

كتاب رقم (1B) من سلسلة برمجة الرسم بلغة VC PLUS PLUS باستخدام OpenGL

هذه النسخة بتاريخ: 2007/11/12

مبادئ برمجة الرسم بلغة

Microsoft VC PLUS PLUS 7.0

باستخدام OpenGL

ففي هذا الكتاب يتم استخدام دوال WIN API
لربط النوافذ بمكتبة OpenGL



برمجة: البراء عبد الرؤوف الرملي

طرابلس / ليبيا

نسخة © 2007 , حقوق الطبع محفوظة SBR



Software Bara Ramli (SBR)

لا يسمح بإعادة طبع هذا الكتاب إلا بإذن خطي
مسبق من المؤلف.

بينما يسمح بنسخه و تصويره في نطاق
الاستعمال الشخصي (الغير تجاري) , ولكن لا
يمكنك الادعاء بأنك من قام بهذا العمل
وعليك الإشارة لمؤلفه الأصلي.

ملاحظة: يقدم هذا الكتاب كما هو من دون
أي كفالة أو ضمان لمحتوياته.

All programs in this book is free software:

you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see < <http://www.gnu.org/licenses/>>.

هذا الكتاب مجاني

مقدمة

في هذا الكتاب قمت بجمع تلخيص وشرح لدوال مكتبة الرسم OpenGL, وقد استخدمت دوال WIN API لربط النوافذ بمكتبة OpenGL, وحاولت أن يكون بسيطاً في أسلوبه, أرجو الله أن ينفع به.

وقد أسميته:

مبادئ برمجة الرسم بلغة Microsoft VC PLUS PLUS 7.0
باستخدام OpenGL

وللعلم فقد استفدت كثيراً من موقع "الفريق العربي للبرمجة":
www.arabteam2000-forum.com

ملاحظة: جميع المكتبات والبرامج المعروضة مرفقة مع الكتاب في مجلد
"الملفات".

وأريد أن أنبه على أن الكتب العربية لازالت تحتاج إلى تطوير وإضافات, وهذا يقع على عاتقنا جميعاً حتى تعم الفائدة, لأنه ما لم نتشارك بأفكارنا, فلن نتقدم خطوة إلى الأمام.

البراء عبد الرؤوف الرملي
opencpp@yahoo.com
طرابلس/ليبيا

يمكنك زيارة موقعي: www.khayma.com/opencpp

الفهرس

الموضوع	
عن مكتبة OpenGL	1
ما الذي يلزم للاستخدام مكتبة OpenGL	2
ما الذي يلزم للاستخدام برامج الكتاب	3
الشكل العام لبرامج الرسم	4
مفاهيم أساسية	5
برامج رسم نقطة وخط مستقيم ومثلث ومستطيل	6
طرق الدوران: برنامج لتدوير مثلث	7
المراجع	8

عن مكتبة OpenGL



قابلة للحمل تنفذ على أنظمة مختلفة مثل:

- Microsoft Windows
- Linux
- UNIX-based systems
- Mac OS X
- game consoles by Nintendo and Sony such as the PlayStation 3

مجال الاستخدام هو الرسومات الاحترافية مثل:

- ArchiCAD
- Autodesk (AutoCAD 2000)
- Quake (GL-Quake, Quake II & III)
- Microstation 95
- MathGL3D (Mathematica)
- Maya (Character animation, Modeling)

الحقوق والسماحيات:

- Open Source

ما الذي يلزم لاستخدام مكتبة OpenGL.H

سنعتمد في هذا الكتاب على Visual C++ 7.0 في بيئة .NET 2003 Microsoft Visual Studio
قم بنسخ الملفات إلى المسارات المبينة (الملفات الثلاثة مرفقة مع الكتاب في مجلد المكتبات)

