

## RAD with Glade

هنشرح مثال على كيفية استخدام Glade مع PyGTK و Gladizer لإنشاء برنامج قاموس يعتمد على wordlist مع عرب ايز

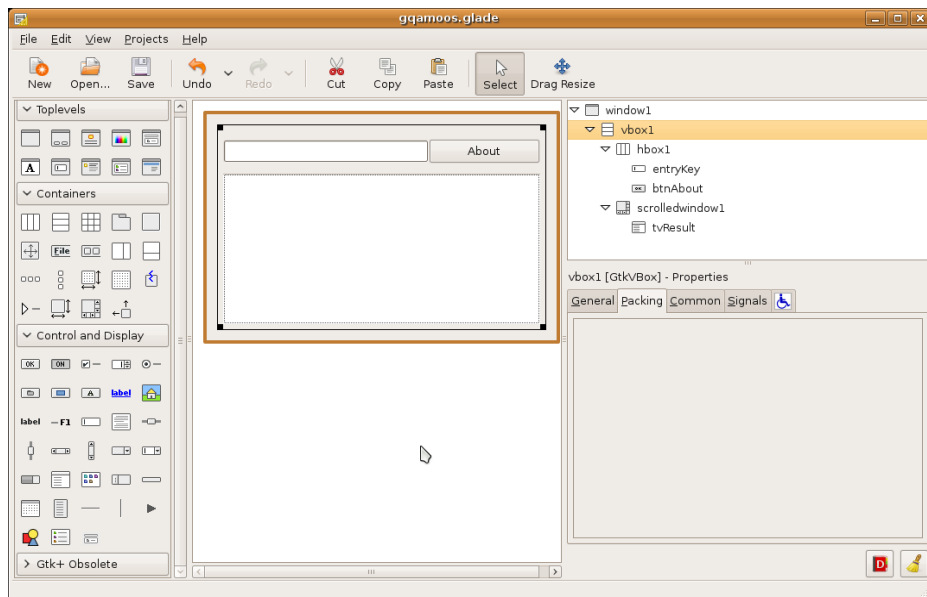


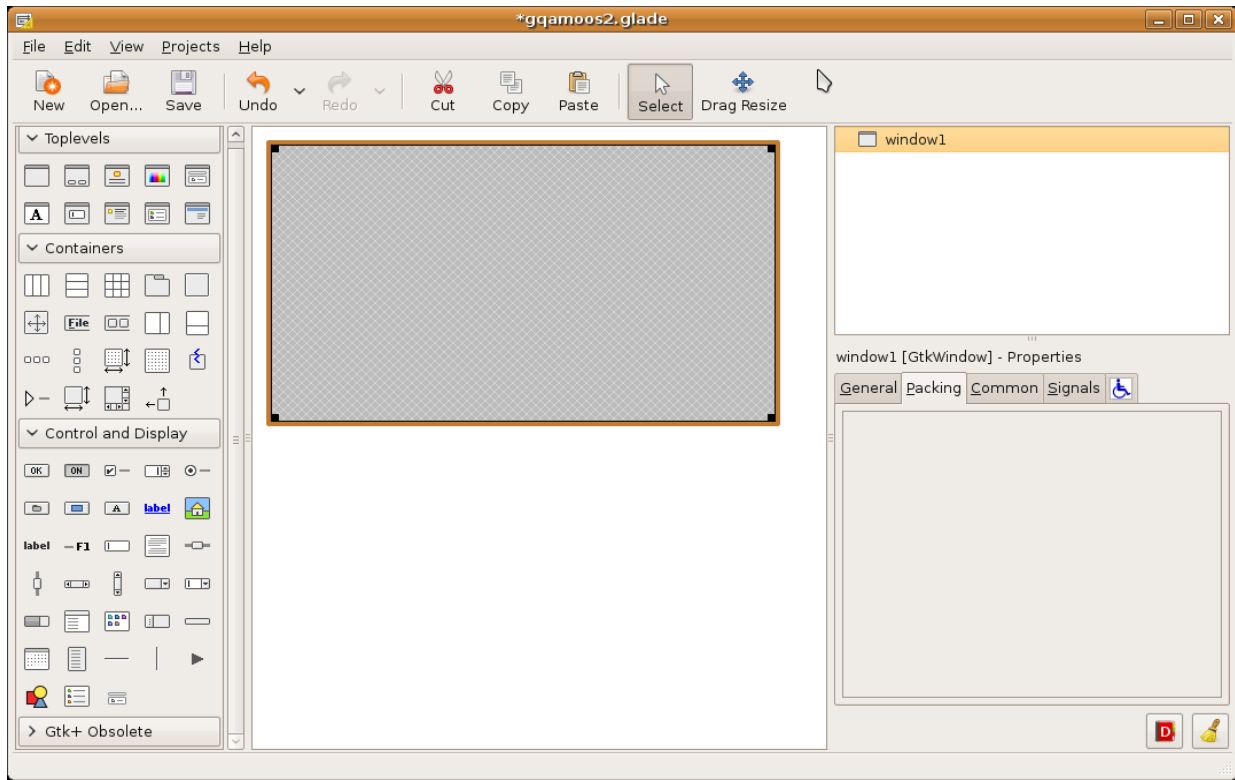
1- حمل ال wordlist

<http://www.arabeyes.org/project.php?proj=Wordlist>

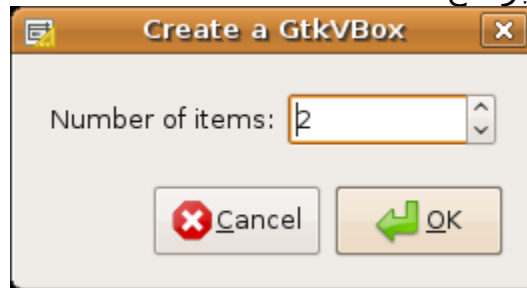
2- افتح glade وانشئ ملف واحفظه ب gqamoos

