

4 Le graphisme (2^{nde} partie)

4.1 Valeur de retour de la fonction main ()

La fonction main peut être déclarée avec une valeur de retour :

```
int main ()
```

ou sans:

```
void main ()
```

Le choix entre ces deux façons réside dans l'utilisation de votre programme :

- . le fait de renvoyer une valeur (un code erreur) permet lors de l'exécution de votre programme par un autre programme de savoir si tout s'est bien passé.

- . le fait de ne pas renvoyer de code erreur rend impossible le fait de savoir si le programme s'est bien terminé (sauf de manière visuelle (en l'exécutant)).

On utilisera généralement void main () pour des programmes de test et int main () pour des vrais programmes.

4.2 Motif de remplissage et contour

Dans tous les exemples qui suivront, j'utiliserai la fonction Initialize () écrite dans le précédent cours.

4.2.1 Exemple

Tapez cet exemple et en vous appuyant sur l'aide en ligne, comprenez l'utilisation des fonctions graphiques setlinestyle, setfillstyle, fillellipse.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<dos.h>
#include<graphics.h>

int main ()
{
    clrscr();

    /* Ouverture du mode graphique */
    if (Initialize ())
        return (1);

    /* Définit la couleur de tracé des traits */
    setcolor (15);

    /* Définit l'épaisseur et le style courants des lignes */
    setlinestyle (DOTTED_LINE, 0, NORM_WIDTH);

    /* Définit le motif et la couleur de remplissage */
    setfillstyle (SOLID_FILL, 12);

    /* Dessine une ellipse de rayon x=15 y=15 donc un cercle rayon 15 */
    fillellipse (30,30,15,15);

    /* Attend l'appui d'une touche : permet de voir ce que l'on fait */
```

```
    getch ();

    /* Termine le mode graphique */
    closegraph();
}
```

4.2.2 Exercices d'application

- 🚧 Changer la couleur du contour
- 🚧 Changer le motif de remplissage.
- 🚧 Changer la couleur du motif de remplissage

4.3 Rectangles et barres

4.3.1 Exemple

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<dos.h>
#include<graphics.h>

int main ()
{
    clrscr();

    /* Ouverture du mode graphique */
    if (Initialize ())
        return (1);

    /* Définit la couleur de tracé des traits */
    setcolor (13);

    /* Définit le motif et la couleur de remplissage */
    setfillstyle (SOLID_FILL, 11);

    /* Barre 3D */
    bar3d (50,50,200,200,13,1);

    /* Attend l'appui d'une touche : permet de voir ce que l'on fait */
    getch ();

    closegraph();
}
```

4.3.2 Explications

void far bar3d (int left, int top, int right, int bottom, int depth, int topflag);

L'aide de **bar3d** n'est pas explicite, en voici donc une explication plus complète (extrait du Manuel de Référence) :

bar3d dessine une barre rectangulaire en trois dimensions (dans l'espace), et la remplit à l'aide du motif et de la couleur de remplissage courants. Le contour dans l'espace de cette barre est tracé dans le style et avec la couleur de tracé courants. La profondeur de la barre est donnée en pixels par l'argument *depth*. L'argument *topflag* sert à

déterminer si la barre possède une face supérieure. Si `topflag` est non nul il y a une face supérieure sinon il n'y en a pas (ce qui permet l'empilage de plusieurs barres). Pour la profondeur de la barre, prendre par exemple 25 % de la barre.

On reconnaît dans `topflag` le fameux type vrai faux (vrai si différent de 0, faux sinon).

4.3.3 Exercices d'application

- ✍️ Changer la couleur de trait
- ✍️ Dessiner un rectangle plein

4.4 Texte graphique : oh les belles lettres ...

4.4.1 Exemple

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<dos.h>
#include<graphics.h>

int main ()
{
    clrscr();

    /* Ouverture du mode graphique */
    if (Initialize ())
        return (1);

    /* Définit la couleur du texte */
    setcolor (10);

    /* Définit le style du texte */
    settextstyle (TRIPLEX_FONT,VERT_DIR,5);

    /* Affiche le texte */
    outtextxy (30,30,"Raven come back");

    /* Attend l'appui d'une touche : permet de voir ce que l'on fait */
    getch ();

