

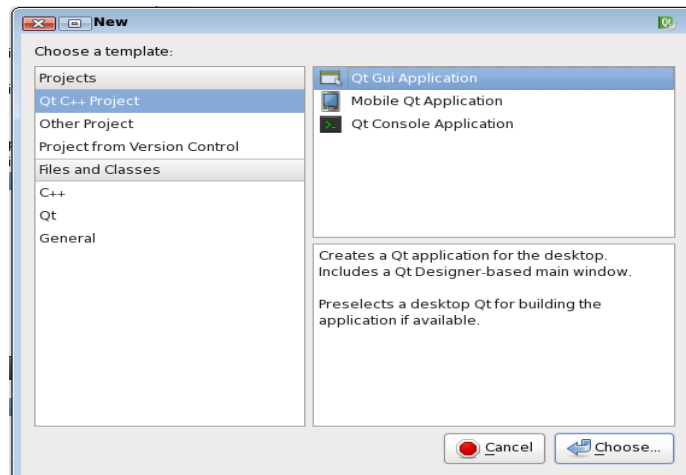
بسم الله الرحمن الرحيم و السلام على أشرف المرسلين محمد رسول
الله صلى الله عليه و سلم
و السلام عليكم

تمثل هذه الوثيقة محاولة تطبيق أدوات qt و opengl معا لإنتاج تطبيق بسيط جدا و
بأسهل الطرق المتوفرة في عالم الواجهات الرسومية.
التطبيق عبارة عن ثلاثة مربعات تختلف في العمق و اللون بحيث يقوم المستعمل
بالنقر على أحد المربعات الثلاثة ليتغير لونه إلى اللون الأحمر , و سيتم إعداد هذا
التطبيق باستعمال الواجهة الرسومية QT Gui
و سيثبت الشرح فقط على المفاهيم الخاصة بال QT لدعم الأوبن جي أل.
بالاعتماد على الصور فأنا أضمن أنه سيكون الشرح واضح جدا

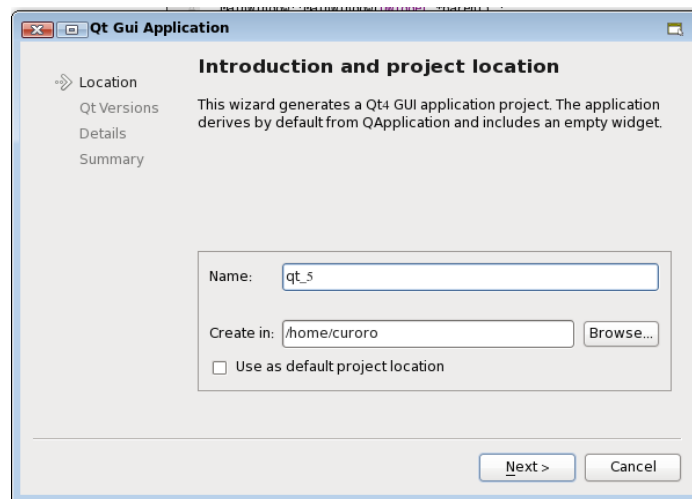
إصدار الكيوتي المستخدمة في هذا التطبيق هي
qt-sdk-linux-x86-opensource-2010.04.bin
يمكن تحميلها من الموقع الرسمي لكيوتي

نظام التشغيل المستخدم هو أعجوبة 4
على فكرة, إن QT مدعوم بشكل كامل من قبل KDE من ناحية الواجهة الرسومية

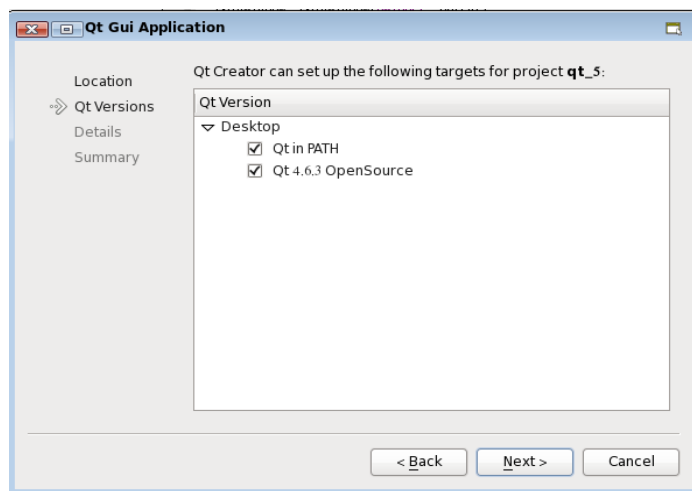
و لنبدأ على بركة الله :
نذهب إلى : File ثم New File or Project ... ثم Qt Gui Application



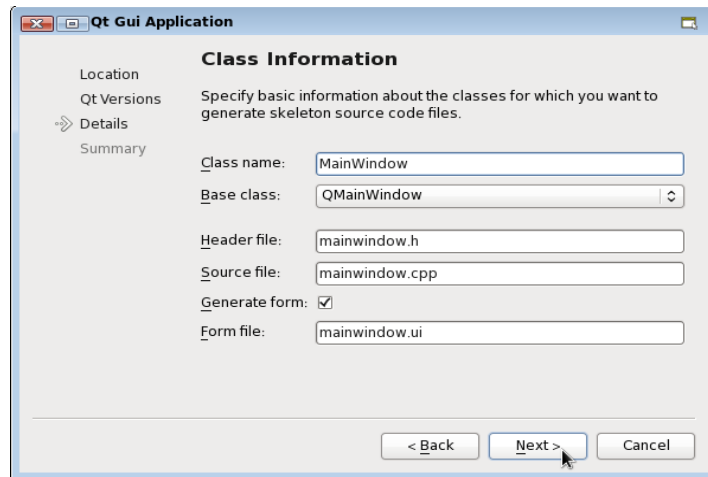
و نختار Qt Gui Application و من ثم choose



