

# PoP C++ Série 6 corrigé

## Exercice 2 (niveau 1) : Layout et Label

La solution ci-dessous choisit de faire afficher un passage à la ligne seulement quand on clique sur le destinataire (Madame ou Monsieur). L'ensemble du code source étant disponible dans un fichier archive GTKLabel, on ne reprend ici que layoutbuttons.cc Remarque la différence d'initialisation de m\_Box\_Top d'une part (Verticalement) et de m\_Box1 et m\_Box2 d'autre part (Horizontalement). Le label m\_Fin apparaît sur 2 lignes dans l'interface grâce à \n

L'organisation spatiale de l'interface est obtenue avec la bonne répartition des widgets entre m\_Box\_Top et les deux autres.



```
#include "layoutbuttons.h"
#include "iostream"
using namespace std;

LayoutButtons::LayoutButtons() :
    m_Box_Top(Gtk::ORIENTATION_VERTICAL),
    m_Box1(Gtk::ORIENTATION_HORIZONTAL, 10),
    m_Box2(Gtk::ORIENTATION_HORIZONTAL, 10),
    m_Label_Salutation("Formule de Salutation"),
    m_Label_Destinataire("A qui s'adresse-t-on ?"),
    m_Label_Fin("Les meilleures choses ont une fin\n"
               " éventuellement sur plusieurs lignes"),
    m_Button_Hello("Hello"),
    m_Button_Bonjour("Bonjour"),
    m_Button_Monsieur("Monsieur"),
    m_Button_Madame("Madame"),
    m_Button_Fin("Fin")
{
    // Set title and border of the window
    set_title("layout buttons");
    set_border_width(0);

    // Add outer box to the window (because the window
    // can only contain a single widget)
    add(m_Box_Top);

    //Put the inner boxes and the separator in the outer box:
    m_Box_Top.pack_start(m_Label_Salutation);
    m_Box_Top.pack_start(m_Box1);
    m_Box_Top.pack_start(m_Separator1);
    m_Box_Top.pack_start(m_Label_Destinataire);
    m_Box_Top.pack_start(m_Box2);
    m_Box_Top.pack_start(m_Separator2);
    m_Box_Top.pack_start(m_Label_Fin);
    m_Box_Top.pack_start(m_Button_Fin);
```

```

// Set the inner boxes' borders
m_Box2.set_border_width(10);
m_Box1.set_border_width(10);

// Put Hello / Bonjour buttons in Box1:
m_Box1.pack_start(m_Button_Hello);
m_Box1.pack_start(m_Button_Bonjour);

// Put Madame / Monsieur buttons in Box2:
m_Box2.pack_start(m_Button_Madame);
m_Box2.pack_start(m_Button_Monsieur);

// Connect the clicked signal of the button to
// thier signal handler
m_Button_Hello.signal_clicked().connect(sigc::mem_fun(*this,
    &LayoutButtons::on_button_clicked_Hello) );

m_Button_Bonjour.signal_clicked().connect(sigc::mem_fun(*this,
    &LayoutButtons::on_button_clicked_Bonjour) );

m_Button_Monsieur.signal_clicked().connect(sigc::mem_fun(*this,
    &LayoutButtons::on_button_clicked_Monsieur) );

m_Button_Madame.signal_clicked().connect(sigc::mem_fun(*this,
    &LayoutButtons::on_button_clicked_Madame) );

m_Button_Fin.signal_clicked().connect(sigc::mem_fun(*this,
    &LayoutButtons::on_button_clicked_Fin) );

// Show all children of the window
show_all_children();
}
LayoutButtons::~~LayoutButtons(){}

void LayoutButtons::on_button_clicked_Hello(){
    cout << "Hello " ;
}

void LayoutButtons::on_button_clicked_Bonjour(){
    cout << "Bonjour " ;
}

void LayoutButtons::on_button_clicked_Monsieur(){
    cout << "Monsieur" << endl;
}

void LayoutButtons::on_button_clicked_Madame(){
    cout << "Madame" << endl;
}

void LayoutButtons::on_button_clicked_Fin(){
    hide(); //to close the application.
}

```