

PoP C++ Série 6

H1 :

1) Usage de GTKmm 3 pour l'interface graphique utilisateur : séparation des fonctionnalités

H2 : MOOC [MOOC Introduction à la programmation orientée objet \(en C++\)](#) Série semaine 5: Polymorphisme

Usage de GTKmm 3 pour l'interface graphique utilisateur : séparation des fonctionnalités

Exercice 1. (niveau 0) : [Rappel pour le dessin](#)

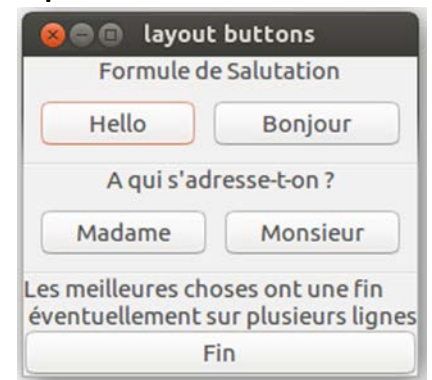
Cet exemple comporte deux modules ; le module **myevent** étend la classe **MyArea** vue la semaine dernière et la combine avec la classe **MyEvent** qui établit un lien entre l'action des boutons et l'affiche.

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

A partir du concept de **Gtk ::Box** exploité dans l'exercice 1 ci-dessus, il s'agit ici de produire un programme qui combine l'affichage de texte avec **Gtk ::Label** et des Buttons sur plusieurs lignes et (éventuellement) colonnes.

Un clic sur l'un des 4 boutons Hello, Bonjour, Madame, Monsieur doit produire l'affichage de ce mot dans le terminal. Un clic sur Fin doit quitter l'application.

reproduire cette interface :



Pour le layout avec **Gtk ::Box** relire l'exercice1 ; il n'y a pas de nouveauté.

Pour afficher du texte le widget **Gtk ::Label** est le type à utiliser.

Pour chaque chaîne de caractère que l'on veut afficher :

- Dans l'interface du module définissant la classe **layoutbuttons.h** :
 - ajouter un attribut dans la liste des « Member widgets » (avec les attributs Buttons). Par ex :
 - **Gtk ::Label m_Label_Salutation ;**

- Dans l'implémentation du module **layoutbuttons.cc**
 - Initialiser le Label dans la liste d'initialisation du constructeur
 - **m_Label_Salutation("Formule de Salutation")**
 - Pour avoir le texte sur plusieurs lignes il suffit d'insérer le caractère de contrôle **\n** dans la chaîne à afficher
 - Utiliser la méthode **pack_start** pour placer le label dans une **Gtk ::Box** comme traité dans l'exercice 1

Un Label n'a pas de `signal_handler`.

Le dernier élément nécessaire pour bien reproduire l'image précédente est un séparateur. On dispose du type **Gtk ::Separator** pour cela. Chaque séparateur doit avoir un attribut dans la liste des « Members widgets ». Par ex : **Gtk ::Separator m_Separator1 ;**

Un séparateur n'a pas besoin d'être initialisé. On doit seulement le placer au bon moment dans l'interface en utilisant **pack_start** sur les **Gtk ::Box**.