

الجلسة الرابعة: عمليات التدوير الأساسية وفصل التحويلات

الهدف من الجلسة

- التعرف بعمليات التدوير الأساسية.
- التعرف بتعليمتي `glPushMatrix` و `glPopMatrix`.

عمليات تدوير مهمة:

تدوير شكل حول نقطة:

لنأخذ المثال التالي:

لديك المثلث بالإحداثيات التالي:

(-3,0)

(0,3)

(3,0)

ويُطلب تدويره حول النقطة (0,3) بزاوية 180° ، الخطوات اللازمة هي:

1. تطبيق عملية إزاحة على الشكل بالإحداثيات التمس تجعل النقطة منطبقة على مبدأ الإحداثيات وحسب النقطة المطلوبة:

`glTranslatef(0,-3,0)`

2. تطبيق عملية دوران على الشكل كله بالزاوية المطلوبة حول المبدأ وهو يوافق المحور Z.
3. إعادة الشكل إلى مكانه الأصلي بتطبيق عملية إزاحة معاكسة للعملية الأولى.

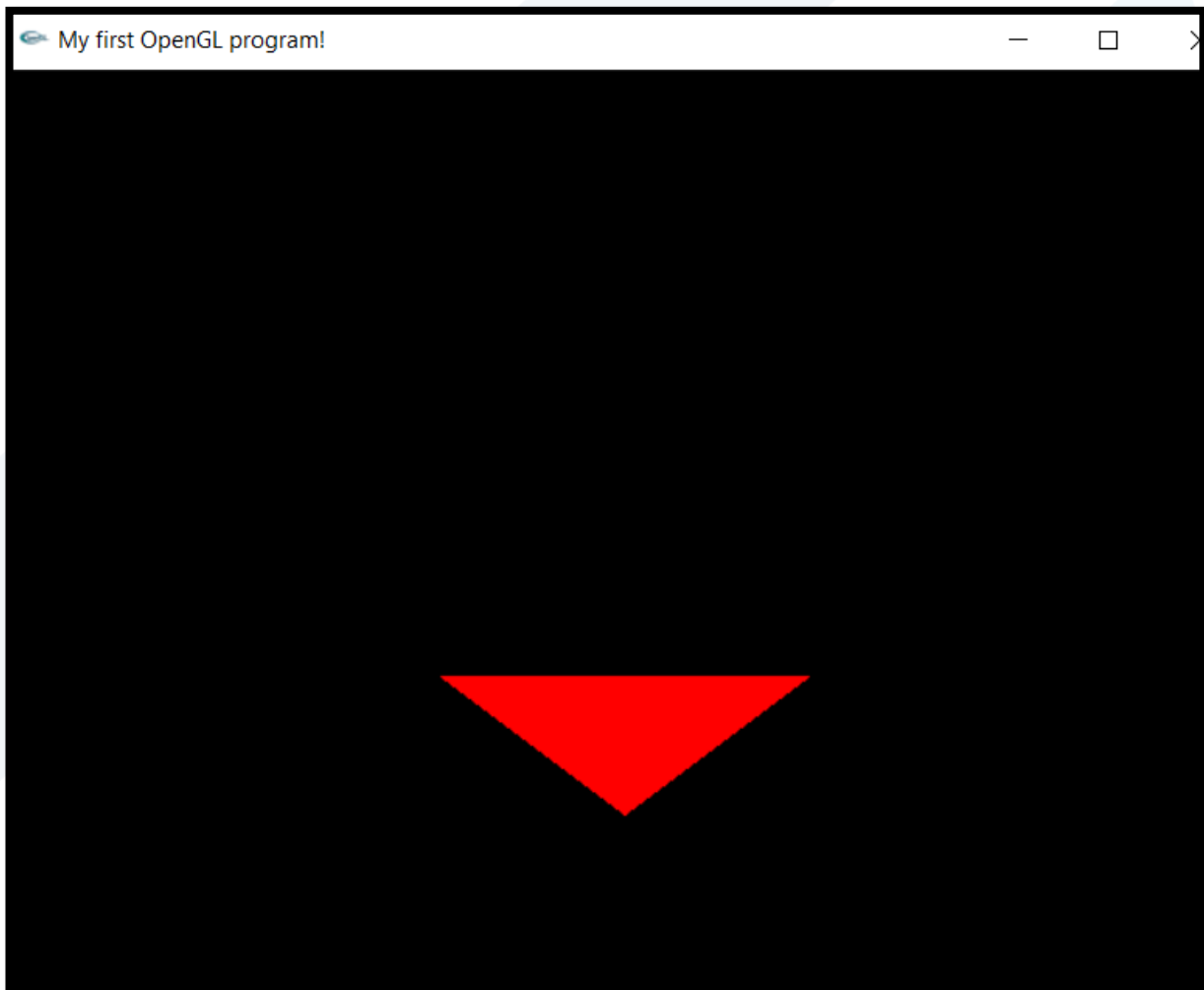
فيكون الكود المطلوب:

```
void draw2DTiangle(){
    glBegin(GL_TRIANGLES);
    glColor3f(1,0,0);
    glVertex2d(-3,0);
    glVertex2d(0,3);
    glVertex2d(3,0);
    glEnd();
}

void myDisplay(void)
{
    glClearColor(0,0,0,0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glTranslatef(0,-3,0);
    glRotatef(180,0,0,1);
}
```

```
draw2DTiangle();  
glTranslatef(0,3,0);  
glutSwapBuffers();  
}
```