لاحظ الواجهة مقسمة للتالى

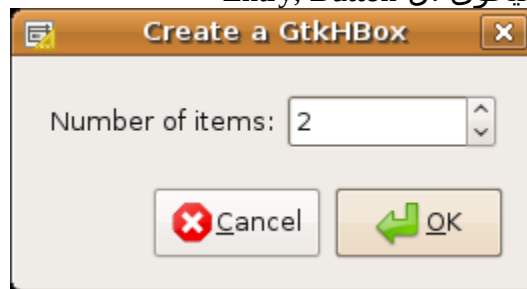




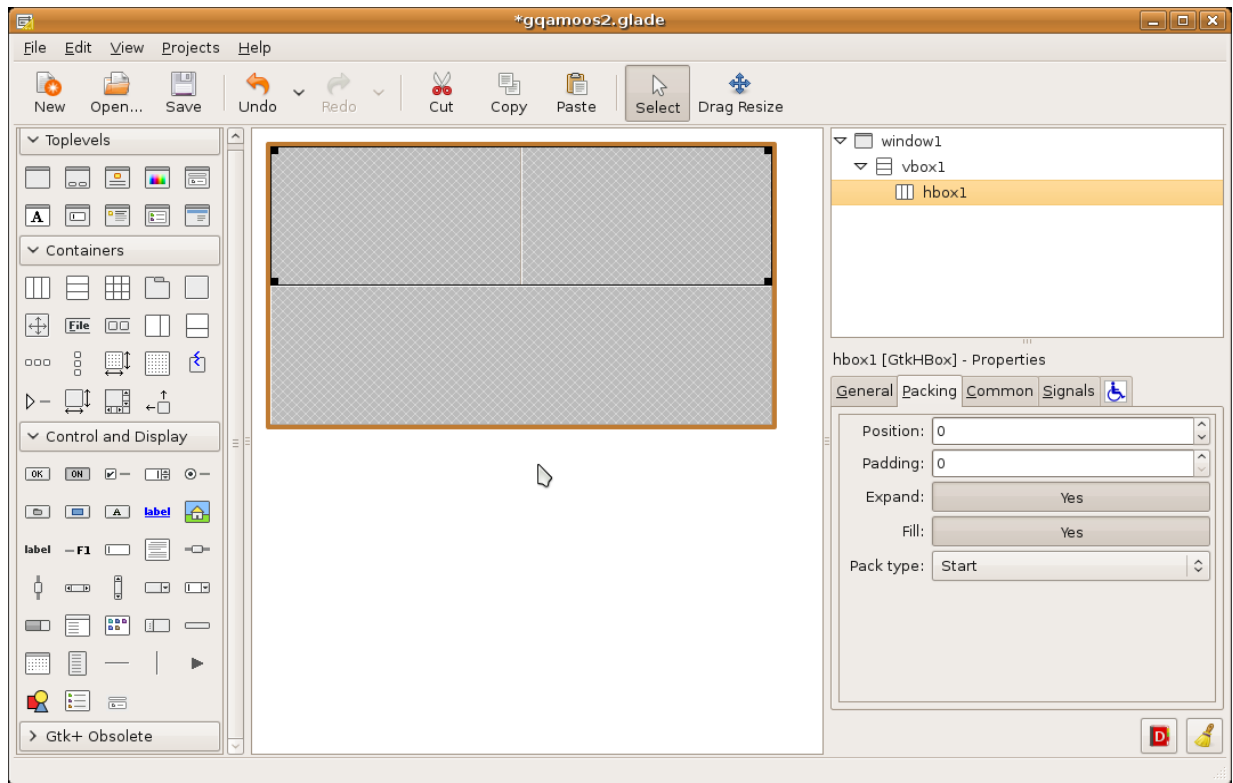
قم بإضافة VBox من اللوحة (لصفين) على النافذة ليضم القسم الأعلى TextEntry, Button  
