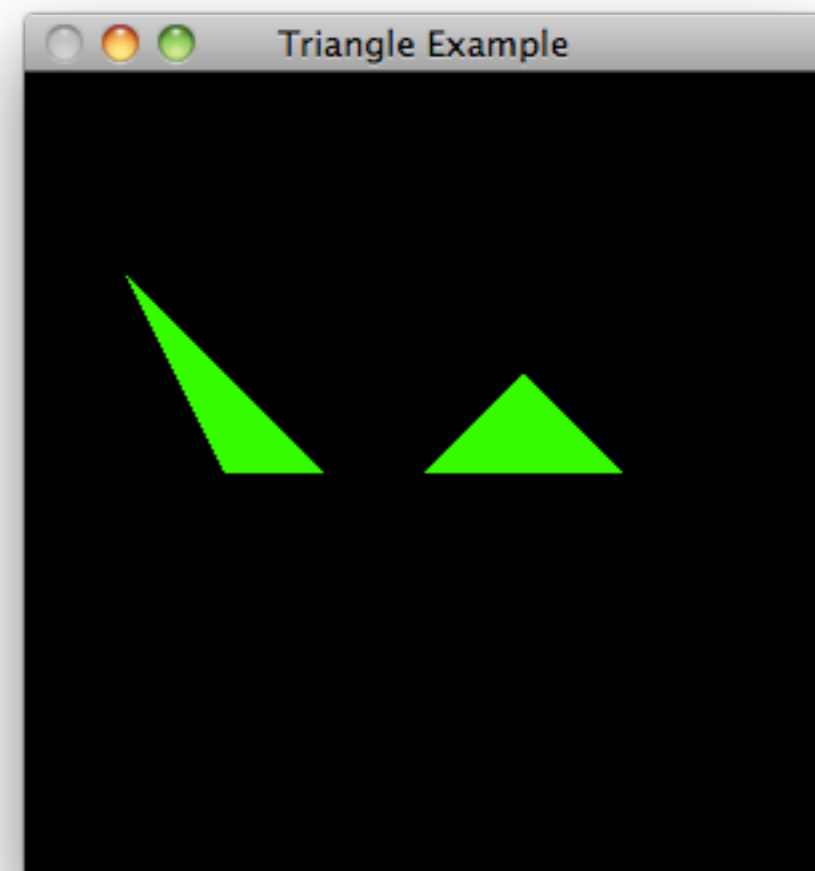
    glBegin(GL_TRIANGLES);

        glVertex2f(0.0f, 0.0f); // V0
        glVertex2f(25.0f, 25.0f); // V1
        glVertex2f(50.0f, 0.0f); // V2

        glVertex2f(-50.0f, 0.0f); // V3
        glVertex2f(-75.0f, 50.0f); // V4
        glVertex2f(-25.0f, 0.0f); // V5

    glEnd();