النتيجة:



تدوير الشكل حول نفسه:

الخطوات اللازمة:

1. حساب مركز ثقل الشكل وهو يتبع المعادلة التالية:

$$\left(\sum_{i=0}^n \frac{x_i}{n}, \sum_{i=0}^n \frac{y_i}{n}, \sum_{i=0}^n \frac{z_i}{n} \right)$$

2. تطبيق عملية إزاحة على الشكل بالإحداثيات التي تجعل مركز الثقل ينطبق على مبدأ الإحداثيات وحسب النقطة المطلوبة.

3. تطبيق عملية دوران على الشكل كله بالزاوية المطلوبة حول المبدأ وهو يوافق المحور Z.

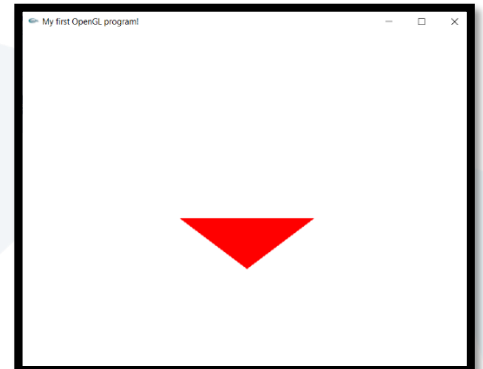
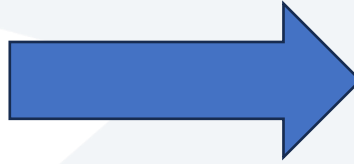
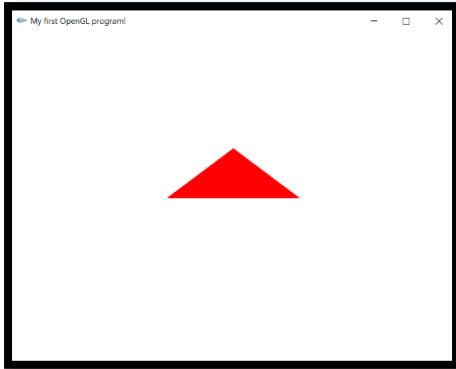
4. إعادة الشكل إلى مكانه الأصلي بتطبيق عملية إزاحة معاكسة للعملية الأولى.

الكود اللازم:

النتيجة:

```
void draw2DTiangle(){
    glBegin(GL_TRIANGLES);
    glColor3f(1,0,0);
    glVertex2d(-3,0);
    glVertex2d(0,3);
    glVertex2d(3,0);
    glEnd();
}

void myDisplay(void)
{
    glClearColor(1,1,1,0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glTranslatef(0,-1,0);
    glRotatef(180,0,0,1);
    draw2DTiangle();
    glTranslatef(0,1,0);
    glutSwapBuffers();
}
```



تعليمتي `glPopMatrix` و `glPushMatrix`:

تُستخدم هتتين التعليمتين لفصل تحويلات شكل عن تحويلات شكل آخر.