C:\Program Files\Microsoft Visual Studio .NET2003\Vc7\PlatformSDK\Lib

glut32.lib
glaux.lib
opengl32.lib
glu32.lib

C:\Program Files\Microsoft Visual Studio
.NET2003\Vc7\PlatformSDK\Include\gl

glut.h
gl.h
glaux.h
glu.h

c:\windows\system32

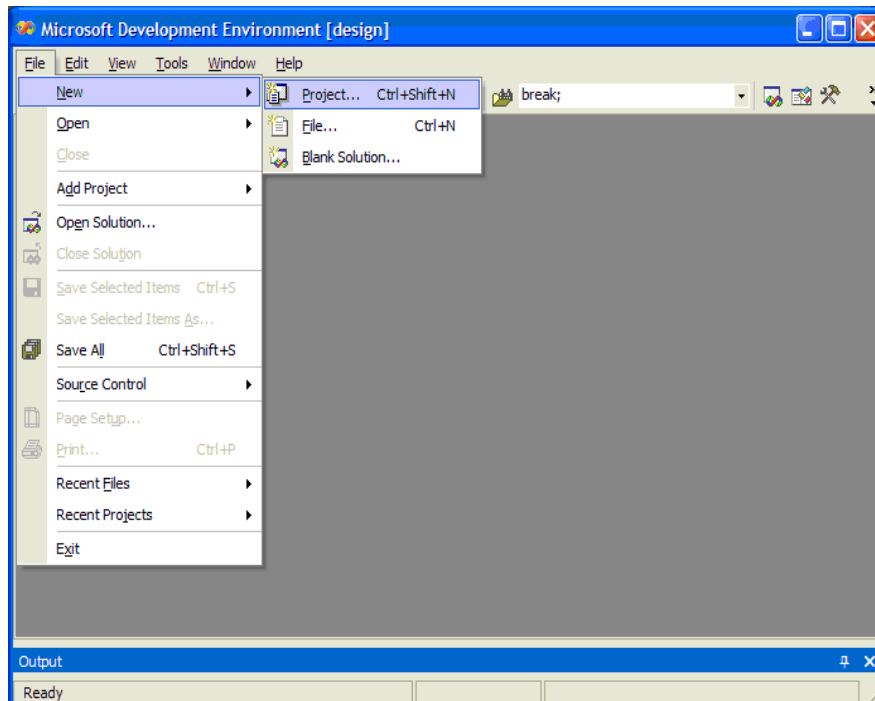
glut32.dll
glu32.dll
opengl32.dll

ما الذي يلزم لاستخدام برامج الكتاب

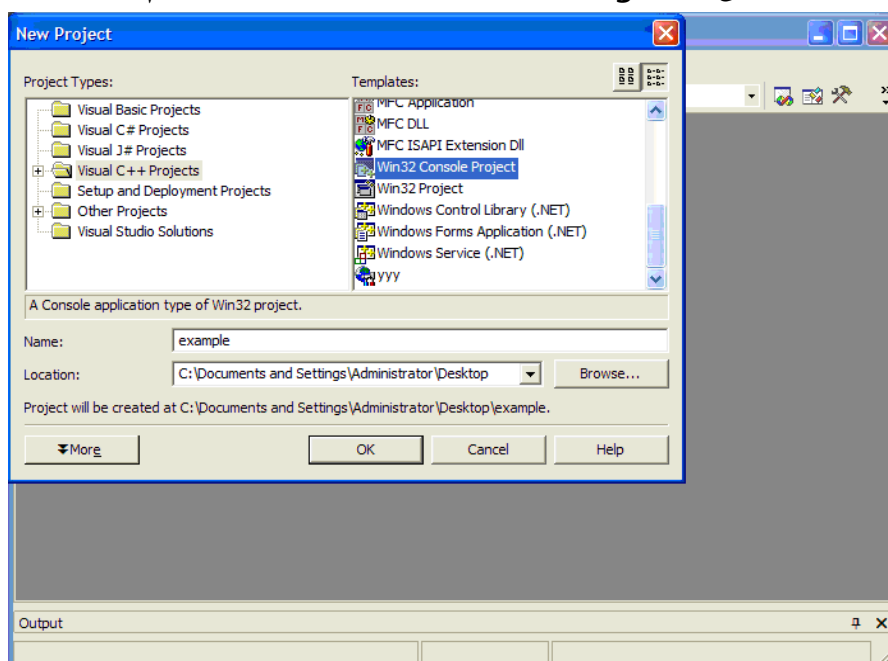
قم بنسخ ملف المكتبة in.h إلى داخل مجلد برنامجك , (الملف مرفق مع الكتاب في مجلد المكتبات)

فإن مشروع جديد في Visual C PLUS PLUS

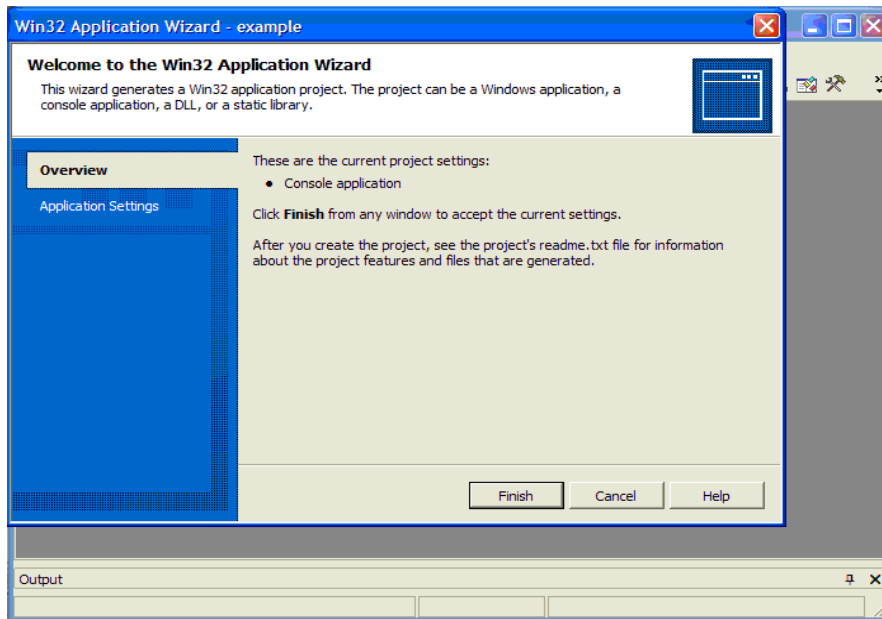
قم بفتح Microsoft Visual Studio .NET 2003 ثم اضغط على
File ثم New ثم Project



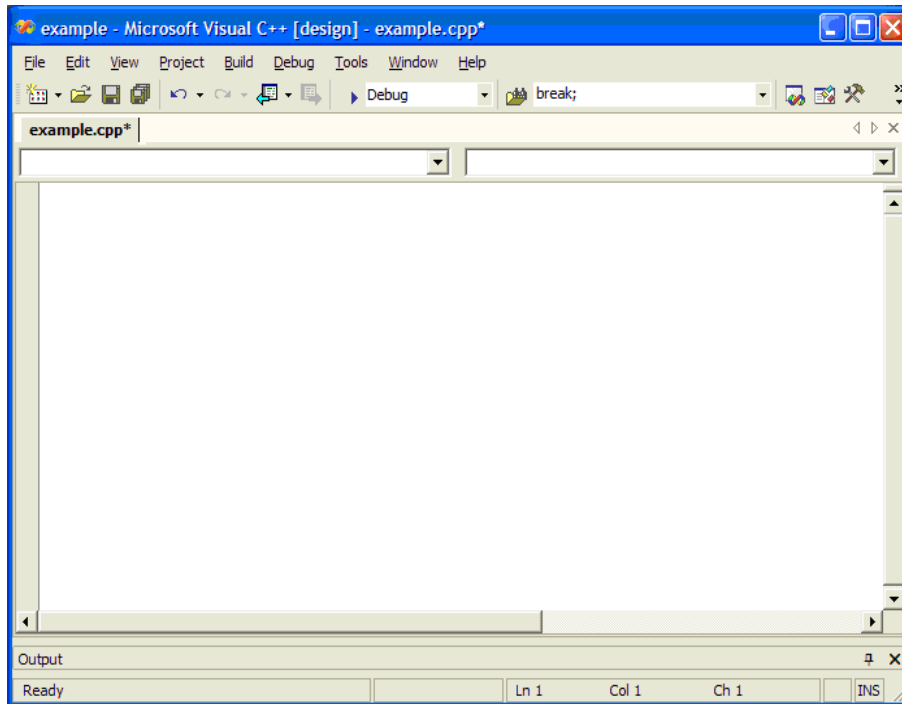
فتظهر لك النافذة السفلية , اختر Visual CPP ثم اكتب اسما للمشروع ثم
اضغط على Win32 Console Project ثم OK



