

---

## *Mini Projet de programmation en langage C*

---

### 1 – Présentation générale, objectifs

Le développement d'un projet est un moyen de progresser dans l'apprentissage d'un langage de programmation. C'est dans cet esprit qu'il vous est proposé.

Il doit vous permettre de consolider les connaissances acquises en TP et d'en acquérir de nouvelles.

Pour cela, vous devrez développer, au cours du 2<sup>ème</sup> semestre, un programme dont vous vous aurez vous même défini les fonctionnalités.

En fin de semestre, votre travail sera évalué par les enseignants, lors d'une séance de présentation orale du travail réalisé. Vous devrez également remettre un mini-dossier récapitulatif de votre travail et faire une démonstration de votre programme.

### 2 - Thème du projet

Cette année, le projet devra obligatoirement mettre en œuvre une bibliothèque de fonctions graphiques. (voir doc annexe sur WINBGI)

Dans ce contexte, il s'agira de réaliser :

- Soit un jeu mettant en œuvre des objets animés (deux objets au moins).
- Soit une application de tracé de courbes liées à des systèmes physiques (exemple, circuit RLC ...)
- Soit une autre application, proposée par vous, et dont le thème aura été préalablement soumis à votre enseignant pour validation.

### 4 – Eléments de programmation à explorer et à mettre en oeuvre

Indépendamment du thème choisi, le programme que vous aurez réalisé devra faire apparaître l'emploi d'éléments du langage C peu ou pas abordés en séance de TP. On peut citer par exemple :

- utilisation de sous-programmes paramétrés,
- tableaux et pointeurs,
- passage de paramètres par adresse,
- structures de données,
- opération sur des fichiers ...

### 5 – Organisation / Calendrier

Le développement du mini projet doit être réalisé en binôme.

Il se déroule en autonomie tout au long du 2<sup>ème</sup> semestre, c'est-à-dire en dehors des séances de TD/TP.

Toutefois, en cas de difficulté vous pourrez poser des questions - en fin de séance- aux enseignants qui encadrent les TP d'Informatique Industrielle ou contacter par E-mail :

M. DESRAT [desrat@ges.univ-montp2.fr](mailto:desrat@ges.univ-montp2.fr)

Ou M. FIOL [fiol@iut-nimes.fr](mailto:fiol@iut-nimes.fr)

En fin de semestre, soit lors de la semaine du 26 au 30 mai 2008, le créneau horaire des TP d'Informatique Industrielle sera consacré à la clôture des projets.

Lors de cette séance, chaque binôme devra présenter son projet à un jury d'enseignant et devra rendre un **mini dossier** (voir contenu ci-après).

La durée de la présentation orale ne devra pas dépasser 10 mn par binôme.

## 6 – Présentation, contenu du mini-dossier

Le mini-dossier à remettre en fin de semestre sera structuré comme suit :

- a) Page de couverture : Titre, auteurs, date
- b) Sommaire paginé
- c) Introduction : Présente le contenu du dossier
- d) Cahier des charges : Développé par les auteurs sur la base du sujet choisi
- e) Analyse : Organigramme général du programme
- f) Description des parties mettant en œuvre les éléments décrits en §4-
- g) Conclusions individuelles

Annexes :

- listing du programme
- CD avec programme source et copie du dossier au format RTF ou Word
- références bibliographiques et/ou de sites Internet utilisés

## 7 – Evaluation

L'évaluation du mini-projet de programmation sera réalisée :

- Au vu du travail réalisé tout au long du semestre. Ainsi, lors des séances de TP les enseignants suivront l'avancement de votre projet, pourront vous poser des questions visant à évaluer votre progression...
- Au vu de l'exposé oral et des réponses aux questions qui feront suite
- Au vu du mini-dossier et plus particulièrement à travers la qualité du programme réalisé. Seront pris en considération aussi bien sa présentation générale que sa structuration et la maîtrise du langage qui en ressortira.

La note attribuée interviendra pour un poids équivalent à celui du contrôle semestriel d'IP du premier semestre.

## ANNEXE : Bibliothèque graphique "winbgim.h" pour Dev-C++ 4.9.9.2

### 1. Introduction

Nous allons utiliser une librairie d'émulation de fonctions graphiques, qui ont été implémentées à l'origine par Borland pour le compilateur Turbo C++. Ces fonctions BGI (Borland Graphics Interface) ont depuis été ré-écrites pour pouvoir être utilisées par différents compilateurs pour des applications Windows. Actuellement, la bibliothèque "winbgim.h" regroupe ainsi des fonctions permettant aussi bien de dessiner des figures géométriques, que de définir des couleurs, ou de déceler la frappe d'une touche.

### 2. Installation de la bibliothèque "winbgim.h"

Dans un premier temps, il faut télécharger le fichier suivant et le dézipper dans un répertoire temporaire [http://apcsteacher.com/reference/cpp/winbgim\\_devcpp4990\\_20041220.zip](http://apcsteacher.com/reference/cpp/winbgim_devcpp4990_20041220.zip). Déplacer les fichiers *winbgim.h* et *conio.h* dans le sous-répertoire **\include** du fichier racine de Dev-C++ et les fichiers *libbgi.a* et *libconio.a* dans le sous-répertoire **\lib**. Par défaut, le fichier racine de Dev-C++ est **C:\Dev-Cpp\**.

Ensuite, il suffit d'inclure la bibliothèque "winbgim.h" dans chaque nouveau programme dans lequel on veut utiliser les fonctions graphiques, via l'instruction **#include <winbgim.h>**.

