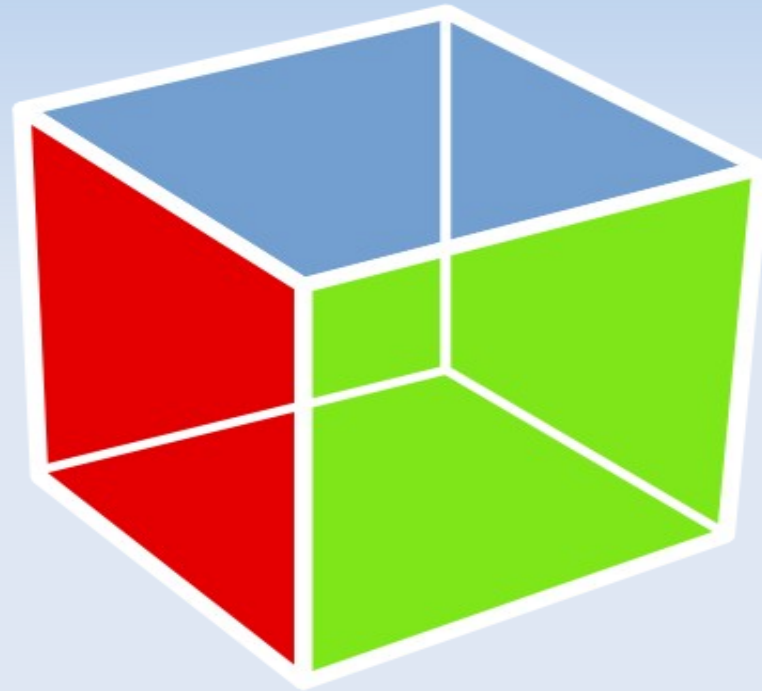


Gtk2



GTK+ (the GIMP ToolKit)
Bratislava, 2008

Gtk2 - Overview

- Graphical toolkit native to X11 (Linux, Unix)
- Ported to Windows and OS X
- Written in C, bindings for Perl, Python, C++, etc
- LGPL license
- Desktops (GNOME, XFCE, ROX, Maemo)
- Installed on all Linux desktops
- UTF-8, i18n, accessibility

Gtk2 – Internals

- Uses an event loop which:
 - Allows nested calls (ex: dialogs, libnotify)
 - Is shared with other libraries (gstreamer, GIO, etc)
- Event based
- Memory management through reference counts
- Object oriented (even in C)
- Hierarchical layout

Gtk2 – Widgets family

- *With a 'window'*
 - Windows, buttons, toggles, data entry and menus
 - MVC (trees, lists, icon views and text views)
- *Without a 'window'*
 - Label, image, system tray
- Containers
 - Boxes, panes, table, notebook, scroll, layout
- Custom
 - Mozilla, terminal, movie player, source view
 - Canvas

Gtk2 – C

```
#include <gtk/gtk.h>
```

```
int main(int argc, char *argv[] ) {
```

```
    gtk_init(&argc, &argv);
```

```
    GtkWidget *window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
```

```
    GtkWidget *label = gtk_label_new("Hello World");
```

```
    gtk_container_add(GTK_CONTAINER(window), label);
```

```
    gtk_widget_show_all(window);
```

```
    // The main event loop
```

```
    gtk_main();
```

```
    return 0;
```

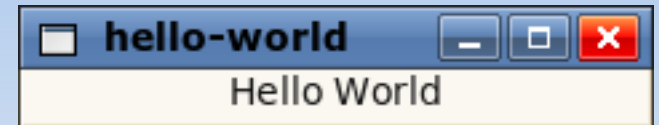
```
}
```

```
/**
```

```
 * Compile with:
```

```
 * gcc -o hello-world -g -O2 -std=c99 `pkg-config --cflags --libs gtk+-2.0` hello-world.c
```

```
 */
```