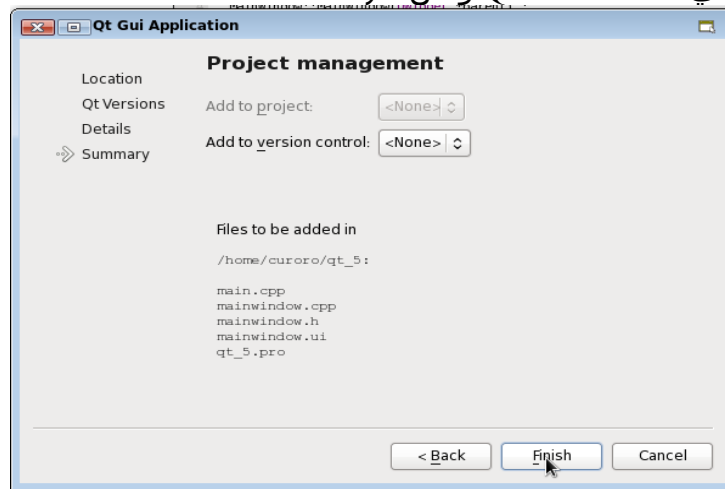
نسمي للمشروع و من ثم Next



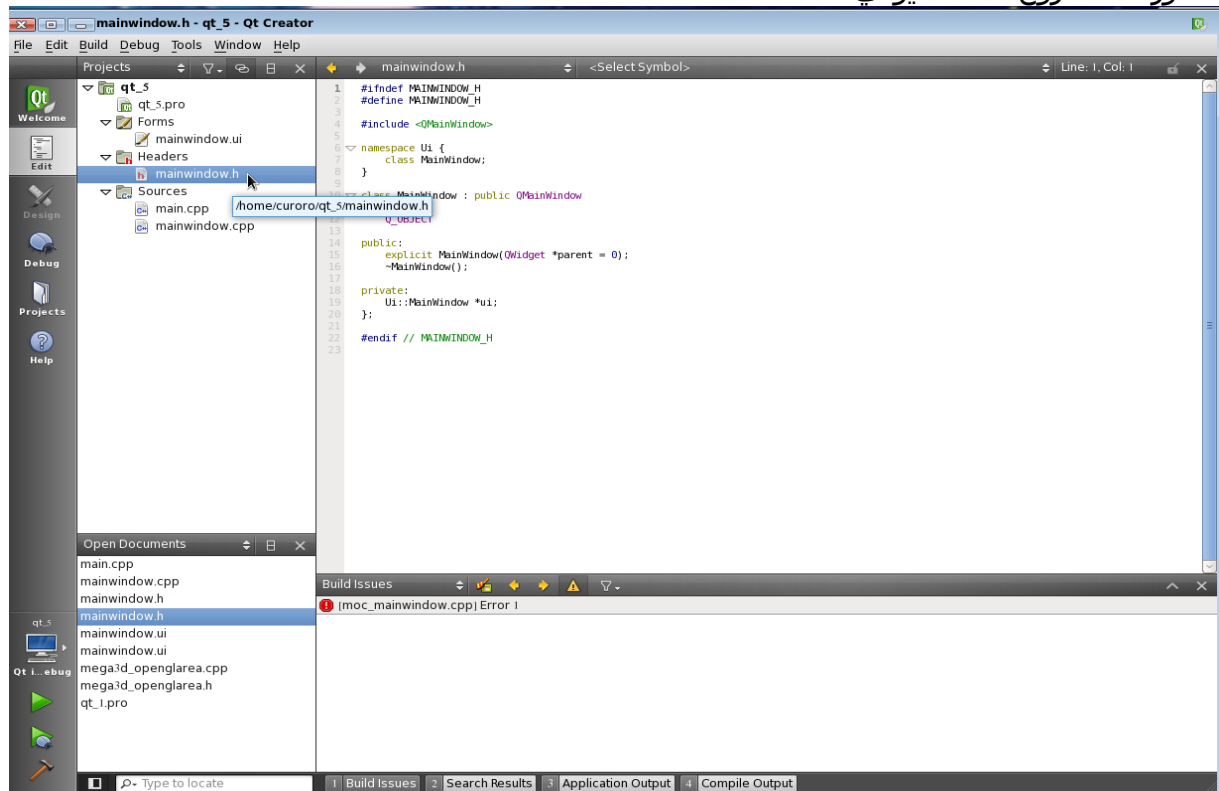
و من ثم Next



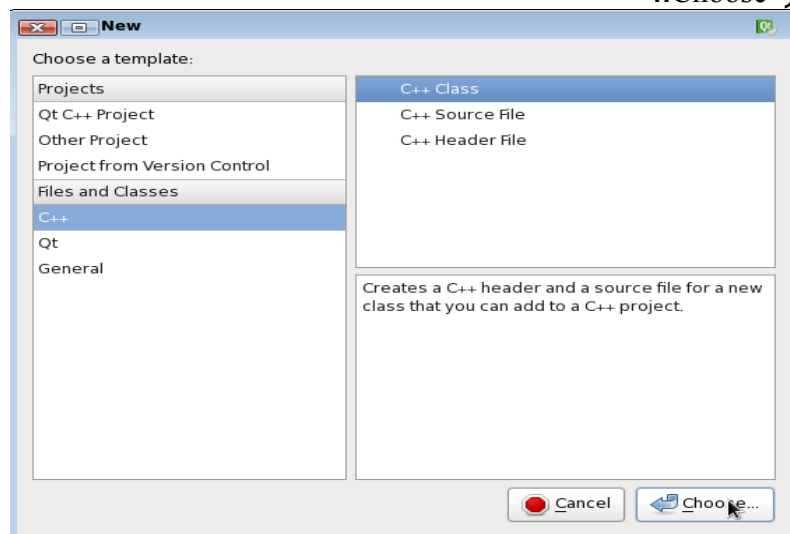
تأكد من أن Base class هي QMainWindow و من ثم Next



و من ثم Finish
و هذه صورة لمشروع نافذة كيوتي :

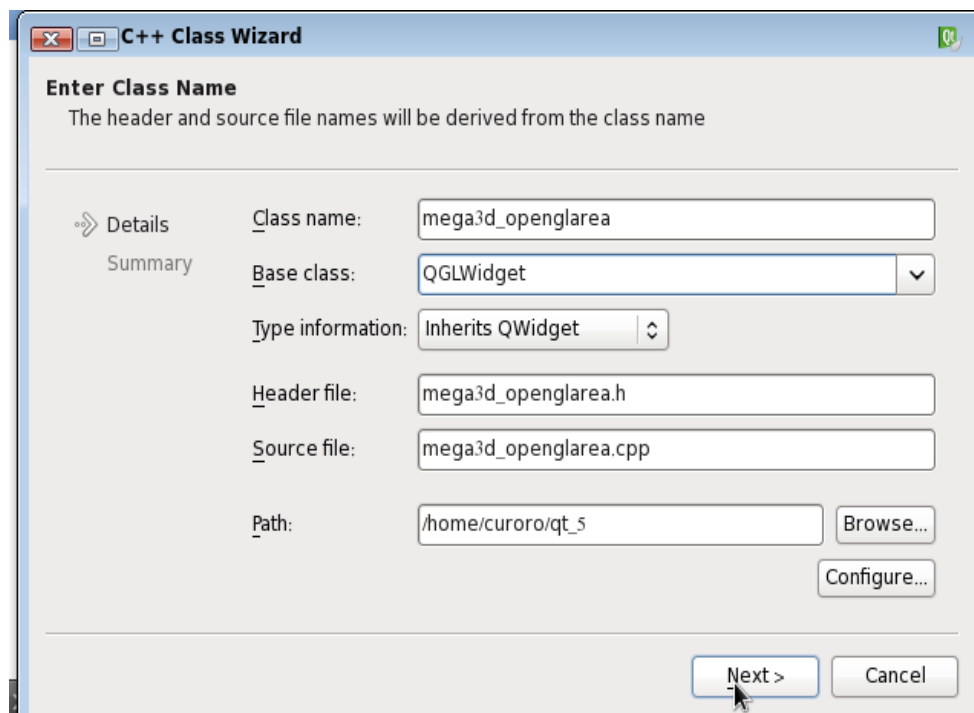


و الآن نضيف فئة للمشروع و نسميها mega3d_openglarea هي تسمية للفئة ليس إلا,
فنذهب إلى File و من ثم New File or Project .. و من ثم ++C من وحدة Files and Classes و من ثم
نحدد C++ Class و ثم Choose..