 ال TextEntry للكلمة المراد البحث عنها  
 ال Button لعرض رسالة حول البرنامج



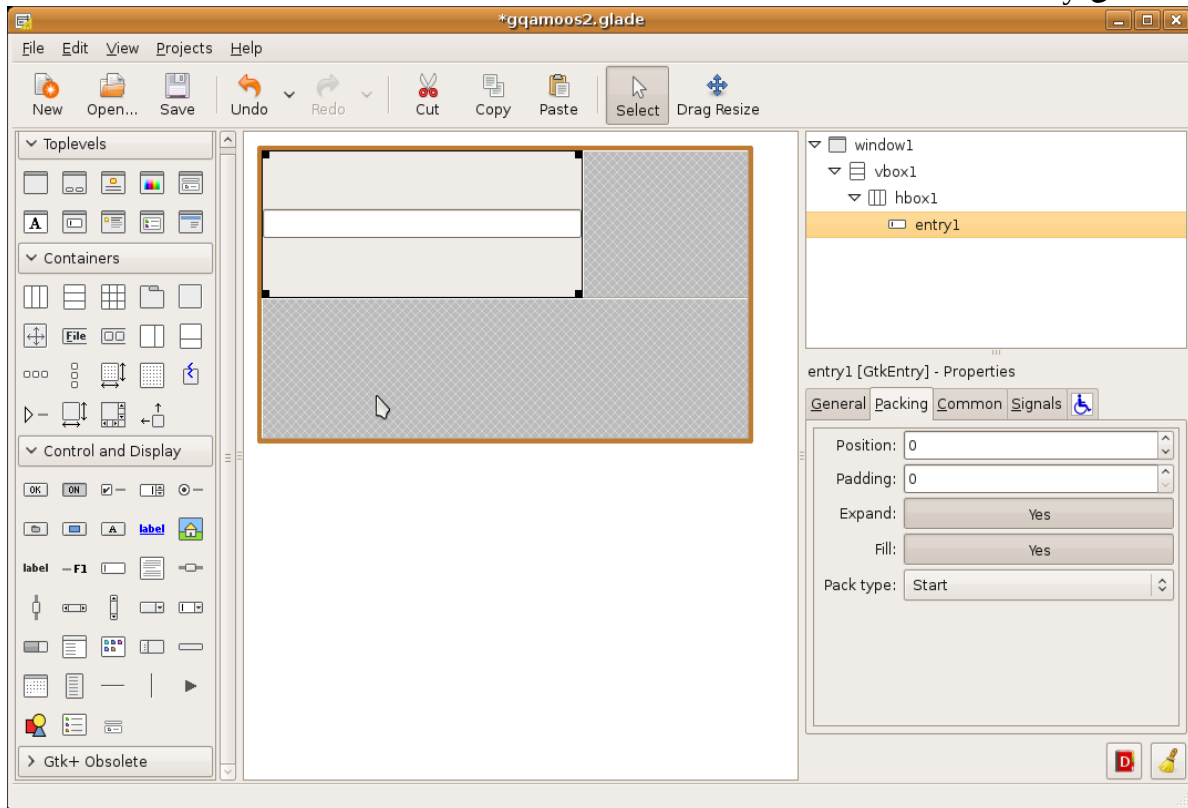
قم بإضافة صندوق افقي Hbox ليحوى ال Entry, Button