Gtk2 – Perl

```
#!/usr/bin/perl

use strict;
use warnings;

use Gtk2 '-init';

exit main();

sub main {

    my $window = Gtk2::Window->new();
    my $label = Gtk2::Label->new('Hello World!');

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

    # Main event loop
    Gtk2->main();

    return 0;
}
```



Gtk2 – Shell script

```
#!/bin/bash
```

```
## SVN graphical user interface
```

```
# Ask which files to commit
```

```
files=$( zenity --list --text="Files to commit" \  
  --column=status --column=file --multiple \  
  --print-column=2 `svn status | grep -v '^[?]'` )
```

```
[ $? -eq 0 ] || exit 1  
[ "$files" = "" ] && exit 2
```

```
# Ask for the commit message
```

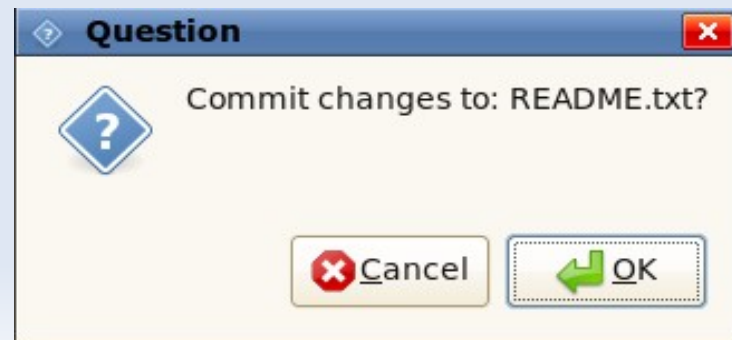
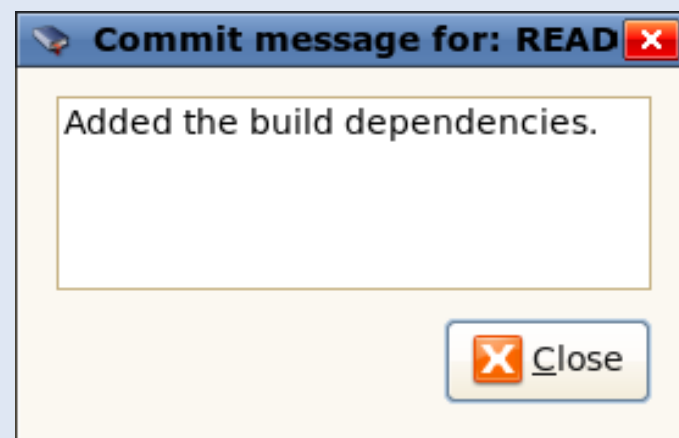
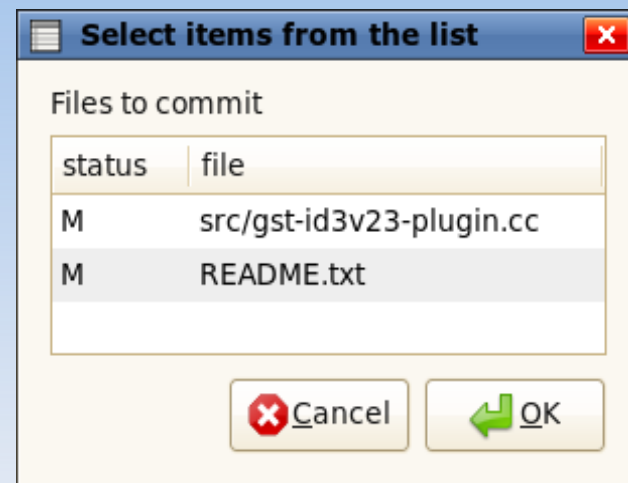
```
message=$( zenity --text-info --editable \  
  --title "Commit message for: $files" )
```

```
[ $? -eq 0 ] || exit 3  
[ "$message" = "" ] && exit 3
```

```
# Confirm the commit
```

```
zenity --question --text="Commit changes to: $files?"  
[ $? -eq 0 ] || exit 4
```

```
echo svn commit -m "$message" $files
```



