

# Developing with GTK+

Owen Taylor  
Red Hat Software  
June 26, 1999

## Outline

- Introducing GTK+
- Principles of GTK+
- Building an Application
- Beyond GTK+

## History of GTK+

**Fall 1996** GTK+ started as part of the GIMP project by Spencer Kimball and Peter Mattis.

**Spring 1998** GTK+ version 1.0 released.  
GIMP version 1.0 released

**Winter 1999** GTK+ version 1.2 released

**Spring 1999** GNOME version 1.0 released

**Summer 1999** Development of version 1.4 of GTK+...

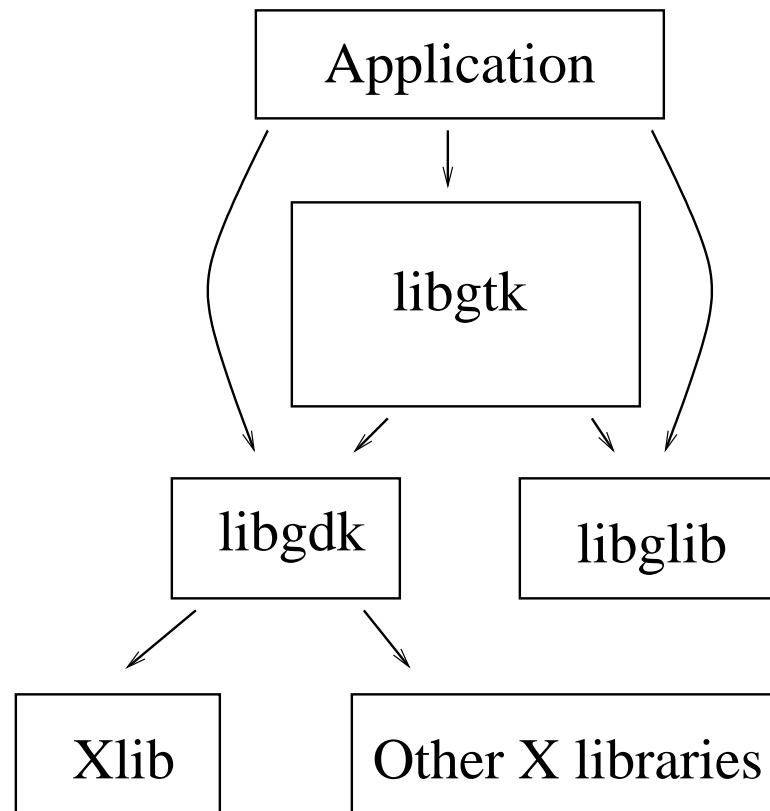
## Benefits of GTK+

- Convenient, but powerful programming interface
- Widespread portability and availability
- Modern appearance
- Unrestrictive Licensing
- Availability of Language Bindings

## Some projects using GTK+

- GIMP (of course...)
- GNOME
- Mozilla
- AbiWord

## The GTK+ Libraries



# Hello World

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

void clicked (GtkWidget *widget, gpointer data)
{
    gtk_main_quit();
}

int main (int argc, char **argv)
{
    GtkWidget *window, *button;

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

```
window = gtk_window_new (GTK_WINDOW_TOPLEVEL);

button = gtk_button_new_with_label ("Hello World");
gtk_container_add (GTK_CONTAINER (window), button);

gtk_signal_connect (GTK_OBJECT (button), "clicked",
                   GTK_SIGNAL_FUNC (clicked), NULL);

gtk_widget_show_all (window);

gtk_main();

return 0;
}
```



## Event Driven Programming

- All actions done within “main loop”
- Receive events from user, dispatch to program
- Callbacks by *signals*

## Object Orientation

- “Methods” take object as first parameter
- Polymorphism - can call methods for parent classes as well as object’s own class.

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

## Signals

```
gtk_signal_connect (GTK_OBJECT (button), "clicked",  
                  GTK_SIGNAL_FUNC (clicked), NULL);
```

- Callbacks within main loop
- Allow *notification* and *customization*
- Can pass in callback data

# Inheritance

## Casting Macros

- Each class has standard macros
- `GTK_[class]()` casts to class *with checking*
- `GTK_WINDOW(button)` will producing warning
- Checks are efficient but can be disabled at compile time

## Geometry Negotiation

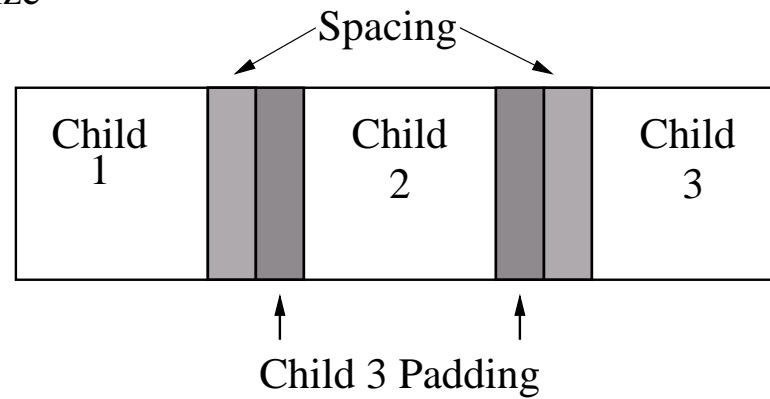
- *requisition* is amount of space widget needs based on its contents
- Containers request space based on size of children
- Then size allocated from top downward
- Each child is given an *allocation*. This may be bigger than the requisition.

# Geometry Example

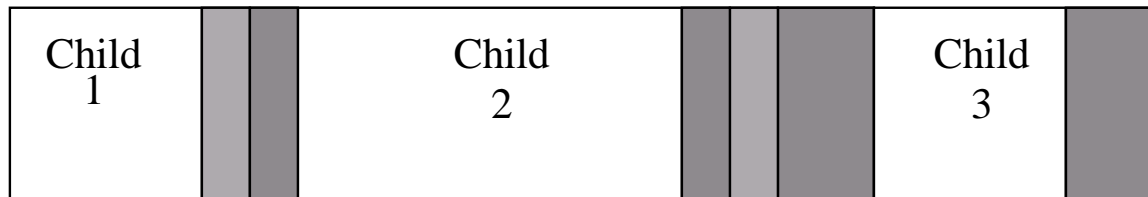
# Packing Boxes

	Expand	Fill
Child 1	FALSE	FALSE
Child 2	TRUE	TRUE
Child 3	TRUE	FALSE

Normal Size



Expanded by User





## Events and Signals

- Lowlevel events sent from X server
- Corresponding signals sent to appropriate widget
- Widgets generate highlevel events
- Event signals have a distinct signature
- Return value determines propagation.

## Reference Counting

- Simple form of *garbage collection*
- Keep track of number of people who care about an object
- Count starts at one, increased by a `ref()` operation, decreased by a `unref()` operation
- When count goes to zero, object is freed
- Initial reference count for GTK+ is special — taken over when object added to container.

## Acknowledgements

- Internationalization of GTK+  
Takashi Matsuda Akira Higuchi
- Original Authors of GTK+  
Peter Mattis Spencer Kimball Josh Macdonald
- The GTK+ Team  
Shawn Amundson Jerome Bolliet Damon Chaplin  
Tony Gale Jeff Garzik Lars Hamann  
Raja Harinath Carsten Haitzler Tim Janik  
Stefan Jeske Elliot Lee Raph Levien  
Ian Main Frederico Meña Paolo Molaro  
Jay Painter Manish Singh