لتصبح الواجهة هكذا

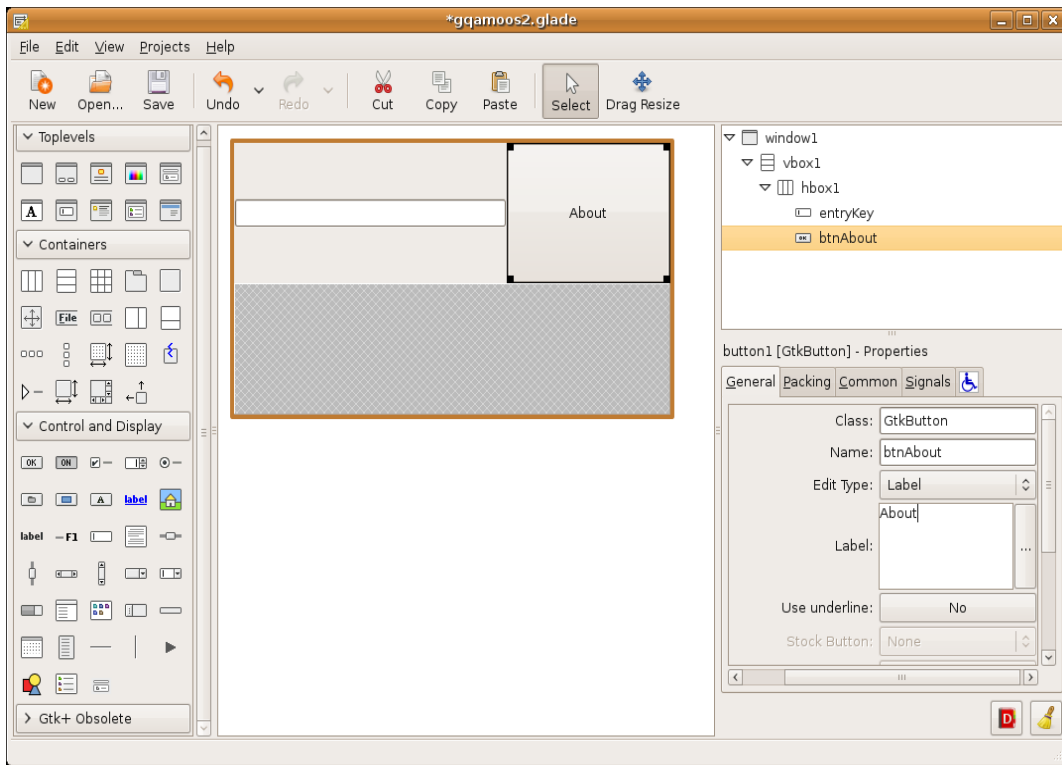


## قم بإضافة ال Text Entry

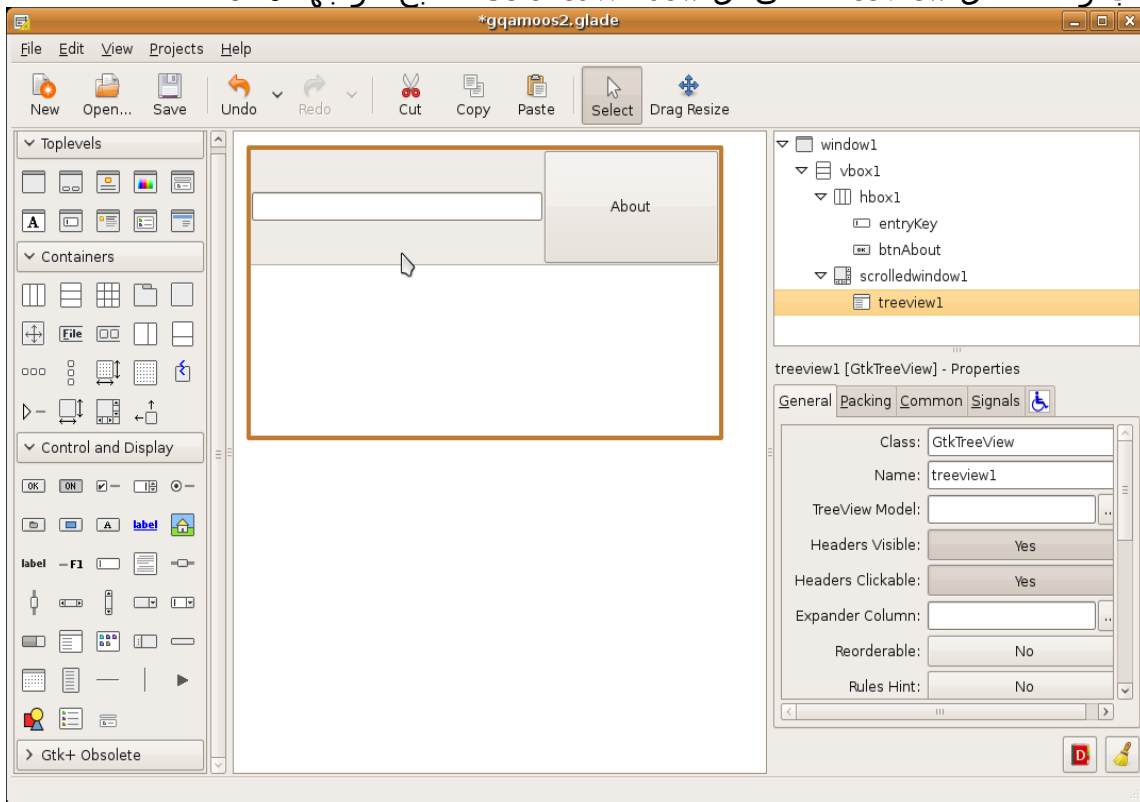


من قسم ال General بأسفل اليمين قم بتغيير اسم الويدجت ل entryKey

قم بإضافة زر Button من اللوحة على اليمين ال Entry



في القسم الأسفل اصف ScrolledWindow ليتيح لنا استخدام ال "Scrollbars" على ال TreeView  
 قم بسحب وإضافة ال TreeView على ال ScrolledWindow لتصبح الواجهة هكذا