Gtk2 Example - Color picker

```
# The color chooser and the main window  
my $color_selection = Gtk2::ColorSelection->new();  
$color_selection->show_all();  
my $win = Gtk2::Window->new('toplevel');  
$win->add($color_selection);
```

```
# Create a system tray icon  
my $tray = Gtk2::StatusIcon->new_from_stock('gtk-color-picker');
```

```
# Set the HTML value of the color selected  
$color_selection->signal_connect(color_changed => sub {  
    # Get the color's HTML value  
    my $color = $color_selection->get_current_color();  
    my $html = $color_selection->palette_to_string($color);
```

```
# Update the clipboards with the HTML value  
my $clipboard = Gtk2::Clipboard->get(Gtk2::Gdk->SELECTION_CLIPBOARD);  
$clipboard->set_text($html_value);  
});
```

```
# Show/hide the main window  
$tray->signal_connect(activate => sub {$win->visible ? $win->hide() : $win->present();});
```

```
Gtk2->main();
```



Gnome Example - Terminal

```
use Glib qw(TRUE FALSE);  
use Gtk2 -init;
```

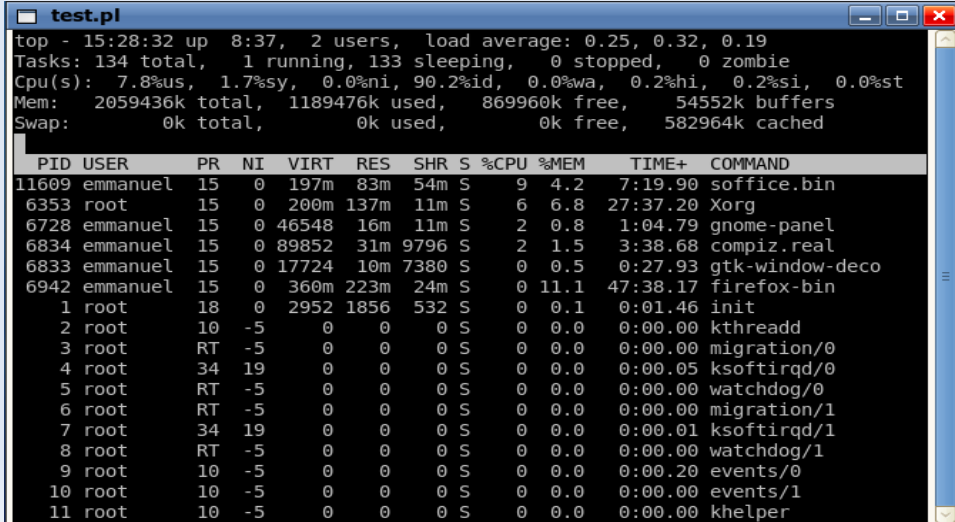
```
use Gnome2::Vte;  
my $terminal = Gnome2::Vte::Terminal->new();
```

```
