

1 Objectifs

Le but de ce TD/TP est de se familiariser avec les tableaux et les menus.

2 Éléments de cours

2.1 Tableau : GtkTable

2.1.1 Filiation

GtkWidget → GtkContainer → GtkTable

2.1.2 Constitution

Un tableau est un container particulier ; précisément, c'est un tableau à m lignes et n colonnes définissant $m \times n$ cases, chaque case pouvant contenir un objet graphique.

2.1.3 Principales fonctions

```
GtkWidget* gtk_table_new(guint nb_ligne, guint nb_col, gboolean homogene);
```

création d'un tableau de $\text{nb_ligne} \times \text{nb_col}$ cases

`homogene == TRUE` ⇒ toutes les cases sont de même taille (et cette taille est définie par le plus grand contenu),

`homogene == FALSE` ⇒ chaque ligne (*resp.*, chaque colonne) prend la hauteur (*resp.*, la largeur) nécessaire à son affichage (en fonction de son plus grand contenu)

```
void gtk_table_attach_defaults(GtkTable* tableau, GtkWidget* widget, guint ind_col_deb, guint ind_col_fin, guint ind_ligne_deb, guint ind_ligne_fin);
```

ajout d'un objet graphique, auquel sont attribuées les cases de coordonnées comprises entre (`ind_col_deb`, `ind_ligne_deb`) et (`ind_col_fin-1`, `ind_ligne_fin-1`)

```
void gtk_table_attach(GtkTable* tableau, GtkWidget* widget, guint ind_col_deb, guint ind_col_fin, guint ind_ligne_deb, guint ind_ligne_fin, GtkAttachOptions xoptions, GtkAttachOptions yoptions, guint xpadding, guint ypadding);
```

ajout d'un objet graphique, en spécifiant certaines options

`xoptions` et `yoptions` :

- `GTK_FILL` : l'objet graphique s'étend dans la ou les cases qui lui sont attribuées ;
- `GTK_SHRINK` : l'objet graphique disparaît si l'espace dont disposent la ou les cases qui lui sont attribuées devient trop petit ;
- `GTK_EXPAND` : le tableau lui-même s'étend de sorte à prendre tout l'espace qu'il peut dans son espace contenant.

`xpadding` et `ypadding` : marge entre le contenu et les bords (gauche/droit et haut/bas) du contenant

```
void gtk_table_set_row_spacing(GtkTable* tableau, guint ind_ligne, guint marge);
```

redéfinition de la marge entre les lignes d'indice ind_ligne et ind_ligne+1

```
void gtk_table_set_col_spacing(GtkTable* tableau, guint ind_col, guint marge);
```

redéfinition de la marge entre les colonnes d'indice ind_col et ind_col+1

```
void gtk_table_set_rows_spacing(GtkTable* tableau, guint marge);
```

définition de la marge entre tout couple de lignes successives

```
void gtk_table_set_cols_spacing(GtkTable* tableau, guint marge);
```

définition de la marge entre tout couple de colonnes successives

```
void gtk_table_resize(GtkTable* tableau, guint nb_ligne, guint nb_col);
```

redéfinition du nombre de lignes et du nombre de colonnes du tableau

2.2 Menu : GtkMenu, GtkMenuItem, GtkMenuBar

2.2.1 Filiation

GtkWidget → GtkContainer → GtkMenuShell → GtkMenuBar

GtkWidget → GtkContainer → GtkMenuShell → GtkMenu

GtkWidget → GtkContainer → GtkBin → GtkItem → GtkMenuItem

Un menu, comme une barre de menu, est une liste d'item (ex : la barre principale contient les item "Fichier" et "A propos de...", le menu "Fichier" contient les item "Nouveau", "Sauver", "Quitter"). À un item est ensuite, le cas échéant, associé un sous-menu (ex. : le menu "Fichier" composé des item "Nouveau", "Sauver", "Quitter" est le sous-menu associé à l'item "Fichier" de la barre de menu).

2.2.2 Constitution

Les principaux éléments d'un menu sont :

- la barre de menu (composant graphique que l'on peut considérer comme un menu particulier) : GtkMenuBar ;
- le menu lui-même (container comportant un nombre arbitraire d'item) : GtkMenu ;
- les item constitutifs du menu et auxquels peuvent être associés des sous-menus : GtkMenuItem.

2.2.3 Principales fonctions

```
GtkWidget* gtk_menu_bar_new(void);
```

création d'une barre de menu - ex. :

```
GtkWidget* pBarreMenu = gtk_menu_bar_new();
```

```
GtkWidget* gtk_menu_new(void);
```

création d'un menu - ex. :

```
GtkWidget* pMenuFichier = gtk_menu_new();
```

```
GtkWidget* pMenuAPropos = gtk_menu_new();
```

```
GtkWidget* gtk_menu_item_new(void);
```

```
GtkWidget* gtk_menu_item_new_with_label(const char* libelle);
```

```
GtkWidget* gtk_menu_item_new_with_mnemonic(const char* libelle);
```