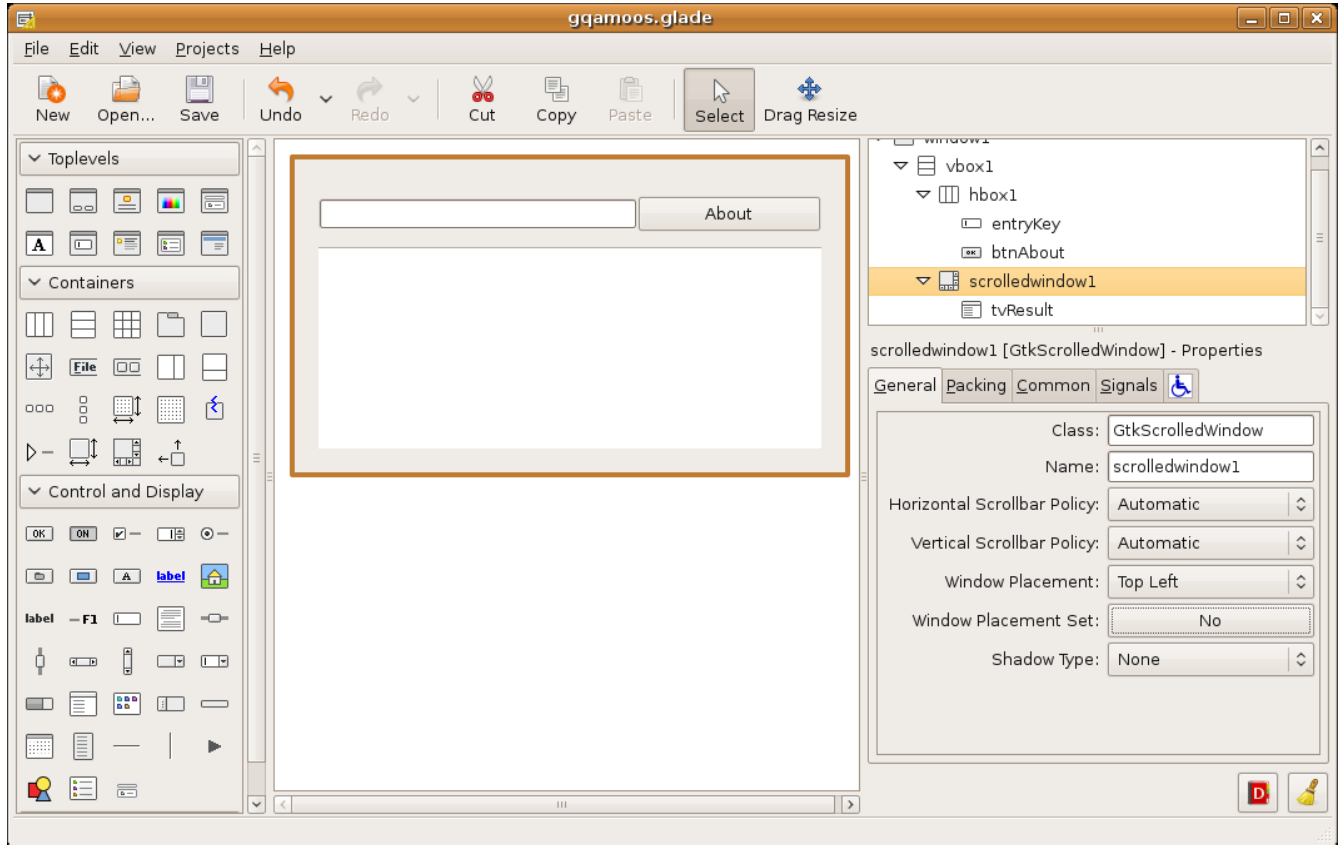
الآن سنقوم ببعض التعديلات على الخواص للويدجتس

قم بتعديل ال Label لل btnAbout إلى About  
ومن قسم ال signals قم بإضافة on\_btnAbout\_clicked signal  
قم بتغيير ال Expand, Fill لل Hbox الى القيم No, No

ولل entryKey قم بإضافة on\_entryKey\_changed signal تحت فرع ال  
"GtkEditable"

