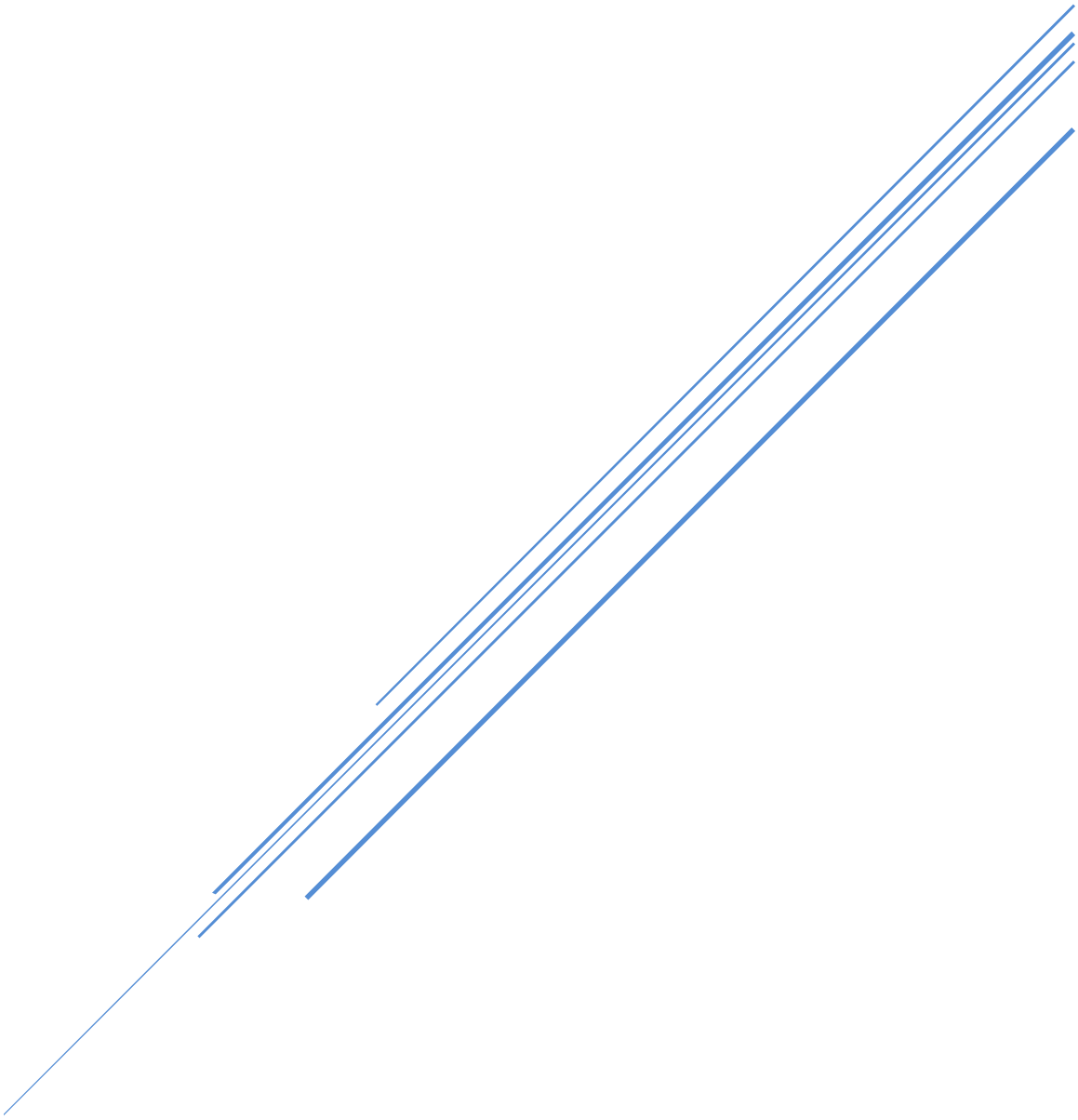
OPEN_GL

By sadek



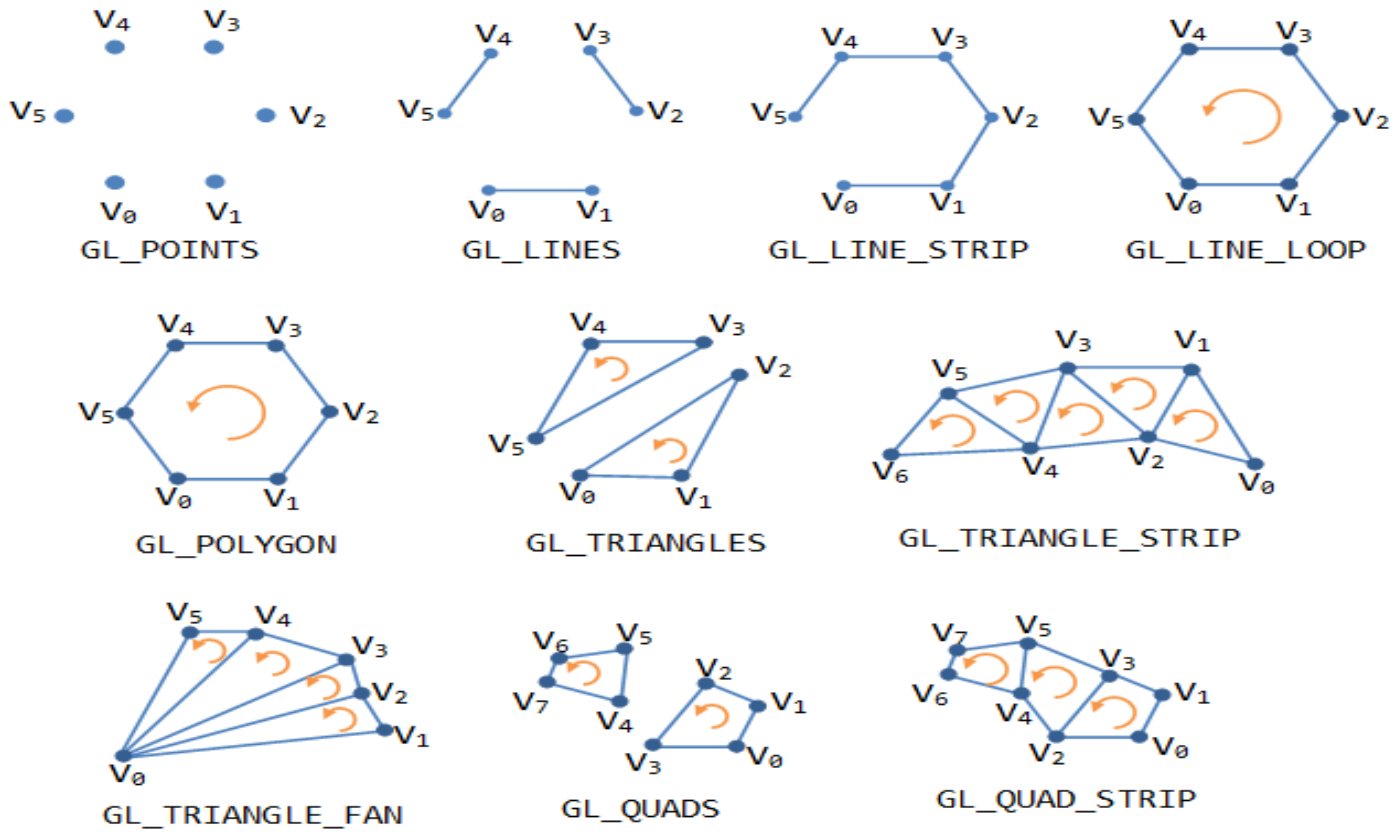
OPEN GL

```
#include <GL/glut.h>
#include <stdlib.h>
#include <iostream>
#include <cmath>
#include <GL/freeglut.h>
using namespace std;



void display (){
   glClearColor(1,1,1,0);
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(1,0,0);
   glPointSize(5);
   glLineWidth(5);

   glFlush();
glutPostRedisplay();
}

void reshape(int width, int height) {
   glViewport(0, 0, width, height);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glOrtho(-width/2, width/2, -height/2, height/2, -1, 1);
   glMatrixMode(GL_MODELVIEW);
}
```

```
int main (int argc, char** argv){
   glutInit(&argc, argv);
glutInitWindowSize(600, 400);
   glutInitWindowPosition(50, 50);
   glutCreateWindow("task1");
   gluOrtho2D(-600,600,-400,400);
   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
   glutMainLoop();
}
```

- لكى نبدا بالرسم يجب ان نضع كود الرسمة بين (glBegin() و glEnd() وهو يعتبر الscoop
- glBegin() تقوم بتحديد ما سيرسم بناء على الargument التى ستاخذه وبسمى الmode على سبيل المثال

mode	Description
GL_POINTS	هذا الmode يستخدم لرسم نقط
GL_LINES , GL_LINE_STRIP , GL_LINE LOOP	يستخدم لرسم خطوط
GL_TRIANGLES , GL_TRIANGLE_STRIP , GL_TRIANGLE_FAN	تستخدم لرسم مثلثات
GL_QUADS , GL_QUAD_STRIP	تستخدم لرسم اشكال رباعية
GL_POLYGON	تستخدم لرسم اى شكل باى عدد اضلاع