création d'un item d'une barre de menu : "simple", avec libellé, avec libellé contenant un raccourci clavier (*cf.*, les fonctions de création de boutons). - ex. :

```
GtkWidget* pItemMenuFichier = gtk_menu_item_new_with_mnemonic("_Fichier");
GtkWidget* pItemMenuAPropos = gtk_menu_item_new_with_mnemonic("_A propos de...");
GtkWidget* pItemMenuNouveau = gtk_menu_item_new_with_mnemonic("_Nouveau");
GtkWidget* pItemMenuSauver = gtk_menu_item_new_with_mnemonic("_Sauver");
GtkWidget* pItemMenuQuitter = gtk_menu_item_new_with_mnemonic("_Quitter");
```

```
void gtk_menu_shell_append(GtkMenuShell* shell_menu, GtkWidget* item);
void gtk_menu_bar_append(GtkMenuBar* barre_menu, GtkWidget* item_menu);
```

ajout d'un item fils à un menu parent (un menu parent pouvant être un menu ou une barre de menu) - ex. :

```
gtk_menu_bar_append(GTK_MENU_BAR(pBarreMenu), pItemMenuFichier);
gtk_menu_bar_append(GTK_MENU_BAR(pBarreMenu), pItemMenuAPropos);
gtk_menu_shell_append(GTK_MENU_SHELL(pMenuFichier), pItemMenuNouveau);
gtk_menu_shell_append(GTK_MENU_SHELL(pMenuFichier), pItemMenuSauver);
gtk_menu_shell_append(GTK_MENU_SHELL(pMenuFichier), pItemMenuQuitter);
```

```
void gtk_menu_item_set_submenu(GtkMenuItem* item, GtkWidget* sous-menu);
```

association d'un sous-menu à un item d'un menu de niveau supérieur (i.e., menu qui s'ouvre lorsque l'utilisateur clique sur l'item) - ex. :

```
gtk_menu_item_set_submenu(GTK_MENU_ITEM(pItemMenuFichier), pMenuFichier);
```

2.2.4 Événementiel

L'événement clé pour la gestion événementielle des menus est l'activation d'un item, par clic sur l'item ou par accès clavier ; le signal "activate" est alors émis. Ex. (bien que cela soit quelque peu violent... il serait préférable d'ouvrir une boîte de dialogue demandant confirmation) :

```
gtk_signal_connect(GTK_OBJECT(pItemMenuQuitter), "activate", G_CALLBACK(gtk_main_quit), NULL);
```

3 Exercices

1. Créer une fenêtre, de titre "Gtk - TD5", non redimensionnable. Définir une constante symbolique N paire (ex. : $N = 16$) et créer un tableau $N \times N$. Créer également un bouton "_Quitter". Ajouter le tableau et le bouton à la fenêtre, en ayant pris soin d'associer la fonction "gtk_main_quit" au signal "clicked" du bouton (ainsi qu'au signal "delete" de la fenêtre).
2. Créer $N \times N$ zones de saisie et placer la zone d'indices (i, j) dans la case (i, j) du tableau. Limiter la largeur des zones de saisie par appel à la fonction "void gtk_entry_set_width_chars(GtkEntry* pZone, guint largeur)", pour largeur == 4. De plus, limiter le nombre de caractères saisissables à 4. Ces zones, non modifiables par l'utilisateur, contiennent leur indice, i.e. : la zone d'indices (i, j) contient l'entier $(N * i) + j$. Par ailleurs, il n'est pas usuel de quitter sauvagement une application après avoir cliqué sur un menu "Quitter" : il est préférable d'ouvrir une boîte de message demandant confirmation (avec un label du type "Voulez-vous vraiment quitter ?" et deux boutons, "Oui" et "Non"). Enfin, Définir dans le tableau une marge de 10 pixels toutes les deux colonnes.
3. Définir $N/2$ nouvelles zones de saisie, ainsi que $N/2$ libellés. Les zones de saisie nouvellement créées sont vides, non éditables, et contiennent au plus 8 caractères. Les libellés sont tous identiques : ils contiennent le texte "somme". Ajouter alors deux lignes au tableau : placer dans l'avant-dernière ligne les libellés nouvellement créés (chaque libellé s'étend sur deux cases consécutives), et les zones de saisie nouvellement créées dans la

dernière ligne (chaque zone s'étend sur deux cases consécutives). Ajouter alors un bouton à la fenêtre (à droite du bouton “_ Quitter”) : ce bouton, de libellé “_ Somme”, calcule pour i de 0 à $N/2$ la somme de tous les nombres présents dans les colonnes $2*i$ et $(2*i)+1$, dans les lignes 0 à $N-1$. Cette somme est alors inscrite dans la zone de saisie nouvellement créée d'indice i .

4. Revenir sur la configuration de l'exercice 2. Ajouter une barre de menu à la fenêtre : cette barre contient les items “_ Jouer” et “_ Paramétrer”. Le sous-menu “Jouer” contient les items “_ Nouvelle partie”, “_ Sauver partie”, “_ Recommencer partie” et “_ Quitter” ; le sous-menu “Paramétrer” contient pour sa part les item “_ Nombre de cases” et “Nombre de coups _ Initiaux”.
5. Associer la boîte de message déjà existante à l'item “_ Quitter”. Faire de sorte que, lorsque l'on clique sur l'item “_ Recommencer partie”, les zones de saisie soient toutes réinitialisées à vide. Le mieux, pour ce faire, est toutefois de demander confirmation à l'utilisateur avant de réinitialiser les zones ! Enfin, définir une variable dans la fonction “main” qui représente le nombre de coups initiaux ; il faut, lorsque l'on clique sur l'item “Nombre de coups _ Initiaux”, que l'utilisateur puisse modifier la valeur de cette variable.
6. Définir d'autres fonctions de callback associées aux menus... des idées ?