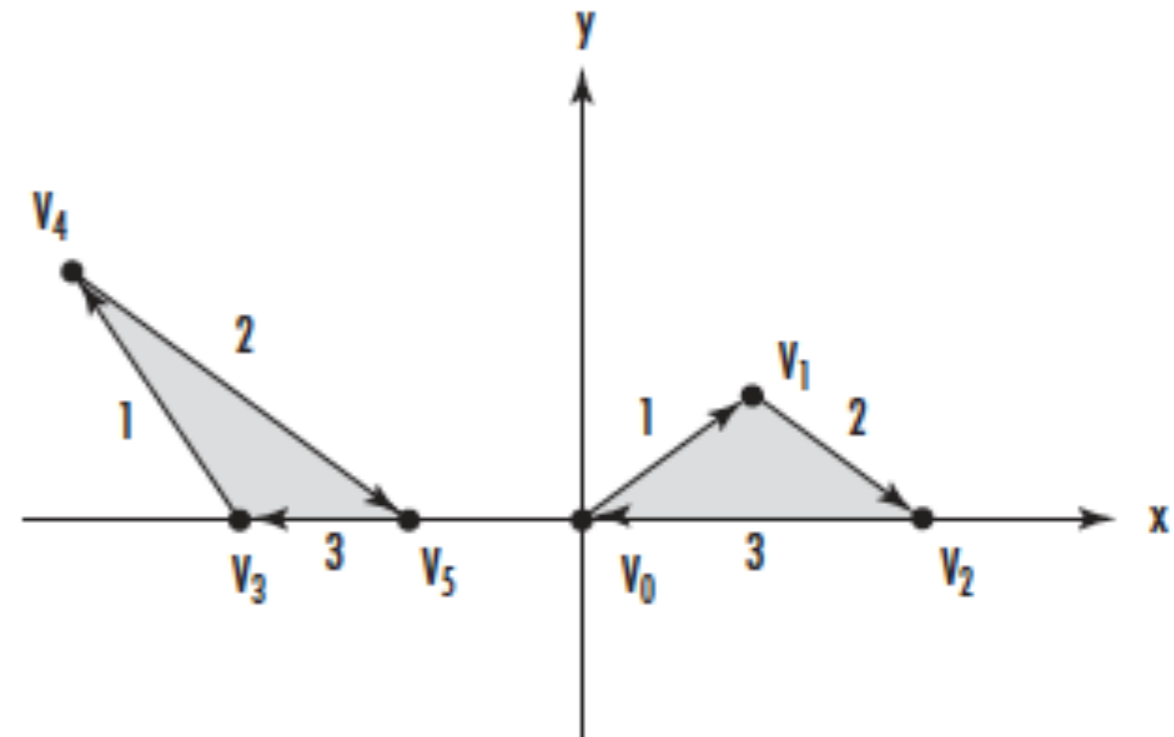
    glutSwapBuffers();
}
```



# Winding

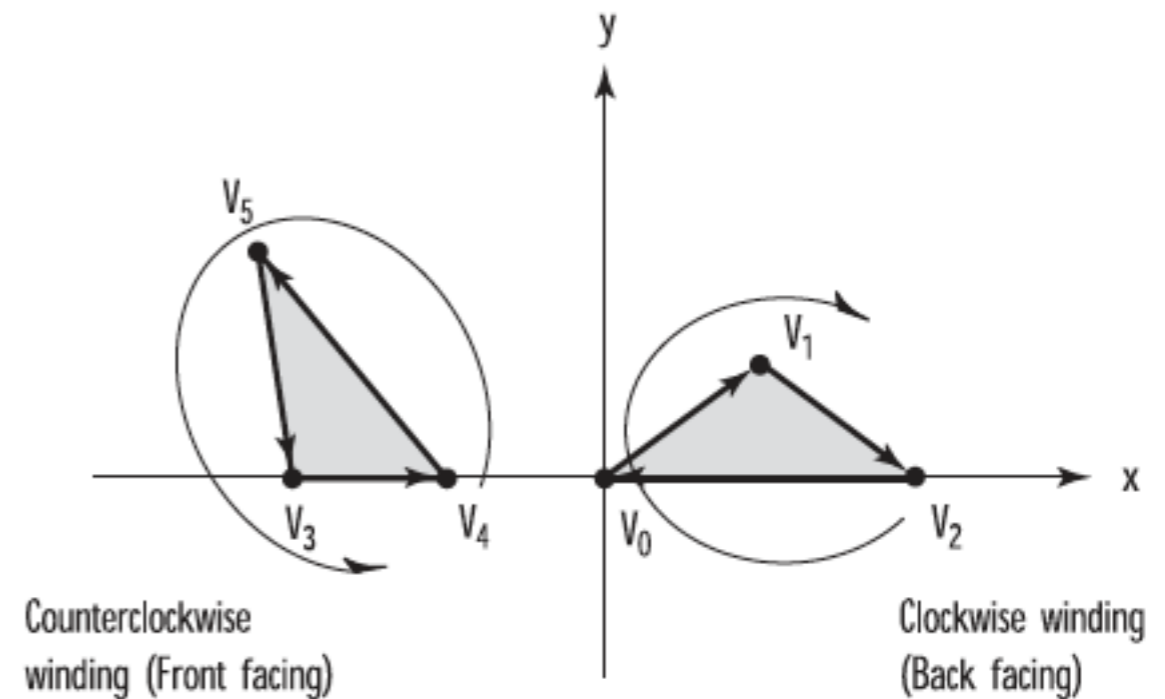
---

- Notice the arrows on the lines that connect the vertices.
- When the first triangle is drawn, the lines are drawn from  $V_0$  to  $V_1$ , then to  $V_2$ , and finally back to  $V_0$  to close the triangle.
- This path is in the order in which the vertices are specified - clockwise in this example.
- The combination of order and direction in which the vertices are specified is called winding.



# Clockwise and Counterclockwise Winding

- OpenGL, by default, considers polygons that have counterclockwise winding to be front facing.
- This means that the triangle on the left shows the front of the triangle, and the one on the right shows the back of the triangle.
- We may want to give the front and back of a polygon different physical characteristics. e.g. hide the back of a polygon altogether or give it a different color and reflective property
- In order to keep the winding of all polygons in a scene consistent, using front-facing polygons to draw the outside surface of any solid objects