    /* Ferme le mode graphique */
    closegraph();
}
```

4.4.2 Taille du texte

Pour connaître la taille des caractères utilisés, nous devons utiliser la fonction :

void gettextsettings (struct textsettingstype far* texttypeinfo) ;

Aïe, quelque chose que je ne connais pas ...

Le mot de syntaxe `struct` nécessiterait un cours à lui tout seul. Nous allons donc utiliser une autre méthode pour connaître la taille des caractères. Relisez l'aide en ligne associée à `settextstyle` et plus spécifiquement l'aide associée au paramètre **charsize**. Eh oui, nous avons

la clé de notre problème de taille de caractères, la taille en pixels des caractères est égal à `charsize * 8`.

Dans l'exemple précédent où `charsize` était de 5 nous avons donc des caractères de 40x40 pixels.

4.4.3 Exercices d'application

- 🐾 Changer le texte ...
- 🐾 Changer la couleur du texte

4.5 Exercice complet et instructif ...

A l'aide de l'exercice 3.4 du cours n°3, réaliser un menu en mode graphique.

Spécifications :

1. Dessiner le menu en mode graphique et en caractères gothiques.
2. Ajoutez un menu texte horizontal qui affiche un texte en horizontal en caractères normaux (DEFAULT_FONT).
3. Ajoutez un menu texte horizontal qui affiche un texte en vertical en caractères normaux (DEFAULT_FONT).

Correction des exercices du chapitre 4

Exercice 4.2.2

-  Changer la couleur du contour
-  Changer le motif de remplissage.
-  Changer la couleur du motif de remplissage

```
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

int Initialize(void)
{
    int    GraphDriver = DETECT;    /* Driver de carte graphique */
    int    ErrorCode;              /* Rapport des codes erreurs */
    int    GraphMode=0;

    initgraph( &GraphDriver, &GraphMode, "c:\\\\tc" );

    ErrorCode = graphresult();
    /* Lecture du résultat de l'initialisation */

    if( ErrorCode != grOk )
    {
        printf("Erreur de système graphique : %s\n",
            grapherrormsg(ErrorCode ));
        return ( 1 );    /* Problème */
    }
}

int main ()
{
    char car = ' ';
    int couleur = 1; /* On évite le 0 car noir sur fond noir !!!! */
    int sortie = 0;

    if (Initialize ())    /* Ouverture du mode graphique */
    {
        printf ("Impossible de passer en mode graphique");
        return (1);
    }

    /* Couleur du contour */
    setcolor (8);

    /* Définition du style de ligne */
    setlinestyle (4, 0, NORM_WIDTH);

    /* Définition du type et couleur de remplissage */
    setfillstyle (SLASH_FILL, 2);

    /* Ellipse pleine */
    fillellipse (30,30,15,15);

    /* Attente */
}
```

```
    getch ();  
    closegraph ();    /* Fermeture du mode graphique */  
    return 0;  
}
```

✍ Exercice 4.3.3

- ✍ Changer la couleur de trait
- ✍ Dessiner un rectangle plein

```
#include <graphics.h>  
#include <conio.h>  
#include <stdio.h>  
#include <stdlib.h>  
  
int Initialize(void)  
{  
    int    GraphDriver = DETECT;    /* Driver de carte graphique */  
    int    ErrorCode;    /* Rapport des codes erreurs */  
    int    GraphMode=0;  
  
    initgraph( &GraphDriver, &GraphMode, "c:\\\\tc" );  
  
    ErrorCode = graphresult();  
    /* Lecture du resultat de l'initialisation */  
  
    if( ErrorCode != grOk )  
    {  
        printf("Erreur de système graphique : %s\n",  
            grapherrormsg(ErrorCode) );  
  
        return ( 1 );    /* Problème */  
    }  
}  
  
int main ()  
{  
    char car = ' ';  
    int couleur = 1; /* On évite le 0 car noir sur fond noir !!!! */  
    int sortie = 0;  
  
    if (Initialize ())    /* Ouverture du mode graphique */  
    {  
        printf ("Impossible de passer en mode graphique");  
        return (1);  
    }  
  
    /* Couleur du contour */  
    setcolor (4);  
  
    /* Définition du type et couleur de remplissage */  
    setfillstyle (SLASH_FILL, 4);  
  
    /* Ellipse pleine */  
    bar3d (50,50,200,200,0,1);  
  
    /* Attente */  
    getch ();  
    closegraph ();    /* Fermeture du mode graphique */  
    return 0;  
}
```

✎ Exercice 4.4.3

🐾 Changer le texte ...

🐾 Changer la couleur du texte

```
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

int Initialize(void)
{
    int    GraphDriver = DETECT;    /* Driver de carte graphique */
    int    ErrorCode;              /* Rapport des codes erreurs */
    int    GraphMode=0;

    initgraph( &GraphDriver, &GraphMode, "c:\\\\tc" );