# Set up scrolling in the terminal  
my $scrollbar = Gtk2::VScrollbar->new();  
$scrollbar->set_adjustment($terminal->get_adjustment());
```

```
# Start a command in the terminal (doesn't have to be a shell)  
$terminal->fork_command('/usr/bin/top', ['top'], undef, '/tmp', FALSE, FALSE, FALSE);  
$terminal->signal_connect(child_exited => sub { Gtk2->main_quit });
```

```
# Layout  
my $window = Gtk2::Window->new();  
my $hbox = Gtk2::HBox->new();  
$hbox->pack_start($terminal, TRUE, TRUE, 0);  
$hbox->pack_start($scrollbar, FALSE, FALSE, 0);  
$window->add($hbox);
```

```
# Start the application  
$window->show_all();  
Gtk2->main();
```



```
test.pl  
top - 15:28:32 up 8:37, 2 users, load average: 0.25, 0.32, 0.19  
Tasks: 134 total, 1 running, 133 sleeping, 0 stopped, 0 zombie  
Cpu(s): 7.8%us, 1.7%sy, 0.0%ni, 90.2%id, 0.0%wa, 0.2%hi, 0.2%si, 0.0%st  
Mem: 2059436k total, 1189476k used, 869960k free, 54552k buffers  
Swap: 0k total, 0k used, 0k free, 582964k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11609	emmanuel	15	0	197m	83m	54m	S	9	4.2	7:19.90	soffice.bin
6353	root	15	0	200m	137m	11m	S	6	6.8	27:37.20	Xorg