وايضا قم بتعديل اسم ال TreeView الى tvResult

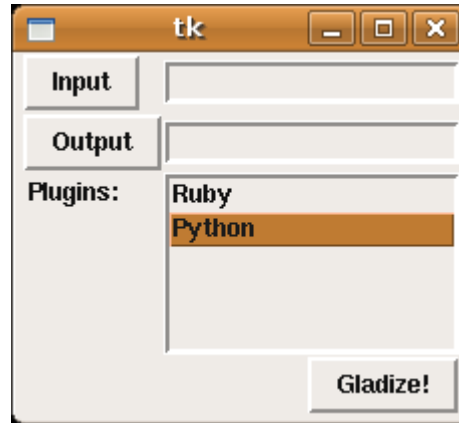
لتصبح الواجهة هكذا



تمام؟ كذا انهينا التصميم  
قم بتشغيل gladizer على ملف gqamoos.glade ليكون الخرج هو gqamoos.py

```
striky@striky-desktop:~$ gladizer.py --file gqamoos.glade -p Python > gqamoos.py
```

او عن طريق gladizerguitk.py



اخيرا قم بتحرير الملف المشابه للتالى

```
#!/bin/python

##CODE GENERATED by gladizer 1.2

import pygtk
pygtk.require('2.0')

import gtk, gtk.glade

class MainWindow(object):

    def __init__(self):

        #Widget tree..
        self.wTree=gtk.glade.XML('gqamoos.glade')

        #connect signals and handlers.
        self.wTree.signal_autoconnect(self)

        self._window1=self.wTree.get_widget('window1')
        self._vbox1=self.wTree.get_widget('vbox1')
        self._hbox1=self.wTree.get_widget('hbox1')
        self._entryKey=self.wTree.get_widget('entryKey')
        self._btnAbout=self.wTree.get_widget('btnAbout')
        self._tvResult=self.wTree.get_widget('tvResult')
        self._window1.show()

    def on_entryKey_changed(self, widget, *args):
        pass

    def on_btnAbout_clicked(self, widget, *args):
        pass

# run main loop
```

```

def main():
    mainwindow = MainWindow()
    #mainwindow.window1.show()
    gtk.main()

if __name__ == "__main__":
    main()

```

وذلك بإضافة ال service looker للبحث فى ال wordlist  
تستطيع الحصول عليها من  
<http://programming-fr34ks.net/pfsoft/looker.py>

```

import looker
import GObject
import gobject

class Looker(GObject):
    def __init__(self):
        self._wc=looker.WordsListLooker()
        self._init_tv()

```

الطريقة \_init\_tv معرفة كالتالى

```

def _init_tv(self):
    encell=gtk.CellRendererText()
    encol=gtk.TreeViewColumn("English", encell, text=0)
    arcell=gtk.CellRendererText()
    arcol=gtk.TreeViewColumn("Arabic", arcell, text=1)
    map(self._tvResult.append_column,[encol, arcol])

```

سيتم ال TreeView باسماء الأعمدة ونوعهم وإضافتهم وذلك من خلال اضافة الأعمدة encol, arcol وهم من النوع gtk.TreeViewColumn وبأخذون معاملات كالتالى

- 1- الهيدر "النص الظاهر"
- 2- ال CellRenderer وهو المسئول عن "رندرة" الخلية (مثل encell وهو "مرندر من النوع النصى" لعرض البيانات النصية)
- 3- الترتيب فى العرض من ال store تابع الطريقة \_get\_model()

```

def _get_model(self):
    lstore=gtk.ListStore(gobject.TYPE_STRING, gobject.TYPE_STRING)
    self._wc.searchKey=self._entryKey.get_text()
    for row in self._wc.lookup():
        iter=lstore.append([row[0], row[1]])

    return lstore

```

هنا نقوم بتحديد نوع البيانات المخزنة. تستطيع استخدام str بدلا من gobject.TYPE\_STRING  
نحصل على النص الظاهر على ال entryKey من خلال الطريقة get\_text ونقوم بإسناده لل searchKey

