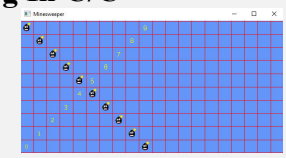
LIFAP1 – TD 9 : Utilisation de la bibliothèque GrAPiC

Objectifs : Présenter les bases d'un affichage graphique en se basant sur GrAPiC

GrAPiC = Graphic for Algo/Prog In C/C++



http://licence-info.univ-lyon1.fr/grapic

GrAPiC fournit un petit nombre de fonctions simples d'utilisation pour l'affichage de formes de base, d'image et de texte ; ainsi que des fonctions pour gérer le clavier et la souris.

Un exemple simple pour commencer

```
int main(int argc, char** argv)
{
    winInit("Simple", 500, 500);   // Créer une fenêtre (taille 500x500)
    winClear();                    // Efface la fenêtre
    line( 10, 0, 10, 499);         // Dessine une ligne verticale en x=10
    winDisplay();                  // Affiche réellement à l'écran
    pressSpace();                  // Attend l'appui sur « espace » pour fermer
    winQuit();                     // Ferme la fenêtre et quitte
    return 0;
}
```

Une fenêtre est constituée de DX x DY pixels, dans notre exemple 500 par 500. Un pixel est un point de la fenêtre.

1. Dessiner une grille où chaque colonne/ligne est espacée de 10 pixels. Vous utiliserez pour cela la procédure `line(int xmin, int ymin, int xmax, int ymax)` qui dessine une ligne entre le point de coordonnées (xmin,ymin) et le point de coordonnées (xmax,ymax).

```
int main(int argc, char** argv)
{
   int i;
   winInit("Simple", 500, 500);
   winClear();
  for(i=0;i<DIMW;i+=10)
   {
      line(i,0,i,DIMW) ; // lignes verticales
      line(0,i,DIMW,i); //lignes horizontales
   }
   winDisplay() ;
   pressSpace();
   winQuit();
   return 0;
}
```

2. Dessiner une série de N lignes verticales, allant de y=0 à y=cos(x). Les fonctions mathématiques sinus et cosinus retournant des valeurs dans [-1 ; 1], un coefficient multiplicateur devra être appliqué.

```
int main(int argc, char** argv)
{
   int i;
   float y;
   winInit("Simple", 500, 500);
   winClear();
     for(i=0;i<DIMW;i+=3)
   {
      y = 0.5*DIMW + 0.5*DIMW*cos(0.05*i);
      line(i,0,i, y);
```

```
    }
    winDisplay() ;
    pressSpace();
    winQuit();
    return 0;
}
```

3. Dessiner 20 carrés de largeur 10 disposés en cercle autour du centre de la fenêtre (cf. image 1). Vous utiliserez pour cela la procédure `rectangle(int xmin, int ymin, int xmax, int ymax)` qui dessine un rectangle entre le point de coordonnées (xmin,ymin) et le point de coordonnées (xmax,ymax).

```
int main(int argc, char** argv)
{
   int i,n=20;
   float angle,rayon=200.0;
   winInit("Simple", 500, 500);
   winClear();
  for (i = 0; i < n; ++i)
   {
     angle = 2.f * M_PI*i/ n;
     rectangle(250+rayon*cos(angle)-10, 250+rayon*sin(angle)-10,
     250+rayon*cos(angle)+10, 250+rayon*sin(angle)+10);
   }
winDisplay() ;
pressSpace();
winQuit();
   return 0;
}
```

4. Dessiner une rosace constituée de 36 cercles dont les centres décrivent un cercle autour du centre de la fenêtre (cf. image 2). On utilisera la procédure `circle(int centre_x, int centre_y, int rayon)`.

```
int main(int argc, char** argv)
{
   int i,n=36;
   float angle,rayon=70.0;
   winInit("Simple", 500, 500);
   winClear();
   for (i = 0; i < n; ++i)
   {
     angle = 2.f * M_PI*i/ n;
     circle(DIMW/2+rayon*cos(angle),DIMW/2+rayon*sin(angle),150);
   }
   winDisplay() ;
   pressSpace();
   winQuit();
   return 0;
}
```