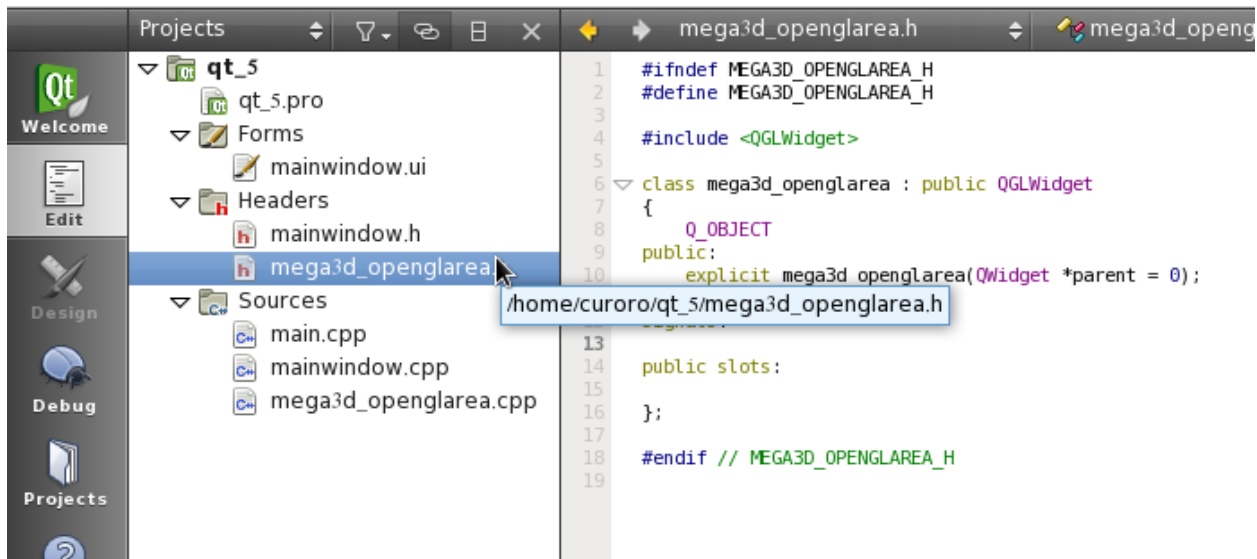
و من ثم نقوم بالإعدادات التالية :

Class name: mega3d_openglarea يمثل إسم الفئة أو الكلاس,
Base class: QWidget يمثل إسم الكلاس الذي سيرث منها بعض الأدوات,
Type information: inherits QWidget يمثل الكلاس الأم لل QWidget.
path هو المجلد الحاوي على المشروع,



و من ثم Next و من ثم Finish.

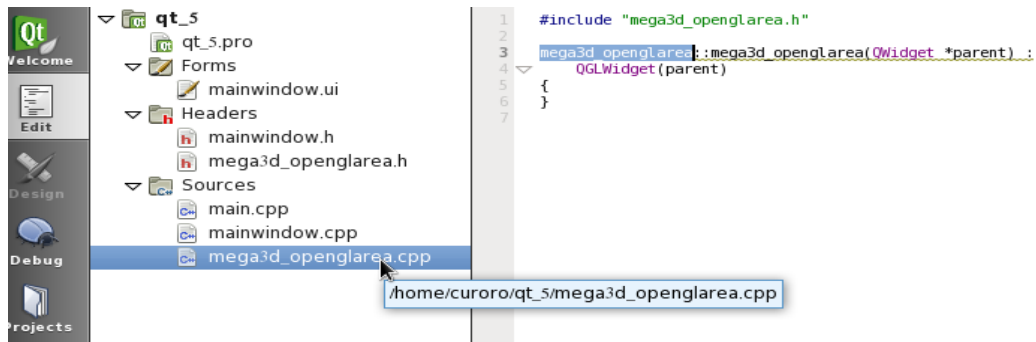
و الآن نذهب إلى فئتنا الرأسية الجديدة mega3d_openglarea.h :



ثم نقوم باستبدال الشيفرة الافتراضية بالشيفرة التالية :

```
1  #ifndef MEGA3D_OPENGLAREA_H
2  #define MEGA3D_OPENGLAREA_H
3
4  #include <QGLWidget>
5  #include <QtGui/QMouseEvent>
6  #include <stdio.h>
7
8
9  class mega3d_openglarea : public QGLWidget
10 {
11     Q_OBJECT
12 public:
13     explicit mega3d_openglarea(QWidget *parent = 0);
14
15 protected:
16     void initializeGL();
17     void resizeGL(int w, int h);
18     void paintGL();
19     void mousePressEvent(QMouseEvent *event);
20     void mouseMoveEvent(QMouseEvent *event);
21     void keyPressEvent(QKeyEvent* event);
22     void drawRects(GLenum mode);
23     void setname(int i) { nameobject = i; }
24     int getname() { return nameobject; }
25
26 signals:
27
28 public slots:
29
30 private:
31     int nameobject;
32     void processHits(GLint hits, GLuint buffer[]);
33
34 };
35
36 #endif // MEGA3D_OPENGLAREA_H
37
```

و مرة ثانية نذهب إلى فئتنا المصدرية الجديدة mega3d_openglarea.cpp :



و من ثم نقوم باستبدال الشيفرة الافتراضية بالشيفرة التالية :

