

## Grafisch programmeren met GTK - Deel 3



door Özcan Güngör  
<ozcangungor(at)netscape.net>

*Over de auteur:*  
Op het ogenblik doe ik mijn legerdienst als Linux- en Oracle beheerder en webprogrammeur.

*Vertaald naar het Nederlands door:*  
Samuel Deros  
<Zenodotus/at/linux.be>



*Kort:*

In deze serie artikelen zullen we leren hoe we graphical user interfaces (GUIs) kunnen schrijven door gebruik te maken van GTK. Ik heb er geen idee van hoe lang deze serie zal duren. Om deze artikelen te begrijpen zou je de volgende mogelijkheden van de C-talen moeten kennen:

- Variabelen
- Functies
- Pointers

Het is aanbevolgen de vorige artikelen te lezen:  
Grafisch programmeren met GTK  
Grafisch programmeren met GTK - Deel 2.  
Dit artikel is wat korter dan de vorige daar ik momenteel mijn legerdienst doe.

---

### Toggle Button

Deze knop lijkt op een normale knop maar heeft twee toestanden: ingedrukt of niet. Om een indrukknop te maken wordt één van de volgende functies gebruikt:

```
GtkWidget *toggle=gtk_toggle_button_new(void);  
GtkWidget *toggle=gtk_toggle_button_new_with_label(const gchar *label);
```

De eerste functie maakt een indrukknop zonder tekstlabel, terwijl de tweede één creëert met de string

label erop.

Zijn status kun je met volgende functie aanpassen:

```
gtk_toggle_button_set_active (GtkToggleButton *toggle_button,  
                             gboolean is_active);
```

Hier is toggle\_button de knop wiens status je wenst aan te passen en active is de status (0 of 1). Wanneer het 0 is, is de knop niet ingedrukt; wanneer het 1 is, is de knop wel ingedrukt.

Om de status van de knop te verkrijgen kan volgende functie gebruikt worden:

```
gboolean gtk_toggle_button_get_active(GtkToggleButton *button);
```

Het "toggled" event kan verbonden worden met een indrukknop.

Hier is een voorbeeld:

```
#include <gtk/gtk.h>  
void togg(GtkWidget *widget, gpointer *data){  
    if (gtk_toggle_button_get_active(GTK_TOGGLE_BUTTON(data)))  
        g_print("State is 1\n");  
    else  
        g_print("State is 0\n");  
}  
int main( int argc, char *argv[] )  
{  
  
    GtkWidget *window;  
    GtkWidget *button;  
    gtk_init (&argc, &argv);  
    /* Maak een nieuw venster */  
    window = gtk_window_new (GTK_WINDOW_TOPLEVEL);  
    gtk_window_set_title (GTK_WINDOW (window), "Toggle Button");  
    /* Verbind een vernietigingsevent aan het venster. */  
    gtk_signal_connect (GTK_OBJECT (window), "destroy",  
                       GTK_SIGNAL_FUNC(gtk_main_quit), NULL);  
  
    /* Maak een toggle knop */  
    button=gtk_toggle_button_new_with_label("I'm a toggle button");  
  
    /* Voeg de knop toe aan het venster */  
    gtk_container_add(GTK_CONTAINER(window),button);  
    /* Verbind het "toggled" event aan de knop */  
    gtk_signal_connect (GTK_OBJECT (button), "toggled",  
                       GTK_SIGNAL_FUNC(togg), (gpointer *)button);  
    gtk_widget_show(button);  
    gtk_widget_show (window);  
  
    gtk_main ();  
    return(0);  
}
```

## Het keuzevakje

Het keuzevakje is een subklasse van de toggle knop. Het kan gebruikt worden om opties te (de)selecteren.

Om een keuzevakje te maken wordt volgende functie gebruikt:

```
GtkWidget* gtk_check_button_new (void);
GtkWidget* gtk_check_button_new_with_label (const gchar *label);
```

De uitleg is dezelfde als voor de indrukknop.

Het voorbeeld:

```
#include <gtk/gtk.h>
void togg(GtkWidget *widget, gpointer *data){
    if (gtk_toggle_button_get_active(GTK_TOGGLE_BUTTON(data)))
        g_print("State is 1\n");
    else
        g_print("State is 0\n");
}
int main( int argc, char *argv[] )
{
    GtkWidget *window;
    GtkWidget *button;
    gtk_init (&argc, &argv);

    /* Maak een nieuw venster */
    window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title (GTK_WINDOW (window), "Check Button");
    /* Verbind een vernietigingsevent aan het venster. */
    gtk_signal_connect (GTK_OBJECT (window), "destroy",
                        GTK_SIGNAL_FUNC(gtk_main_quit), NULL);

    /* Maak een keuzevakje */
    button=gtk_check_button_new_with_label("I'm a check button");

    /* Voeg het vakje toe aan het venster */
    gtk_container_add(GTK_CONTAINER(window),button);
    /* Verbind het "toggled" event aan de knop */
    gtk_signal_connect (GTK_OBJECT (button), "toggled",
                        GTK_SIGNAL_FUNC(togg), (gpointer *)button);
    gtk_widget_show(button);
    gtk_widget_show (window);

    gtk_main ();
    return(0);
}
```

## Label

Labels laten je toe om waar dan ook in een venster tekst te plaatsen.

