

# الجلسة الثالثة: الرسم ثلاثي الأبعاد والعمليات على الأشكال

## الهدف من الجلسة

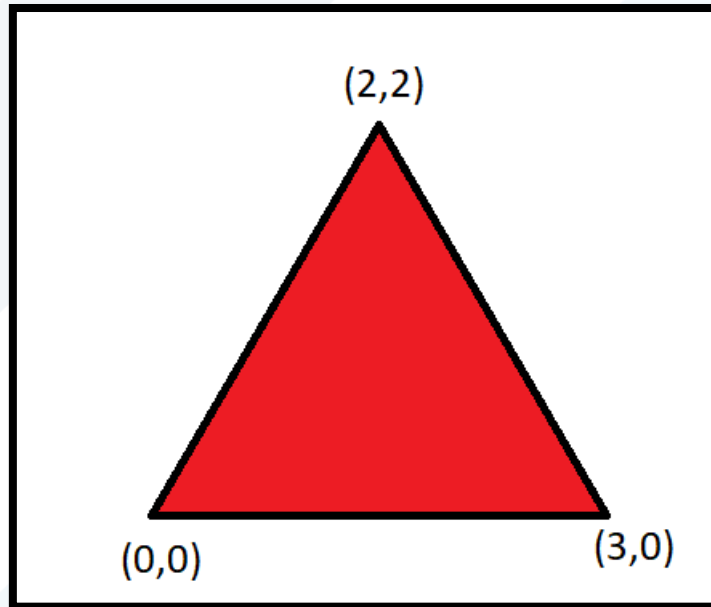
- التعريف بالرسم ثلاثي الأبعاد.
- التعريف بالعمليات المختلفة لتطبيقه على أشكال ثنائية البعد.
- حل تمارين.

## الرسم في الفضاء ثلاثي الأبعاد

- يستخدم هذا الرسم المحور z إضافةً إلى المحورين x و y.
- تظهر في الشكل ثلاثي الأبعاد أوجه إضافية.

مثال:

إذا كان لدينا المثلث التالي المرسوم في الفضاء ثنائي البعد:



لتحويله إلى شكل ثلاثي الأبعاد، سيكون شكلاً هرمياً، وهذا يتطلب 4 أوجه، كل وجه له 4 نقاط أي الشكل كاملٌ سيملك 12 نقطة، وقد يظهر وجه واحد عالشاشة.