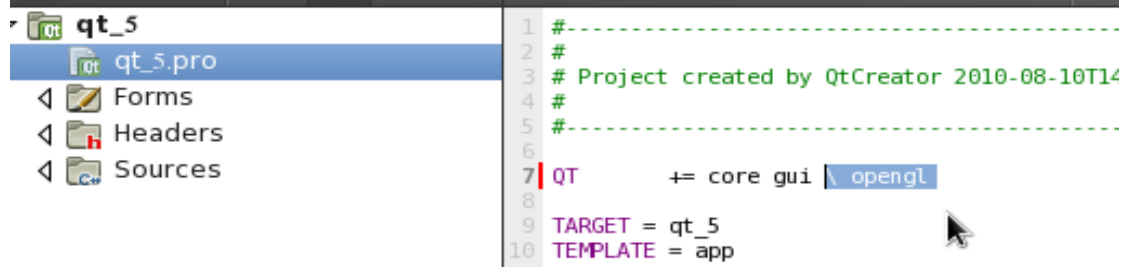
```
1 #include "mega3d_openglarea.h"
2
3 mega3d_openglarea::mega3d_openglarea(QWidget *parent) :
4     QWidget(parent)
5 {
6     nameobject = 0;
7 }
8 void mega3d_openglarea::initializeGL() {
9     glClearColor(0.0, 0.0, 0.0, 0.0);
10    glEnable(GL_DEPTH_TEST);
11    glShadeModel(GL_FLAT);
12    glDepthRange(0.0, 1.0);
13 }
14 void mega3d_openglarea::resizeGL(int w, int h) {
15     glViewport(0, 0, (GLsizei) w, (GLsizei) h);
16     glMatrixMode(GL_PROJECTION);
17     glLoadIdentity();
18     glOrtho(0.0, 8.0, 0.0, 8.0, -0.5, 2.5);
19     glMatrixMode(GL_MODELVIEW);
20     glLoadIdentity();
21 }
22 void mega3d_openglarea::paintGL() {
23     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
24     drawRects(GL_RENDER);
25     glFlush();
26 }
27 void mega3d_openglarea::mousePressEvent(QMouseEvent *event) {
28     GLuint selectBuf[512];
29     GLint hits;
30     GLint viewport[4];
31
32     if (event->buttons() & Qt::RightButton)
33         return;
34
35     glGetIntegerv(GL_VIEWPORT, viewport);
36
37     glSelectBuffer(512, selectBuf);
38     (void) glRenderMode(GL_SELECT);
39
40     glInitNames();
41     glPushName(0);
42
43     glMatrixMode(GL_PROJECTION);
44     glPushMatrix();
45     glLoadIdentity();
46     /* create 5x5 pixel picking region near cursor location */
47     gluPickMatrix((GLdouble) event->x(), (GLdouble) (viewport[3] - event->y()),
48                 5.0, 5.0, viewport);
49     glOrtho(0.0, 8.0, 0.0, 8.0, -0.5, 2.5);
50     drawRects(GL_SELECT);
51     glPopMatrix();
52     glFlush();
53
54     hits = glRenderMode(GL_RENDER);
55     processHits(hits, selectBuf);
56 }
57
58 void mega3d_openglarea::mouseMoveEvent(QMouseEvent *event) {
59     printf("%d, %d\n", event->x(), event->y());
60 }
61
```

```

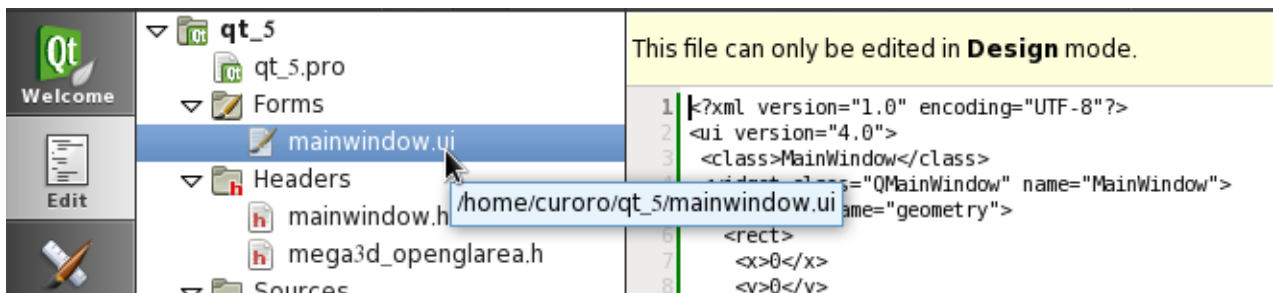
62 void mega3d_openglarea::keyPressEvent(QKeyEvent* event) {
63     switch(event->key()) {
64         case Qt::Key_Escape:
65             close();
66             break;
67         default:
68             event->ignore();
69             break;
70     }
71 }
72 void mega3d_openglarea::processHits(GLint hits, GLuint buffer[])
73 {
74     GLuint *ptr;
75     ptr = (GLuint *) buffer;
76     if(hits == 0) setname(0); ;
77     if(hits!=0)
78     {
79         ptr+=((3+(*ptr)))-1;
80         printf("    the name is %d",*ptr);
81         setname(*ptr);
82         printf("\n");
83     }
84
85     updateGL();
86 }
87 void mega3d_openglarea::drawRects(GLenum mode)
88 {
89     if (mode == GL_SELECT)
90         glLoadName(1);
91     glBegin(GL_QUADS);
92     glColor3f(1.0, 1.0, 0.0);
93     if(getname() == 1) glColor3f(1.0, 0.0, 0.0);
94     glVertex3i(2, 0, 0);
95     glVertex3i(2, 6, 0);
96     glVertex3i(6, 6, 0);
97     glVertex3i(6, 0, 0);
98     glEnd();
99     if (mode == GL_SELECT)
100         glLoadName(2);
101     glBegin(GL_QUADS);
102     glColor3f(0.0, 1.0, 1.0);
103     if(getname() == 2) glColor3f(1.0, 0.0, 0.0);
104     glVertex3i(3, 2, -1);
105     glVertex3i(3, 8, -1);
106     glVertex3i(8, 8, -1);
107     glVertex3i(8, 2, -1);
108     glEnd();
109     if (mode == GL_SELECT)
110         glLoadName(3);
111     glBegin(GL_QUADS);
112     glColor3f(1.0, 0.0, 1.0);
113     if(getname() == 3) glColor3f(1.0, 0.0, 0.0);
114     glVertex3i(0, 2, -2);
115     glVertex3i(0, 7, -2);
116     glVertex3i(5, 7, -2);
117     glVertex3i(5, 2, -2);
118     glEnd();
119 }
120

```

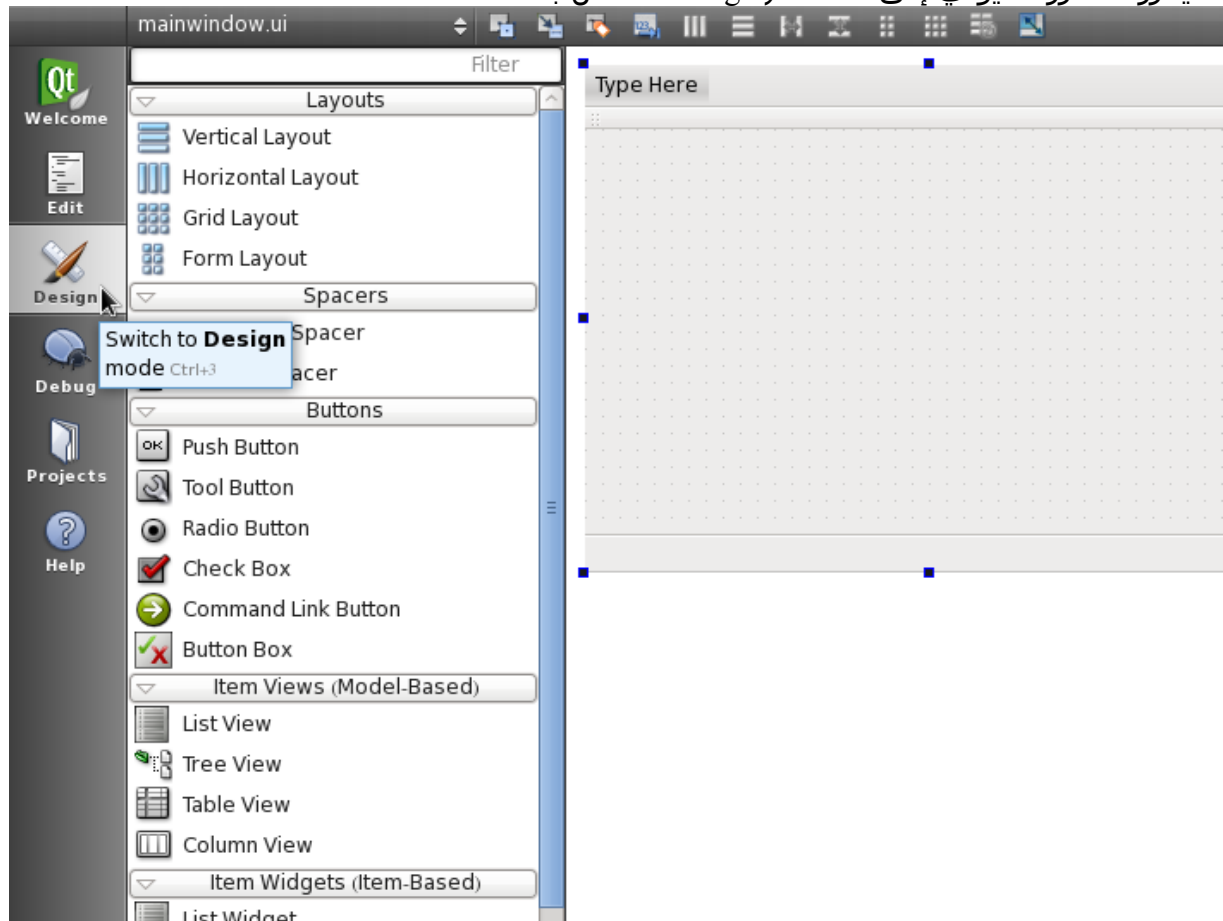
ثم نذهب إلى qt 5,pro و من ثم نضيف opengl \ قرب += core gui QT هكذا :



