

Portada

Sobre GTK

Bucle
principal

Señales

Callbacks

Glade y
builder

Ejemplo
simple

Juego de la
vida

main
gui object
drawing area
Timer
Ejercicio

Curso de programación en C moderno (II Edición)

Neira Ayuso, Pablo Falgueras García, Carlos

Tema 14

GTK

Índice

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main
gui object
drawing area
Timer
Ejercicio

- 1 Sobre GTK
- 2 Bucle principal
- 3 Señales
- 4 Callbacks
- 5 Glade y builder
- 6 Ejemplo simple
- 7 Juego de la vida
 - main
 - gui object
 - drawing area
 - Timer
 - Ejercicio

Sobre GTK

Portada

Sobre GTK

Bucle
principal

Señales

Callbacks

Glade y
builder

Ejemplo
simple

Juego de la
vida

main
gui object
drawing area
Timer
Ejercicio

- **GTK** (Gimp ToolKit) es una biblioteca libre y multiplataforma para el desarrollo de interfaces gráficas de usuario
- Está escrita en C, aunque tiene *bindings* a una infinidad de lenguajes distintos (C++, C#, Python, Lua, Haskell, ...)
- Se trata de una biblioteca muy popular, con mucha comunidad alrededor y muchos años de madurez
- Las interfaces se construyen mediante la anidación de diversos **widget**. Tiene una enorme variedad de widgets (ejecutar “gtk3-widget-factory”)
- Dispone de una documentación detallada y de calidad (<https://developer.gnome.org/gtk3/stable/>). Además de numerosos tutoriales y ejemplos desarrollados por su gran comunidad.

Bucle principal

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

- Toda la lógica de la interfaz se realiza dentro de un bucle infinito `gtk_main()`;
- Antes del bucle la interfaz no es funcional
- Después del bucle no existe interfaz
- No podemos modificar el interior del bucle
- **¿Cómo interacciono con la interfaz gráfica?**

```
1 #include <gtk/gtk.h>
2
3 int main(int argc, char **argv)
4 {
5     GtkWidget *window;
6
7     gtk_init(&argc, &argv);
8     window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
9     gtk_widget_show_all(window);
10    gtk_main();
11
12    return 0;
13 }
```

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

- Una señal puede verse como un **mensaje** que nos envía GTK cada vez que ocurre un **evento**
- Un **evento** puede ser cualquier acción del usuario: un click en un botón, mover el ratón sobre una ventana, arrastar algún objeto, También hay eventos internos de la interfaz gráfica: se acaba de crear/destruir la ventana, se va a redibujar algún objeto, ...
- En GTK se implementan mediante **callbacks**

Callbacks

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

- Un callback es una función que pasamos a alguna librería, para que ella se encargue de llamarla cuando convenga
- GTK guarda una lista de punteros a funciones por cada posible evento
- Como usuarios de GTK, **conectamos** una o más funciones de callback a una señal (pulsar un botón). Cuando ocurra el evento oportuno, GTK se encargará de llamarlas a todas

Glade y builder

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

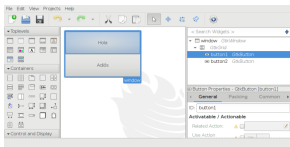
Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

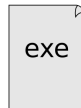
- Crear una interfaz compleja directamente en un lenguaje de programación es complejo y poco versátil
- GTK puede crear e inicializar todos los objetos necesarios a partir de un XML con la descripción de la interfaz
- **Glade** es un programa que nos ayuda a construir este XML de manera gráfica



Glade



builder.ui



gol

Ejemplo simple

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <gtk/gtk.h>
4
5 #define BUILDER_FILE "builder.ui"
6
7 int main(int argc, char **argv)
8 {
9     GtkBuilder *builder;
10    GtkWidget *window;
11    GtkWidget *button1;
12    GtkWidget *button2;
13
14    gtk_init(&argc, &argv);
15
16    /* builder */
17    builder = gtk_builder_new();
18    if (!gtk_builder_add_from_file(builder, BUILDER_FILE, NULL)) {
19        fprintf(stderr, "Can't open file \"%s\"\n", BUILDER_FILE);
20        exit(EXIT_FAILURE);
21    }
22
23    /* objects */
24    window = GTK_WIDGET(gtk_builder_get_object(builder, "window"));
25    button1 = GTK_WIDGET(gtk_builder_get_object(builder, "button1"));
26    button2 = GTK_WIDGET(gtk_builder_get_object(builder, "button2"));
```


Ejemplo simple

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

```
27
28  /* signals */
29  g_signal_connect(window, "destroy", G_CALLBACK(gtk_main_quit),
    NULL);
30  g_signal_connect(button1, "clicked", G_CALLBACK(button_cb), "Hola"
    );
31  g_signal_connect(button2, "clicked", G_CALLBACK(button_cb), "Adios"
    );
32  gtk_builder_connect_signals(builder, NULL);
33
34  gtk_widget_show_all(window);
35  gtk_main();
36
37  return 0;
38 }
39
40 static void button_cb(GtkWidget *widget, const char *str)
41 {
42     printf("%s mundo!\n", str);
43 }
```