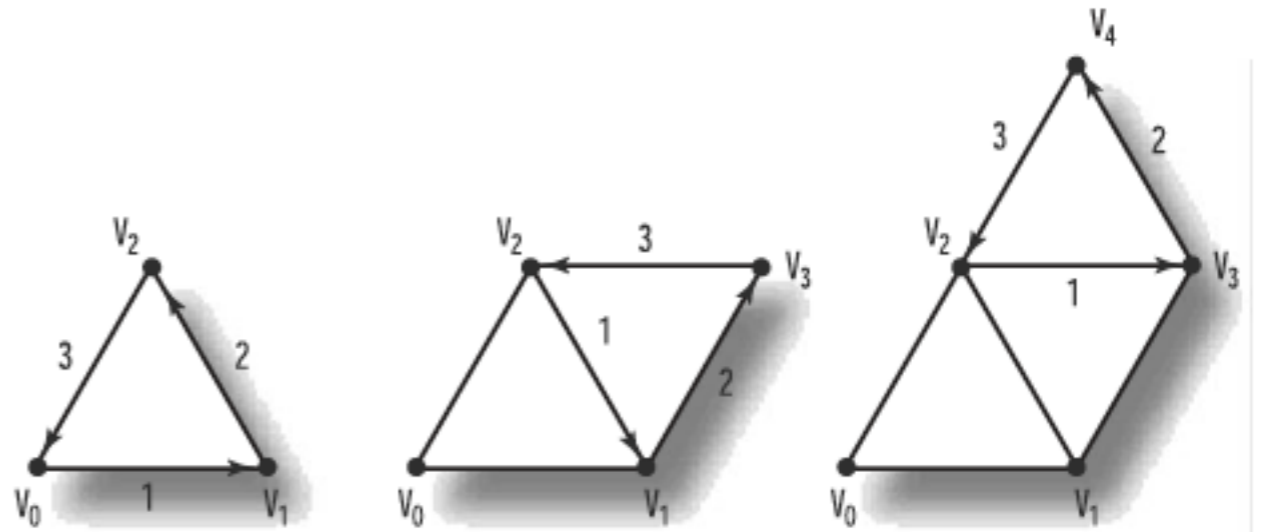
# Default Front/Back

---

- If you need to reverse the default behavior of OpenGL, you can do so by calling the following function:
- `glFrontFace(GL_CW);`
- The `GL_CW` parameter tells OpenGL that clockwise-wound polygons are to be considered front facing. To change back to counterclockwise winding for the front face, use `GL_CCW`.

# Triangle Strips

---



- For many surfaces and shapes, you need to draw several connected triangles.
- Can be drawn as a strip of connected triangles with the `GL_TRIANGLE_STRIP` primitive.
- Progression of a strip of three triangles specified by a set of five vertices numbered  $V_0$  through  $V_4$ . The vertices are not necessarily traversed in the same order in which they were specified.
- Preserve the winding (counterclockwise) of each triangle. The pattern is  $V_0, V_1, V_2$ ; then  $V_2, V_1, V_3$ ; then  $V_2, V_3, V_4$ ; and so on

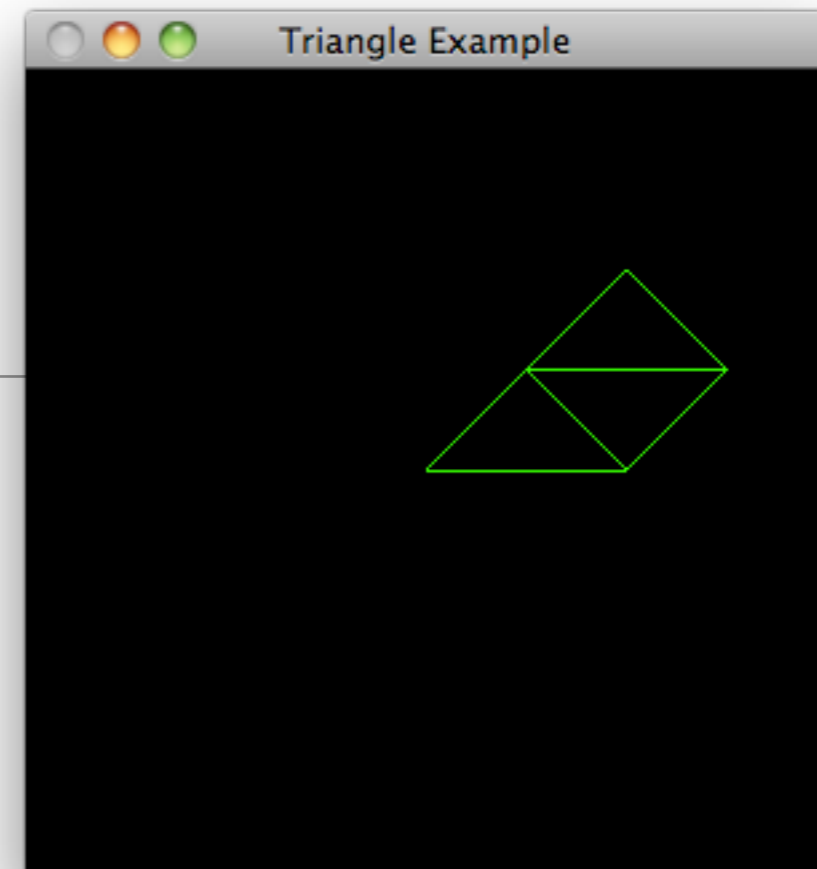


# Triangle Strips