فتظهر لك النافذة السفلية فاضط Finish



فتظهر لك صفحة المشروع , امسح النص الموجود بها لتكتب برنامجك



الشكل العام لبرامج الرسم

#include "stdafx.h"	افتراضية من Microsoft VC
# pragma comment (lib, "opengl32.lib") # pragma comment (lib, "glu32.lib") # pragma comment (lib, "glaux.lib")	ربط مكتبة OpenGL
#include <windows.h> #include <gl\gl.h> #include <gl\glu.h> #include <gl\glaux.h>	استدعاء مكتبة OpenGL
#include "in.h"	مكتبة قمنا بتجهيزها بدوال إعداد بيئة الرسم و نافذة الرسم , كي لا تطول صفحة البرنامج لذا قم بنسخ ملف مكتبة in.h بجانب برنامجك
int DrawGLScene(GLvoid) { glClear(GL_COLOR_BUFFER_BIT GL_DEPTH_BUFFER_BIT); glLoadIdentity(); glTranslatef(-1.5f,0.0f,-6.0f); دوال الرسم نكتب هنا return TRUE; }	دالة بيانات الرسم
int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevIn- stance,LPSTR lpCmdLine,int nCmdShow) { char NAME_WINDOW[]=""; int X_WINDOW=640,Y_WINDOW=480,T=16; MSG msg; BOOL done=FALSE; //if(MessageBox(NULL,"FULL?","SBR",MB_YESNO MB_ICONQUESTION)==IDNO) // { fullscreen=FALSE; // } if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WINDOW,T,fullscreen)) {return 0;} while(!done) { if (PeekMessage(&msg,NULL,0,0,PM_REMOVE)) { if (msg.message==WM_QUIT) {done=TRUE;} else { TranslateMessage(&msg); DispatchMessage(&msg); } } }	الدالة الرئيسية


```

else
{
    if ((active && !DrawGLScene()) || keys[VK_ESCAPE])
        {done=TRUE;}
    else {SwapBuffers(hDC);}
    if (keys[VK_F1])
    {
        keys[VK_F1]=FALSE;
        KillGLWindow();
        fullscreen=!fullscreen;

    if
(!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WINDOW,T,fullscreen))
{return 0;}

    }

}
KillGLWindow();
return (msg.wParam);
}

```

مفاهيم أساسية

أولاً يجب أن تعلم أن نافذة الرسم قسمة إلى 4 أقسام بالنسبة للدالة `glVertex2f` وأبوابها بحيث تكون نقطة الأصل في مركز النافذة كما يظهر

في الشكل التالي:



لرسم الأشكال الأساسية (النقطة-خط مستقيم-مثلث-مستطيل-وغيرها) يلزم 3 دوال:

- دالة لتحديد نوع الشكل المراد رسمه (عدد النقاط)

- دوال لتحديد نقاط الشكل

- دالة للدلالة على انتهاء النقاط

دالة لتحديد نوع الشكل المراد رسمه (عدد النقاط)
glBegin(NAME);

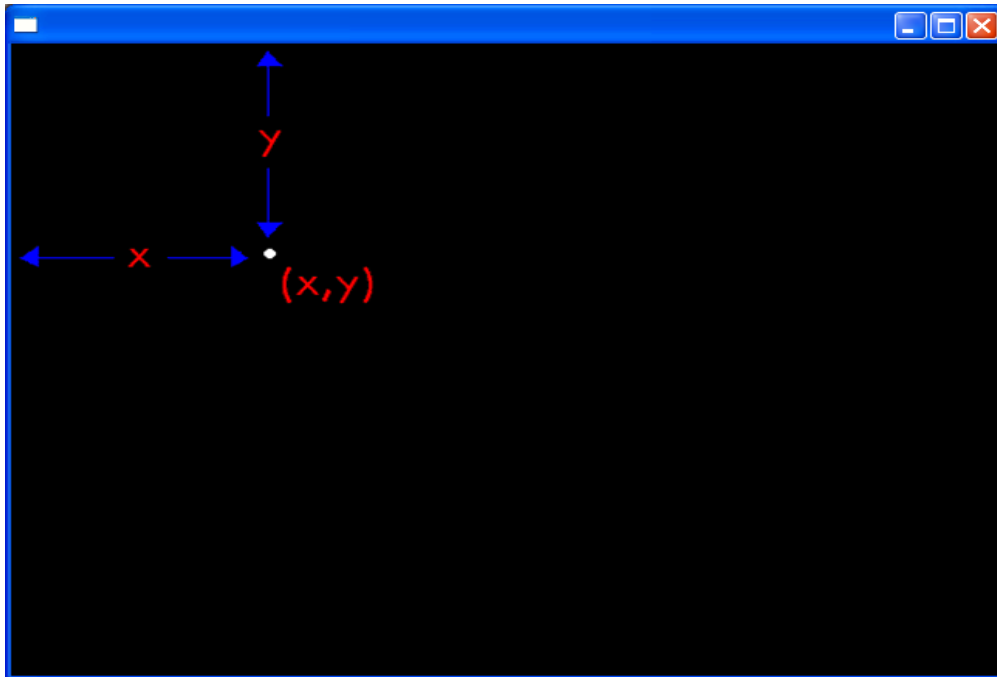
الوصف	NAME
لرسم خط .. وهنا يلزمك نقطتين	GL_LINES
رسم خطوط متصلة	GL_LINE_STRIP
نفس الأمر السابق لكن يتم وصل آخر نقطة بأول نقطة ليتم رسم شكل مغلق	GL_LINE_LOOP
لرسم مثلث .. وتحتاج إلى ثلاث نقاط أو مضاعفات الثلاثة	GL_TRIANGLES
لرسم مثلثات متصلة	GL_TRIANGLE_STRIP
لرسم مثلثات تتصل بنقطة المنتصف (غالباً تستخدم لرسم الأشكال الدائرية)	GL_TRIANGLE_FAN
لرسم أشكال مربعة أو مستطيلة (يعني أشكال ذات أربع رؤوس) وتحتاج إلى أربع نقاط لرسم شكل رباعي واحد .. وثمان نقاط لرسم شكلين رباعيين	GL_QUADS
لرسم أشكال رباعية متصلة	GL_QUAD_STRIP
لرسم مضلع .	GL_POLYGON