Enfin, avant la première compilation de votre programme, il est nécessaire de modifier les options du compilateur (ligne de commandes à destination de l'éditeur de liens) de la manière suivante. Dans l'environnement Dev-C++, allez dans le menu **Outils**, puis **Options du compilateur**, cochez la case **Ajouter ces commandes lors de l'appel à l'éditeur de liens** et tapez la ligne:

```
-lbgi -lgdi32 -luser32
```

dans le champ en-dessous. Cliquez OK.

### 3. Première application graphique

Dans un nouveau fichier, copiez le programme ci-dessous et enregistrez-le. Compilez et exécutez. Si tout se déroule correctement, une fenêtre Windows BGI doit s'ouvrir et afficher le texte BONJOUR de différentes couleurs.

```
#include <winbgim.h>
main()
{
    int i;
    initwindow(200,100);
    setbkcolor(WHITE);
    cleardevice();
    for (i=0;i<=15;i++)
    {
        setcolor(i);
        settextstyle(4,0,2);
        outtextxy(25,35,"BONJOUR");
        delay(500);
    }
    closegraph();
}
```

### 4. Fonctions principales de la bibliothèque "winbgim.h"

La bibliothèque "winbgim.h" contient une centaine de fonctions. Nous décrivons ici les fonctions principales regroupées par thèmes pour des applications graphiques simples. La liste complète des fonctions est accessible sur : <http://www.cs.colorado.edu/%7Emain/cs1300/doc/bgi/index.html>

#### - Fenêtre graphique

**int initwindow(int width, int height);**

Fonction indispensable au début du programme. Elle ouvre une fenêtre graphique de largeur **width** et de hauteur **height**, exprimées en pixels. Note: pour toutes les fonctions suivantes, les dimensions et les positions seront toujours exprimées en pixels.

**void closegraph(void);**

Fonction associée, qui permet de fermer la fenêtre graphique précédemment ouverte. Elle se place généralement à la fin du programme.

**void cleardevice(void);**

Fonction qui efface la fenêtre graphique (équivalente à la commande "CLS" sous DOS). Son appel applique la couleur de fond définie par **setbkcolor()**.

- Couleurs

Par défaut, 16 couleurs de base sont prédéfinies, que l'on peut accéder, soit par un nombre entier compris entre 0 et 15, soit par le nom de couleur correspondant (en anglais et en majuscule). Exemples: 0=BLACK, 1=BLUE, 2=GREEN,...,14=YELLOW, 15=WHITE. Il est possible de définir d'autres couleurs en mode RGB.

**void setbkcolor(int color);**

Fonction qui définit la couleur de fond de la fenêtre graphique. **color** prend une valeur entre 0 et 15.

**void setcolor(int color);**

Fonction qui définit la couleur active pour les figures et le texte. Il faut donc appeler cette fonction chaque fois que l'on veut changer de couleur.

- Texte

**void settextstyle(int font, int direction, int charsize);**

Fonction qui définit le style actif du texte: **font** prend une valeur entière entre 0 et 11 et fait référence à l'une des 12 fontes prédéfinies (Script, Sans sérification, Gras, Gothique, etc), **direction** est égale à 0 ou 1 pour afficher le texte horizontalement ou verticalement, **charsize** fixe la taille de la police.

**void outtextxy(int x, int y, char \*textstring);**

Fonction affichant le texte **\*textstring** à la position (x,y) de la fenêtre graphique. Exemple: **outtextxy**(0, 0, "bonjour"); affiche bonjour dans le coin en haut à gauche.

- Dessin de figures élémentaires

**void setlinestyle(int linestyle, unsigned upattern, int thickness);**

Fonction qui définit le style actif des traits: **linestyle** peut prendre les valeurs 0:trait continu, 1: trait pointillé, etc **upattern** est égal à 0 et **thickness** fixe la largeur du trait.

**void setfillstyle(int pattern, int color);**

Fonction qui définit le style actif du remplissage des figures: **pattern** est un nombre entier compris entre 0 et 11 (0: vide, 1: plein, 2: rayures horizontales, etc) et **color** une des 16 couleurs de base.

**void circle(int x, int y, int radius);**

Fonction qui trace un cercle vide de rayon **radius** et dont les coordonnées du centre sont (x,y).

**void fillellipse(int x, int y, int xradius, int yradius);**

Fonction qui dessine une ellipse de rayon horizontal **xradius** et de rayon vertical **yradius** en (x,y).

**void rectangle(int left, int top, int right, int bottom);**

Fonction qui dessine un rectangle dont le bord gauche se situe à la position **left**, le bord supérieur à la position **top**, le bord droit à la position **right** et le bord inférieur à la position **bottom**.

**void line(int x1, int y1, int x2, int y2);**

Fonction qui trace une ligne entre les 2 points (x1,y1) et (x2,y2).

- Touches du clavier

**int getch(void);**

Fonction qui attend la frappe d'une touche au clavier, tel scanf() en mode console, et renvoie la valeur du caractère en ASCII. Il est possible de stocker ce caractère dans une variable via l'instruction c=(char) **getch()**;

**int kbhit(void);**

Fonction qui renvoie une valeur non nulle lorsqu'une touche du clavier, n'importe laquelle, est actionnée. Si aucune touche n'est frappée, la fonction renvoie la valeur nulle, et passe à l'instruction suivante. Cette fonction est souvent associée à **getch()**.

- Autres fonctions

**void delay(int milliseconds);**

Fonction qui temporise pendant un certain temps donné en millisecondes.

De nombreuses autres fonctions sont définies dans la bibliothèque "winbgim.h", concernant entre autres la manipulation d'images ou la gestion d'évènements liés à la souris. Le fichier *demobgi.cpp* présente une vue globale des multiples applications graphiques pouvant être réalisées.