مثلاً: إذا كان لدينا الكود التالي:

```
#include<GL\glut.h>
void drawAxis(){
glBegin(GL_LINES);
glColor3f(0,0,0);
glVertex2d(-10,0);
glVertex2d(10,0);
glVertex2d(0,-10);
glVertex2d(0,10);
glEnd();
}

void draw2DTiangle1(){
glBegin(GL_TRIANGLES);
glColor3f(1,0,0);
glVertex2d(0,0);
glVertex2d(3,3);
glVertex2d(6,0);
glEnd();
}
```

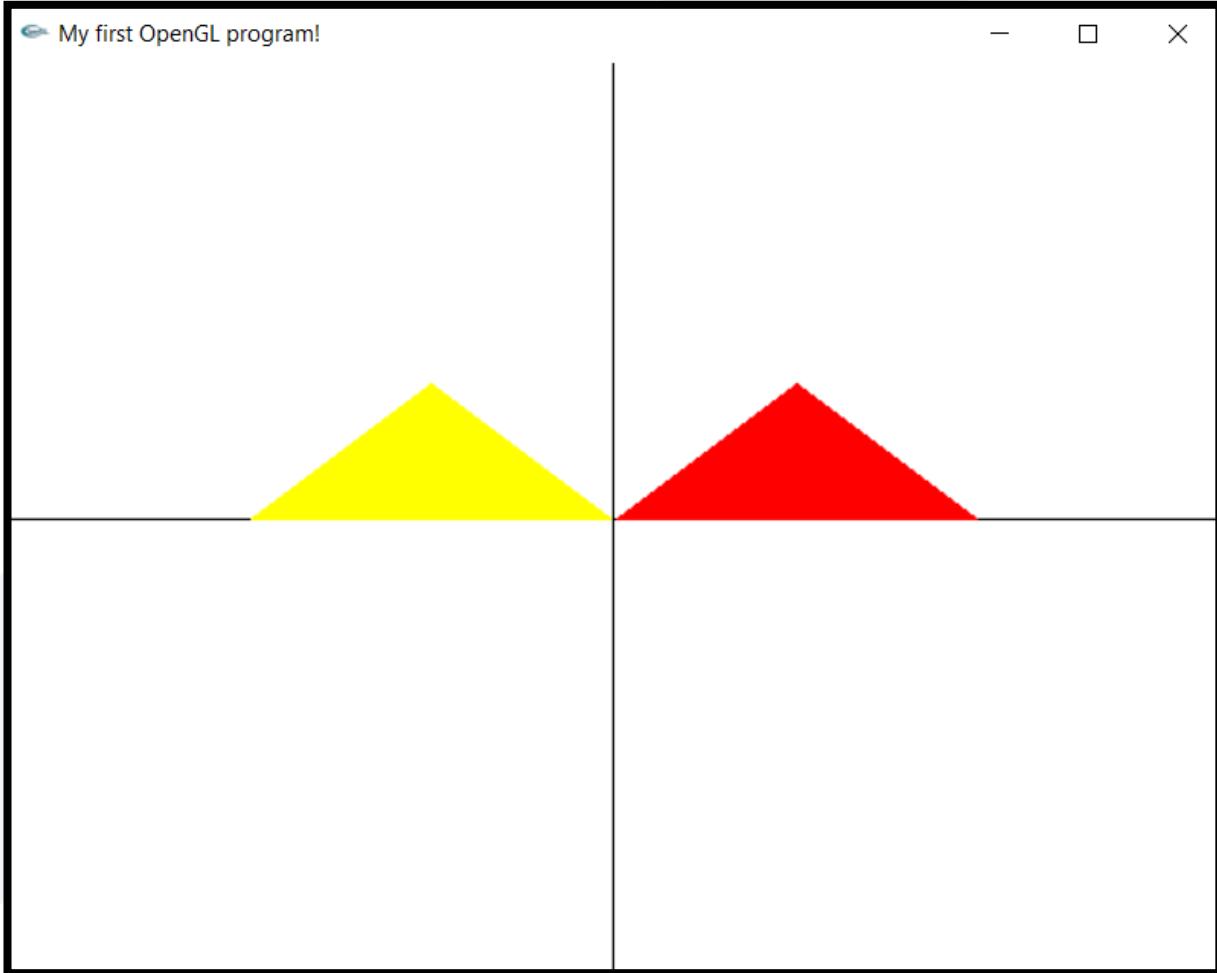
```
void draw2DTiangle2(){
    glBegin(GL_TRIANGLES);
    glColor3f(1,1,0);
    glVertex2d(-6,0);
    glVertex2d(-3,3);
    glVertex2d(0,0);
    glEnd();
}

void myDisplay(void)
{
    glClearColor(1,1,1,0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    drawAxis();
    draw2DTiangle1();
    draw2DTiangle2();
    glutSwapBuffers();
}

void main (int argc, char **argv)
{
    glutInit (&argc, argv); // to initialize the toolkit;
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH); //sets the display mode
    glutInitWindowSize (640, 480); // sets the window size
    glutInitWindowPosition (10, 10); // sets the starting position for the window
    glutCreateWindow ("My first OpenGL program!"); //creates the window and sets the title
    glOrtho(-10,10,-10,10,-10,10);
    glutDisplayFunc (myDisplay);
}
```

```
glutMainLoop(); // go into a loop until event occurs  
}
```

