



GUI Programming with Perl / GTK

Anuradha Weeraman

`anu@taprobane.org`

<http://www.linux.lk/~anu/>

23 May 2006



Contents

- Overview
- GUI Toolkits
- Hello World
- Layout
- C -> Perl
- GUI Builders
- CPAN



Overview

- What's a widget?
- What's a GUI?
- What's a GUI toolkit?
- What's GTK?
- What's GTK-Perl?
- How is GUI programming different?



GUI Toolkits

- Athena Widget Library
- OSF Motif
- Xforms
- FLTK
- the GIMP Toolkit
- Qt Toolkit
- LessTif



Hello World

```
#!/usr/bin/perl

use Gtk2 '-init';

$window = Gtk2::Window->new;

$label = Gtk2::Label->new ("Hello World");

$window->add ($label);
$window->show_all;

Gtk2->main;
```





Hello World – Part 2

```
#!/usr/bin/perl

use Gtk2 '-init';

$window = Gtk2::Window->new;

$window->signal_connect(
    destroy => sub { Gtk2->main_quit }
);

$label = Gtk2::Label->new ("Hello World");

$window->add ($label);
$window->show_all;

Gtk2->main;
```





Hello World – Part 3

```
#!/usr/bin/perl

use Gtk2 '-init';

$window = Gtk2::Window->new;
$window->set_title("Hello");
$window->signal_connect(destroy => sub { Gtk2->main_quit });

$button = Gtk2::Button->new ("Greetings Earthling");
$button->signal_connect(clicked => sub { Gtk2->main_quit });

$window->add ($button);
$window->show_all;

Gtk2->main;
```





Hello World – Part 4

```
#!/usr/bin/perl

use Gtk2 '-init';

sub quit_program {
    Gtk2->main_quit;
    print "Program has stopped.\n";
}

$window = Gtk2::Window->new;
$window->set_title("Hello");
$window->signal_connect(destroy => \&quit_program);

$button = Gtk2::Button->new ("Greetings Earthling");
$button->signal_connect(clicked => \&quit_program);

$window->add ($button);
$window->show_all;

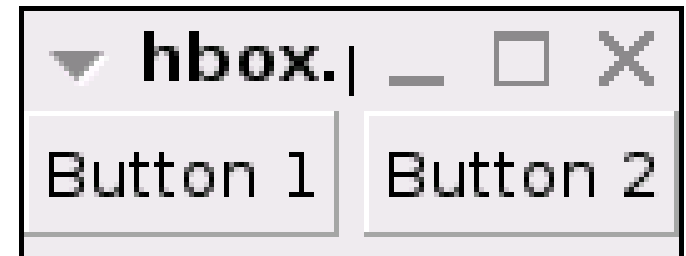
Gtk2->main;
```





Layout - HBox

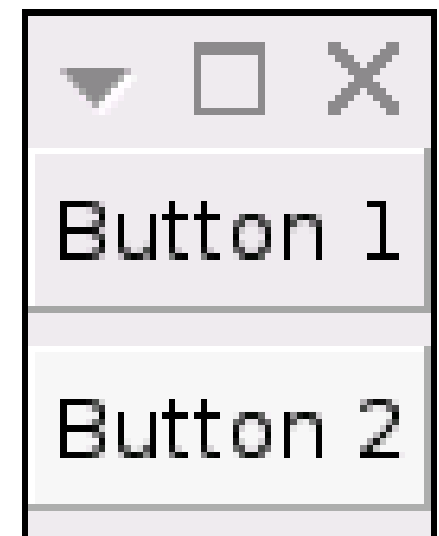
```
$window = Gtk2::Window->new;  
$hbox = Gtk2::HBox->new;  
  
$button_1 = Gtk2::Button->new ("Button 1");  
$button_2 = Gtk2::Button->new ("Button 2");  
  
$hbox->pack_start ($button_1, 0, 0, 0);  
$hbox->pack_start ($button_2, 0, 0, 0);  
  
$window->add ($hbox);
```





Layout - VBox

```
$window = Gtk2::Window->new;  
$vbox = Gtk2::VBox->new;  
  
$button_1 = Gtk2::Button->new ("Button 1");  
$button_2 = Gtk2::Button->new ("Button 2");  
  
$vbox->pack_start ($button_1, 0, 0, 0);  
$vbox->pack_start ($button_2, 0, 0, 0);  
  
$window->add ($vbox);
```





C -> Perl

- Consistent naming
- One-to-one mapping
- Object-oriented
- Transparently handles type-casting, reference counting etc.
- Exceptions allowed where Perl capabilities afford a cleaner API – multiple return values, string / array function parameters



Function Name Translation

<code>g_</code>	<code>-></code>	<code>Glib</code>
<code>gtk_</code>	<code>-></code>	<code>Gtk2</code>
<code>gdk_</code>	<code>-></code>	<code>Gtk2::Gdk</code>
<code>gdk_pixbuf_</code>	<code>-></code>	<code>Gtk2::Gdk::Pixbuf</code>
<code>pango_</code>	<code>-></code>	<code>Gtk2::Pango</code>



Function Name Translation

<code>gtk_window_</code>	<code>-></code>	<code>Gtk2::Window</code>
<code>gtk_button_</code>	<code>-></code>	<code>Gtk2::Button</code>
<code>gtk_window_new</code>	<code>-></code>	<code>Gtk2::Window->new</code>
<code>gtk_button_new</code>	<code>-></code>	<code>Gtk2::Button->new</code>



Function Parameters

```
gtk_window_set_title  
(GtkWindow *window, gchar string)
```

becomes

```
$window->set_title ($string)
```



Function Parameters

GList replaced by variable number of arguments:

```
gtk_window_set_icon_list (GtkWindow * window, GList * list)
>window->set_icon_list (@icons)
```