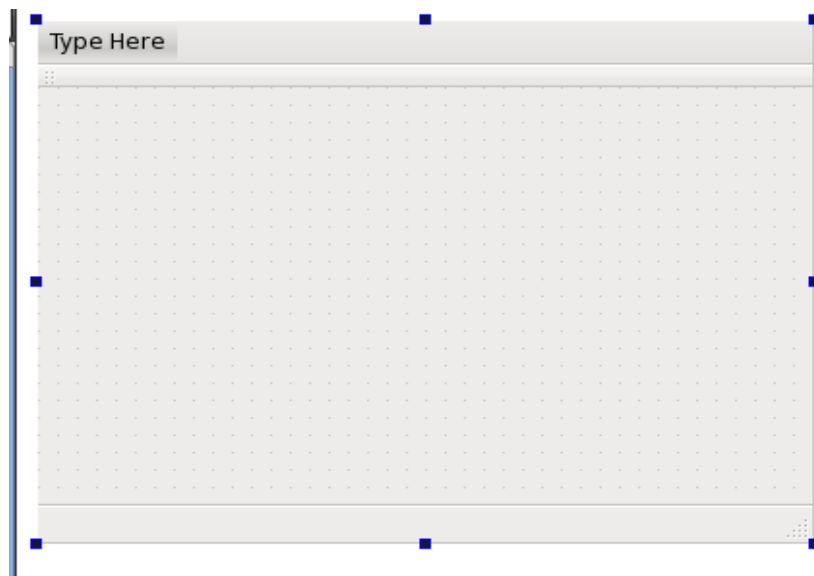
و الآن بعد أن إنتهينا من إعداد شيفرة الفئة mega3d-openglarea نذهب لإعداد النافذة MainWindow.ui : إذن نذهب إلى MainWindow.ui و نقوم بالنقر عليها مرتين



و هكذا يمررنا محرر الكيوتي إلى المصمم Design الخاص بنافذنا

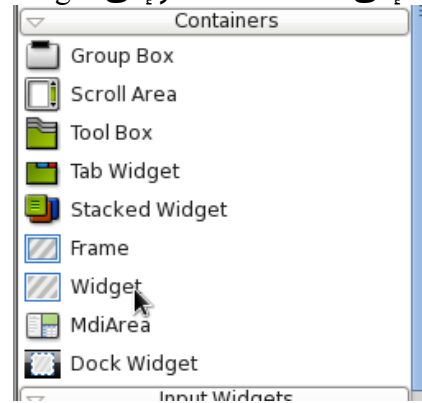


و هذه نافذتنا الفارغة :

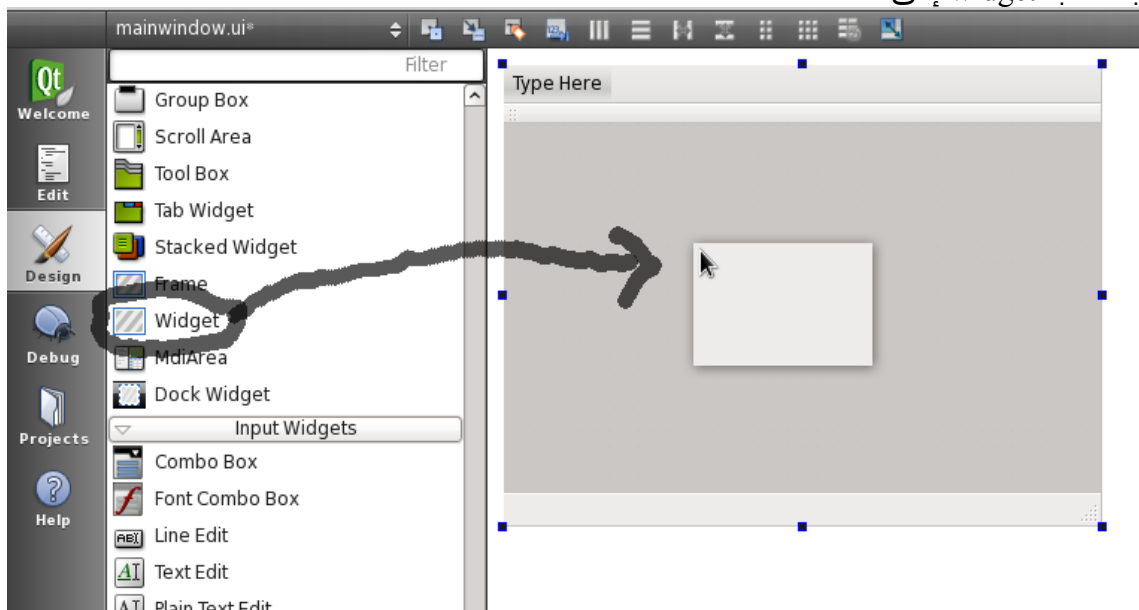


و الآن جاء دور إعداد النافذة بالمتطلبات الضرورية لتفاعل مع الأوبن جي أل :

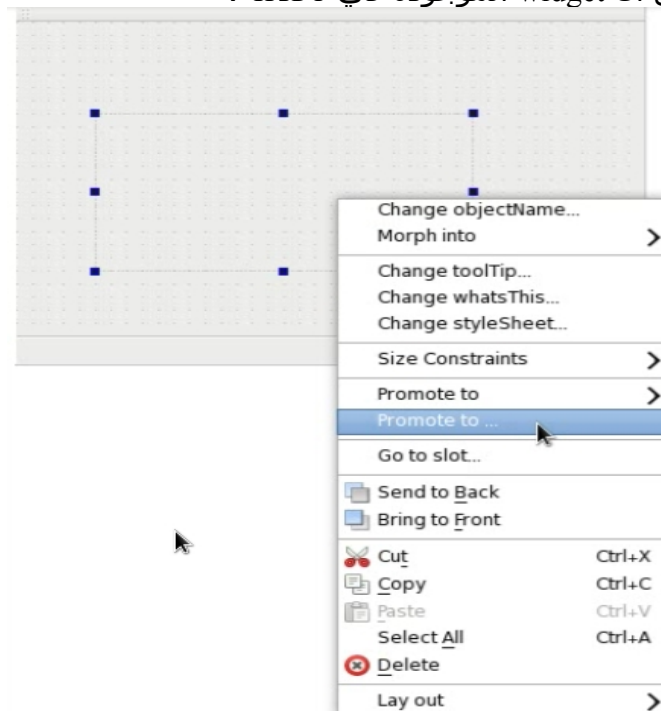
من اليسار أو من يمين الكيوتي نذهب إلى Containers ثم إلى widget



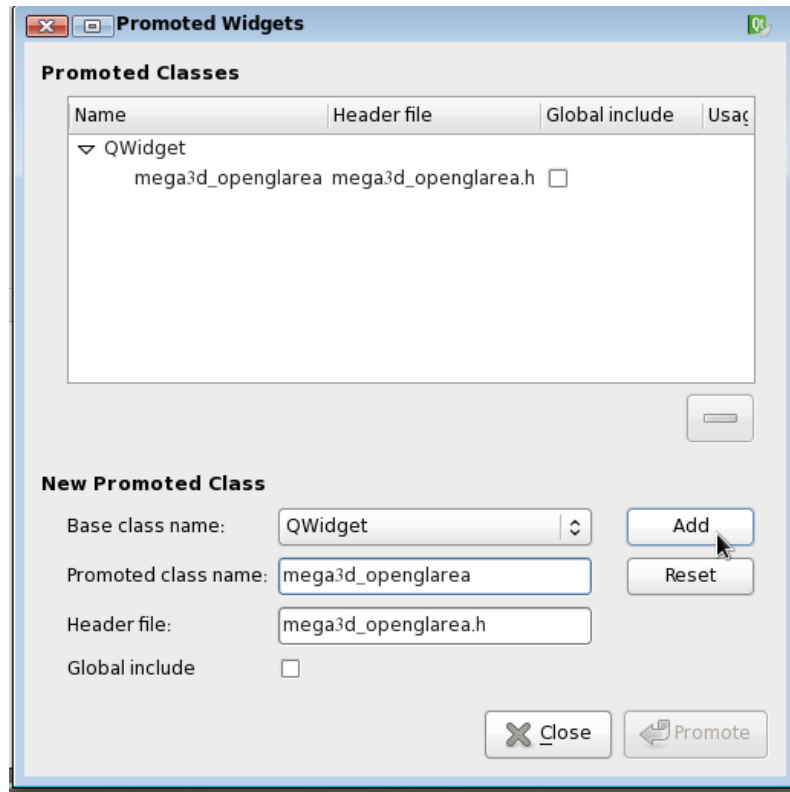
ثم نقوم بسحب widget إلى النافذة



ثم نقر على يسار الفأرة على ال widget الموجودة في نافذتنا :