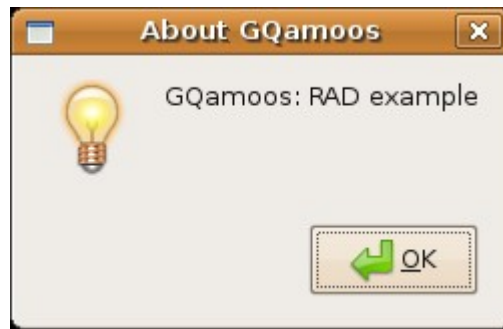
”looker service ال الأوبجكت المسئول عن استغلال ال self.\_wc لل property  
 نقوم بإضافة نتائج البحث من خلال lstore.append فى داخل list ونقوم بإعادة ذلك ال  
 نقوم بربط التغيير فى النص الظاهر على ال entryKey بتعديل ال model الخاص بال TreeView لتغيير  
 نتائج البحث مع كل تغيير وذلك بتعديل on\_entryKey\_changed callback

```
def on_entryKey_changed(self, widget, *args):
    self._tvResult.set_model(self._get_model())
```

نقوم بتحديد ال model بإستخدام الطريقة set\_model الخاصة بال tvResult ال TreeView

تحديد رسالة حول البرنامج قم بتعديل ال on\_btnAbout\_clicked المسئولة عن معالجة ال clicked signal  
 للزر btnAbout

```
def on_btnAbout_clicked(self, widget, *args):
    md=gtk.MessageDialog(self._window1, gtk.DIALOG_DESTROY_WITH_PARENT,
    gtk.MESSAGE_INFO, gtk.BUTTONS_OK, "GQamoos: RAD example")
    md.set_title("About GQamoos")
    md.run()
    md.destroy()
```



قم بتشغيل البرنامج



ستلاحظ اننا لم نحدد عنوان النافذة..تستطيع تعديل ذلك اما بإستخدام الطريقة set\_title  
 self.\_window1 او من خلال ملف glade بتعديل ال Title لل window1 إلى Gqamoos



الكود النهائي

```
#!/bin/python

##CODE GENERATED by gladizer 1.2

import pygtk
pygtk.require('2.0')

import looker

import gobject
import gtk, gtk.glade

class MainWindow(object):

    def __init__(self):

        #Widget tree..
        self.wTree=gtk.glade.XML('gqamoos.glade')

        #connect signals and handlers.
        self.wTree.signal_autoconnect(self)

        self._window1=self.wTree.get_widget('window1')
        self._vbox1=self.wTree.get_widget('vbox1')
        self._hbox1=self.wTree.get_widget('hbox1')
        self._entryKey=self.wTree.get_widget('entryKey')
        self._btnAbout=self.wTree.get_widget('btnAbout')
        self._tvResult=self.wTree.get_widget('tvResult')
        self._window1.show()
        self._wc=looker.WordsListLooker()

        self._init_tv()
```

```

def _init_tv(self):
    encell=gtk.CellRendererText()
    encol=gtk.TreeViewColumn("English", encell, text=0)
    arcell=gtk.CellRendererText()
    arcol=gtk.TreeViewColumn("Arabic", arcell, text=1)
    map(self._tvResult.append_column,[encol, arcol])

def _get_model(self):
    lstore=gtk.ListStore(str, str) #gobject.TYPE_STRING,
gobject.TYPE_STRING
    self._wc.searchKey=self._entryKey.get_text()
    for row in self._wc.lookup():
        iter=lstore.append([row[0], row[1]])

    return lstore

def on_entryKey_changed(self, widget, *args):
    self._tvResult.set_model(self._get_model())

def on_btnAbout_clicked(self, widget, *args):
    md=gtk.MessageDialog(self._window1, gtk.DIALOG_DESTROY_WITH_PARENT,
gtk.MESSAGE_INFO, gtk.BUTTONS_OK, "GQamoos: RAD example")
    md.set_title("About GQamoos")
    md.run()
    md.destroy()
    #pass

# run main loop

def main():
    mainwindow = MainWindow()
    #mainwindow.window1.show()
    gtk.main()

if __name__ == "__main__":
    main()

```

By Ahmed Youssef