6728	emmanuel	15	0	46548	16m	11m	S	2	0.8	1:04.79	gnome-panel
6834	emmanuel	15	0	89852	31m	9796	S	2	1.5	3:38.68	compiz.real
6833	emmanuel	15	0	17724	10m	7380	S	0	0.5	0:27.93	gtk-window-deco
6942	emmanuel	15	0	360m	223m	24m	S	0	11.1	47:38.17	firefox-bin
1	root	18	0	2952	1856	532	S	0	0.1	0:01.46	init
2	root	10	-5	0	0	0	S	0	0.0	0:00.00	kthreadd
3	root	RT	-5	0	0	0	S	0	0.0	0:00.00	migration/0
4	root	34	19	0	0	0	S	0	0.0	0:00.05	ksoftirqd/0
5	root	RT	-5	0	0	0	S	0	0.0	0:00.00	watchdog/0
6	root	RT	-5	0	0	0	S	0	0.0	0:00.00	migration/1
7	root	34	19	0	0	0	S	0	0.0	0:00.01	ksoftirqd/1
8	root	RT	-5	0	0	0	S	0	0.0	0:00.00	watchdog/1
9	root	10	-5	0	0	0	S	0	0.0	0:00.20	events/0
10	root	10	-5	0	0	0	S	0	0.0	0:00.00	events/1
11	root	10	-5	0	0	0	S	0	0.0	0:00.00	khelper

Widget properties and layout

- Too many properties
- Temporary layout containers
- Callback setting is repetitive

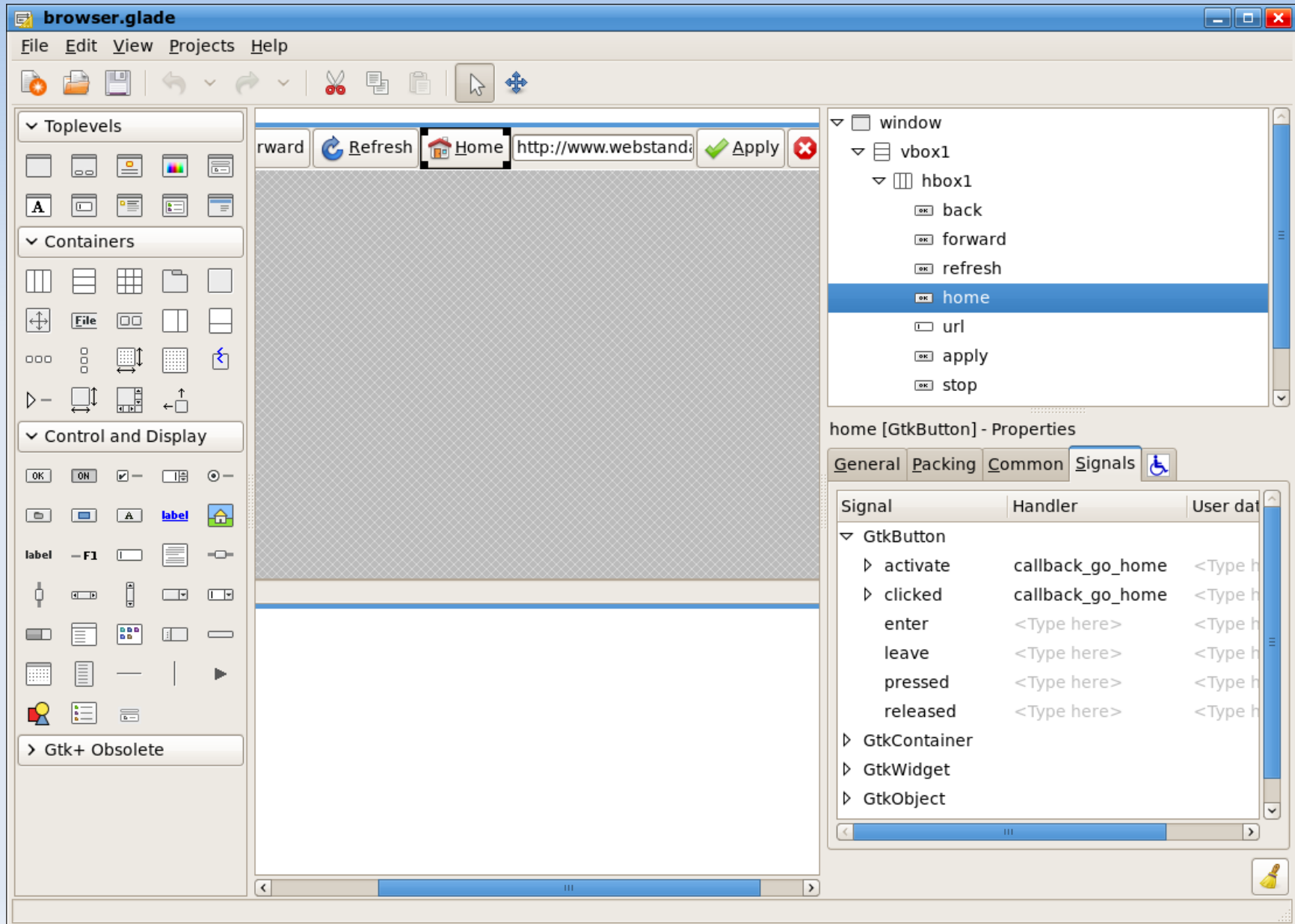
```
my $vbox = Gtk2::VBox->new;  
$vbox->set_border_width(2);  
$vbox->set_spacing(2);  
$vbox->pack_start($hbox, FALSE, FALSE, 0);  
$vbox->pack_start($pane, TRUE, TRUE, 0);  
$vbox->pack_start($status, FALSE, FALSE, 0);
```