ثم نختار Promote to .. لنذهب إلى



و نقوم بإعدادها على الشكل :

Base class name: QWidget

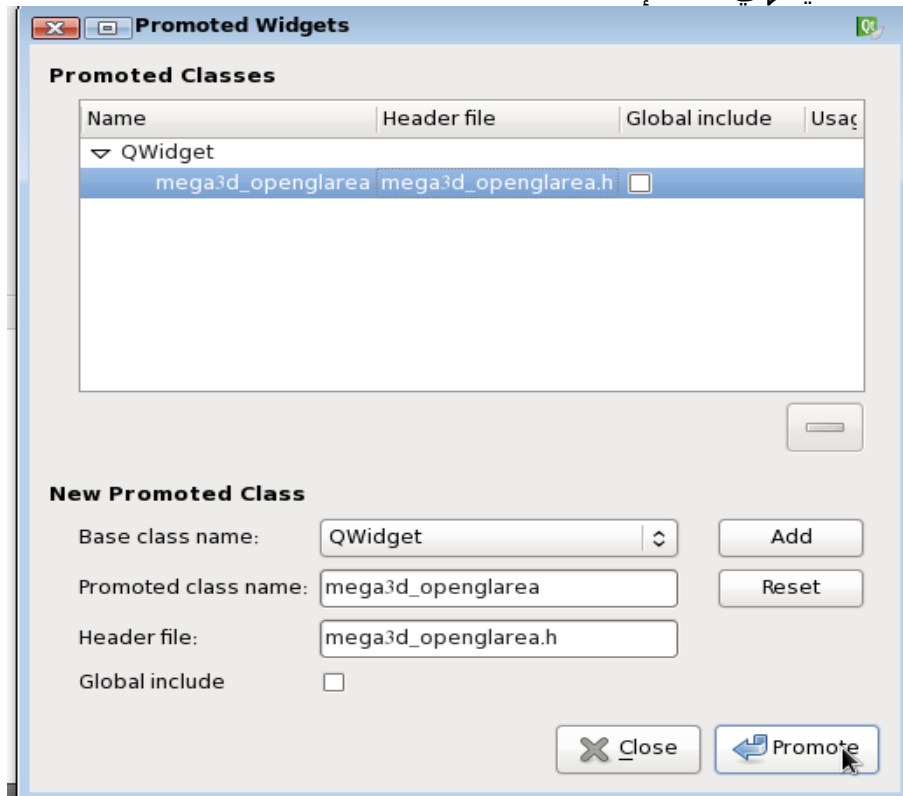
Promote class name: mega3d_openglarea

Header file: mega3d_openglarea.h

و من ثم نقوم بالنقر على Add لتظهر لنا الترقية على ال QWidget في الأعلى على Promoted Classes.

و من ثم نقوم بتحديد Qwidget [] mega3d_openglarea mega3d_openglarea.h

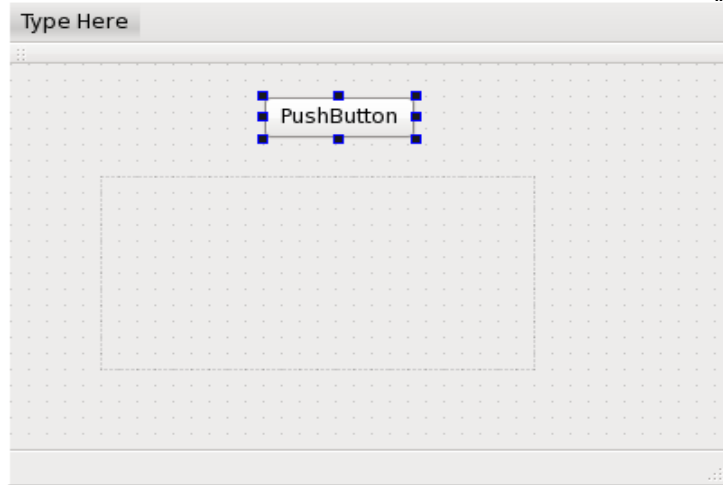
ثم ننقر على Promote لينتهي هذا الإعداد.



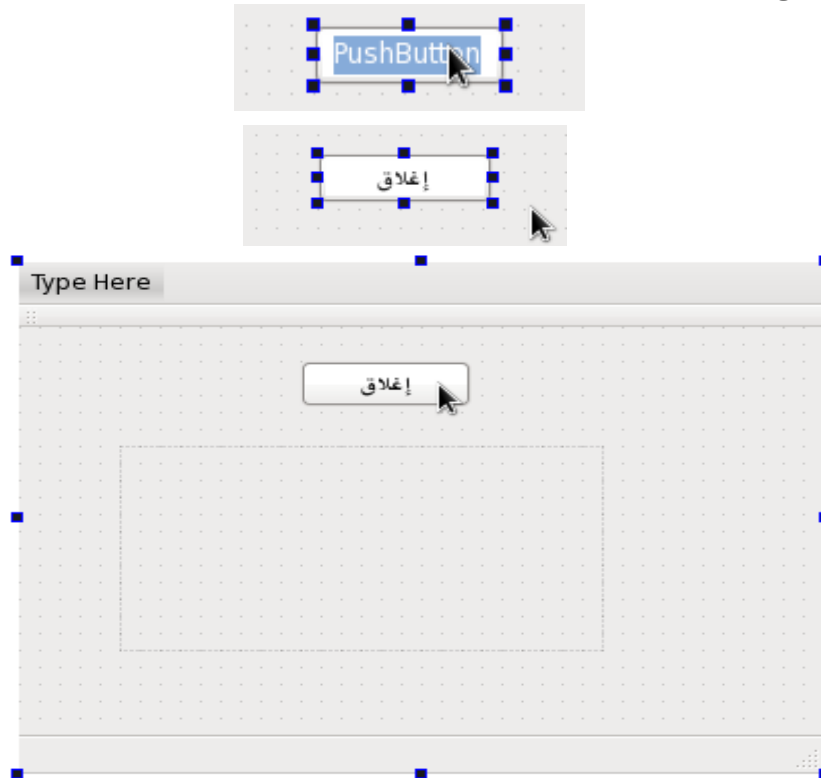
و الآن سنقوم ببعض الإضافات على النافذة :
نذهب إلى الخيارات Buttons ثم نسحب PushButton نحو نافذتنا ,



لتظهر على الشكل التالي :



نغير الإسم من PushButton نحو إغلاق و ذاك عبر النقر مرتين على الزر الظاهر بإسم PushButton
لتظهر النافذة على الشكل :



و الآن بقي إعدادات signals & slots :
نذهب إلى الإشارة + الظاهرة باللون الأخضر و نقوم بالنقر عليها ليظهر لنا signals & slots جديد في الأسفل , و يتكون من Sender و Receiver أي مرسل و مستقبل ,



هنا سنجعل الزر "إغلاق" هو المرسل و نافذتنا MainWindow هي المستقبل, و سنختار لكليهما الحدث الخاص به.

