

## مبادئ الرسم باستخدام OpenGL

### 1 مفردات الجلسة:

- ✓ التعليمات الأساسية الخاصة بإعدادات الرسم باستخدام OpenGL
- ✓ رسم الأغراض الأساسية
- ✓ تمرين للتدريب

### 2 التعليمات الأساسية الخاصة بإعدادات الرسم باستخدام OpenGL:

يوحد العديد من التعليمات التي تستخدم لإعداد لوحة الرسم باستخدام OpenGL والتي تتناسب مع خصائص اللوحة المراد رسمها سواء من حيث انها ثنائية أم ثلاثية البعد والألوان والخلفية وغيرها نذكر منها:

- ✓ تغيير لون الخلفية: `glClearColor(R,G,B,A)`، حيث تعبر المتغيرات R, G, B عن الألوان الأحمر، الأخضر والأزرق على التوالي والتي تأخذ قيمها بين الصفر والواحد
- ✓ تغيير لون خط الرسم: `glColor3f(R,G,B)`
- ✓ تغيير حجم خط الرسم: `glPointSize(1.0)`
- ✓ تحديد أي مصفوفة هي المصفوفة الحالية المستخدمة: `glMatrixMode(GLenum mode)`
  - تطبيق عمليات مصفوفات التحويل المتتالية على نموذج مكس العرض `GL_MODELVIEW`
  - تطبيق عمليات مصفوفات التحويل المتتالية على نموذج مكس الإسقاط `GL_PROJECTION`
  - تطبيق عمليات مصفوفات التحويل المتتالية على نموذج مكس التركيب `GL_TEXTURE`
  - تطبيق عمليات مصفوفات التحويل المتتالية على نموذج مكس الألوان `GL_COLOR`
- ✓ استبدال المصفوفة الحالية بالمصفوفة المنجزة: `glLoadIdentity()`
- ✓ اعداد وتحديد مصفوفة ثنائية البعد للعرض `gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)`؛
- ✓ يفضل استخدام تابع مستقل للرسم
- ✓ يتم تحديد بداية ونهاية رسم أي غرض بالتعليمة التالية: `glEnd` ..... `glBegin(GLenum mode)` والتي تقوم بتحديد نقاط الغرض المراد رسمه بالإضافة إلى نوع الغرض لسهولة الرسم
- ✓ يتم تحديد احداثيات نقاط الأغراض بالتعليمة التالية: `glVertex2f(x, y)`, `glVertex2i(x, y)` وهنا في مجال الاحداثيات ثنائية البعد، حيث تستخدم الأولى قيمة عددية صحيحة والثانية حقيقية
- ✓ يتم تصفير ال Buffers وتتهيئته بقيم ابتدائية بالتعليمة التالية: `glClear(GLbitfield mask)` هنا يمكن الجمع بين أكثر من mask عن طريق عملية ال OR بينهم
- ✓ يوجد أربعة أقنعة mask:

- GL\_COLOR\_BUFFER\_BIT ➤
- GL\_DEPTH\_BUFFER\_BIT ➤
- GL\_ACCUM\_BUFFER\_BIT ➤
- GL\_STENCIL\_BUFFER\_BIT ➤

✓ يتم فرض تنفيذ أوامر الـ GL في وقت محدد بالتعليمة التالية: glFlush();

### 3 رسم الأغراض الأساسية:

يوجد العديد من الأغراض التي يمكن استخدامها في عملية الرسم

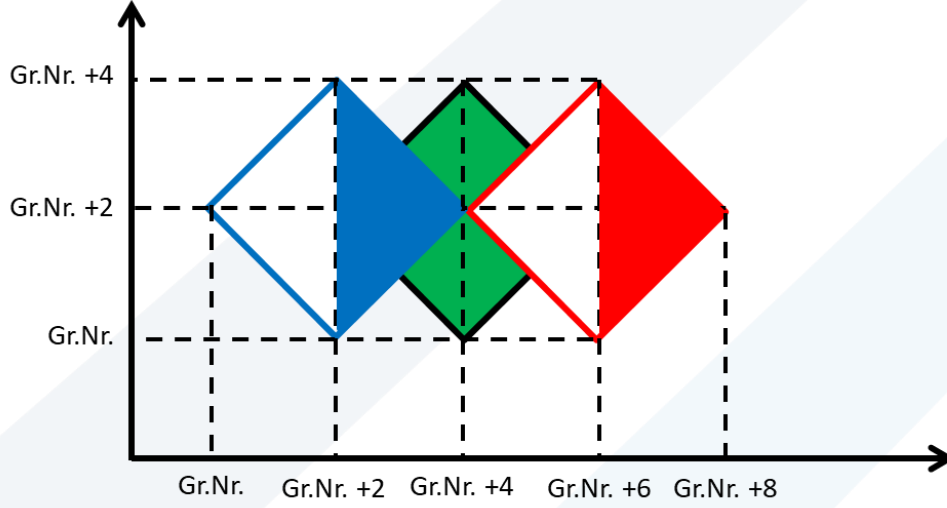
الوصف	NAME
لرسم خط .. ولها يلزمه نقطتين	GL_LINES
رسم خطوط متصلة	GL_LINE_STRIP
نفس الأمر السابق لكن يتم وصل آخر نقطة بأول نقطة ليتم رسم شكل مغلق	GL_LINE_LOOP
لرسم مثلث .. وتحتاج إلى ثلاث نقاط أو مضاعفات الثلاثة	GL_TRIANGLES
لرسم مثلثات متصلة	GL_TRIANGLE_STRIP
لرسم مثلثات تتصل بنقطة المنتصف (غالباً تستخدم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

✓ وبالتالي تصبح بنية وهيكلية برنامج الرسم على الشكل التالي:

<pre>void MyInit() { glClearColor(0.3,0.8,1.0,0.0); glColor3f(1.0,1.0,1.0); glPointSize(1.0); glMatrixMode(GL_PROJECTION); glLoadIdentity(); gluOrtho2D(0 , 700 , 0 , 600); }</pre>	إعداد بيئة الرسم
<pre>void graph() { glClear(GL_COLOR_BUFFER_BIT); // دوال الرسم نكتب هنا glFlush(); }</pre>	دالة بيانات الرسم

#### 4 تمرين للتدريب:

المطلوب كتابة برنامج بلغة الـ C++ يقوم برسم الشكل التالي باستخدام مكتبات الـ OpenGL



المطلوب:

- 1- تغيير خلفية اللوحة
- 2- تغيير حجم اللوحة
- 3- تغيير الألوان وعرض خط الرسم
- 4- تغيير مصفوفة الاستخدام