دالة لتحديد نقطة ثنائية البعد
glVertex2f(x,y);

أما لتحديد نقطة ثلاثية البعد
glVertex2f(x,y,z);

دالة للدلالة على انتهاء النقاط
glEnd();

برنامج لرسم نقطة



فمثلا لرسم نقطة واحدة يلزم 3 دوال:

glBegin(GL_POINTS);	تحديد نوع الشكل وفي هذه الحالة نختار نقاط منفردة
glVertex2f(x,y);	دالة لتحديد نقطة حيث (x,y) متغيرين صحيحين يمثلان إحداثي النقطة
glEnd();	دالة للدلالة على انتهاء النقاط

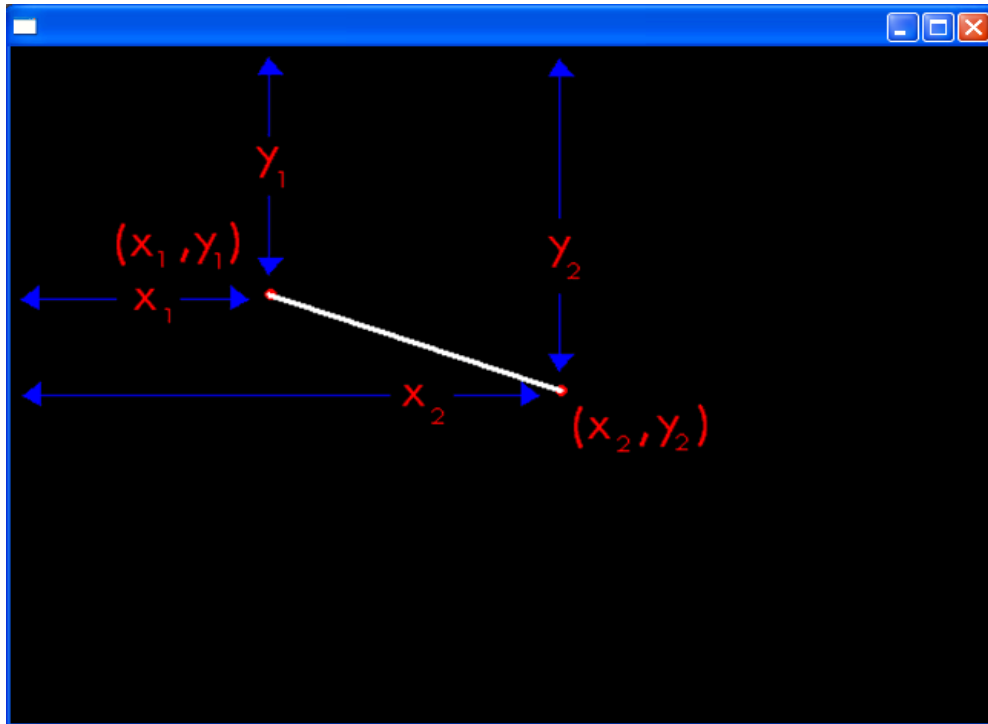
لون الرسم الافتراضي هو الأبيض ما لم يتم تعديله

فقمنا بجمع الـ 3 دوال داخل دالة باسم **point** ومن ثم نقوم باستدعائها:

#include "stdafx.h"	افتراضية من Microsoft VC
# pragma comment (lib, "opengl32.lib") # pragma comment (lib, "glu32.lib") # pragma comment (lib, "glaux.lib")	ربط مكتبة OpenGL
#include <windows.h> #include <gl\gl.h> #include <gl\glu.h> #include <gl\glaux.h>	استدعاء مكتبة OpenGL
#include "in.h"	مكتبة قمنا بتجهيزها بدوال إعداد بيئة الرسم ونافذة الرسم , كي لا تطول صفحة البرنامج , لذا قم بنسخ ملف مكتبة in.h بجانب برنامجك
void point(float x,float y) {	دالة لرسم نقطة وتلوينها
glBegin(GL_POINTS);	تصريح نوع الرسم (نقطي)
glVertex2f(x,y);	لرسم نقطة
glEnd();	إنهاء إدخال النقط

