

Programmation d'interfaces

Cours 5 - Dessiner avec Cairo

H. Djerroud

LIASD - Université Paris 8

Automne 2020

Plan de cours

- Créer une zone de dessin
- Utilisation de Cairo

Gtk Drawing Area

- Zone de dessin avec GTK

Exemple GtkDrawingArea (1)

```
int main(int argc, char* argv[]){
    GtkWidget *win, *draw_area;
    gtk_init(&argc, &argv);
    win = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_container_set_border_width (GTK_CONTAINER (win), 10);
    g_signal_connect (G_OBJECT(win), "delete_event",
                     G_CALLBACK(gtk_main_quit) ,NULL);
    draw_area = gtk_drawing_area_new ();
    gtk_widget_set_size_request (draw_area, 400, 400);
    gtk_container_add(GTK_CONTAINER(win), draw_area);
    g_signal_connect (G_OBJECT (draw_area), "draw",
                     G_CALLBACK (refresh_draw), NULL);
    gtk_widget_show_all(win);
    gtk_main();
    return 0;
}
```

Exemple GtkDrawingArea (2)

```
gboolean refresh_draw (GtkWidget *widget, cairo_t *cr, gpointer data){
    guint width, height;
    GtkStyleContext *context;
    context = gtk_widget_get_style_context (widget);
    width = gtk_widget_get_allocated_width (widget);
    height = gtk_widget_get_allocated_height (widget);
    gtk_render_background (context, cr, 0, 0, width, height);
    cairo_set_source_rgb (cr,0,0,0);
    cairo_set_line_width(cr, 9);
    cairo_move_to (cr, 50,50);
    cairo_line_to (cr, 350,350);
    cairo_move_to (cr, 50,350);
    cairo_line_to (cr, 350,50);
    cairo_stroke (cr);
    return FALSE;
}
```

Introduction

Pour bien commencer avec la librairie Cairo il faut connaître le jargon :

- Contexte (Context)
- Chemin (Path)
- Source
- Surface
- Msque (Mask)

Context

- Le dessin avec Caire se fait via un contexte (context). Le contexte Cairo contient tous les paramètres graphique qui décrivent comment le dessin doit être fait par exemple : la largeur de la ligne, la couleur, la surface sur laquelle dessiner, etc.
- Cela permet de réduire le nombre de paramètres des fonctions de dessin.
- Tout dessin avec Cairo est toujours fait sur un objet cairo_t.
- Un contexte du Caire est lié à une surface spécifique. Un PDF, SVG, PNG, GtkWindow etc.

Contexte (suite)

Récupérer un contexte depuis un widget GTK :

```
....  
GtkStyleContext *context;  
context = gtk_widget_get_style_context (widget);  
...
```

On peut aussi récupérer certains paramètres des widgets GTK qui ne sont pas dans le contexte, par exemple :

```
....  
gint width, height;  
width = gtk_widget_get_allocated_width (widget);  
height = gtk_widget_get_allocated_height (widget);  
...
```

Path

- Un chemin est composé d'une ou plusieurs lignes. Ces lignes sont reliées par deux ou plusieurs points d'ancrage. Les chemins peuvent être constitués de lignes droites et de courbes. Il existe deux types de chemins : **ouverts** et **fermés**.
- Dans un chemin fermé, les points de départ et d'arrivée se rencontrent.
- Dans un chemin ouvert, les points de départ et d'arrivée ne se rencontrent pas.
- Pour dessiner avec Cairo on part toujours d'un chemin vide. Puis il faut définir les chemins, les rendre visibles et les remplir.
- Après chaque appel de fonction `cairo_stroke()` ou `cairo_fill()`, le chemin est vidé.

Source

- La source est la peinture que nous utilisons dans le dessin.
- Il existe quatre types de sources de base :
 - les couleurs
 - les dégradés
 - les motifs
 - les images

Surface

- Une surface est un support sur lequel on souhaite dessiner.
- On peut par exemple dessiner sur les surfaces PDF, PostScript png, etc. Ou dessiner directement sur les surfaces Xlib (les fenêtres) par exemple.

