



<https://www.cairographics.org>

Biblioteka grafiki wektorowej 2D

Źródło: <https://zetcode.com/gfx/cairo/>

Context

Rysowanie w Cairo odbywa się za pośrednictwem kontekstu. Kontekst Cairo zawiera wszystkie parametry, które opisują sposób rysowania (szerokość linii, kolor, powierzchnia do rysowania). Wszystkie rysunki w Cairo są zawsze wykonywane na obiekcie **cairo_t**. Kontekst jest powiązany z określoną powierzchnią (PDF, SVG, PNG, GtkWindow).

Surface

Powierzchnia to miejsce docelowe w którym rysujemy. Możemy renderować dokumenty (m.in.) za pomocą powierzchni PDF, PostScript lub SVG, lub rysować na powierzchni GtkWindow (GtkDrawingArea).

Path

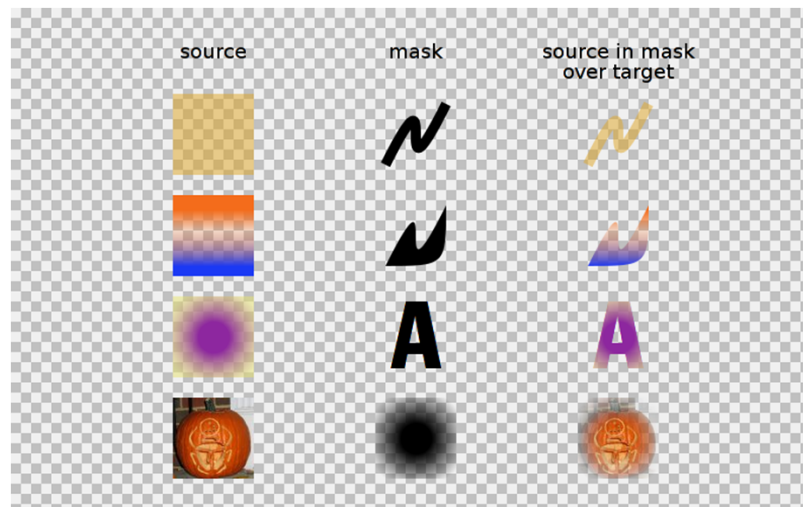
Ścieżka składa się z jednej lub więcej linii. Ścieżki mogą składać się z linii prostych i krzywych. Istnieją dwa rodzaje ścieżek: otwarte i zamknięte. Najpierw definiujemy ścieżkę, a następnie rysujemy wzdłuż niej lub wypełniamy. Po każdym wywołaniu funkcji **cairo_stroke()** lub **cairo_fill()** ścieżka jest czyszczona.

Source

Źródło możemy porównać do długopisu lub tuszu (farby), którego używamy do rysowania i wypełniania kształtów. Istnieją cztery rodzaje podstawowych źródeł: kolory, gradienty, wzory i obrazy.

Mask

Zanim źródło zostanie nałożone na powierzchnię, jest najpierw filtrowane. Maska służy jako filtr - określa, gdzie źródło jest stosowane, a gdzie nie.



https://cworth.org/cworth/papers/cairo_ddc2005/html/cairo-003.html

PNG image

```
#include <cairo.h>

int main(void)
{
    cairo_surface_t *surface;
    cairo_t *cr;

    surface = cairo_image_surface_create(CAIRO_FORMAT_ARGB32, 390,
60);
    cr = cairo_create(surface);

    cairo_set_source_rgb(cr, 0, 0, 0);
    cairo_select_font_face(cr, "Sans", CAIRO_FONT_SLANT_NORMAL,
        CAIRO_FONT_WEIGHT_NORMAL);
    cairo_set_font_size(cr, 40.0);

    cairo_move_to(cr, 10.0, 50.0);
    cairo_show_text(cr, "Ala ma kota.");

    cairo_surface_write_to_png(surface, "image.png");

    cairo_destroy(cr);
    cairo_surface_destroy(surface);

    return 0;
}
```

Ala ma kota.

PDF file

```
#include <cairo.h>
#include <cairo-pdf.h>

int main(void)
{
    cairo_surface_t *surface;
    cairo_t *cr;

    surface = cairo_pdf_surface_create("pdffile.pdf", 504, 648);
    ..

    cairo_show_page(cr);

    cairo_surface_destroy(surface);
    cairo_destroy(cr);

    return 0;
}
```

SVG file

```
#include <cairo.h>
#include <cairo-svg.h>

int main(void)
{
    cairo_surface_t *surface;
    cairo_t *cr;

    surface = cairo_svg_surface_create("svgfile.svg", 390, 60);
    ..

    cairo_surface_destroy(surface);
    cairo_destroy(cr);

    return 0;
}
```

GTK Window

```
#include <cairo.h>
#include <gtk/gtk.h>

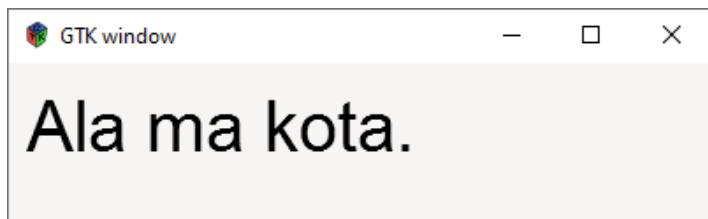
static void do_drawing(cairo_t *);

static gboolean on_draw_event(GtkWidget *widget, cairo_t *cr, gpointer user_data)
{
    do_drawing(cr);

    return FALSE;
}

static void do_drawing(cairo_t *cr)
{
    cairo_set_source_rgb(cr, 0, 0, 0);
    cairo_select_font_face(cr, "Sans", CAIRO_FONT_SLANT_NORMAL,
        CAIRO_FONT_WEIGHT_NORMAL);
    cairo_set_font_size(cr, 40.0);

    cairo_move_to(cr, 10.0, 50.0);
    cairo_show_text(cr, "Ala ma kota.");
}
```



```
int main(int argc, char *argv[])
{
    GtkWidget *window;
    GtkWidget *darea;

    gtk_init(&argc, &argv);

    window = gtk_window_new(GTK_WINDOW_TOPLEVEL);

    darea = gtk_drawing_area_new();
    gtk_container_add(GTK_CONTAINER(window), darea);

    g_signal_connect(G_OBJECT(darea), "draw", G_CALLBACK(on_draw_event), NULL);
    g_signal_connect(window, "destroy", G_CALLBACK(gtk_main_quit), NULL);

    gtk_window_set_position(GTK_WINDOW(window), GTK_WIN_POS_CENTER);
    gtk_window_set_default_size(GTK_WINDOW(window), 400, 90);
    gtk_window_set_title(GTK_WINDOW(window), "GTK window");

    gtk_widget_show_all(window);

    gtk_main();

    return 0;
}
```