OpenGL Primitives

- يمكن استخدام اكثر من glBegin فى نفس الfunction على سبيل المثال واحدة لالنقط وواحدة لالخطوط
- **glvertexNT()** تقوم بوضع نقطة او vertex فى الشكل

glvertexNT ()

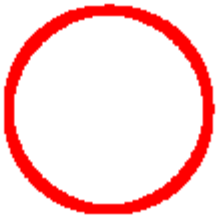
الT هو نوع الArgument الذى سيعطى كا نقط الاحدثيات على سبيل المثال هل هو float ام Int

الN هنا هو عدد الاحدثيات

- من اشكال الglvertex هم **glvertex3f()** و **glvertex2d()** (**الf تعنى float و d تعنى integer**)
- فى حالة اننا اردنا رسم شكل دائرى او بيضاوى او يحتوى على انحناء فانه يفضل استخدام sin و الcos لاتمام هذه العملية

- من امثلة استخدام الsin و الcos هو هذا الكود المستخدم لرسم دائرة :

```
float x,y;
for (float angle=0; angle <90 ; angle+=.1){
   x=100*cos(angle);
   y=100*sin(angle);
   glVertex2f(x,y);
}
```

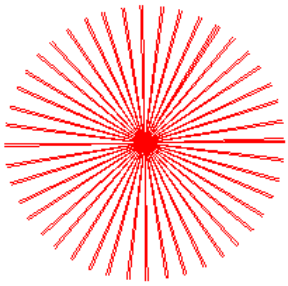


- رسمة اخرى :

```
float x,y;
for (float angle=0; angle <360 ; angle+=1){
   x=20*(16*pow(sin(angle),3));
   y=20*((13*cos(angle))+(-5*cos(2*angle))+(-2*cos(3*angle))+(-cos(4*angle)));
   glVertex2f(x,y);
}
```

- رسمة اخرى

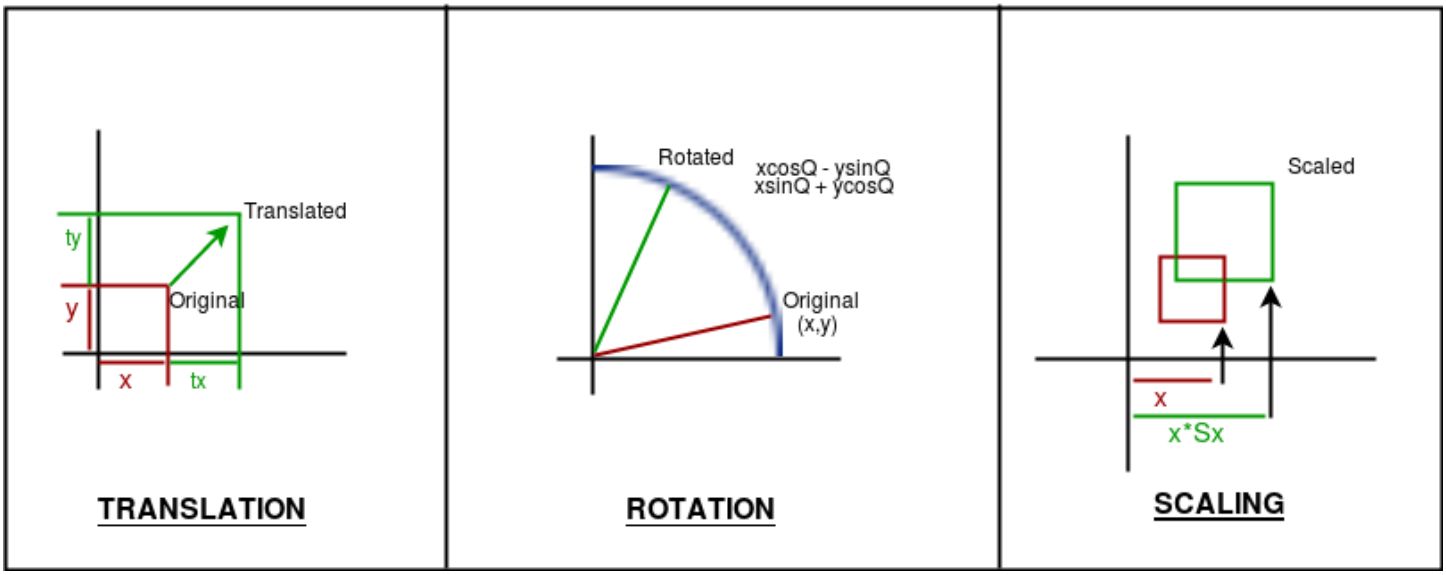
```
float x,y;
for (float angle=0; angle <90 ; angle+=1){
   x=200*cos(angle);
   y=200*sin(angle);
   glVertex2f(0,0);
   glVertex2f(x,y); }
```



