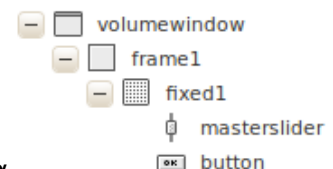


Σε αυτό το άρθρο θα παρουσιάσουμε πως μπορεί η εφαρμογή μας να έχει ένα εικονίδιο στον δίσκο συστήματος (systray) και πως κάνοντάς του κλικ μας εμφανίζει το κεντρικό παράθυρο της εφαρμογής ή ένα popup μενού για έλεγχο της.

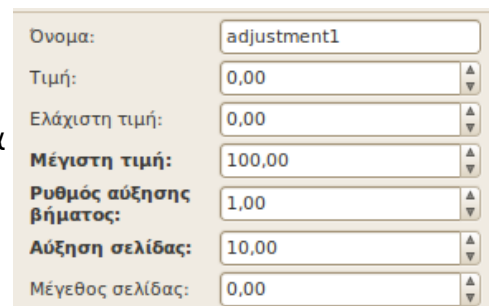
Θα φτιάξουμε ένα μικρό slider που θα ελέγχει την ένταση του ήχου, όπως αυτό που υπάρχει ήδη στα περισσότερα συστήματα linux. Έχει καθαρά εκπαιδευτικούς σκοπούς δείχνοντας την αλληλοεπίδραση μεταξύ των σημάτων του systray και του προγράμματος. Για τον έλεγχο του ήχου θα χρησιμοποιήσουμε είτε την βιβλιοθήκη alsaaudio της python είτε την εντολή amixer του συστήματος της ALSA.

Δημιουργούμε το παράθυρο μας στο Glade και του δίνουμε όνομα volumewindow. Στον *Τύπο παραθύρου* επιλέγουμε Αναδυόμενο (popup). Αυτού του είδους τα παράθυρα δεν έχουν πλαίσιο και διακοσμήσεις και δεν τα ελέγχει ο window manager του συστήματος. Στις αιτήσεις πλάτους και ύψους δίνουμε αντίστοιχα 60 και 170 pixels, ενώ στο *Πλάτος πλαισίου* δίνουμε τιμή 2. Αυτό είναι το περιθώριο που θα έχουν όλα τα αντικείμενα του παραθύρου από τις άκρες του.

Στην συνέχεια προσθέτουμε ένα frame και αφαιρούμε τα alignment και label που δημιουργεί. Επιλέγουμε την σκίαση της αρεσκείας μας και προσθέτουμε μέσα του ένα πλαίσιο *fixed*. Αυτό το πλαίσιο μας επιτρέπει να τοποθετήσουμε αντικείμενα σε συγκεκριμένες θέσεις μέσα του. Βλέπουμε ότι έχουμε και αυτήν την δυνατότητα εκτός από την τοποθέτηση σε γραμμές και στήλες. Στην συγκεκριμένη περίπτωση είναι χρήσιμο γιατί το παράθυρο δεν μπορεί να αλλάξει μέγεθος και έτσι με λίγο πειραματισμό μπορούμε να τοποθετήσουμε τα διάφορα widget όπου ακριβώς θέλουμε και να είμαστε σίγουροι ότι πάντα θα φαίνονται έτσι.

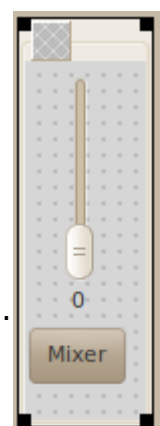


Προσθέτουμε μία *Κατακόρυφη κλιμακα (slider)* και την ονομάζουμε masterslider. Θέτουμε τα *Ψηφία* στο μηδέν, την *Θέση τιμής* κάτω και την ορίζουμε σαν *Αντεστραμμένη*. Προσθέτοντας μια νέα *Ρύθμιση* το glade την ονομάζει αυτόματα adjustment1 και της δίνει ορισμένες αρχικές τιμές. Αλλάζουμε την *Μέγιστη τιμή* σε 100, τον *Ρυθμό αύξησης βήματος* σε 1, την *Αύξηση σελίδας* σε 10 και το *Μέγεθος σελίδας* σε 0.



Επιστρέφοντας στο masterslider και στο tab packing βλέπουμε ότι τα συνηθισμένα Γέμισμα και Ανάπτυξη έχουν αντικατασταθεί από τα *Θέση Χ* και *Θέση Υ*, λόγω του ότι βρίσκετε μέσα σε fixed πλαίσιο. Σε αυτά βάζουμε αντίστοιχα τις τιμές 10 και 6. Στα *Σήματά* του και συγκεκριμένα στο value-changed δίνουμε σαν handler την μέθοδο masterslider_change. Αυτή θα αλλάζει την ένταση του ήχου ανάλογα με την θέση του slider.