<pre> } int DrawGLScene(GLvoid) { glClear(GL_COLOR_BUFFER_BIT GL_DEPTH_BUFFER_BIT); glLoadIdentity(); glTranslatef(0.0f,0.0f,-1.0f); point(0.0f,0.0f); return TRUE; } </pre>	<p>دالة بيانات الرسم</p>
<pre> int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nCmdShow) { char NAME_WINDOW[]=""; int X_WINDOW=640,Y_WINDOW=480,T=16; MSG msg; BOOL done=FALSE; //if(MessageBox(NULL,"FULL?","SBR",MB_YESNO MB_ICONQUES- TION)==IDNO) // { // fullscreen=FALSE; // } if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WIN- DOW,T,fullscreen)) {return 0;} while(!done) { if (PeekMessage(&msg,NULL,0,0,PM_REMOVE)) { if (msg.message==WM_QUIT) {done=TRUE;} else { TranslateMessage(&msg); DispatchMessage(&msg); } } else { if ((active && !DrawGLScene()) keys[VK_ESCAPE]) {done=TRUE;} else {SwapBuffers(hDC);} if (keys[VK_F1]) { keys[VK_F1]=FALSE; KillGLWindow(); fullscreen=!fullscreen; if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WIN- DOW,T,fullscreen)) {return 0;} } } } KillGLWindow(); return (msg.wParam); } </pre>	<p>الدالة الرئيسية</p>

برنامج لرسم خط مستقيم



رسم مستقيم من الإحداثي $(x1, y1)$ إلى الإحداثي $(x2, y2)$ يلزم 4 دوال:

glBegin(GL_LINES);	نوع الشكل خط مستقيم
glVertex2f(x1,y1); glVertex2f(x2,y2);	لتحديد نقطة بداية ونهاية المستقيم
glEnd();	دالة للدلالة على انتهاء النقاط

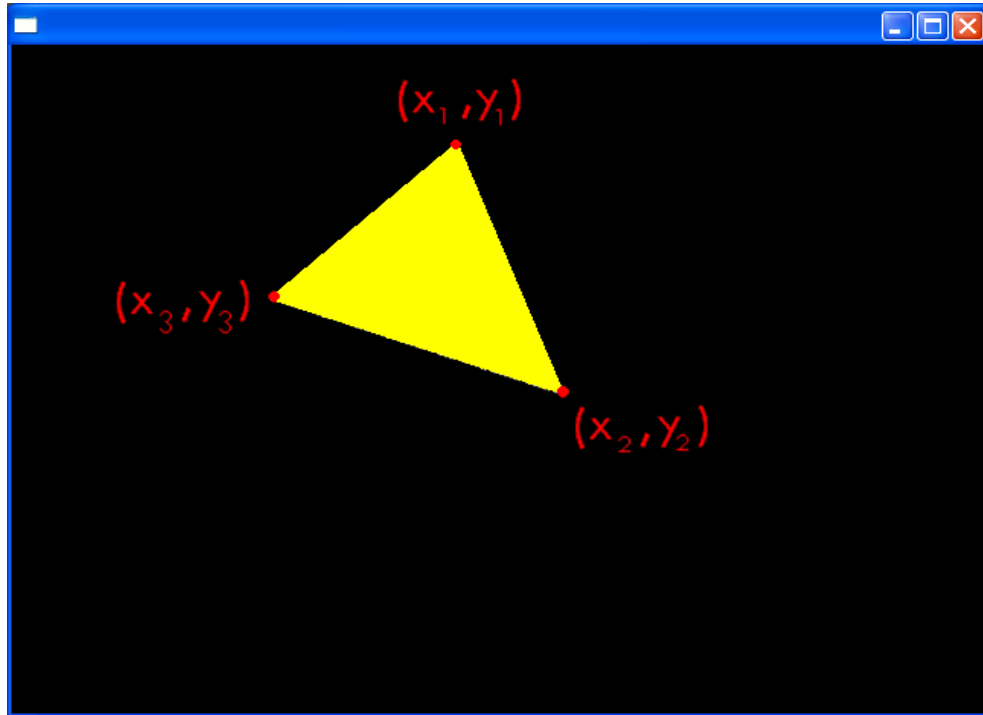
لون الرسم الافتراضي هو الأبيض ما لم يتم تعديله.

فقمنا بجمع الـ 4 دوال داخل دالة باسم **line** ومن ثم نقوم باستدعائها:

#include "stdafx.h"	افتراضية من Microsoft VC
# pragma comment (lib, "opengl32.lib") # pragma comment (lib, "glu32.lib") # pragma comment (lib, "glaux.lib")	ربط مكتبة OpenGL
#include <windows.h> #include <gl\gl.h> #include <gl\glu.h> #include <gl\glaux.h>	استدعاء مكتبة OpenGL
#include "in.h"	مكتبة قمنا بتجهيزها بدوال إعداد بيئة الرسم ونافذة الرسم , كي لا تطول صفحة البرنامج , لذا قم بنسخ ملف مكتبة in.h بجانب برنامجك
void line(float x1,float y1,float x2,float y2) {	دالة لرسم خط مستقيم وتلوينه
glBegin(GL_LINES);	تعيين نوع الرسم (خطي)
glVertex2f(x1,y1);	لتعيين النقطة الأولى

<code>glVertex2f(x2,y2);</code>	لتصوير النقطة الثانية
<code>glEnd();</code> <code>}</code>	إنهاء إدخال النقط
<pre> int DrawGLScene(GLvoid) { glClear(GL_COLOR_BUFFER_BIT GL_DEPTH_BUFFER_BIT); glLoadIdentity(); glTranslatef(0.0f,0.0f,-1.0f); line(0.0f,0.0f,1.0f,1.0f); return TRUE; } </pre>	دالة بيانات الرسم
<pre> int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nCmdShow) { char NAME_WINDOW[]=""; int X_WINDOW=640,Y_WINDOW=480,T=16; MSG msg; BOOL done=FALSE; //if(MessageBox(NULL,"FULL?","SBR",MB_YESNO MB_ICONQUES- TION)==IDNO) // { // fullscreen=FALSE; // } if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WIN- DOW,T,fullscreen)) {return 0;} while(!done) { if (PeekMessage(&msg,NULL,0,0,PM_REMOVE)) { if (msg.message==WM_QUIT) {done=TRUE;} else { TranslateMessage(&msg); DispatchMessage(&msg); } } else { if ((active && !DrawGLScene()) keys[VK_ESCAPE]) {done=TRUE;} else {SwapBuffers(hDC);} if (keys[VK_F1]) { keys[VK_F1]=FALSE; KillGLWindow(); fullscreen=!fullscreen; if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WIN- DOW,T,fullscreen)) {return 0;} } } } KillGLWindow(); return (msg.wParam); } </pre>	الدالة الرئيسية