Same with the array moved to the end of the parameter list:

```
gtk_list_insert_items (GtkList *list, GList *items, gint position)
$list->insert_items ($position, @items)
```

Array parameter and integer with the size of that array, replaced by variable number of arguments:

```
gtk_curve_set_vector (GtkCurve *curve, int veclen, gfloat vector[])
$curve->set_vector (@vector)
```

Same with the array moved to the end of parameter list:

```
gtk_item_factory_create_items (GtkItemFactory * ifactory,
    guint n_entries, GtkItemFactoryEntry * entries,
    gpointer callback_data)
$itemfactory->create_items ($callback_data, @entries)
```



Return Values

```
gtk_window_get_size (GtkWindow *window, gint *width,  
                    gint *height)
```

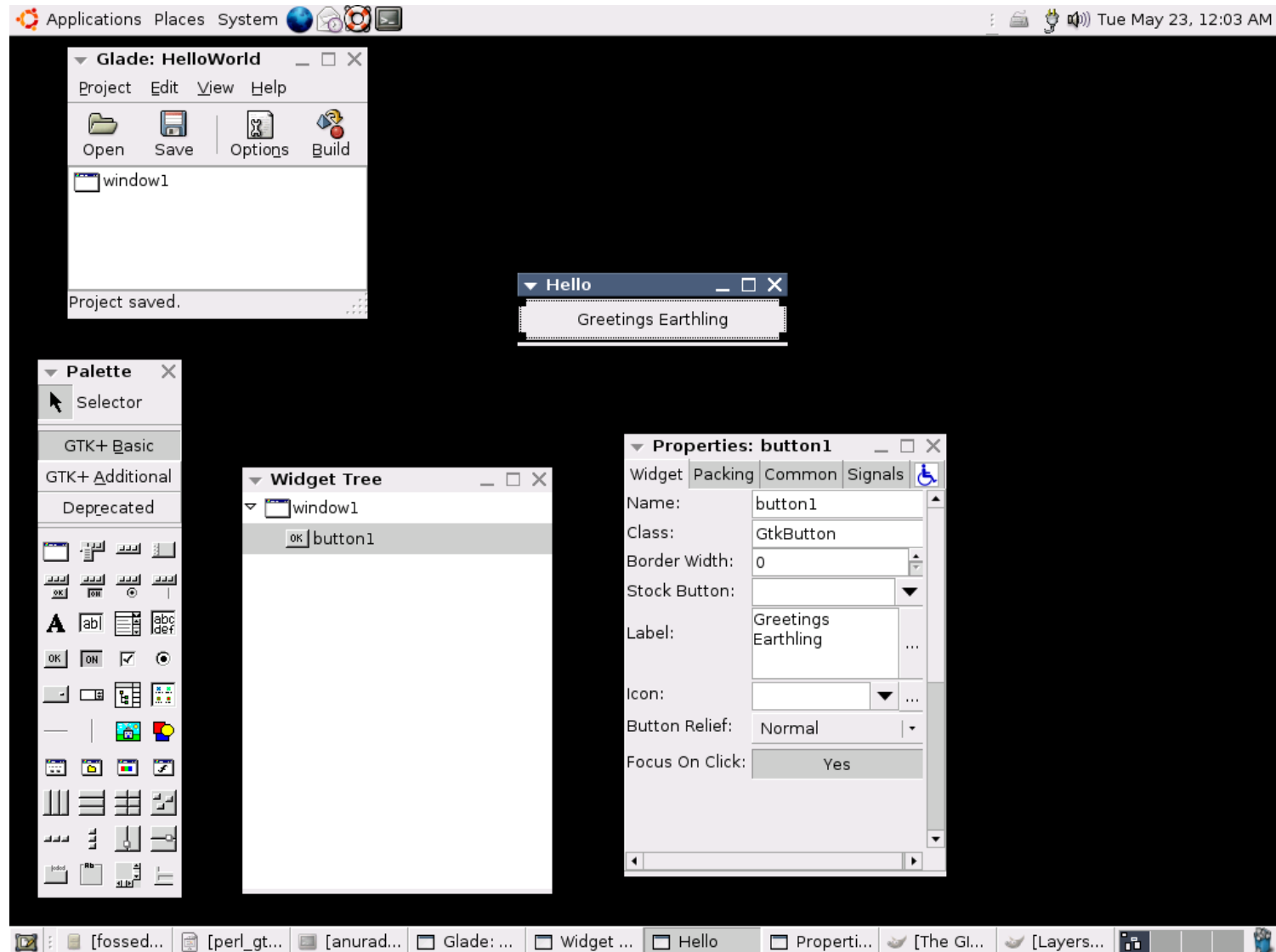
```
($width, $height) = $window->get_size
```

```
gtk_calendar_get_date (GtkCalendar * calendar,  
                     guint year, guint month, guint day)
```

```
($year, $month, $day) = $calendar->get_date
```



GUI Builders - Glade





Installing Modules

Download `foo-module.tar.gz`

```
$ tar zxvf foo-module.tar.gz
```

```
$ cd foo-module
```

```
$ perl Configure.PL
```

```
$ make
```

```
$ make test
```

```
# make install
```

OR

use CPAN.



CPAN

- CPAN.org
- Comprehensive Perl Archive Network
- Mirrors all over the world
- Command line shell
- Bundled with standard Perl distribution
- Intuitive module management



CPAN

perl -MCPAN -e shell

cpan> install Term::ReadKey

cpan> install Term::ReadLine

cpan> install Bundle::CPAN

cpan> h or ?



Thank You!

For more information, visit:

<http://www.gtk.org>

<http://www.gtk2-perl.sourceforge.net>

You can download this presentation
from <http://www.linux.lk/~anu>