

gtkmm

CS 247

University of Waterloo

cs247@student.cs.uwaterloo.ca

July 4, 2014

Overview

- 1 Basics
- 2 Events
- 3 Containers
- 4 Useful Widgets
- 5 Images
- 6 Dialog Boxes
- 7 Menus and Toolbars

The Main Program

```
#include <gtkmm.h>

int main(int argc, char * argv[]) {
    Gtk::Main kit(argc, argv);
    Gtk::Window window;
    Gtk::Main::run(window);
    return 0;
}
```

Initialize Gtk, create the window for your program and run it. A common design choice is to inherit from `Gtk::Window`.

Buttons are a type of widget.

- You may set the label via the constructor:
`Gtk::Button(std::string label)`
- Or with `Gtk::Button::set_label(std::string label)`
- Stock buttons are available
- Usually used with events

Events allow the user to interact with the program. The code:

```
button.signal_clicked.connect(sigc::mem_fun(  
    *this, &onButtonClicked));}
```

links the `Gtk::button` button's click event to the function `onButtonClicked`.

Sometimes, we want to pass arguments to our event handlers. For this, wrap `sigc::bind` around `sigc::mem_fun`.

```
button.signal_clicked().connect(  
    sigc::bind(  
        sigc::mem_fun(*this, &onButtonClicked),  
        "I'm an argument!");
```

This code will call `onButtonClicked` with the argument "I'm an argument" when the button is clicked.

Most widgets in gtkmm can contain only one child (including `Gtk::Window`). To add multiple elements to a widget, you must pack them first in a container. Typical containers:

- Horizontal box: `Gtk::HBox`
- Vertical box: `Gtk::VBox`
- Table: `Gtk::Table`

Add items to the box containers with `Gtk::Box::pack_start()` and `Gtk::Box::pack_end()` Add items to the table with `Gtk::Table::attach()`

Useful Widgets

- `Gtk::Frame`
- `Gtk::Label`
- `Gtk::CheckButton`
- `Gtk::RadioButton`
- `Gtk::Entry`

- `Gtk::Image`
- `Gdk::Pixbuf` (note: `Gdk`)
- Use `create_from_file()` on `Pixbuf`
- Use `set()` on `Images` to change the `Pixbuf`
- Create images with either a filename as a string or a `Glib::RefPtr` to a `PixBuf` as an argument.
- `Gtk::Image` does not receive events but can be embedded in a button.

Dialog Boxes

Dialog boxes are broken into two segments: a vbox and an action area.

- Access vbox: `get_vbox`
- Access action area: `get_action_area`

Transfer control and show the dialog with `run`.

Menus and Toolbars

- Actions and action groups are shared between menus and toolbars
- Layout is specified via xml-style string
- Create Action Group via: `Gtk::ActionGroup::create()`
- Create Actions via: `Gtk::Action::create()` takes an identity and the display name as parameters, accepts stock types (e.g. Cut, Paste, etc)
- `set sensitive()` disables or enables actions / action groups.

End