برنامج لرسم مثلث وتلوينه



رسم مثلث رؤوسه (x_1, y_1) (x_2, y_2) (x_3, y_3) وتلوينه يلزم 6 دوال:

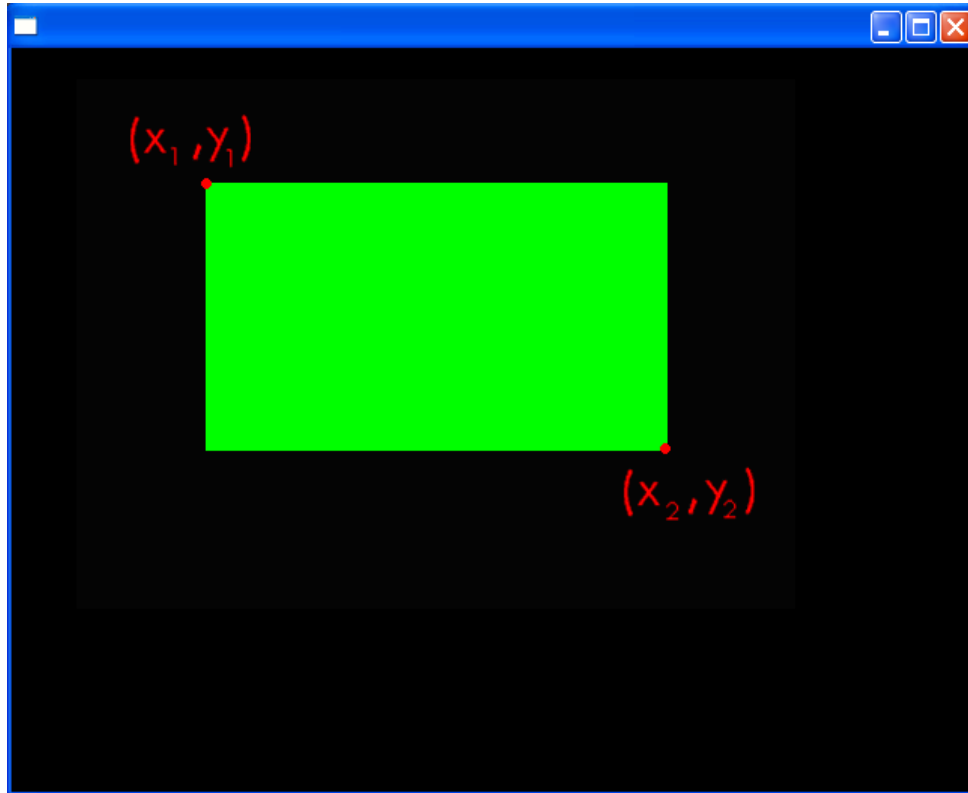
<code>glColor3f(0.7, 0.7, 0.0);</code>	لون المثلث وهو هنا الأصفر
<code>glBegin(GL_TRIANGLES);</code>	نوع الشكل مثلث
<code>glVertex2f(x1,y1);</code> <code>glVertex2f(x2,y2);</code> <code>glVertex2f(x3,y3);</code>	لتحديد إحداثيات رؤوس المثلث
<code>glEnd();</code>	دالة للدلالة على انتهاء النقاط

فقمنا بجمع الـ 5 دوال داخل دالة باسم **triangle** ومن ثم نقوم باستدعائها:

<code>#include "stdafx.h"</code>	افتراضية من Microsoft VC
<code># pragma comment (lib, "opengl32.lib")</code> <code># pragma comment (lib, "glu32.lib")</code> <code># pragma comment (lib, "glaux.lib")</code>	ربط مكتبة OpenGL
<code>#include <windows.h></code> <code>#include <gl\gl.h></code> <code>#include <gl\glu.h></code> <code>#include <gl\glaux.h></code>	استدعاء مكتبة OpenGL
<code>#include "in.h"</code>	مكتبة قمنا بتجهيزها بدوال إعداد بيئة الرسم ونافذة الرسم , كي لا تطول صفحة البرنامج , لذا قم بنسخ ملف مكتبة in.h بجانب برنامجك
<code>void triangle(float x1,float y1,float x2,float y2,float x3,float y3)</code> {	دالة لرسم مثلث وتلوينه
<code>glColor3f(0.7, 0.7, 0.0);</code>	دالة لتعريف لون للرسم
<code>glBegin(GL_TRIANGLES);</code>	تعريف نوع الرسم (مثلثي)

<code>glVertex2f(x1,y1);</code>	لتصوير النقطة الأولى
<code>glVertex2f(x2,y2);</code>	لتصوير النقطة الثانية
<code>glVertex2f(x3,y3);</code>	لتصوير النقطة الثالثة
<code>glEnd();</code> <code>}</code>	إنهاء إكمال النقط
<pre> int DrawGLScene(GLvoid) { glClear(GL_COLOR_BUFFER_BIT GL_DEPTH_BUFFER_BIT); glLoadIdentity(); glTranslatef(0.0f,0.0f,-3.0f); triangle(0.0f,0.0f,1.0f,1.0f,1.0f,0.0f); return TRUE; } </pre>	دالة بيانات الرسم
<pre> int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nCmdShow) { char NAME_WINDOW[]=""; int X_WINDOW=640,Y_WINDOW=480,T=16; MSG msg; BOOL done=FALSE; //if(MessageBox(NULL,"FULL?","SBR",MB_YESNO MB_ICONQUES- TION)==IDNO) // { // fullscreen=FALSE; // } if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WINDOW,T,fullscreen)) {return 0;} while(!done) { if (PeekMessage(&msg,NULL,0,0,PM_REMOVE)) { if (msg.message==WM_QUIT) {done=TRUE;} else { TranslateMessage(&msg); DispatchMessage(&msg); } } else { if ((active && !DrawGLScene()) keys[VK_ESCAPE]) {done=TRUE;} else {SwapBuffers(hdc);} if (keys[VK_F1]) { keys[VK_F1]=FALSE; KillGLWindow(); fullscreen=!fullscreen; if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WINDOW,T,fullscreen)) {return 0;} } } } KillGLWindow(); return (msg.wParam); } </pre>	الدالة الرئيسية

