

PoP C++ Série 5 corrigé



Exercice 4 (niveau 1) : dessiner le nouveau Logo de l'EPFL =>

La proposition de solution de Martin Simic a été ajustée pour prendre en compte l'épaisseur de trait différente pour la barre verticale du E et du F. Voici le signal handler on_draw :

```
bool Logo::on_draw(const Cairo::RefPtr<Cairo::Context>& cr)
{
    cr->set_source_rgb(0.92, 0.0, 0.0);

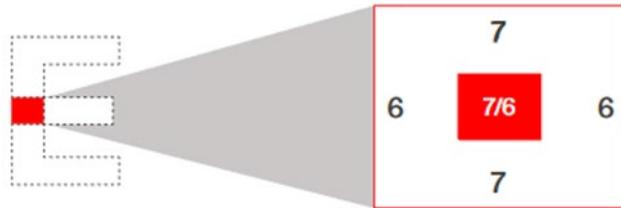
    // E
    cr->set_line_width(58.0); //~(7/6)50
    cr->move_to(50, 50);
    cr->line_to(50, 150);
    cr->stroke();
    cr->move_to(50, 200);
    cr->line_to(50, 300);
    cr->stroke();
    cr->set_line_width(50.0);
    cr->move_to(50, 275);
    cr->line_to(194, 275);
    cr->stroke();
    cr->move_to(79, 175); //E milieu
    cr->line_to(185, 175);
    cr->stroke();
    cr->move_to(50, 75);
    cr->line_to(194, 75);
    cr->stroke();

    //P
    cr->move_to(250, 50);
    cr->line_to(250, 300);
    cr->stroke();
    cr->move_to(275, 75);
    cr->line_to(325, 75);
    cr->stroke();
    cr->move_to(275, 175);
    cr->line_to(325, 175);
    cr->stroke();
    cr->arc(325, 125, 50, -M_PI/2, M_PI/2);
    cr->stroke();

    // F
    cr->set_line_width(58.0); //~(7/6)50
    cr->move_to(450, 50);
    cr->line_to(450, 150);
    cr->stroke();
    cr->move_to(450, 200);
    cr->line_to(450, 300);
    cr->stroke();
    cr->set_line_width(50.0);
    cr->move_to(479, 175); //F milieu
    cr->line_to(585, 175);
    cr->stroke();
    cr->move_to(450, 75);
    cr->line_to(594, 75);
    cr->stroke();

    //L
    cr->move_to(650, 50);
    cr->line_to(650, 300);
    cr->stroke();
    cr->move_to(650, 275);
    cr->line_to(790, 275);
    cr->stroke();

    return true;
}
```



On remarquera les points suivants :

1. L'arrondi de la lettre P est produit avec un seul appel de la méthode `arc()`
2. Ce dessin est exprimé dans l'espace GTKmm avec des coordonnées absolues indépendantes de la taille de la fenêtre (exprimée par **width** et **height**). Dans l'exercice 3 (niveau 0) le dessin s'adapte à la taille de la fenêtre car toutes les coordonnées des points sont soit l'origine soit fonction de **width** et **height** dont la valeur est remise à jour avant de refaire le dessin. Dans le cas présent le choix de coordonnées absolues fait que le dessin n'est pas modifié quand on change la taille de la fenêtre. C'est une bonne chose pour un logo pour lequel les proportions sont importantes. D'un autre côté si la taille de la fenêtre devient trop petite on ne le voit plus en entier.
3. Pour préserver les proportions quand on change la taille de la fenêtre il faudrait exprimer le dessin du logo dans un espace normalisé spécifique au logo puis mettre en place une transformation de coordonnées du type de celle vue en cours qui s'assure que :
 - a. Le cadrage $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$ est dans les mêmes proportions que la fenêtre.
 - b. Le cadrage ajusté permette de voir l'ensemble du logo dans la fenêtre