```
my $window = Gtk2::Window->new('toplevel');  
$window->set_position('center');  
$window->set_default_size(600, 400);  
$window->set_border_width(0);  
$window->set_title($NAME);  
$window->signal_connect(  
    delete_event => \&close_program  
);  
$window->add_accel_group($accels);
```

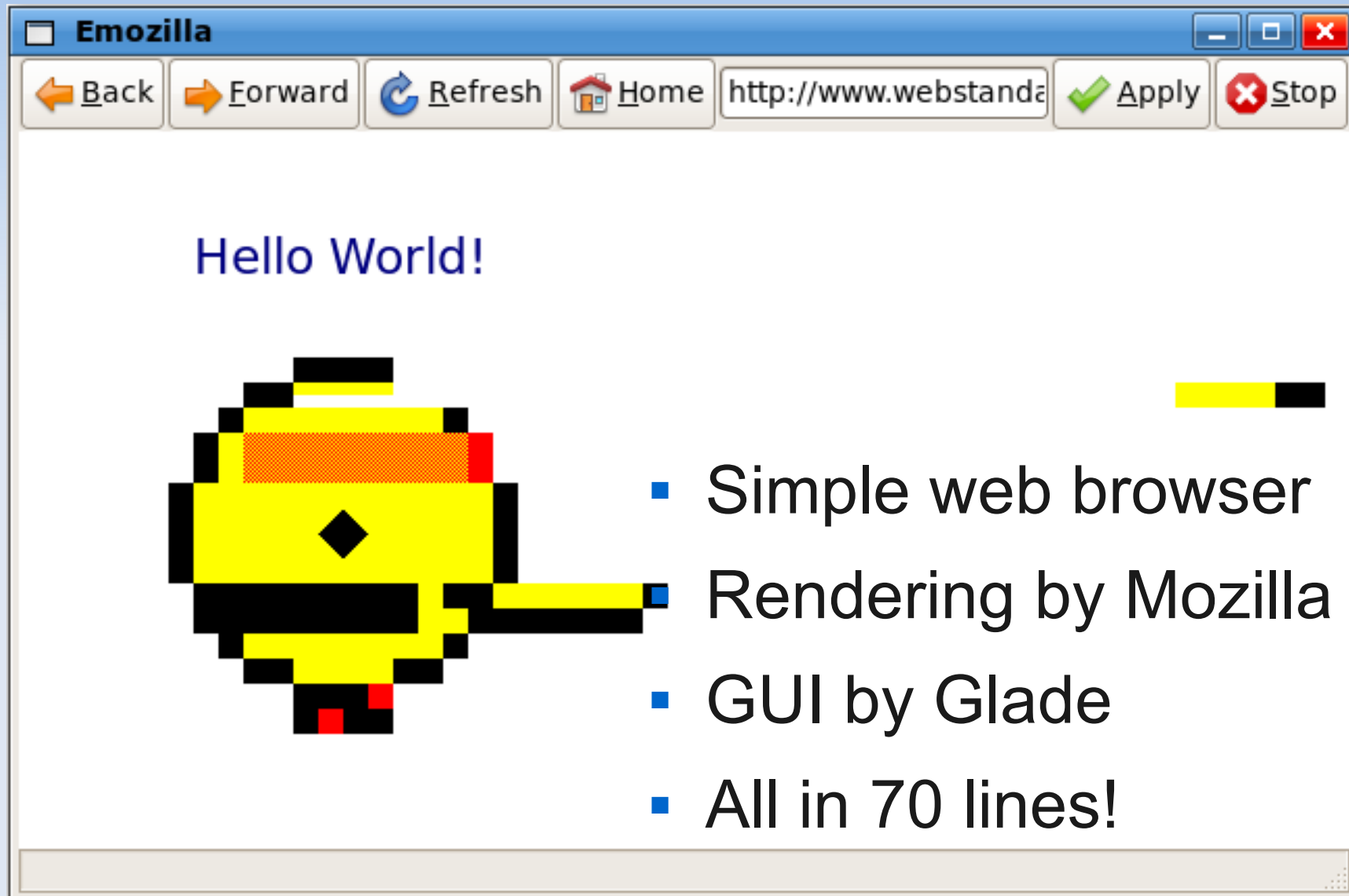
Gtk2::GladeXML

- Easy graphical GUI builder
- All Gtk2 and Gnome widgets are available
