

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include <gsl_statistics_double.h>
5  #include <gsl_statistics.h>
6  #include <stdlib.h>
7  #include <gtk/gtk.h>
8
9
10
11 void on_activate_entry(GtkWidget *pEntry0, gpointer data);
12 void on_copier_button2(GtkWidget *pButton_valeur, gpointer data);
13
14
15
16 /// Fenêtre d'acquisition ///
17
18 int main (int argc, char **argv)
19 {
20
21 GtkWidget *pWindow;
22 GtkWidget *pVBox;
23 GtkWidget *pHBox;
24 GtkWidget *pHBox2;
25 GtkWidget *pFrame1;
26 GtkWidget *pFrame2;
27 GtkWidget *pFrame3;
28
29
30 GtkWidget *pVBoxFrame;
31 GtkWidget *pVBoxFrame2;
32 GtkWidget *pSeparator;
33 GtkWidget *pEntry;
34 GtkWidget *pEntry0;
35 GtkWidget *pLabel;
36 GtkWidget *pLabel0;
37 GtkWidget *pLabel2;
38 GtkWidget *pLabel3;
39 GtkWidget *pLabel4;
40 GtkWidget *pLabel5;
41 GtkWidget *pLabel6;
42 GtkWidget *pLabel7;
43
44
45 GtkWidget *pButton_valeur;
46 float number2;
47 const gchar *sText;
48
49
50 gtk_init(&argc,&argv);
51
52 pWindow = gtk_window_new(GTK_WINDOW_TOPLEVEL);
53 gtk_window_set_title(GTK_WINDOW(pWindow), "Les GtkBox");
54 gtk_window_set_default_size(GTK_WINDOW(pWindow), 620, 400);
55 g_signal_connect(G_OBJECT(pWindow), "destroy", G_CALLBACK(gtk_main_quit),NULL);
56
57 /* Creation de la GtkBox verticale */
58 pVBox = gtk_vbox_new(TRUE, 0);
59 /* Ajout de la GtkVBox dans la fenetre */
60 gtk_container_add(GTK_CONTAINER(pWindow), pVBox);
61
62
63 /// Les éléments 1er ligne
64
65 pLabel0= gtk_label_new("Valeur");
66 pButton_valeur= gtk_button_new_with_label("Valider");
67 pEntry0 = gtk_entry_new();
68
69 /* Recuperation du texte contenu dans le GtkEntry */
70 sText = gtk_entry_get_text(GTK_ENTRY(pEntry0));
71 /* Conversion chaîne -> valeur numérique */
72 number2 = g_strtod (sText, NULL);
73
74 printf ("number :%f", number2);
75
76 /* Creation de la GtkBox horizontale: ligne 0 */
77 pHBox = gtk_hbox_new(TRUE, 0);
78 gtk_box_pack_start(GTK_BOX(pVBox), pHBox, TRUE, TRUE, 0);
79
80
81 gtk_box_pack_start(GTK_BOX(pHBox), pEntry0, TRUE, FALSE, 0); /* NB éléments */
82 gtk_box_pack_start(GTK_BOX(pHBox), pButton_valeur, FALSE, TRUE, 80);
83 gtk_box_pack_start(GTK_BOX(pHBox), pLabel0, TRUE, FALSE, 0);
84
85
86
87
88
89
90
91
92 pLabel2= gtk_label_new(NULL);
93 pLabel3= gtk_label_new(NULL);
94 pLabel4= gtk_label_new(NULL);
95 pLabel5= gtk_label_new(NULL);
96 pLabel6= gtk_label_new(NULL);
97 pLabel7= gtk_label_new(NULL);
98
99
100 /// Les Mesures ///
101 /* Ligne 1*/
102 pHBox2 = gtk_hbox_new(TRUE, 0);
103 gtk_box_pack_start(GTK_BOX(pVBox), pHBox2, TRUE, TRUE, 0);
104
105
106 /// Mesures 1 ///
107 /* Creation du premier GtkFrame */
108 pFrame1 = gtk_frame_new("Mesure 1");
109 gtk_box_pack_start(GTK_BOX(pHBox2), pFrame1, TRUE, FALSE, 0);
110 /* Creation et insertion d'une boîte pour le premier GtkFrame */
111 pVBoxFrame = gtk_vbox_new(TRUE, 0);
112 gtk_container_add(GTK_CONTAINER(pFrame1), pVBoxFrame);

```