Les type de surfaces)

```
typedef enum _cairo_surface_type {  
    CAIRO_SURFACE_TYPE_IMAGE,  
    CAIRO_SURFACE_TYPE_PDF,  
    CAIRO_SURFACE_TYPE_PS,  
    CAIRO_SURFACE_TYPE_XLIB,  
    CAIRO_SURFACE_TYPE_XCB,  
    CAIRO_SURFACE_TYPE_GLITZ,  
    CAIRO_SURFACE_TYPE_QUARTZ,  
    CAIRO_SURFACE_TYPE_WIN32,  
    CAIRO_SURFACE_TYPE_BEOS,  
    CAIRO_SURFACE_TYPE_DIRECTFB,  
    CAIRO_SURFACE_TYPE_SVG,  
    CAIRO_SURFACE_TYPE_OS2  
} cairo_surface_type_t;
```

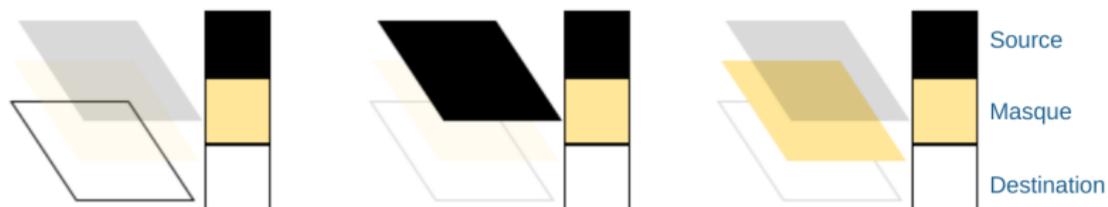
Mask

- Les masques ressemblent aux pochoirs que les graphistes utilisent pour dessiner.
- Avant que la source ne soit appliquée à la surface, elle est d'abord passé sur le pochoir. Il est utilisé comme filtre. Le masque détermine où la source est appliquée où pas.
- Les parties opaques du masque permettent de d'appliquer la source sur la surface. Les parties transparentes ne permettent pas.

Récapitulatif

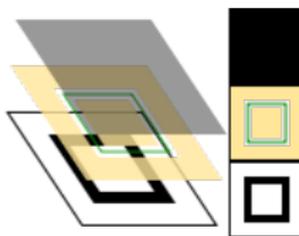
Dessin extrait de :

<https://www.cairographics.org/tutorial/>



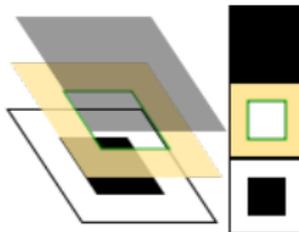
L'opération Stroke

- Dessine en suivant les lignes tracées dans le masque.



L'opération Fill

- Fait un remplissage en suivant les lignes dessinées dans le masque.



Le point de référence

- Le point $(0,0)$ se situe de haut à gauche.
- L'axe des X se prolonge horizontalement de gauche à droite.
- L'axe des Y se prolonge verticalement de haut en bas.
- Il est possible de changer le point de référence pour faciliter le dessin en utilisant les coordonnées relatifs, avec la fonction `cairo_translate(contexte, pos_x, pos_y)`

Les lignes

- Les lignes sont des objets vectoriels très basiques. Pour tracer une ligne, il faut utiliser deux appels de fonction.
- Le point de départ avec l'appel `cairo_move_to()`.
- Le point de fin d'une ligne avec l'appel `cairo_line_to()`.

Exemple (tracer les lignes d'un carré)

```
...
cairo_set_source_rgb (cr,0,0,0);
cairo_set_line_width(cr, 3);
cairo_move_to (cr, 50,50);
cairo_line_to (cr, 50,350);
cairo_line_to (cr, 350,350);
cairo_line_to (cr, 350,50);
cairo_line_to (cr, 50,50);

//tracer la ligne
cairo_stroke (cr);
...
```

Exemple (Remplissage d'un carré)

```
...
cairo_set_source_rgb (cr,0,0,0);
cairo_set_line_width(cr, 3);
cairo_move_to (cr, 50,50);
cairo_line_to (cr, 50,350);
cairo_line_to (cr, 350,350);
cairo_line_to (cr, 350,50);
cairo_line_to (cr, 50,50);

// On trace la ligne mais on garde le chemin
cairo_stroke_preserve(cr);

// On change la couleur pour le remplissage
cairo_fill (cr);
...
```

Les cercles et les arcs

- Avec Cairo on dessine pas des cercles mais des arcs.
- Un cercle est une exception d'un arc.
- Pour dessiner un arc on utilise la fonction :
`cairo_arc(contexte, pos_x, pos_y, angle_debut, angle_fin)`

Exemple (dessin d'un cercle)

```
...  
width = gtk_widget_get_allocated_width (widget);  
height = gtk_widget_get_allocated_height (widget);  
...  
// Changer de référence au centre  
cairo_translate(cr, width/2, height/2);  
cairo_arc(cr, 0, 0, 150, 0, 2 * G_PI);  
cairo_stroke_preserve(cr);  
// On change la couleur pour le remplissage  
cairo_set_source_rgb (cr,1,0,0);  
cairo_fill (cr);  
...
```

Les rectangles

- Il est possible de dessiner directement des rectangles au lieu de dessiner plusieurs lignes avec la fonction `cairo_rectangle(contexte, pos_x1, pos_y1, pos_x2, pos_y2)`