```
gcc -g $(pkg-config --cflags gtk+-3.0 --libs gtk+-3.0)main.c
```

Juego de la vida

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

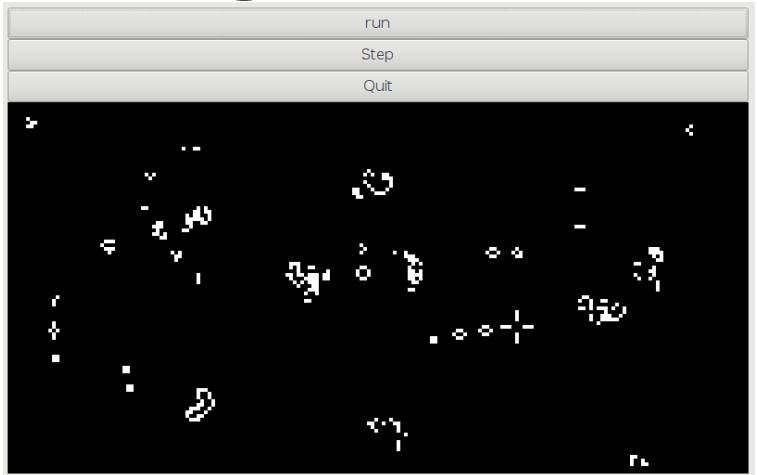
Glade y builder

Ejemplo simple

Juego de la vida

main
gui object
drawing area
Timer
Ejercicio

Juego de la vida



main

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

```
1 int main(int argc, char **argv)
2 {
3     struct world *w;
4     struct gui *g;
5
6     gtk_init(&argc, &argv);
7
8     w = world_alloc(WORLD_X, WORLD_Y);
9     if (!w) {
10         perror("Can't allocate world");
11         exit(EXIT_FAILURE);
12     }
13     world_init(w);
14
15     g = gui_alloc("builder.ui", w, WINDOW_X, WINDOW_Y);
16     if (!g) {
17         perror("Can't create gui");
18         exit(EXIT_FAILURE);
19     }
20
21     gtk_main();
22
23     world_free(w);
24     gui_free(g);
25
26     return 0;
27 }
```

gui object

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

```
1 struct gui {  
2     GtkBuilder *builder;  
3  
4     GtkWidget *window;  
5     GtkWidget *btn_quit;  
6     GtkWidget *btn_step;  
7     GtkWidget *btn_pause;  
8     GtkGrid *grid;  
9     GtkWidget *drawing_area;  
10  
11     struct {  
12         double x;  
13         double y;  
14     } cell_size;  
15  
16     bool run;  
17  
18     struct world *world;  
19 };
```

drawing area

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area Timer Ejercicio

```

1  /* drawing area */
2  g->drawing_area = gtk_drawing_area_new();
3  // Tamaño de la ventana de dibujado
4  gtk_widget_set_size_request(g->drawing_area, ws_x, ws_y);
5  // Necesario para capturar los click del raton
6  gtk_widget_set_events(g->drawing_area,
7    gtk_widget_get_events(g->drawing_area) | GDK_BUTTON_PRESS_MASK);
8  // Anyadimos el nuevo widget al grid
9  gtk_grid_attach_next_to(g->grid, g->drawing_area, g->btn_quit,
10    GTK_POS_BOTTOM, 1, 1);

```

```

1  static gboolean draw_cb(GtkWidget *widget, cairo_t *cr, struct gui *
    g)
2  {
3    /* Clear screen */
4    cairo_set_source_rgb(cr, 0, 0, 0);
5    cairo_paint(cr);
6
7    /* TODO
8     * Dibuja el mundo sabiendo que el código de abajo dibuja un
        rectangulo
9     * blanco en las coordenadas (2, 3) de 5x4 pixeles de tamaño
10    */
11    cairo_set_source_rgb(cr, 1, 1, 1);
12    cairo_rectangle(cr, 2, 3, 5, 4);
13    cairo_fill(cr);
14
15    return false;
16 }

```

Timer

Portada

Sobre GTK

Bucle principal

Señales

Callbacks

Glade y builder

Ejemplo simple

Juego de la vida

main gui object drawing area **Timer** Ejercicio

```
1 static void pause_cb(GtkWidget *widget, struct gui *g)
2 {
3     if (!g->run)
4         g_timeout_add(VEL_MS, timer_cb, g); // Conectamos la senyal de
        nuevo
5     g->run = !g->run;
6 }
```

```
1 static gboolean timer_cb(gpointer gui)
2 {
3     struct gui *g = (struct gui *)gui;
4     step_cb(NULL, g);
5     /* Si devuelve false, la senyal se desconecta y el callback no se
        vuelve
6     * a llamar
7     */
8     return g->run;
9 }
```

Ejercicio

Portada

Sobre GTK

Bucle
principal

Señales

Callbacks

Glade y
builder

Ejemplo
simple

Juego de la
vida

main
gui object
drawing area
Timer
Ejercicio

- 1 Estudia detenidamente el código proporcionado
- 2 Crea el fichero “builder.ui” con **Glade**. Presta atención al identificador de cada widget
- 3 Completa el código de “gui.c”. Las zonas a completar están marcadas con un TODO