    ErrorCode = graphresult();
    /* Lecture du résultat de l'initialisation */

    if( ErrorCode != grOk )
    {
        printf("Erreur de système graphique : %s\n",
            grapherrormsg(ErrorCode) );

        return ( 1 );    /* Problème */
    }
}

int main ()
{
    char car = ' ';
    int couleur = 1; /* On évite le 0 car noir sur fond noir !!!! */
    int sortie = 0;

    if (Initialize ())    /* Ouverture du mode graphique */
    {
        printf ("Impossible de passer en mode graphique");
        return (1);
    }

    /* Couleur du contour */
    setcolor (4);

    /* Définition du type et couleur de remplissage */
    settextstyle (TRIPLEX_FONT, VERT_DIR, 5);

    /* Affichage du texte */
    outtextxy (30,30, "Coucou");

    /* Attente */
    getch ();

    /* Fermeture du mode graphique */
    closegraph ();
    return 0;
}
```

✎ Exercice 4.5

Dessiner le menu en mode graphique et en caractères gothiques.

Ajoutez un menu texte horizontal qui affiche un texte horizontal en caractères normaux.

Ajoutez un menu texte horizontal qui affiche un texte vertical en caractères normaux.

```
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

int Initialize(void)
{
    int    GraphDriver = DETECT;    /* Driver de carte graphique */
    int    ErrorCode;              /* Rapport des codes erreurs */
    int    GraphMode=0;

    initgraph( &GraphDriver, &GraphMode, "c:\\\\tc" );

    ErrorCode = graphresult();
    /* Lecture du resultat de l'initialisation */

    if( ErrorCode != grOk )
    {
        printf("Erreur de système graphique : %s\n",
            grapherrormsg(ErrorCode) );

        return ( 1 );    /* Problème */
    }
}

int main ()
{
    char car = ' ';
    int couleur = 1; /* On évite le 0 car noir sur fond noir !!!! */
    int sortie = 0;

    if (Initialize ())    /* Ouverture du mode graphique */
    {
        printf ("Impossible de passer en mode graphique");
        return (1);
    }

    /* Choix de la police de caractère */
    settextstyle (GOTHIC_FONT, HORIZ_DIR, 3);

    /*
    La taille des caractères est maintenant de 3*8 = 24 pixels
    Afin de laisser de la place pour mon rectangle j',cris donc
    une ligne tous les 30 pixels et ... 10 pixels du bord de l',cran
    */

    do
    {
        /* Affichage du menu */
        cleardevice ();
        setcolor (couleur);
        outtextxy (10,30,"1. Dessiner une ligne.");
        outtextxy (10,60,"2. Dessiner un rectangle.");
        outtextxy (10,90,"3. Dessiner un cercle.");
        outtextxy (10,120,"4. Changer la couleur.");
    }
```

*Initiation au langage C.
Niveau 2.*

```
outtextxy (10,150,"5. Texte horizontal.");
outtextxy (10,180,"6. Texte vertical.");
outtextxy (10,250,"9. Sortir.");

car = getch ();
switch (car)
{
    case '1':
        cleardevice ();
        line (100,100,250,370);
        outtextxy (200,400,"Appuyez sur une touche");
        getch ();
        break;

    case '2':
        cleardevice ();
        rectangle (100,200,350,400);
        outtextxy (200,400,"Appuyez sur une touche");
        getch ();
        break;

    case '3':
        cleardevice ();
        circle (150,250,100);
        outtextxy (200,400,"Appuyez sur une touche");
        getch ();
        break;

    case '4':
        couleur ++;
        if (couleur > 15)
            couleur = 1;
        break;

    case '5':
        cleardevice ();

        /* On passe en mode normal */
        settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
        outtextxy (100,100, "Ceci est un texte");

        /* On revient au style gothique */
        settextstyle (GOTHIC_FONT, HORIZ_DIR, 3);

        outtextxy (200,400, "Appuyez sur une touche");
        getch ();
        break;

    case '6':
        cleardevice ();

        /* On passe en mode normal */
        settextstyle (DEFAULT_FONT, VERT_DIR, 1);
        outtextxy (100,100, "Ceci est un texte");

        /* On revient au style gothique */
        settextstyle (GOTHIC_FONT, HORIZ_DIR, 3);

        outtextxy (200,400, "Appuyez sur une touche");
        getch ();
        break;
}
```

```
        case '9':  
            sortie = 1;  
            break;  
        }  
    }  
while (!sortie);  
  
closegraph ();    /* Fermeture du mode graphique */  
return 0;  
}
```