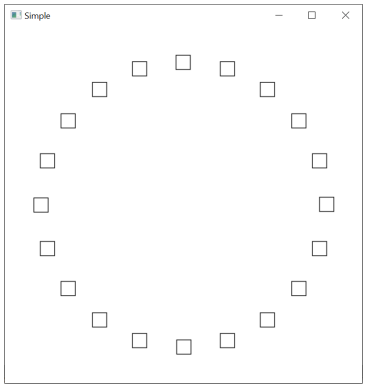


image 1

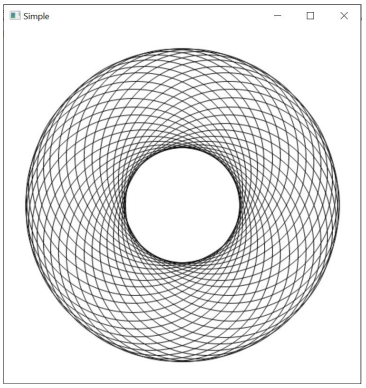


image 2

Les couleurs

Une couleur est représentée par un triplet de valeurs : (Rouge, Vert et Bleu).
(0,0,0) représente le noir
(255, 0, 0) représente le rouge
(0, 255, 0) représente le vert
(0, 0, 255) représente le bleu
(255, 255, 0) représente le jaune

```
#include <Grapic.h>
using namespace grapic;

int main(int argc, char** argv)
{
    bool stop=false ;
    winInit("Simple", 500, 500);   // Créer une fenêtre (taille 500x500)
    backgroundColor( 0, 0, 255 );  // Couleur de fond à bleu
    winClear();                    // Efface la fenêtre avec du bleu

    color(255,0,0);                // Couleur du pinceau rouge (RGB)
    rectangleFill( 10, 10, 490, 490);  // Dessine un rectangle rouge
    winDisplay() ;
    pressSpace();
    winQuit();

    return 0;
}
```

5. Donner les valeurs de RGB pour obtenir du blanc ? du gris ?

```
Blanc = color (255,255,255) ;
Gris = color (127,127,127) ;
```

6. Dessiner un dégradé allant du noir au blanc à l'aide de la procédure `rectangleFill(int xmin, int ymin, int xmax, int ymax)` (cf. image 3).

```
int main(int argc, char** argv)
{
   winInit("Simple", 500, 500);
   winClear();
       color(125,125,125);
   int i;
   int n=250;
   for(i=0;i<n;i++)
   {
      color(i,i,i);
      rectangleFill( 2*i,0,2*i+1,500);
   }
   winDisplay();
   pressSpace();
   winQuit();
   return 0;
}
```

7. Dessiner un dégradé de cercles colorés allant du noir (cercle externe) au rouge (cercle interne) à l'aide de la procédure `circleFill(int centre_x, int centre_y, int rayon)` (cf. image 4).

```
int main(int argc, char** argv)
{
   int i,n=50;
   float angle,rayon=70.0;
   winInit("Simple", 500, 500);
   winClear();
   for (i = 0; i < n; ++i)
```

```
    {
      angle = 2.f * M_PI*i/ n;
      color(255*i/n,0,0);
      circleFill(DIMW/2,DIMW/2,200-200/n*i);
    }
   winDisplay();
   pressSpace();
    winQuit();
    return 0;
}
```