برنامج لرسم مستطيل وتلوينه



لرسم مستطيل ركنيه (x_1, y_1) (x_2, y_2) وتلوينه يلزم 6 دوال:

<code>glColor3f(0.0, 1.0, 0.0);</code>	لون المستطيل وهو هنا الأخضر
<code>glBegin(GL_QUADS);</code>	نوع الشكل مستطيل
<code>glVertex2f(x1,y1);</code> <code>glVertex2f(x2,y1);</code> <code>glVertex2f(x2,y2);</code> <code>glVertex2f(x1,y2);</code>	لتحديد إحداثيات أركان المستطيل
<code>glEnd();</code>	دالة للدلالة على انتهاء النقاط

فقمنا بجمع الـ 5 دوال داخل دالة باسم `rectangle` ومن ثم نقوم باستدعائها:

<code>#include "stdafx.h"</code>	افتراضية من Microsoft VC
<code># pragma comment (lib, "opengl32.lib")</code> <code># pragma comment (lib, "glu32.lib")</code> <code># pragma comment (lib, "glaux.lib")</code>	ربط مكتبة OpenGL
<code>#include <windows.h></code> <code>#include <gl\gl.h></code> <code>#include <gl\glu.h></code> <code>#include <gl\glaux.h></code>	استدعاء مكتبة OpenGL
<code>#include "in.h"</code>	مكتبة قمنا بتجهيزها بدوال إعداد بيئة الرسم ونافذة الرسم , كي لا تطول صفحة البرنامج , لذا قم بنسخ ملف مكتبة <code>in.h</code> بجانب برنامجك
<code>void rectangle(float x1, float y1, float x2, float y2)</code> {	دالة لرسم مثلث وتلوينه
<code>glColor3f(0.0, 1.0, 0.0);</code>	دالة لتعريف لون للرسم
<code>glBegin(GL_QUADS);</code>	تعريف نوع الرسم (مثلثي)
<code>glVertex2f(x1,y1);</code>	لتعريف النقطة الأولى

<code>glVertex2f(x2,y1);</code>	لتصوير النقطة الثانية
<code>glVertex2f(x2,y2);</code>	لتصوير النقطة الثالثة
<code>glVertex2f(x1,y2);</code>	لتصوير النقطة الرابعة
<code>glEnd();</code> <code>}</code>	إنهاء إكمال النقط
<pre> int DrawGLScene(GLvoid) { glClear(GL_COLOR_BUFFER_BIT GL_DEPTH_BUFFER_BIT); glLoadIdentity(); glTranslatef(0.0f,0.0f,-3.0f); rectangle(0.0f,0.0f,1.5f,1.0f); return TRUE; } </pre>	دالة بيانات الرسم
<pre> int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nCmdShow) { char NAME_WINDOW[]=""; int X_WINDOW=640,Y_WINDOW=480,T=16; MSG msg; BOOL done=FALSE; //if(MessageBox(NULL,"FULL?","SBR",MB_YESNO MB_ICONQUES- TION)==IDNO) // { fullscreen=FALSE; // } if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WIN- DOW,T,fullscreen)) {return 0;} while(!done) { if (PeekMessage(&msg,NULL,0,0,PM_REMOVE)) { if (msg.message==WM_QUIT) {done=TRUE;} else { TranslateMessage(&msg); DispatchMessage(&msg); } } else { if ((active && !DrawGLScene()) keys[VK_ESCAPE]) {done=TRUE;} else {SwapBuffers(hDC);} if (keys[VK_F1]) { keys[VK_F1]=FALSE; KillGLWindow(); fullscreen=!fullscreen; if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WIN- DOW,T,fullscreen)) {return 0;} } } } KillGLWindow(); return (msg.wParam); } </pre>	الدالة الرئيسية

طرق تدوير الأشكال

glRotatef(turn,X,Y,Z);