Om een label te maken gebruik je gewoon de volgende functie:

```
GtkWidget* gtk_label_new(const gchar *text);
```

Met de functie

```
gtk_label_set_text(GtkLabel *label, gchar *text);
```

kun je ten allen tijde de string op een label aanpassen.

```
gtk_label_set_justify(GtkLabel *label, GtkJustification jtype);
```

De `gtk_label_set_justify` functie wordt gebruikt om de tekst op een label uit te vullen. `jtype` kan de volgende waarden aannemen:

- `GTK_JUSTIFY_LEFT` om de tekst links uit te vullen
- `GTK_JUSTIFY_RIGHT` om de tekst rechts te plaatsen
- `GTK_JUSTIFY_CENTER` om de tekst te centreren
- `GTK_JUSTIFY_FILL` om de tekst het gehele label te laten vullen.

```
gtk_label_set_line_wrap (GtkLabel *label, gboolean wrap);
```

wordt gebruikt om de tekst verdeelbaar te maken (ook wel *terugloop* genoemd), zodat het verdeeld kan worden in verschillende stukken wanneer de tekst langer is dan de plaats die het moet innemen. De rest van de regel gaat dan verder op de volgende lijn. Wanneer `wrap` 1 is, zal de tekst overgebracht worden naar de volgende lijn, anders niet.

```
gtk_label_get(GtkLabel *label, gchar **str)
```

wordt gebruikt om de tekst op het label als `str` te krijgen.

## ToolTips

Een Tooltip is een tekst die verschijnt wanneer de muis over een widget beweegt. Je kunt bijvoorbeeld een tip op een knop plaatsen en wanneer je de muis over die knop beweegt dan verschijnt bijvoorbeeld de tekst "verstuur de informatie".

Om dit te doen dient eerst een `GtkToolTips` widget gemaakt te worden:

```
GtkToolTips* gtk_tooltips_new();
```

Vervolgens wordt de tooltip aan een bestaande widget gehecht:

```
gtk_tooltips_set_tip(GtkTooltips *tooltips, GtkWidget *widget,  
                    const gchar *tip_text, const gchar *tip_private);
```

Een klein voorbeeldje:

```
#include <gtk/gtk.h>  
int main( int argc, char *argv[] )  
{  
    GtkWidget *window;  
    GtkWidget *button;  
    GtkTooltips *tip;
```

```

gtk_init (&argc, &argv);
/* Maak een nieuw venster */
window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
gtk_window_set_title (GTK_WINDOW (window), "Tooltips");
/* Verbind een vernietigingsevent aan het venster */
gtk_signal_connect (GTK_OBJECT (window), "destroy",
                    GTK_SIGNAL_FUNC (gtk_main_quit), NULL);
/* Maak een knop */
button=gtk_button_new_with_label("I'm a Button");
/* Voeg de knop toe aan het venster */
gtk_container_add(GTK_CONTAINER(window),button);
/* Maak een tooltips aan*/
tip=gtk_tooltips_new();
/* Verbind deze tooltip aan de knop met tekst */
gtk_tooltips_set_tip(tip, button, "Click me!",NULL);
gtk_widget_show(button);
gtk_widget_show(window);
gtk_main ();
return(0);
}

```

Nog enkele andere functies:

```
gtk_tooltips_enable (GtkTooltips *tooltips);
```

stelt de tooltips in werking.

```
gtk_tooltips_disable (GtkTooltips *tooltips);
```

Stelt de tooltips buiten werking.

Om welke tooltipdata van een widget dan ook te verkrijgen, hebben we het volgende nodig

```
GtkTooltipsData* gtk_tooltips_get_data(GtkWidget *widget);
```

GtkTooltipsData is een struct die er als volgt uitziet:

```

struct _GtkTooltipsData
{
    GtkTooltips *tooltips;
    GtkWidget *widget;
    gchar *tip_text;
    gchar *tip_private;
    GdkFont *font;
    gint width;
    GList *row;
};

```

Om de vertraging van de verschijnende tekst aan te passen:

```
gtk_tooltips_set_delay (GtkTooltips *tip, guint delay)
```

## Combo Box

Een combo box is een aanpasbaar tekstveld gecombineerd met een pull-down menu. Je kunt een waarde ingeven of er één selecteren uit een uitklapbare lijst.