```
void renderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_TRIANGLE_STRIP);
        glVertex2f(0.0f, 0.0f); // V0
        glVertex2f(50.0f, 0.0f); // V1
        glVertex2f(25.0f, 25.0f); // V2
        glVertex2f(75.0f, 25.0f); // V3
        glVertex2f(50.0f, 50.0f); // V4
    glEnd();

    glutSwapBuffers();
}
```

```
void setupRC()
{
    //...
    glPolygonMode(GL_FRONT, GL_LINE);
}
```

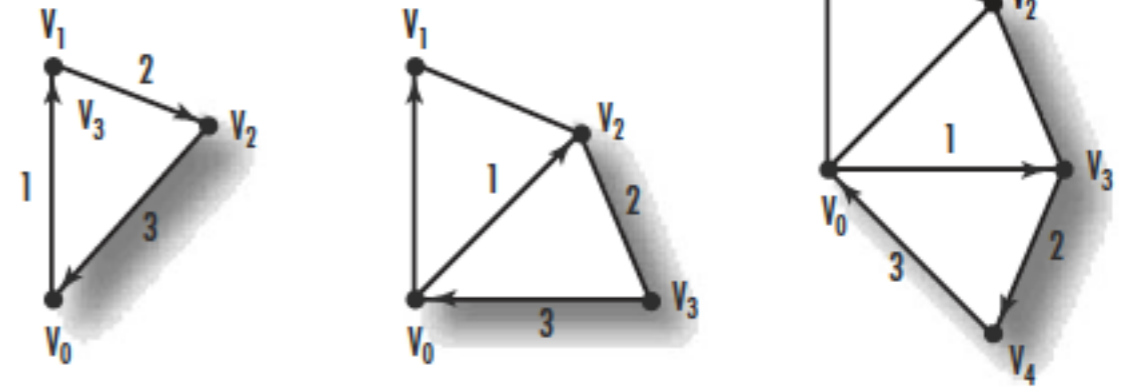


- Advantages:

1. Only need to specify only a single point for each additional triangle.
2. Fewer vertices means a faster transfer from computer's memory to graphics card and fewer vertex transformations

# Triangle Fans (1)

---



- Produce a group of connected triangles that fan around a central point
- The first vertex,  $V_0$ , forms the origin of the fan.
- After the first three vertices are used to draw the initial triangle, all subsequent vertices are used with the origin ( $V_0$ ) and the vertex immediately preceding it ( $V_{n-1}$ ) to form the next triangle.

# Triangle Fans (2)

```
void renderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_TRIANGLE_FAN);
        glVertex2f(0.0f, 0.0f);
        glVertex2f(0.0f, 50.0f);
        glVertex2f(25.0f, 30.0f);
        glVertex2f(40.0f, 0.0f);
        glVertex2f(25.0f, -30.0f);
    glEnd();

    glutSwapBuffers();
}
```

```
void retupRC()
{
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

    glColor3f(0.0f, 1.0f, 0.0f);

    glOrtho (-100.0f, 100.0f, -100.0f, 100.0f, -100.0f, 100.0f);

    glPolygonMode(GL_FRONT, GL_LINE);
    glPolygonMode(GL_BACK, GL_LINE);
}
```