Στην συνέχεια προσθέτουμε ένα button με το όνομα button, και στην ετικέτα του βάζουμε Mixer. Στις *θέσεις Χ και Υ* βάζουμε 1 και 132 αντίστοιχα, ενώ στο σήμα *clicked* δίνουμε σαν handler button_clicked. Αυτή η μέθοδος θα εμφανίζει το gnome-alsamixer, αν υπάρχει στο σύστημα.

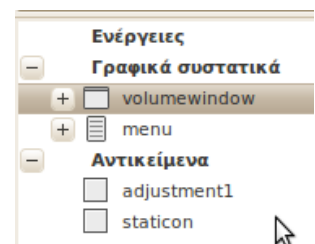


Με το μικρό μας παράθυρο έτοιμο ας προσθέσουμε τα υπόλοιπα αντικείμενα που χρειαζόμαστε για το πρόγραμμα μας. Επιλέγουμε ένα *Αναδυόμενο μενού (popup)* και του δίνουμε το όνομα *menu*. Με το *menu* επιλεγμένο πατάμε *edit* στην μπάρα εργαλείων και προσθέτουμε δύο *Στοιχεία εικόνας*. Για το πρώτο επιλέγουμε το έτοιμο κουμπί *stock Περί* και για το άλλο το *Έξοδος*. Στα *activate* σήματα τους βάζουμε σαν *handler* αντίστοιχα τις μεθόδους *about dialog* και *close_window*.

Το βασικότερο μας αντικείμενο είναι ένα *Εικονίδιο γραμμής κατάστασης* το οποίο θα το βρούμε στα *Διάφορα*. Προσθέτουμε ένα και το ονομάζουμε *staticon*. Στο *Pixbuf* γράφουμε το όνομα του αρχείου του εικονιδίου που θα χρησιμοποιήσουμε. Υπάρχουν πολλά εικονίδια που μπορείτε να χρησιμοποιήσετε στον κατάλογο */usr/share/icons/gnome*. Αντιγράψτε ένα της αρεσκείας σας στον ίδιο κατάλογο με τα αρχεία *glade* και *python*. Στον *Τίτλο* δίνουμε *SysAudio*. Αυτό θα εμφανίζεται όταν περνάει ο δείκτης του ποντικιού από πάνω του. Στα *Σήματα* του δίνουμε στο *activate* την μέθοδο *window_show* και στο *popup-menu* την *menu_popup*. Η πρώτη μέθοδος εκτελείται όταν ο χρήστης κάνει αριστερό κλικ πάνω στο εικονίδιο και θα εμφανίσει το παράθυρό μας με τον *slider*, ενώ η δεύτερη όταν κάνει ένα από τα υπόλοιπα και συγκεκριμένα δεξί κλικ, εμφανίζοντας το μενού της εφαρμογής.

Τα βασικά αντικείμενα της εφαρμογής είναι τα εξής:

Σε αυτό το σημείο το αρχείο *glade* είναι έτοιμο, έχουμε προσδιορίσει ότι θα χρειαστούμε και το σώζουμε σαν *volumecontrol.glade*. Το πρόγραμμα μας σε *python* είναι απλό και περιέχει στοιχεία που έχουμε δει ήδη.



```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

```
import os
try:
    import alsaaudio
    HAVEALSA=True
except:
    import subprocess
    import commands
    HAVEALSA=False
import pygtk
pygtk.require('2.0')
import gtk

class VolumeControl():

    def __init__(self):
        builder=gtk.Builder()
        builder.add_from_file("volumecontrol.glade")
        builder.connect_signals(self)
        self.window = builder.get_object("volumewindow")
        self.window_position = (0, 0)
        self.staticon = builder.get_object("staticon")
```

```

self.menu = builder.get_object("menu")
self.masterslider = builder.get_object("masterslider")

def close_window(self, widget, data=None):
    gtk.main_quit()

def button_clicked(self, widget):
    os.system('gnome-alsamixer')

def window_show(self, widget, data=None):
    if self.window.get_property('visible'):
        self.window.hide()
    else:
        self.masterslider.set_value(self.get_master_volume())
        self.set_window_position()
        self.window.move(self.window_position[0], self.window_position[1])
        self.window.show_all()
        self.window.present()

def menu_popup(self, widget, button, time, data = None):
    if button == 3:
        self.menu.show_all()
        self.menu.popup(None, None, None, 3, time)

def masterslider_change(self, widget):
    volume = int(widget.get_value())
    if HAVEALSA:
        mix = alsaaudio.Mixer()
        mix.setvolume(volume)
    else:
        proc = subprocess.Popen('/usr/bin/amixer sset Master ' \
                                + str(volume) + '%', shell=True, stdout=subprocess.PIPE)
        proc.wait()