Een combo box kan gemaakt worden met

```
GtkWidget *gtk_combo_new();
```

En we hebben een lijst met opties nodig in de vorm van een GList struct.

```
GList *glist=NULL;
```

De opties kunnen op volgende manier aan de lijst worden toegevoegd

```
GList *g_list_append(GList *list, gchar *option);
```

Dan dient de lijst aan de combo box toegevoegd te worden

```
gtk_combo_set_popdown_strings(GtkCombo *combo, GList *List);
```

De combo box is klaar. Om de geselecteerde opties te lezen gebruik je:

```
gchar *gtk_entry_get_text(GtkEntry *entry);
```

*entry* is in dit geval GTK\_ENTRY(GTK\_COMBO(combo)->entry)).

```
gtk_combo_set_use_arrows(GtkCombo *combo,gint val);
```

Deze functie wordt gebruikt om ervoor te zorgen dat de opwaartse/neerwaartse pijltjestoetsen op het toetsenbord de geselecteerde waarde in een combo box kunnen wijzigen. Wanneer val 0 is, zullen deze toetsen niet werken, anders zullen deze toetsen de waarde wijzigen. Maar wanneer de waarde in de combo box nieuw is (niet in de lijst staat), dan zullen deze toetsen niet werken.

```
gtk_combo_set_use_arrows_always(GtkCombo *combo,gint val);
```

Deze functie is dezelfde als `gtk_combo_set_use_arrows` maar deze versie zorgt ervoor dat ook bij een nieuwe waarde, de pijltjestoetsen blijven werken.

```
gtk_combo_set_value_in_list(GtkCombo *combo, gboolean val,  
                             gboolean ok_if_empty);
```

Wanneer val 1 is, kun je een waarde in de lijst ingeven. Wanneer `ok_if_empty` 1 is mag de waarde leeg gelaten worden.

```
#include <gtk/gtk.h>  
void act(GtkWidget *widget, gpointer *data){  
    g_print((gchar *)data);  
}  
int main( int argc,char *argv[] ) {  
    GtkWidget *window;  
    GtkWidget *combo;  
    GtkWidget *button;  
    GtkWidget *box;  
    GList *list=NULL;  
    list=g_list_append(list,"Slackware");
```

```

list=g_list_append(list,"RedHat");
list=g_list_append(list,"SuSE");
gtk_init (&argc, &argv);

/* Maak een nieuw venster */
window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
gtk_window_set_title (GTK_WINDOW (window), "Combo Box");

/* Verbind een vernietigingsevent aan het venster */
gtk_signal_connect (GTK_OBJECT (window), "destroy",
                    GTK_SIGNAL_FUNC(gtk_main_quit), NULL);

/* Maak een horizontaal vak */
box=gtk_hbox_new(1,0);
gtk_container_add(GTK_CONTAINER(window),box);

/* Maak een combo box */
combo=gtk_combo_new();
/* Stel de lijst in */
gtk_combo_set_popdown_strings(GTK_COMBO(combo),list);
/* Zorg ervoor dat de pijltjestoetsen werken */
gtk_combo_set_use_arrows_always(GTK_COMBO(combo),1);
gtk_box_pack_start(GTK_BOX(box), combo,1,1,1);
button=gtk_button_new_with_label("Write it");
gtk_signal_connect(GTK_OBJECT(button),"clicked",GTK_SIGNAL_FUNC(act),
                  gtk_entry_get_text(GTK_ENTRY(GTK_COMBO(combo)->entry)));
gtk_box_pack_start(GTK_BOX(box), button,1,1,1);
gtk_widget_show(box);
gtk_widget_show(combo);
gtk_widget_show(button);
gtk_widget_show (window);

gtk_main ();
return(0);
}

```

Commentaren zijn altijd welkom.

<p>Site onderhouden door het LinuxFocus editors team © Özcan Güngör "some rights reserved" see <a href="http://www.LinuxFocus.org">linuxfocus.org/license/ http://www.LinuxFocus.org</a></p>	<p>Vertaling info: tr --&gt; -- : Özcan Güngör &lt;<a href="mailto:ozcangungor(at)netscape.net">ozcangungor(at)netscape.net</a>&gt; en --&gt; tr: Özcan Güngör &lt;<a href="mailto:ozcangungor%28at%29netscape.net">ozcangungor%28at%29netscape.net</a>&gt; en --&gt; nl: Samuel Derosus &lt;<a href="mailto:Zenodotus/at/linux.be">Zenodotus/at/linux.be</a>&gt;</p>
--	---