إذن سنقوم بالإعدادات التالية:

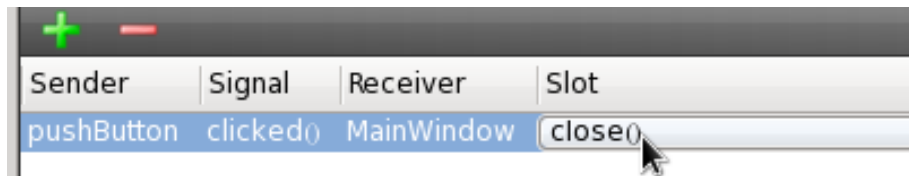
Sender = pushButton

Signal = clicked

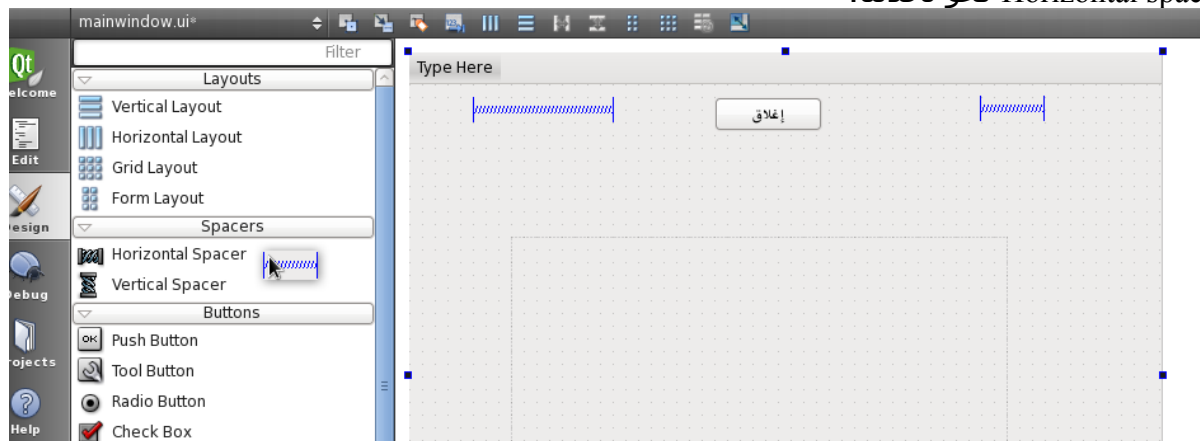
Receiver = MainWindow

Slot = close

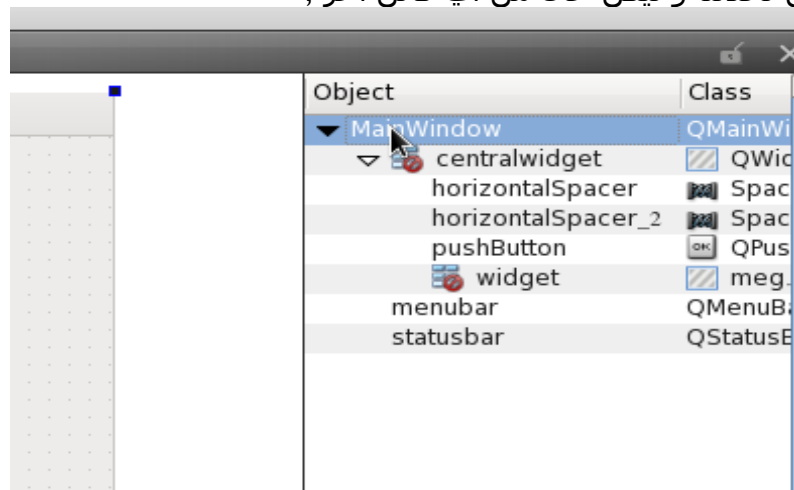
بهذا الشكل :



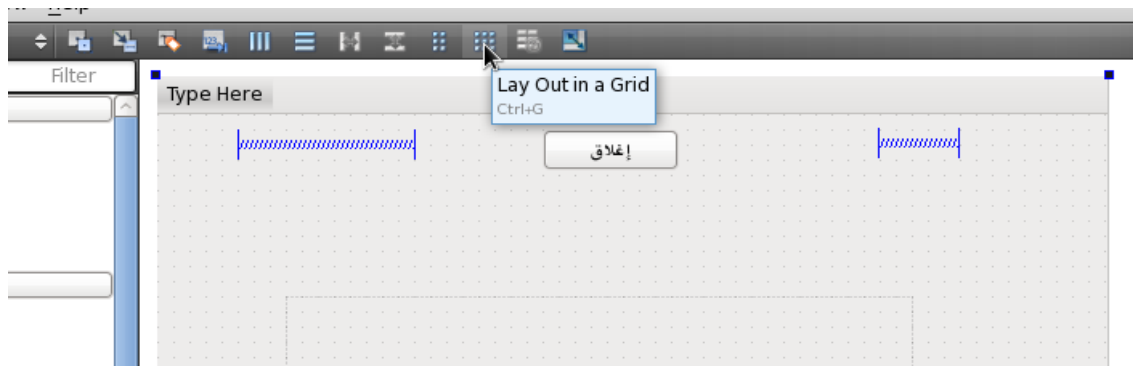
ثم نسحب الخيار Spacers :
من Horizontal spacer نحو نافذتنا.



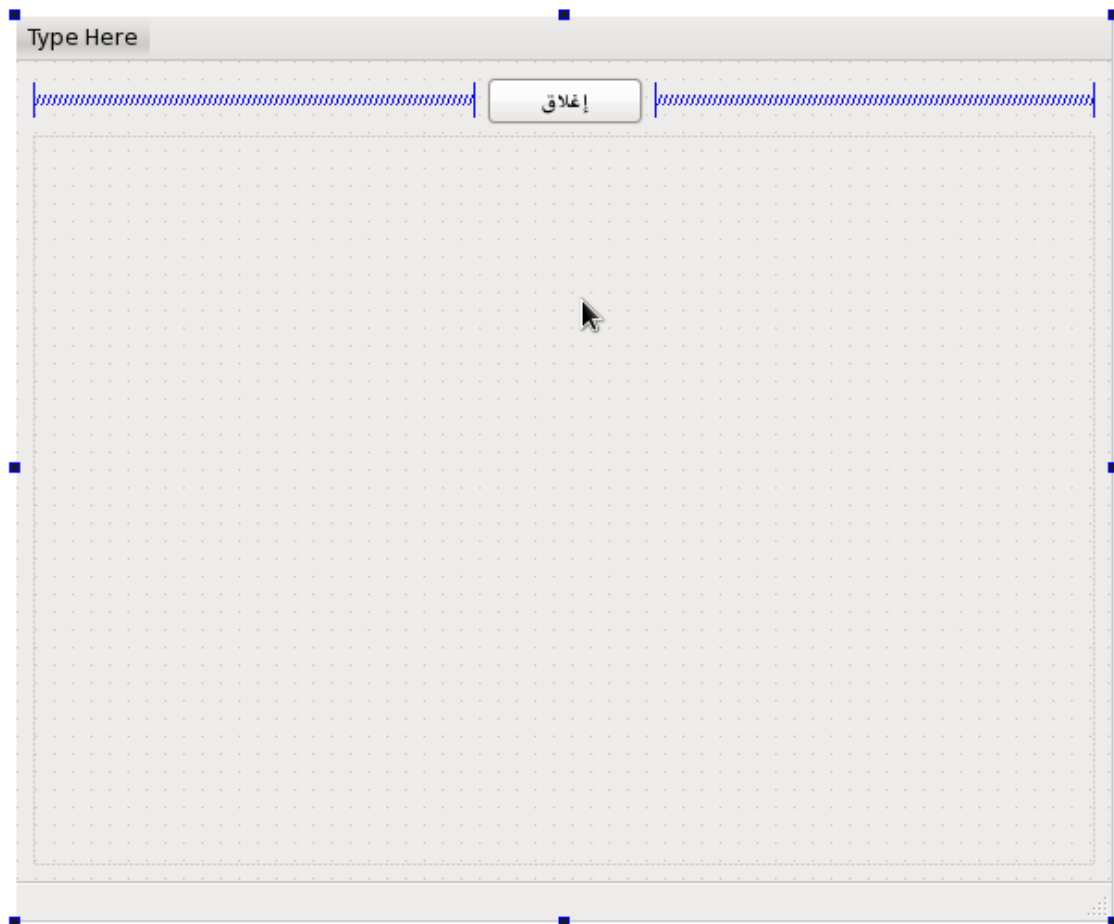
واحد على يمين الزر "إغلاق" و الآخر على يساره.
ثم نذهب إلى الكائن MainWindow و من ثم نحدد ليظهر باللون الأزرق , الغرض من هذا تحديد نافذتنا, أو النقر في أي جزء من نافذتنا و ليكن خال من أي كائن آخر ,