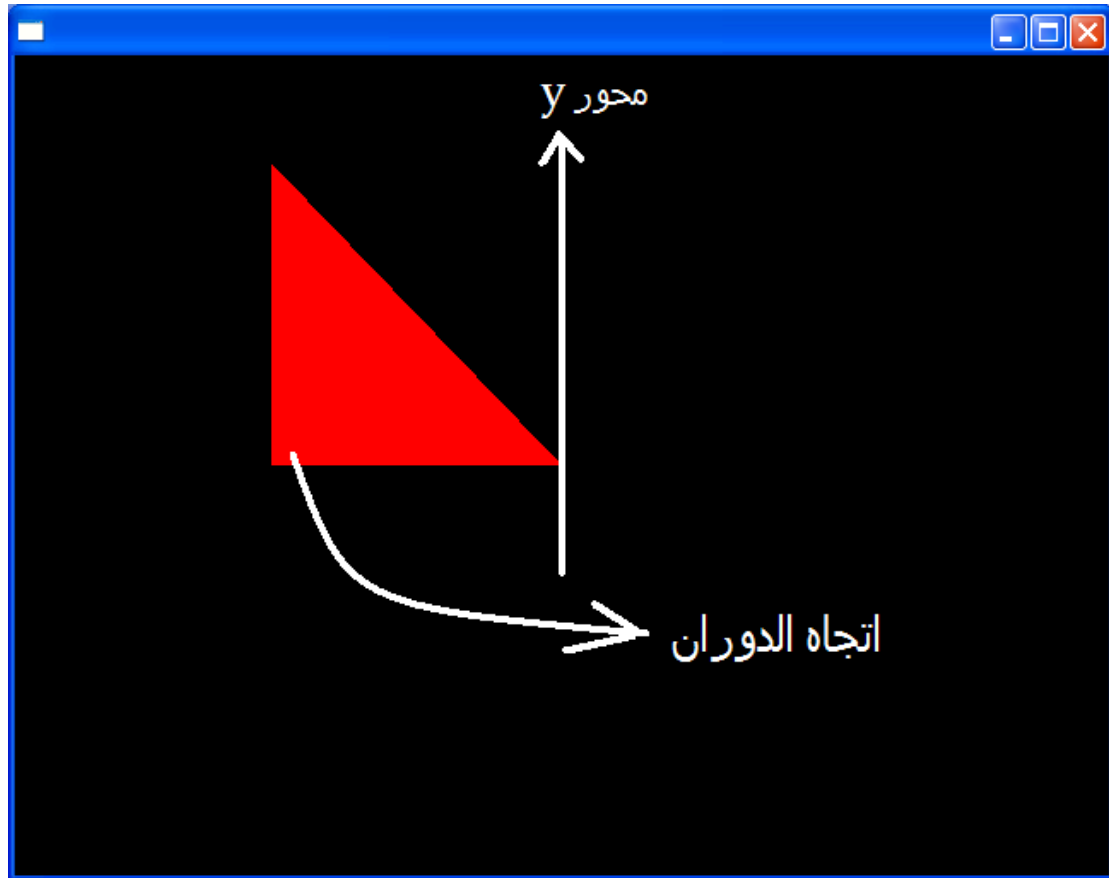
وهي الدالة لتدوير الكائن حول المحاور:

المتغيرات الثلاث تعبر عن كل من المحور X ، والمحور Y ، والمحور Z عند وضع قيمة $X=1$ والباقي أصفار فإن الدوران يتم حول محور X وأما إذا وضعت قيمة $X=-1$ فإن الدوران يتم في اتجاه عكسي حول محور X .
عند وضع قيمة $Y=1$ والباقي أصفار فإن الدوران يتم حول محور Y وأما إذا وضعت قيمة $Y=-1$ فإن الدوران يتم في اتجاه عكسي حول محور Y .
عند وضع قيمة $Z=1$ والباقي أصفار فإن الدوران يتم حول محور Z وأما إذا وضعت قيمة $Z=-1$ فإن الدوران يتم في اتجاه عكسي حول محور Z .

سوف نعرف متغيرين من نوع **GLfloat** هما المتغير t لتدوير المثلث

```
GLfloat t;  
glRotatef(t,0.0f,1.0f,0.0f);  
t+=0.2f;
```

برنامج لندوير مثلث حول محور Y



#include "stdafx.h"	افتراضية من Microsoft VC
# pragma comment (lib, "opengl32.lib") # pragma comment (lib, "glu32.lib") # pragma comment (lib, "glaux.lib")	ربط مكتبة OpenGL
#include <windows.h> #include <gl\gl.h> #include <gl\glu.h> #include <gl\glaux.h>	استدعاء مكتبة OpenGL
#include "in.h"	مكتبة قمنا بتجهيزها بدوال إعداد بيئة الرسم ونافذة الرسم , كي لا تطول صفحة البرنامج , لذا قم بنسخ ملف مكتبة in.h بجانب برنامجك
GLfloat t=0.0f;	
void triangle(float x1,float y1,float x2,float y2,float x3,float y3) {	دالة لرسم مثلث وتلوينه
glColor3f(1.0, 0.0, 0.0);	دالة لتعريف لون للرسم
glBegin(GL_TRIANGLES);	تعريف نوع الرسم (مثلثي)
glVertex2f(x1,y1);	لتعريف النقطة الأولى
glVertex2f(x2,y2);	لتعريف النقطة الثانية
glVertex2f(x3,y3);	لتعريف النقطة الثالثة
glEnd(); }	إنهاء إدخال النقط
int DrawGLScene(GLvoid) { glClear(GL_COLOR_BUFFER_BIT GL_DEPTH_BUFFER_BIT);	دالة بيانات الرسم

<pre> glLoadIdentity(); glTranslatef(0.0f,0.0f,-3.0f); glRotatef(t,0.0,1.0,0.0); triangle(0.0f,0.0f,1.0f,1.0f,1.0f,0.0f); t+=1.0f; return TRUE; } </pre>	
<pre> int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nCmdShow) { char NAME_WINDOW[]=""; int X_WINDOW=640,Y_WINDOW=480,T=16; MSG msg; BOOL done=FALSE; //if(MessageBox(NULL,"FULL?","SBR",MB_YESNO MB_ICONQUES- TION)==IDNO) // { fullscreen=FALSE; // } if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WINDOW,T,ful lScreen)) {return 0;} while(!done) { if (PeekMessage(&msg,NULL,0,0,PM_REMOVE)) { if (msg.message==WM_QUIT) {done=TRUE;} else { TranslateMessage(&msg); DispatchMessage(&msg); } } else { if ((active && !DrawGLScene()) keys[VK_ES- CAPE]) {done=TRUE;} else {SwapBuffers(hDC);} if (keys[VK_F1]) { keys[VK_F1]=FALSE; KillGLWindow(); fullscreen=!fullscreen; } if (!CreateGLWindow(NAME_WINDOW,X_WINDOW,Y_WINDOW,T,ful lScreen)) {return 0;} } } KillGLWindow(); return (msg.wParam); } </pre>	<p>الحالة الرئيسية</p>

المراجع

المصدر	الصفة
الفريق العربي للبرمجة بقسم برمجة الألعاب والجرافيكس باستخدام مكتبة OpenGL	مشاركات الشمري - مشرف قسم برمجة الألعاب والجرافيكس
	مشاركات: CompuM4n - عضو
	دروس OpenGL المترجمة: نعلي دعيح و Basha
جامعة حلب - الهندسة الإلكترونية	كتاب إلكتروني