```

113 /* Creation et insertion des elements contenus dans le premier GtkFrame */
114 pLabel = gtk_label_new("X1 :");
115 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pLabel, TRUE, FALSE, 0);
116 pEntry = gtk_entry_new();
117 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pEntry, TRUE, FALSE, 0);
118 pEntry = gtk_entry_new();
119 /* Creation d un GtkHSeparator */
120 pSeparator = gtk_hseparator_new();
121
122 /* Box de contrôle final */
123 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pLabel2, TRUE, FALSE, 0);
124
125
126 /* Creation d un GtkHSeparator */
127 pSeparator = gtk_hseparator_new();
128 pLabel = gtk_label_new("Y1 :");
129 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pLabel, TRUE, FALSE, 0);
130 pEntry = gtk_entry_new();
131 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pEntry, TRUE, FALSE, 0);
132 pEntry = gtk_entry_new();
133 /* Creation d un GtkHSeparator */
134 pSeparator = gtk_hseparator_new();
135
136
137 /* Box de contrôle final */
138 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pLabel3, TRUE, FALSE, 0);
139
140
141
142
143
144 /// Mesures 2 ///
145 /* Creation du premier GtkFrame */
146 pFrame2 = gtk_frame_new("Mesure 2");
147 gtk_box_pack_start(GTK_BOX(pHBox2), pFrame2, TRUE, FALSE, 0);
148 /* Creation et insertion d une boite pour le premier GtkFrame */
149 pVBoxFrame = gtk_vbox_new(TRUE, 0);
150 gtk_container_add(GTK_CONTAINER(pFrame2), pVBoxFrame);
151 pLabel = gtk_label_new("X2 :");
152 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pLabel, TRUE, FALSE, 0);
153 pEntry = gtk_entry_new();
154 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pEntry, TRUE, FALSE, 0);
155 pEntry = gtk_entry_new();
156 /* Creation d un GtkHSeparator */
157 pSeparator = gtk_hseparator_new();
158
159 /* Box de contrôle final */
160 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pLabel4, TRUE, FALSE, 0);
161
162 pLabel = gtk_label_new("Y2 :");
163 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pLabel, TRUE, FALSE, 0);
164 pEntry = gtk_entry_new();
165 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pEntry, TRUE, FALSE, 0);
166 pEntry = gtk_entry_new();
167 /* Creation d un GtkHSeparator */
168 pSeparator = gtk_hseparator_new();
169
170 /* Box de contrôle final */
171 gtk_box_pack_start(GTK_BOX(pVBoxFrame), pLabel5, TRUE, FALSE, 0);
172
173
174
175
176 /// Mesures 3 ///
177 /* Creation du premier GtkFrame */
178 pFrame3 = gtk_frame_new("Mesure 3");
179 gtk_box_pack_start(GTK_BOX(pHBox2), pFrame3, TRUE, FALSE, 0);
180 /* Creation et insertion d une boite pour le premier GtkFrame */
181 pVBoxFrame2 = gtk_vbox_new(TRUE, 0);
182 gtk_container_add(GTK_CONTAINER(pFrame3), pVBoxFrame2);
183 pLabel = gtk_label_new("X3 :");
184 gtk_box_pack_start(GTK_BOX(pVBoxFrame2), pLabel, TRUE, FALSE, 0);
185 pEntry = gtk_entry_new();
186 gtk_box_pack_start(GTK_BOX(pVBoxFrame2), pEntry, TRUE, FALSE, 0);
187 pEntry = gtk_entry_new();
188 /* Creation d un GtkHSeparator */
189 pSeparator = gtk_hseparator_new();
190 /* Box de contrôle final */
191 gtk_box_pack_start(GTK_BOX(pVBoxFrame2), pLabel6, TRUE, FALSE, 0);
192
193 pLabel = gtk_label_new("Y3 :");
194 gtk_box_pack_start(GTK_BOX(pVBoxFrame2), pLabel, TRUE, FALSE, 0);
195 pEntry = gtk_entry_new();
196 gtk_box_pack_start(GTK_BOX(pVBoxFrame2), pEntry, TRUE, FALSE, 0);
197 pEntry = gtk_entry_new();
198 /* Creation d un GtkHSeparator */
199 pSeparator = gtk_hseparator_new();
200 /* Box de contrôle final */
201 gtk_box_pack_start(GTK_BOX(pVBoxFrame2), pLabel7, TRUE, FALSE, 0);
202
203
204
205
206
207
208 /* Connexion du signal "activate" du GtkEntry */
209 g_signal_connect(G_OBJECT(pEntry), "activate", G_CALLBACK(on_activate_entry), (GtkWidget*) pLabel2);
210
211
212 /* Connexion du signal "clicked" du GtkButton */ /* Copie des X - Y */
213 g_signal_connect(G_OBJECT(pButton_valeur), "clicked", G_CALLBACK(on_copier_button2), (GtkWidget*) pVBoxFrame2);
214
215
216 gtk_widget_show_all(pWindow);
217 gtk_main();
218
219 return EXIT_SUCCESS;
220 }
221
222
223 /* Fonction callback execute lors du signal "activate" */
224 void on_activate_entry(GtkWidget *pEntry00, gpointer data)

```

```

225 {
226
227 }
228
229
230
231 /* Fonction callback executee lors du signal "clicked" */
232 /* Option qui recopie la variable sur le label en dessous */
233 void on_copier_button2(GtkWidget *pButton_valeur, gpointer data)
234 {
235 GtkWidget *pTempEntry0;
236 GtkWidget *pTempLabel;
237 GList *pList;
238 const gchar *sText;
239 gchar *modified_text;
240 gdouble number;
241 float number2;
242
243 /* Recuperation de la liste des elements que contient la GtkVBox */
244 pList = gtk_container_get_children(GTK_CONTAINER((GtkWidget*)data));
245
246 /* Passage a l'element suivant : le Label */
247 pList = g_list_next(pList);
248
249 /* Le premier element est le GtkEntry */
250 pTempEntry0 = GTK_WIDGET(pList->data);
251
252 /* Cet element est le GtkLabel */
253 pTempLabel = GTK_WIDGET(pList->data);
254
255
256
257
258
259
260 /* Recuperation du texte contenu dans le GtkEntry */
261 sText = gtk_entry_get_text(GTK_ENTRY(pTempEntry0));
262
263 /* Conversion chaine -> valeur numerique */
264 number = g_strtod (sText, NULL);
265
266 /* Modification du nombre */
267 number = number*2;
268
269 /* Conversion valeur numerique -> chaine */
270 modified_text = g_strdup_printf ("%0.2f", number);
271
272
273 /* Modification du texte contenu dans le GtkLabel */
274 gtk_label_set_text(GTK_LABEL(pTempLabel), modified_text);
275
276 /* Liberation de la memoire utilisee par la liste */
277 g_list_free(pList);
278
279
280 /* Libération de la chaine temporaire */
281 g_free (modified_text);
282
283 printf ("number :%lf", number2);
284 }
285
286
287
288
289
290

```