مثال برمجى:

```
#include<GL/glut.h>
void myDisplay(void)
{
glClearColor(0,0,0,0);
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glBegin(GL_TRIANGLES);
// Red triangle (front face)
```

```
glColor3f(1.0f, 0.0f, 0.0f);
glVertex3f(-3, 0, 0);
glVertex3f( 3, 0, 0);
glVertex3f( 0, 2, -1);

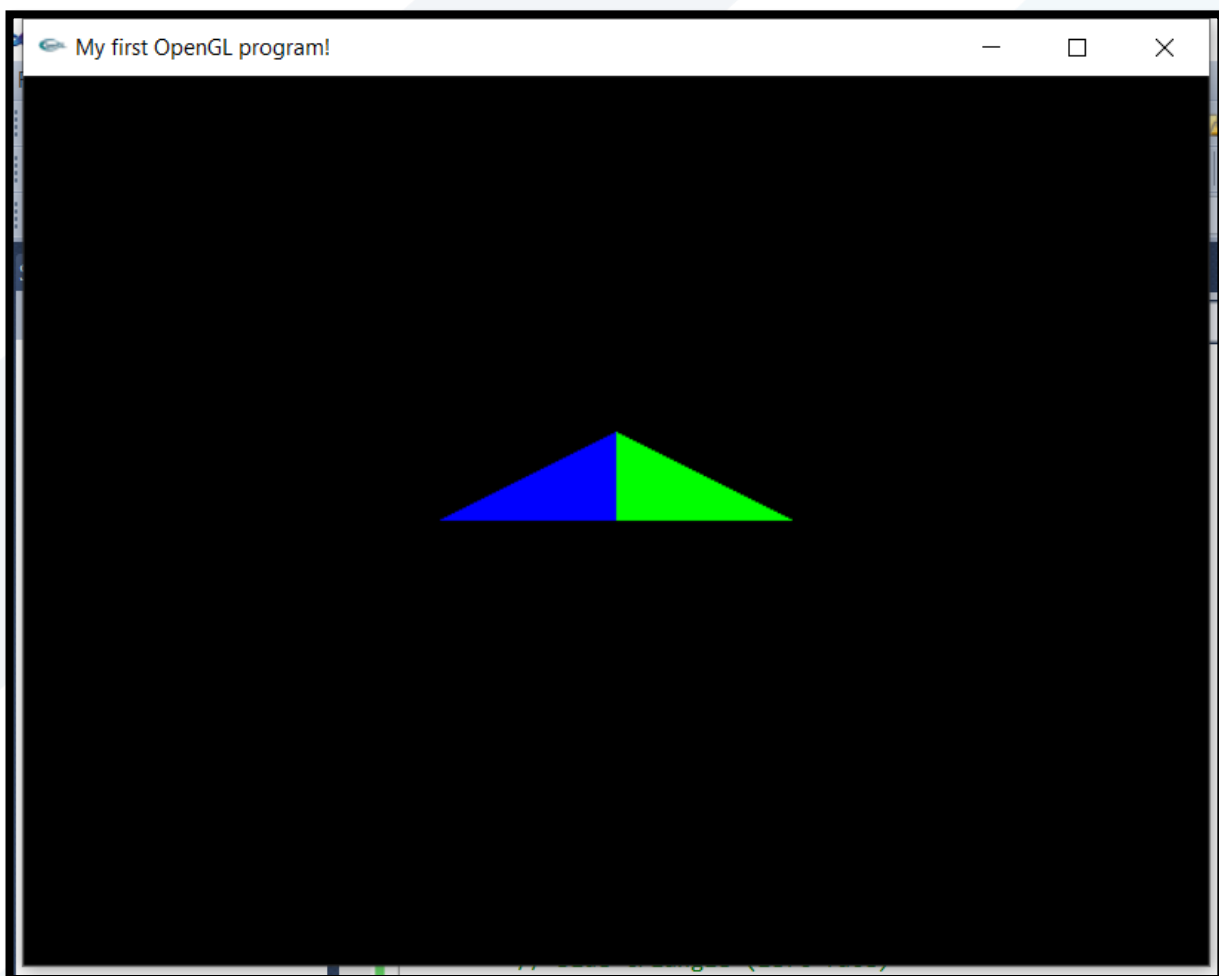
// Green triangle (right face)
glColor3f(0.0f, 1.0f, 0.0f);
glVertex3f( 3, 0, 0);
glVertex3f( 0, 0, -3);
glVertex3f( 0, 2, -1);

// Blue triangle (left face)
glColor3f(0.0f, 0.0f, 1.0f);
glVertex3f(-3, 0, 0);
glVertex3f( 0, 0, -3);
glVertex3f( 0, 2, -1);

// Cyan triangle (base)
glColor3f(0.0f, 1.0f, 1.0f);
glVertex3f(-3, 0, 0);
glVertex3f( 3, 0, 0);
glVertex3f( 0, 0, -3);
glEnd();
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  
}
```

خرج الكود السابق:



## التحويلات الأساسية:

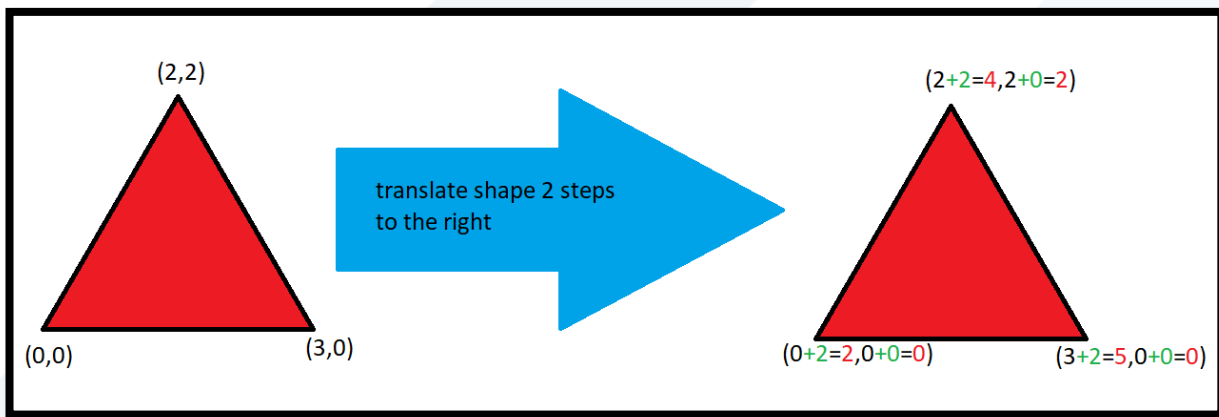
- الإزاحة.
- الدوران.

- التحجيم.

## الإزاحة translation:

- هي عملية تحريك الشكل على المحاور (x) في الفضاء ثنائي البعد، أو تحريكه على المحاور (x, y) في الفضاء ثلاثي الأبعاد.
- يمثل عملية الجمع رياضياً.

مثال:



برمجياً:

- نستخدم تابع `glTranslatef(x_step, y_step, z_step)`.
- يُكتب تابع الإزاحة قبل الشكل المراد تحريكه.

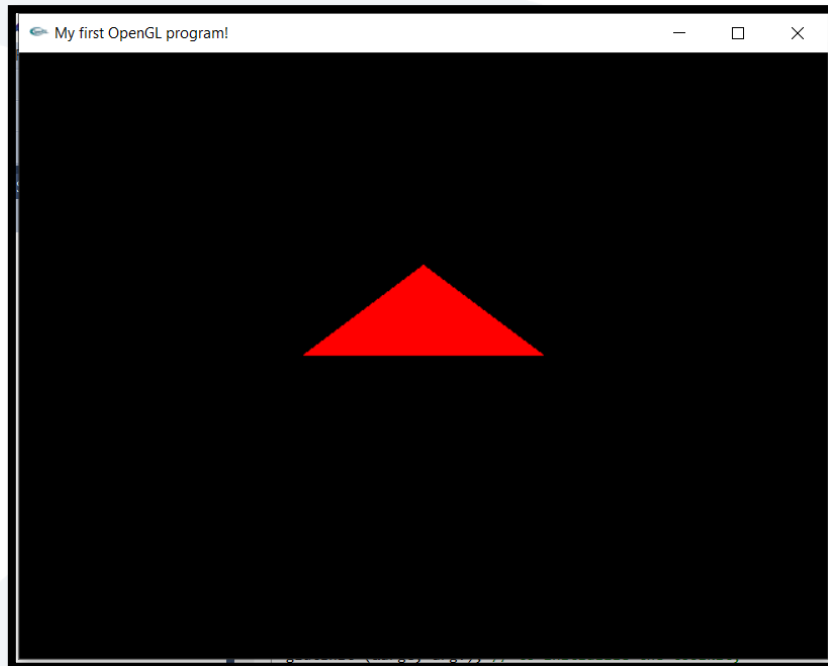
مثال برمجي:

ليكن لدينا الكود التالي:

```
#include<GL/glut.h>
void myDisplay(void)
{
    glClearColor(0,0,0,0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
    glColor3f(1,0,0);
    glVertex2f(-3,0);
```

```
glVertex2f(3,0);  
glVertex2f(0,3);  
glEnd();  
  
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  
}
```

خرج الكود السابق:



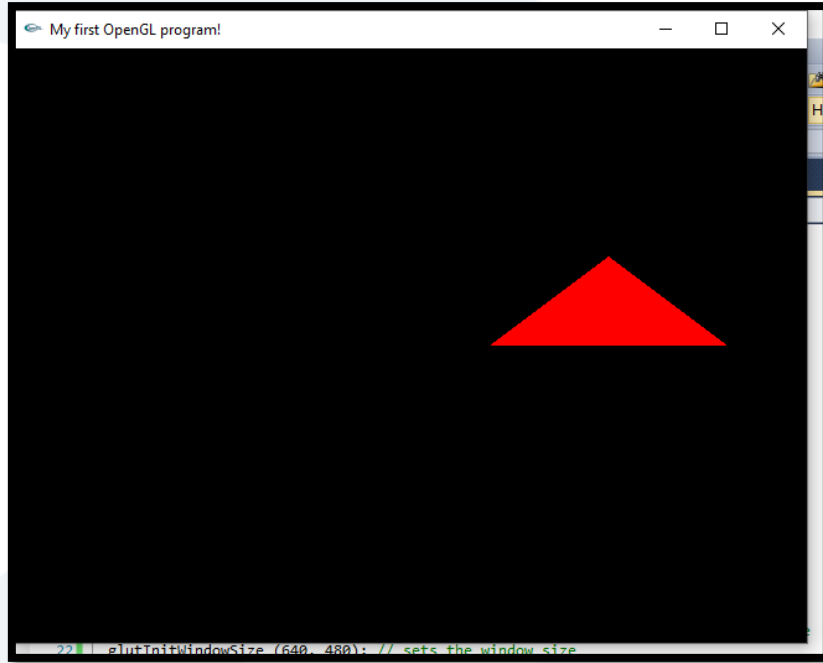
لتطبيق تعليمة الإزاحة بالشكل التالي:

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

void myDisplay(void)
{
glClearColor(0,0,0,0);
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glTranslatef(5,0,0);
glBegin(GL_TRIANGLES);
glColor3f(1,0,0);
glVertex2f(-3,0);
glVertex2f(3,0);
glVertex2f(0,3);
glEnd
glutSwapBuffers();
}
```

**Translation  
method**

يكون الخرج بالشكل التالي:



## تمرين 1:

ارسم مربع وحركه إلى الأسفل بمقدار 3 خطوات.

## تمرين 2:

ارسم مكعب وحركه على المحور x نحو اليمين بمقدار 5.

## الدوران Rotation:

هي عملية تدوير الشكل حول أحد المحاور، يتم تطبيق مصفوفات التدوير حسب كل محور دوران.

يحتاج لتحديد:

- محور الدوران.
- زاوية الدوران.
- اتجاه الدوران.

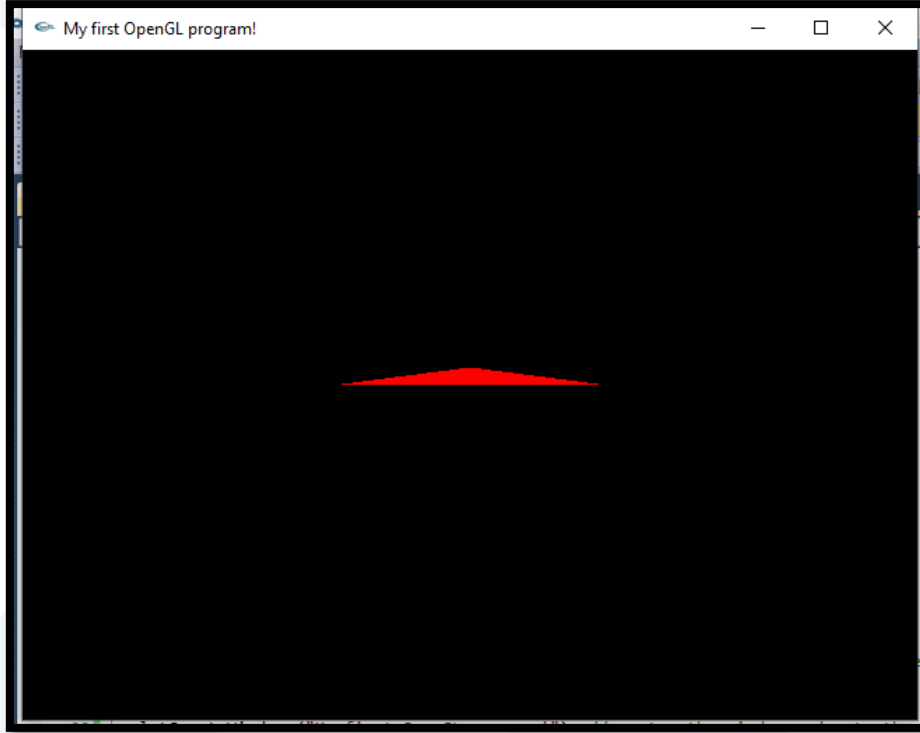
### مثال برمجى:

لتدوير المثلث السابق حول محور الـ X بزاوية  $80^\circ$  مع عقارب الساعة، نطبق الكود التالي:

```
void myDisplay(void)
{
glClearColor(0,0,0,0);
```

```
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);  
glRotatef(-80,1,0,0);  
glBegin(GL_TRIANGLES);  
glColor3f(1,0,0);  
glVertex2f(-3,0);  
glVertex2f(3,0);  
glVertex2f(0,3);  
glEnd();  
glutSwapBuffers();  
}
```

خرج الكود السابق:



### تمرين 3:

قم بتدوير المكعب في التمرين 2 حول المحور z بزاوية  $90^\circ$  مع عقارب الساعة.

### التحجيم scaling

هو عملية تغيير حجم الشكل إما بتكبيره أو تصغيره.

يمثل عملية الضرب رياضياً.

برمجياً:

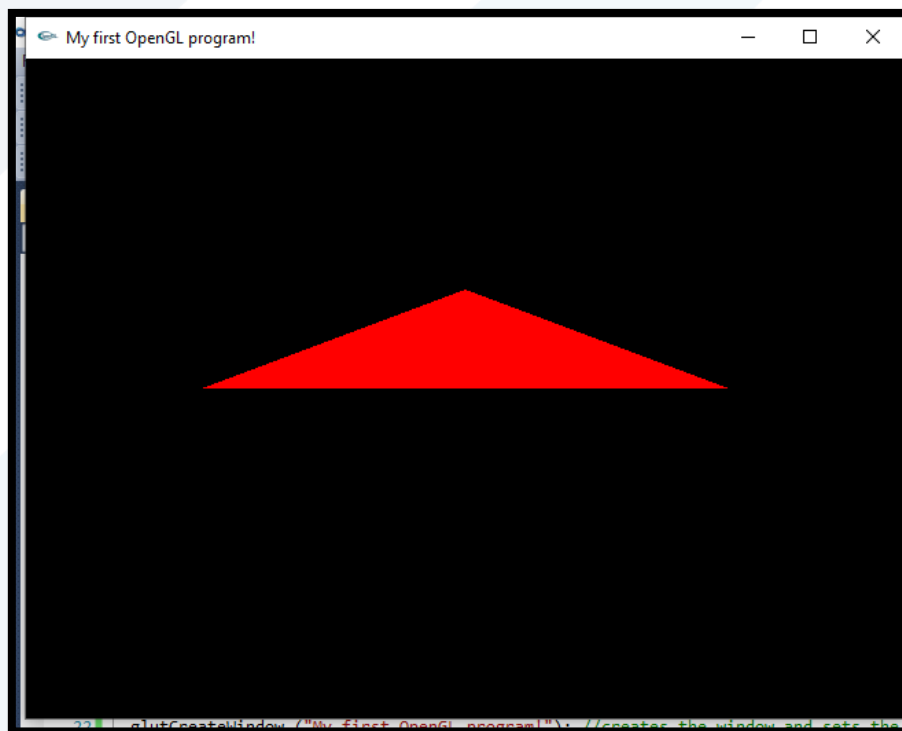
نستخدم التابع `glScalef(x_scale_amount, y_scale_amount, z_scale_amount)`

مثال برمجي:

```
#include<GL/glut.h>\nvoid myDisplay(void)
```

```
{  
glClearColor(0,0,0,0);  
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);  
glScalef(2,1,1);  
glBegin(GL_TRIANGLES);  
glColor3f(1,0,0);  
glVertex2f(-3,0);  
glVertex2f(3,0);  
glVertex2f(0,3);  
glEnd();  
glutSwapBuffers();  
}
```

خروج الكود السابق:



#### تمرين 4:

قم بتكبير المكعب المرسوم في التمرين 2 إلى الضعف.

#### تمرين 5:

قم بتصغير المكعب المرسوم في التمرين 2 إلى الثلث.

#### ملاحظات هامة:

توابع التحويلات تُكتب قبل رسم الشكل المراد تحريكه

تُطبق تعليمات التحويلات على كل الأشكال التي تُكتب تحتها

في الإزاحة: حيادي الجمع هو 0

في التحجيم: حيادي الضرب هو 1

#### حل التمارين السابقة:

[https://drive.google.com/drive/folders/1NL\\_C3CPCEkhILNHW-YE3dvgxp7TSzwr?usp=sharing](https://drive.google.com/drive/folders/1NL_C3CPCEkhILNHW-YE3dvgxp7TSzwr?usp=sharing)

#### تمرين إضافي:

نفذ الكود التالي:

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

```
void axis(){
```

```
//a method to draw X, Y axis
```

```
glBegin(GL_LINES);
```

```
glColor3f(0,0,0);
glVertex3d(-10,0,0);
glVertex3d(10,0,0);

glVertex3d(0,-10,0);
glVertex3d(0,10,0);

glVertex3d(0,0,-10);
glVertex3d(0,0,10);
glEnd();

}

void myDisplay(void)
{
glClearColor(1,1,1,0);
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
//glTranslatef(-2,
axis();
glBegin(GL_TRIANGLES);
glColor3f(1,0,0);
glVertex2f(-3,0);
glVertex2f(0,0);
glVertex2f(-1,3);
glEnd();

glBegin(GL_TRIANGLES);
glColor3f(0,1,0);
glVertex2f(3,0);
glVertex2f(6,0);
glVertex2f(4,3);
```

```

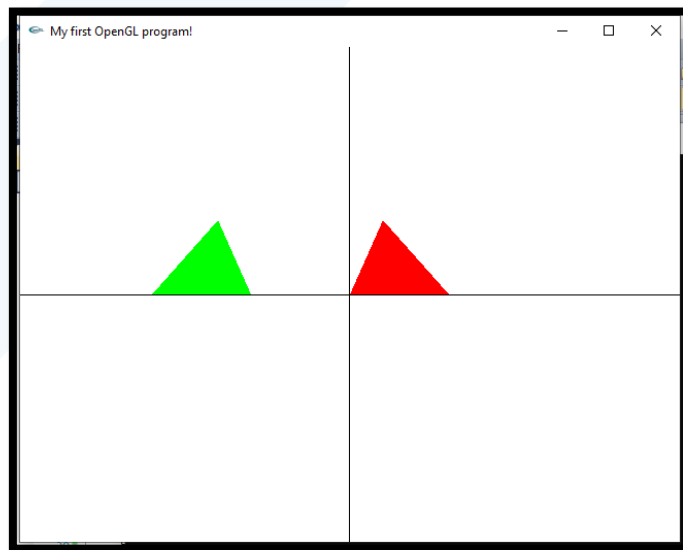
glEnd();

glutSwapBuffers();
}

void main (int argc, char **argv)
{
    initialize the toolkit; glutInit (&argc, argv); // to initial
    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
}

```

عدل ما يلزم للحصول على الشكل التالي:



اجعل الشكل المثلث الأحمر يتحرك نحو الأعلى بمقادير 3.

والمثلث الأخضر يزداد حجماً بمقدار 3.  
ملاحظة:

ابحث عن التوابع `glPopMatrix` و `glPopMatrix`