- يمكن التحكم فى حجم النقطة او حجم الخط من خلال glPointSize() و glLineWidth()

Transformation

- التحريك بانواعه يقوم على تغير قيم الاحدثيات الخاصة بالشكل او على تغير محاور الاحدثيات بمعنى تغير مكان نقطة الاصل سيقوم بتغير مكان الرسمة كلها
- الميزة فى طريقة تغير محاور الاحدثيات هى انها ستقوم بتغيير جميع النقط دفعة واحدة بدلا من تغييرها نقطة بنقطة
- من خلال الطريقة الثانية نقوم باستخدام مجموعة من الfunction التى تقوم بعمل الtransformation مثل glTranslate**T**(x,y,z) وهى واحد من ثلاث طرق خاصة بالتحريك
- الfunctions الخاصة بالtransformation توضع خارج الglBigen و الglEnd



- الT فى الامر السابق تشير الى نوع القيم الموضوعة كا احدثيات والتى يمكن ان تكون F وتعنى float او d وتعنى integer
- gltranslatef() يقوم بتغير نقطة الاصل الخاصة بالرسمة والتى تؤدى الى تحركها من مكان الى اخر ويمكن اعتبرها انها تقوم باضافة التغيير الى رسمة معينة بمعنى ان مايضاف كا parameter لها هو dx و dy و dz

على سبيل المثال gltranslatef(1,0,0) تعنى انه سيقوم باضافة الى الx الخاص بالرسمة كلها (بجميع نقطها دفعة واحدة) قيمة 1 وبذالك فالنقطة (2,3,5) تصبح (3,3,5) وهكذا مع باقى النقاط

- عند استخدام امر من اوامر الtransformation فيجب استدعاء مصفوفة الوحدة وهى المصفوفة التى نقوم بتغييرها لتحريك الشكل ويتم استدعائها بامر gloadidentity
- لعمل دوران نستخدم امر glRotatet(ceta,x,y,z) اى اننا نستخدم الامر ونضع الزاوية ومحور الدوران او نقطة الدوران
- الدوران بزاوية موجبة يعنى الدوران عكس عقارب الساعة والدوران بزاوية سالبة هو الدوران مع عقارب الساعة
- لكى نرى عملية التحريك بشكل سلس فاننا نضع مكان التغير variable ونقوم بتغيير قيمته بشكل صغير ويجب الانتباه بان يتم تعريف الvariable خارج دالة الرسم
- لعمل scale نستخدم امر glScaled(x,y,z)
- اذا كانت قيم الx و y و z قيم اكبر من الواحد فان الشكل يصبح اكبر واذا كانت اقل من الواحد فان الشكل يصبح اصغر ولتركه كما هو نضعه ب 1

مثال :

```
glLoadIdentity();

   glRotatef(ceta,0,0,1);

   glScaled(ceta*0.001,ceta*0.001,0);

   glBegin(GL_LINES);
```

```
glVertex2f(0.0,0.0);
glVertex2f(-100,0);

glVertex2f(0.0,0.0);
glVertex2f(0,100);

glVertex2f(0.0,0.0);
glVertex2f(100,0);

glVertex2f(0.0,0.0);
glVertex2f(0,-100);
glEnd();
ceta+=0.1;
```

- اختلاف ترتيب الtranslate و الscale و الrotate يؤدى الى نتائج مختلفة

اذا قمنا بالعمل بهؤلاء الطرق فقط فستوجهنا بعض المشاكل مثل ان اى transformation باى من الاشكال السابقة سيحدث لجميع الاشكال وليس لاشكال معينة

- لحل المشكلة السابقة سنقوم باستخدام two functions وهما glPushMatrix() و glPopMatrix() وهم يجعلوا ما بينهم لايؤثر على باقى الكود ويتم استخدامهم كتالى

```
glLoadIdentity();
glRotatef(ceta,0,0,1);              تتم على الكل//
glBegin(GL_LINES);
        glVertex2f(0.0,0.0);
        glVertex2f(-100,0);
        glVertex2f(0.0,0.0);
        glVertex2f(0,-100);
glEnd();
glPushMatrix();
        glScaled(ceta*0.001,ceta*0.001,0);        تتم فقط على من بداخل//
```

```
        glBegin(GL_LINES);
                glVertex2f(0.0,0.0);
                glVertex2f(0,100);
                glVertex2f(0.0,0.0);
                glVertex2f(100,0);
        glEnd();
glPopMatrix();
ceta+=0.1;
```