

LIFAP1 – TP8 : Introduction à GRaPiC

Objectifs : Prise en main de la bibliothèque graphique GRaPiC

Prise en main de l'outil GRaPiC : Graphics for Algo/Prog in C/C++

Pour réaliser les exercices demandés, reprenez vos notes du TD 9 et n'hésitez pas à consulter les tutoriels disponibles en ligne : <http://liris.cnrs.fr/alexandre.meyer/grapic/html/index.html> .

1. Téléchargez la dernière version de GRaPiC et installez-la **sur votre compte (pas sur le bureau)**.

Ouvrez ensuite le fichier `Graptic/build/windows/grapic_workspace` avec Codeblocks. Sélectionnez dans la partie gauche de la fenêtre le projet que vous souhaitez utiliser (LIFAP1_TP3) puis à l'aide du bouton droit, choisissez `Activate project`.

Vous trouverez une vue des résultats qu'on souhaite obtenir dans les questions suivantes en bas de cette page.

2. Ecrivez un sous-programme qui affiche au centre de la fenêtre un carré dont la taille sera passée en paramètre. Vous utiliserez la procédure GRaPiC `Rectangle`.

```
void draw_carre(int taille)
{
    int i;
    rectangle( DIMW/2 - taille/2, DIMW/2 - taille/2, DIMW/2 + taille/2, DIMW/2 + taille/2);
}

```

3. Réutilisez la procédure précédente pour afficher `n` (passé en paramètres) carrés imbriqués.

```
void draw_carre_imbrique(int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        draw_carre(10*i);
    }
}

```

4. Ecrivez un sous-programme qui dessine un cercle plein dont le rayon et la couleur sont passés en paramètres. Vous utiliserez pour cela `CircleFill`.

```
void draw_cercle(int rayon, int couleur_R, int couleur_G, int couleur_B)
{
    int i;

    color(couleur_R,couleur_G, couleur_B);
    circleFill( DIMW/2, DIMW/2, rayon);
}

```

5. En utilisant la procédure précédente, dessinez à présent une cible (alternance de cercles concentriques noirs et blancs).

```
void draw_cible(int taille)
{
    int i, couleur;
    for(i=0;i<taille;i++)
    {
        if (i%2 == 0)
            couleur=0;
        else
            couleur = 255;
        color(couleur,couleur, couleur);
        circleFill(DIMW/2, DIMW/2, DIMW/2 - i*DIMW/2/taille);
    }
}

```

6. Dessinez une grille de carrés produisant un dégradé de couleur du vert (dans le coin supérieur gauche) au rouge (dans le coin inférieur droit).

```
void draw_degrade()
{
    int i,j;
    const int l = DIMW/10;
    for(i=0;i<10;++i)
        for(j=0;j<10;++j)
        {
            color( 255*i*10/100, 255*j*10/100, 0);
            rectangleFill( i*l+1, j*l+1, (i+1)*l-1, (j+1)*l-1 );
        }
}
```

7. Ecrivez un sous-programme qui affiche une image dont la taille diminue au cours du temps.

```
void draw_retrecit(Image im)
{
    int i;
    const int nb_frame = 30;
    for (i=0;i<nb_frame;i++)
    {
        winClear();
        image_draw( im, DIMW/2-(400-i*400/nb_frame)/2, DIMW/2-(400-
i*400/nb_frame)/2, 400-i*400/nb_frame, 400-i*400/nb_frame);
        winDisplay();
        delay(2000/nb_frame);
    }
}
```

8. En utilisant 3 images différentes, générez une grille d'images positionnées aléatoirement.

```
void draw_grille_image(Image im1,Image im2,Image im3)
{
    int i,j,num_image;
    const int nb_line = 6;
    const int nb_row = 6;
    for(i=0;i<nb_line;i++)
        for(j=0;j<nb_row;j++)
        {
            num_image = rand () % 3;
            if (num_image ==0) image_draw( im1, i*(DIMW/nb_line), j*(DIMW/nb_row),
(DIMW/nb_line), (DIMW/nb_row));
            else if (num_image ==1) image_draw(im2, i*(DIMW/nb_line), j*(DIMW/nb_row),
(DIMW/nb_line), (DIMW/nb_row));
            else image_draw(im3, i*(DIMW/nb_line), j*(DIMW/nb_row), (DIMW/nb_line),
(DIMW/nb_row));
        }
    delay(5000);
}
```

9. Sans utiliser la procédure Circle de GRaPiC, dessinez un cercle plein dont le rayon r est passé en paramètre et contenant le caractère c également passé en paramètre. En supposant que le centre du cercle est au centre de la fenêtre, vous utiliserez le calcul de la distance au centre de tout point de la fenêtre pour déterminer s'il appartient ou non au cercle.

Rappel : Soient A (x, y) et O (x₀, y₀) alors la distance entre A et O est $\sqrt{(x-x_0)^2 + (y-y_0)^2}$

```
void draw_cercle_caractere(int r, char c[1])
{
    int i,j;
    float abscisse, ordonnee,dist;

    for(i=0;i<DIMW;i+=10)
    {
        for(j=0;j<DIMW;j+=10)
        {
            dist = sqrt(pow(i-DIMW/2,2)+pow(j-DIMW/2,2));
            if (dist<=r)
            {
                color(255,0,0);
                print(i,j,c);
            }
        }
    }
}
```