سيظهر الشكل التالي:



إذا أردنا نقل المثلث الأحمر نحو اليمين خطوتين والمثلث الأصفر نحو اليسار خطوتين، و قمنا بتطبيق تعلمتي تحويل بالشكل التالي:

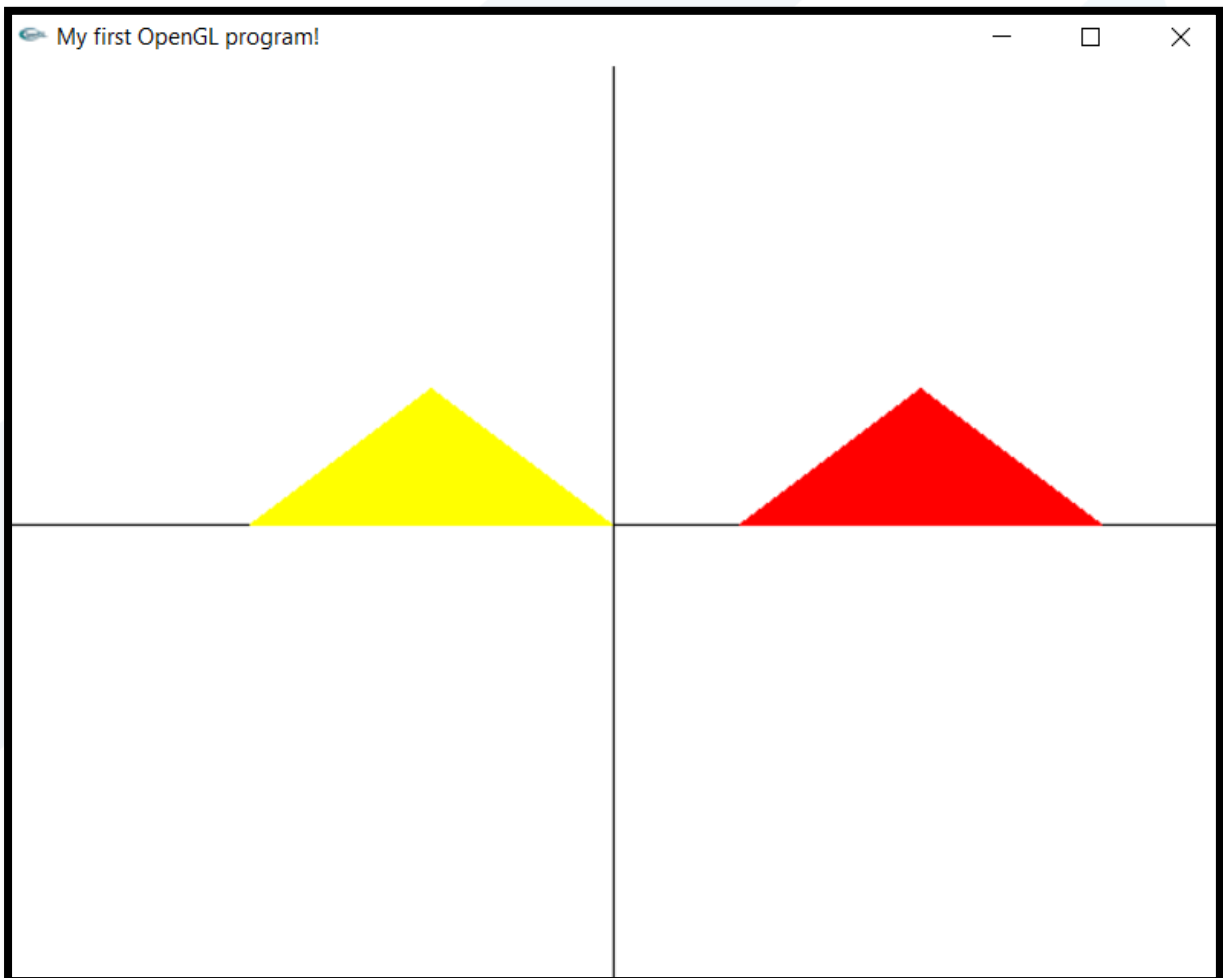
```
void myDisplay(void)  
{  
    glClearColor(1,1,1,0);  
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);  
    drawAxis();  
    glTranslatef(2,0,0);  
    draw2DTiangle1();  
}
```

```

glTranslatef(-2,0,0);
draw2DTiangle2();
glutSwapBuffers();
}

```

النتيجة:



تعلیمة التحویل الأولى طبقت على جميع الأشكال، لفصل التحويلات عن بعضها نستخدم تعلیمتی: `glPushMatrix()` وتعلیمتی `glPopMatrix()`.