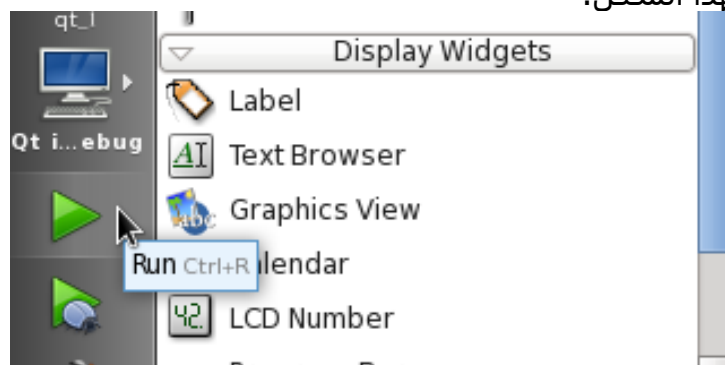
ثم نذهب إلى الخيار Lay Out in a Grid و من ثم نقوم بالنقر عليه



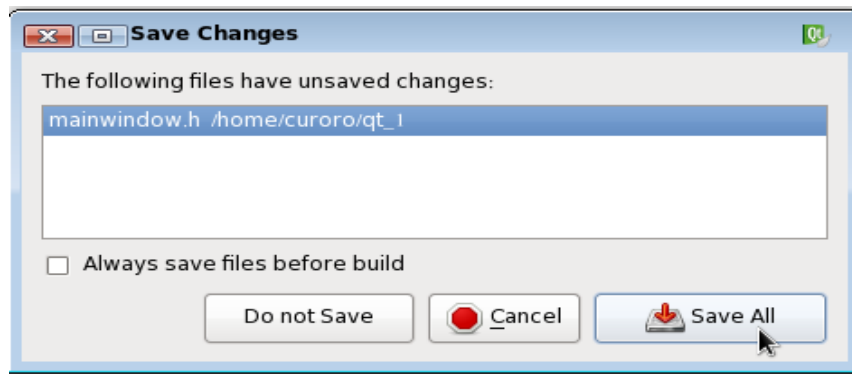
و النتيجة هي :



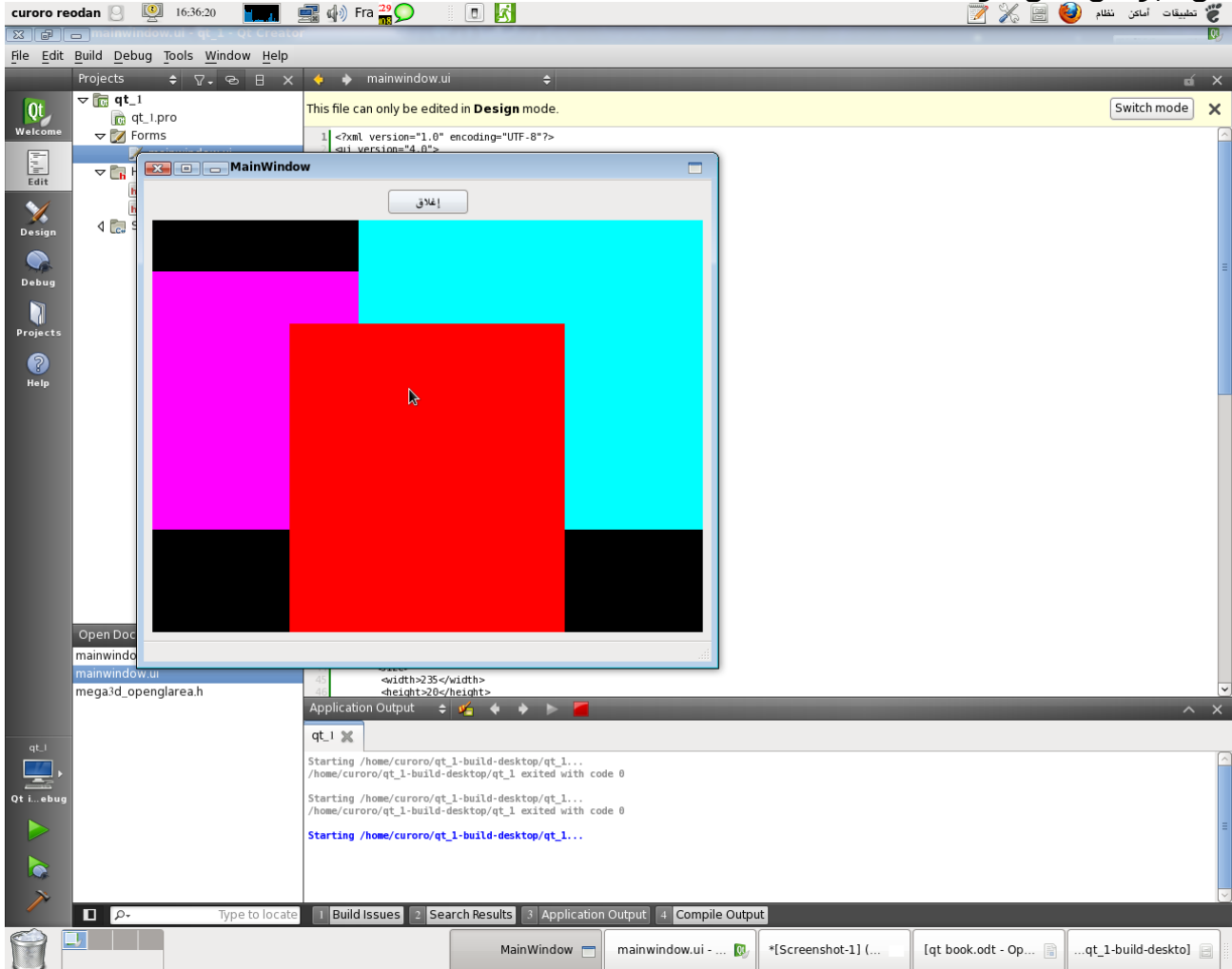
و الآن جاء دور تنفيذ البرنامج :
من القائمة نذهب إلى Build و من ثم Run
أو
نقوم بالنقر على Run , بهذا الشكل:



و سيطلب منك المحرر كيوتي تثبيت حفظ المشروع عندها نقر على الزر Save All .



و ناتج البرنامج ككل هو :



انتهى،