def set_window_position(self):
    staticon_geometry = self.staticon.get_geometry()[1]
    if staticon_geometry.y <= 200:
        y_coords = staticon_geometry.y
    else:
        y_coords = staticon_geometry.y-180
    self.window_position = (staticon_geometry.x+20, y_coords+10)

def get_master_volume(self):
    if HAVEALSA:
        mix = alsaaudio.Mixer()
        return mix.getvolume()[0]
    else:
        output = commands.getoutput('/usr/bin/amixer sget Master | grep "%")
        master = output.split('\n')[0]
        start = master.find('(') + 1
        end = master.find('%', start)
        return float(master[start:end])

```

```

def aboutdialog(self, widget, data=None):
    about = gtk.AboutDialog()
    about.set_program_name('Volume Control')
    about.set_version('1.0')
    about.set_comments(u'Πρόγραμμα ελέγχου της στάθμης του ήχου.')
    about.run()
    about.destroy()

def main(self):
    gtk.main()

if __name__ == "__main__":
    app=VolumeControl()
    app.main()

```

Στην αρχή προσπαθούμε να φορτώσουμε την βιβλιοθήκη alsaaudio και αν αυτό αποτύχει φορτώνουμε τις subprocess και command για να μπορέσουμε να εκτελέσουμε εντολή του συστήματος και να πάρουμε το αποτέλεσμα της. Ορίζουμε και μια μεταβλητή για να ξέρουμε αν έχουμε φορτώσει την alsaaudio ή όχι. Οι βασικές μέθοδοι είναι γνωστές από τα προηγούμενα άρθρα. Στην __init__ φορτώνουμε το glade αρχείο και ορίζουμε τις μεταβλητές που θα χρησιμοποιήσουμε. Η aboutdialog δημιουργεί και δείχνει το παράθυρο “Περί” όταν πατήσει ο χρήστης την αντίστοιχη επιλογή στο μενού και η close_window τερματίζει το πρόγραμμα.

Η get_master_volume επιστρέφει την τιμή που έχει η ένταση του ήχου κατά τη στιγμή της κλήσης της. Χρησιμοποιεί είτε την getvolume της alsaaudio είτε το αποτέλεσμα της εντολής “amixer sget Master”. Η set_window_position θέτει την θέση του παραθύρου λίγο πιο κάτω και λίγο πιο δεξιά από το εικονίδιο.

Η window_show εκτελείται όταν ο χρήστης κάνει αριστερό κλικ στο εικονίδιο και το κρύβει αν αυτό είναι ήδη ορατό ή το εμφανίζει αφού πρώτα υπολογίσει την τιμή του ήχου και την θέση του παραθύρου. Η present που καλείται στο τέλος έχει σαν σκοπό να εμφανίζει το παράθυρο αν αυτό καλύπτεται από κάποιο άλλο.

Η master_slider_change εκτελείται όταν ο χρήστης αλλάξει την θέση του slider και αναλόγως αυτής θέτει την τιμή της έντασης είτε με την setvolume είτε με την “amixer sset Master”.

Η menu_popup εμφανίζει το μενού της εφαρμογής. Τα popup menu χρειάζονται για να εμφανιστούν τον αριθμό του κουμπιού του ποντικιού που πατήθηκε και την χρονική στιγμή που έγινε το γεγονός. Αυτά και τα δύο μας τα δίνει το σήμα popup-menu του εικονιδίου, εμείς μένει μόνο να ελέγξουμε αν το κουμπί που πατήθηκε είναι το δεξί.

Η button_clicked εμφανίζει το gnome-alsamixer, αν είναι εγκατεστημένο, αλλιώς θα εμφανίσει μήνυμα λάθους στην γραμμή εντολών. Εδώ μπορείτε, με τις πληροφορίες από τα προηγούμενα άρθρα, να προσθέσετε ένα ErrorDialog για να ενημερώνεται ο χρήστης.

Είναι μια απλή εφαρμογή αλλά δείχνει τα στοιχεία που χρειάζονται για την δημιουργία ενός systray προγράμματος. Με τον ίδιο τρόπο διάφορα προγράμματα, όπως η Opera και το Transmission, βάζουν ένα εικονίδιο στο systray και το πρόγραμμα δεν τερματίζει αν ο χρήστης πατήσει το X στο παράθυρο του αλλά κρύβεται απλώς και εμφανίζεται πάλι αν ο χρήστης κάνει αριστερό κλικ στο εικονίδιο, ενώ με δεξί εμφανίζει είτε μενού είτε πληροφορίες του προγράμματος.