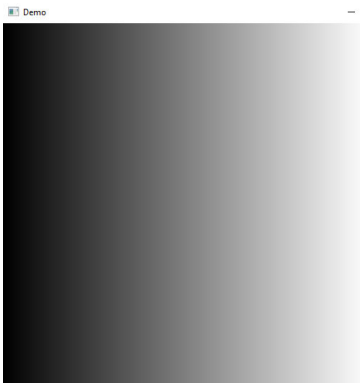


Image 3

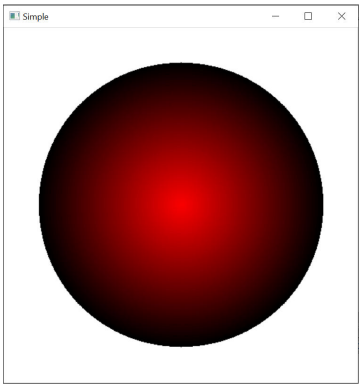


Image 4

Les images et le texte

```
#include <Grapic.h>
using namespace grapic;

int main(int argc, char** argv)
{
    bool stop=false;
    winInit("Simple", 500, 500);        // Créer une fenêtre (taille 500x500)
    backgroundColor( 0, 0, 255 );       // Couleur de fond à bleu
    color(255,0,0);                     // Couleur du pinceau rouge (RGB)
    Image im = image("data/grapic.png");
    winClear();                         // Efface la fenêtre avec du bleu
    image_draw( im, 50, 50, 400, 400); // Dessine l'image, coin inférieur gauche
                                        // (50,50), image redimensionnée en 400x400
    print(10,10, "Oh !") ;              // Affiche la chaine "Oh !" en (10,10)
    print(30,30, 99) ;                  // Affiche le nombre 99 en (30,30)
    winDisplay() ;
    pressSpace();
    winQuit();                          // Ferme la fenêtre et quitte
    return 0;
}
```

8. Dessiner une horloge (en prenant exemple sur l'exercice 3) à l'aide de la procédure `print(int x, int y, const char *txt)` où `const char *txt` est une chaine de caractères (cf. image 5).

```
int main(int argc, char** argv)
{
   winInit("Simple", 500, 500);
   winClear();
   int i;
   float a;
```

```
    for(i=0;i<12;++i)
    {
        a = -(-0.5f*M_PI + 2.f*M_PI*float(i+1)/12);
        color(0,0,0);
        print( 250+100*cos(a), 250+100*sin(a), i+1);
    }
    winDisplay() ;
    pressSpace();
    winQuit();
    return 0;
}
```

9. Ecrire un programme qui affiche un damier de deux images, chaque case du damier fera 50x50, il y aura donc 10x10 cases au total car la fenêtre fait 500x500. (cf. image 6). On utilisera les fonctions `image(const char *emplamcement)` qui charge en mémoire l'image se trouvant à l'emplacement donné et `image_draw(Image im, int x, int y, int taille_x, int taille_y)` qui affiche l'image à l'écran aux coordonnées (x,y) selon la taille fournie.

```
int main(int argc, char** argv)
{
    winInit("Simple", 500, 500);
    winClear();
    Image im1=image("data/pacman/pacman.png");
    Image im2=image("data/pacman/fantome.png");

    int i,j;
    //image_draw(im1,100,100,20,20);
    for(i=0;i<10;i++)
        for(j=0;j<10;j++)
        {
            if ( (i+j)%2==0)
                image_draw(im1 , i*50, j*50, 50, 50);
                else image_draw( im2, i*50, j*50, 50, 50);
        }
    winDisplay() ;
    pressSpace();
    winQuit();
    return 0;
}
```

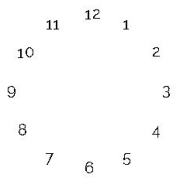


Image 5

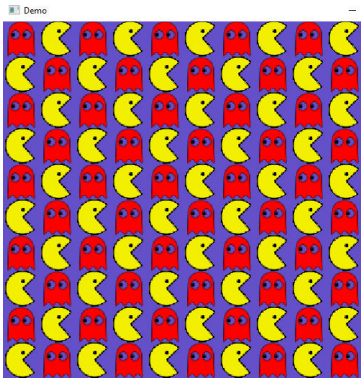


Image 6