 - Properties
 - Signals
 - Packing
- XML file format
 - Simple to read, edit and manipulate
 - The same for all bindings (C/C++, Perl, Python)

Glade



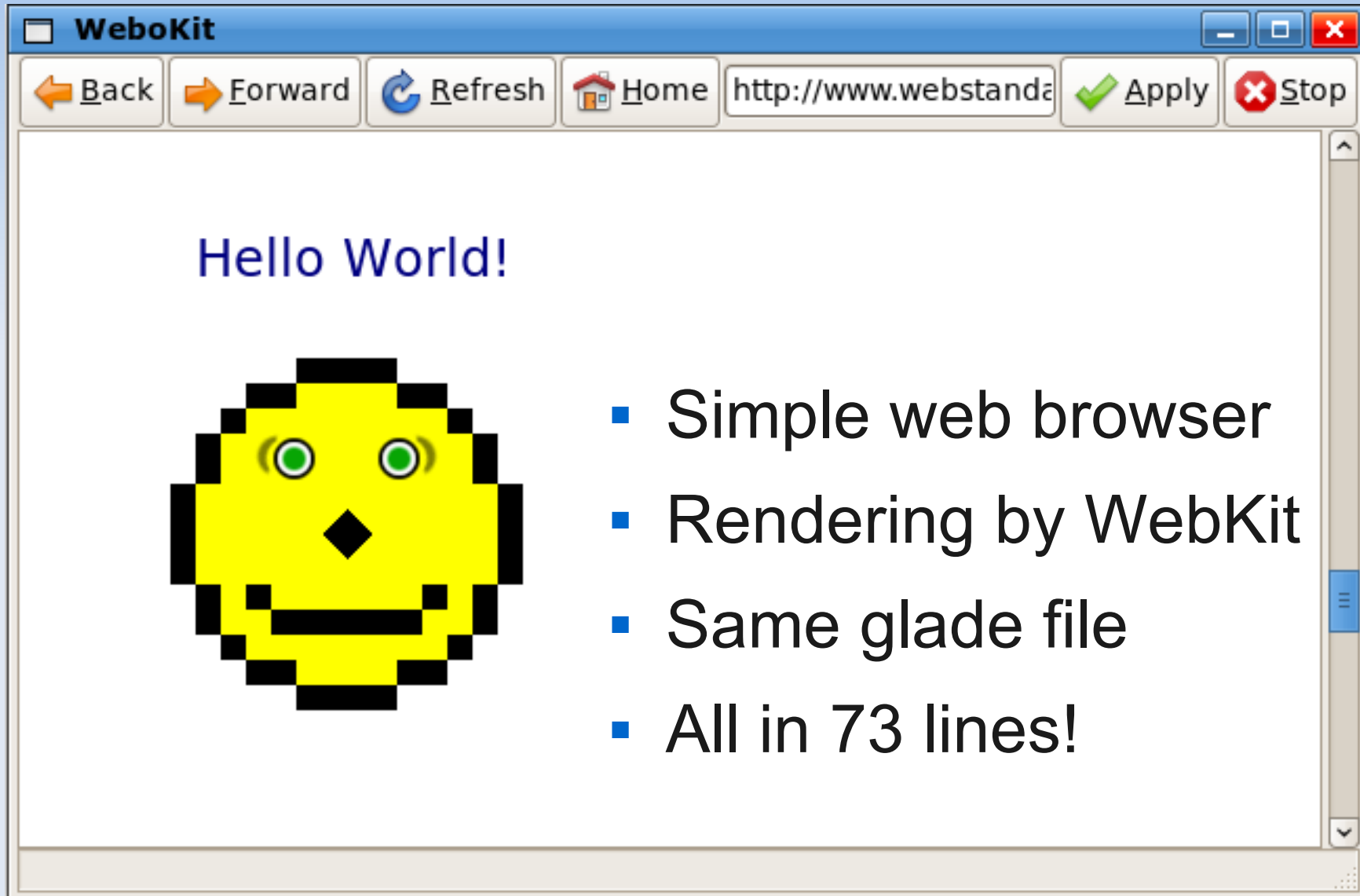
Glade Example - MozEmbed



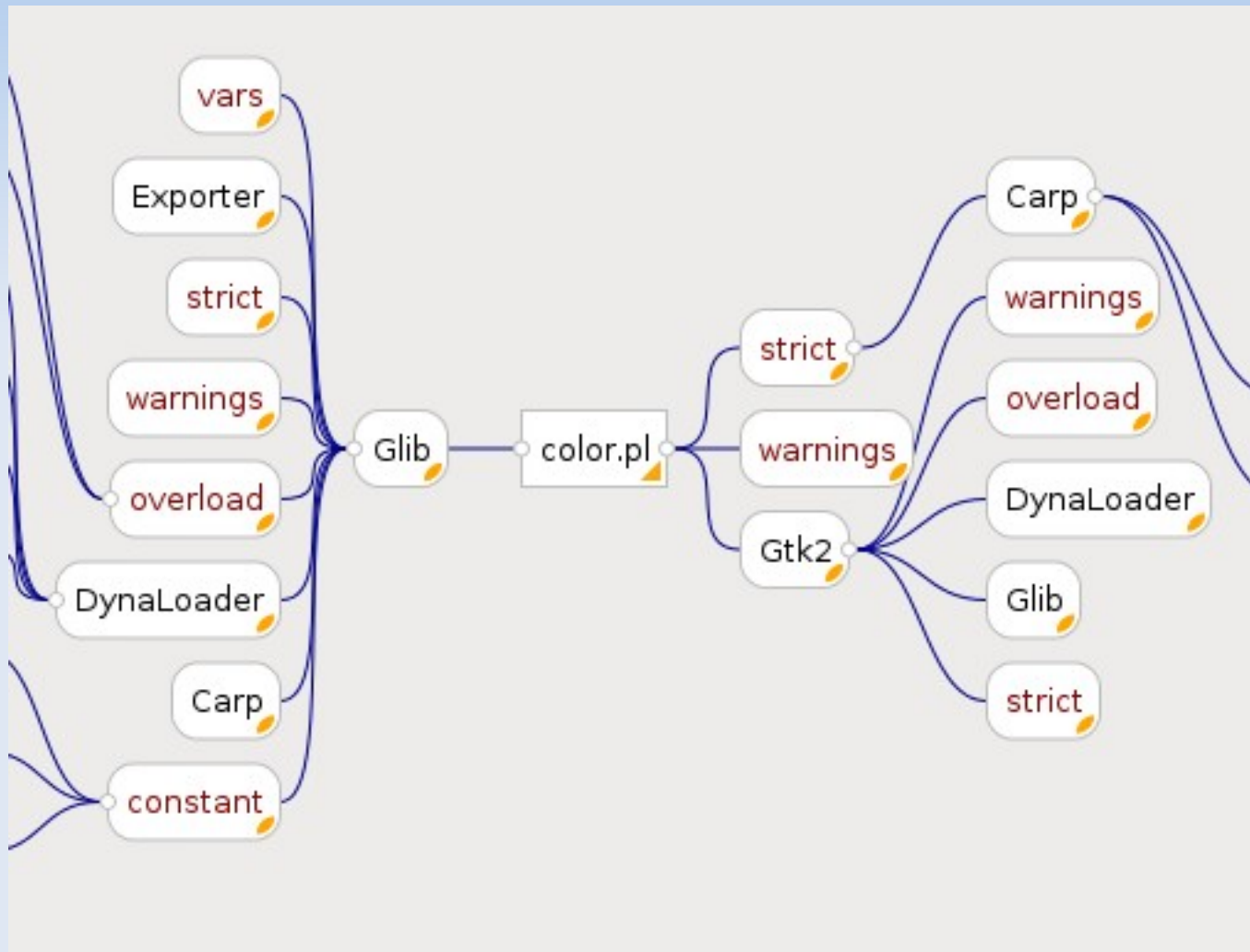
The screenshot shows a web browser window titled "Emozilla". The address bar contains "http://www.webstanda". The browser interface includes buttons for Back, Forward, Refresh, Home, Apply, and Stop. The main content area displays "Hello World!" in blue text. Below the text is a pixelated yellow character with a black diamond for an eye and a long yellow tongue sticking out. To the right of the character is a small yellow and black horizontal bar.

- Simple web browser
- Rendering by Mozilla
- GUI by Glade
- All in 70 lines!

Glade Example - WebKit



Mind Map Example



Gtk2->main_quit();