كما يلي:

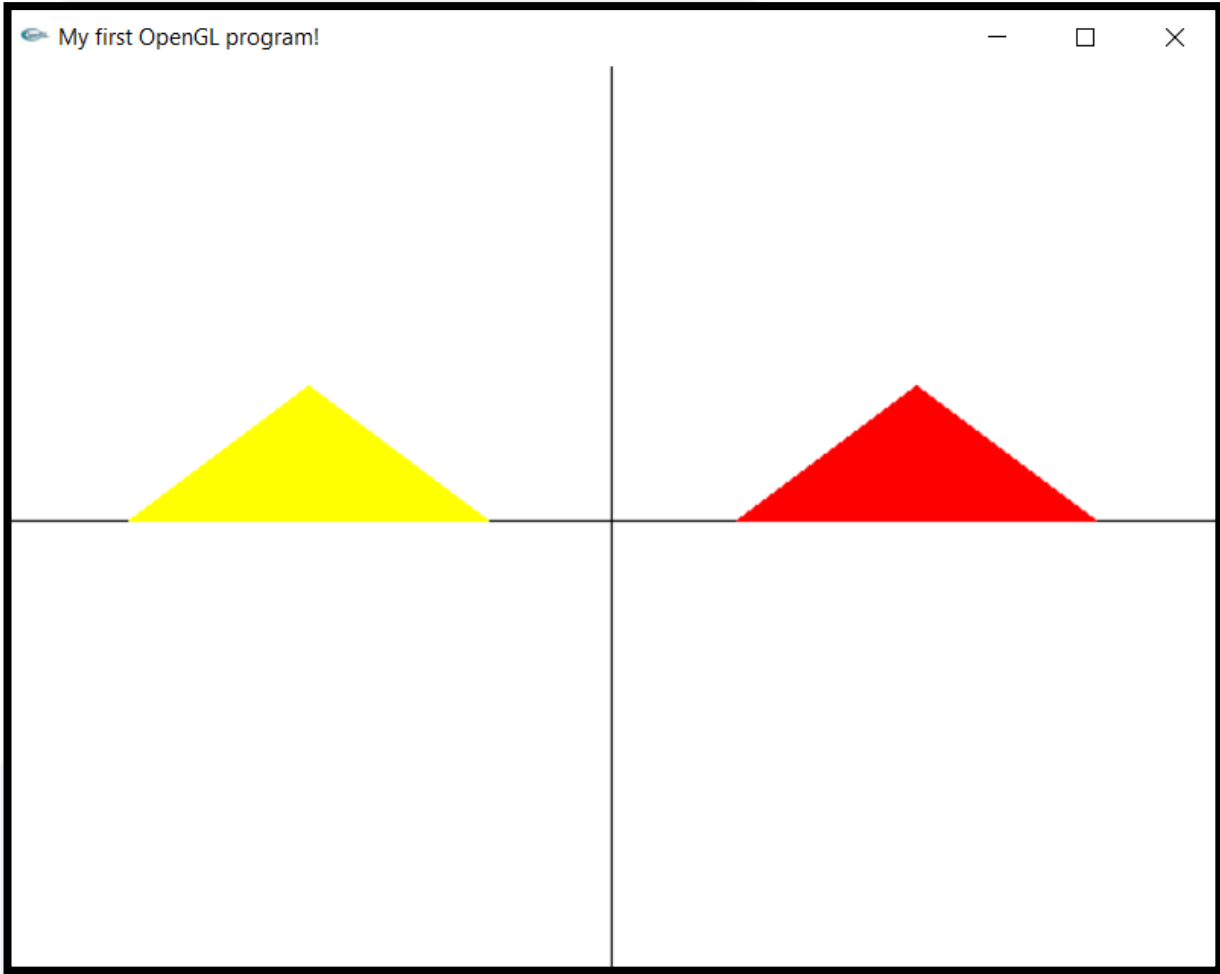
```

void myDisplay(void)
{
    glClearColor(1,1,1,0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
}

```

```
drawAxis();  
glPushMatrix();  
glTranslatef(2,0,0);  
draw2DTiangle1();  
glPopMatrix();  
glPushMatrix();  
glTranslatef(-2,0,0);  
draw2DTiangle2();  
glPopMatrix();  
glutSwapBuffers();  
}
```

يظهر الخرج التالي:



انتهت الجلسة