Introduction à la Programmation

—

TP #2

—

G1: monasse*(at)*imagine.enpc.fr G2: nicolas.audebert*(at)*onera.fr
G3: alexandre.boulch*(at)*onera.fr G4: maxime.ferrera*(at)*onera.fr
G5: laurent.bulteau*(at)*u-pem.fr G6: pierre-alain.langlois*(at)*eleves.enpc.fr

1 Variables, boucles, conditions, fonctions

1.1 Premier programme avec fonctions

1. *Récupérer le programme exemple :*
   Télécharger l'archive `Tp2.zip` sur la page du cours, la décompresser sur le bureau et ouvrir la solution dans Visual. Etudier le projet `Hop` dont voici les sources :

```
1  #include <iostream>
2  using namespace std;
3
4  int plus(int a,int b)
5  {
6      int c;
7      c=a+b;
8      return c;
9  }
10
11 void triple1(int a)
12 {
13     a=a*3;
14 }
15
16 void triple2(int& a)
17 {
18     a=a*3;
19 }
20
21 int main()
22 {
23     int i,j=2,k;
24     i=3;
25     k=plus(i,j);
26     triple1(i);
27     triple2(i);
28     return 0;
29 }
```

2. *Debugger :*
   Exécuter le programme pas à pas et étudier la façon dont les variables changent. Souvenez-vous que si vous n'utilisez pas Visual, vous devez utiliser `Cmake` avec l'option `-DCMAKE_BUILD_TYPE=Debug`.

1.2 Premier programme graphique avec Imagine++

Dans ce TP et les suivants, nous utiliserons la librairie graphique d'Imagine++ (cf annexe du polycopié). Elle permet de gérer très simplement le fenêtrage, le dessin, et les entrées-sorties clavier et souris.

1. *Programme de départ :*
   Etudier le programme du projet `Tennis` dont voici le source :

```
#include <Imagine/Graphics.h>
using namespace Imagine;
...

////////////////////////////////////////////////////////
// Fonction principale
int main()
{
    // Ouverture de la fenetre
    openWindow(256,256);
    // Position et vitesse de la balle
    int xb=128,
        yb=20,
        ub=2,
        vb=3;
    // Boucle principale
    while (true) {
        // Affichage de la balle
        fillRect(xb-3,yb-3,7,7,Red);
        // Temporisation
        milliSleep(20);
        // Effacement de la balle
        fillRect(xb-3,yb-3,7,7,White);
        // Rebond
        if (xb+ub>253)
            ub=-ub;
        // Mise a jour de la position de la balle
        xb+=ub;
        yb+=vb;
    }
    endGraphics();
    return 0;
}
```

   Ne pas s'intéresser à la fonction Clavier()

   Générer puis exécuter la solution. Que se passe-t-il ?

2. *Aide de Visual Studio* A tout moment, la touche `F1` permet d'accéder à la documentation du mot clé situé sous le curseur. Tester cette fonctionnalité sur les mots-clés if, while et return.

   Touche utile : F1 = Accéder à la documentation

3. *Comprendre le fonctionnement du programme :*
   Identifier la boucle principale du programme. Elle se décompose ainsi :
   (a) Affichage de la balle
   (b) Temporisation de quelques millisecondes pour que la balle ne se déplace pas trop vite
   (c) Effacement de la balle
   (d) Gestion des rebonds
   (e) Mise à jour des coordonnées de la balle

   Pourquoi la ligne comportant l'instruction while suscite-t-elle un warning ? A quoi sert la condition formulée par l'instruction if ?

4. *Gestion de tous les rebonds :*
   Compléter le programme afin de gérer tous les rebonds. Par exemple, il faut inverser la vitesse horizontale ub quand la balle va toucher les bords gauche ou droit de la fenêtre.

5. *Variables globales :*
   Doubler la hauteur de la fenêtre. Modifier la taille de la balle. Cela nécessite de modifier le code à plusieurs endroits. Aussi, à la place de valeurs numériques "en dur", il vaut mieux définir des variables. Afin de simplifier et bien que ça ne soit pas toujours conseillé, utiliser des variables globales constantes. Pour cela, insérer tout de suite après les deux lignes d'include le code suivant

```
const int width = 256;    // Largeur de la fenetre
const int height = 256;   // Hauteur de la fenetre
const int ball_size = 3; // Rayon de la balle
```

et reformuler les valeurs numériques du programmes à l'aide de ces variables. Le mot clé const indique que ces variables ne peuvent être modifiées après leur initialisation. Essayer de rajouter la ligne width=300; au début de la fonction main et constater que cela provoque une erreur de compilation.

6. *Utilisation de fonctions :*
   La balle est dessinée deux fois au cours de la boucle, la première fois en rouge et la deuxième fois en blanc pour l'effacer. Ici le dessin de la balle ne nécessite qu'une ligne mais cela pourrait être beaucoup plus si sa forme était plus complexe. Aussi, pour que le programme soit mieux structuré et plus lisible, et que le code comporte le moins possible de duplications, regrouper l'affichage de la balle et son effacement dans une fonction DessineBalle définie avant la fonction main :

   ```
   void DessineBalle(int x,int y,Color col) {
       ...
   }
   ```

   De même, définir une fonction
   void BougeBalle(int &x,int &y,int &u,int &v)
   pour gérer les rebonds et le déplacement de la balle.

1.3 Jeu de Tennis

Nous allons rendre ce programme plus ludique en y ajoutant deux raquettes se déplaçant horizontalement en haut et en bas de l'écran, et commandées par les touches du clavier.

1. *Affichage des raquettes :*
   Ajouter dans la fonction main des variables xr1,yr1,xr2,yr2 dédiées à la position des deux raquettes. Puis définir une fonction DessineRaquette en prenant modèle sur DessineBalle. Placer les appels de ces fonctions aux endroits appropriés dans la boucle principale.
2. *Gestion du clavier :*
   La gestion du clavier est réalisée pour vous par la fonction Clavier dont nous ignorerons le contenu pour l'instant. Cette fonction nous permet de savoir directement si une des touches qui nous intéressent (q et s pour le déplacement de la première raquette, k et l pour la deuxième) sont enfoncées ou non. Cette fonction, Clavier(int& sens1, int& sens2), retourne dans sens1 et sens2, les valeurs 0, -1 ou 1 (0 : pas de déplacement, -1 : vers la gauche, 1 : vers la droite).
3. *Déplacement des raquettes :*
   Coder le déplacement d'une raquette dans une fonction
   void BougeRaquette(int &x, int sens)
   puis appeler cette fonction dans la boucle principale pour chacune des deux raquettes. Evidemment, faire en sorte que les raquettes ne puissent sortir de la fenêtre.
4. *Rebonds sur les raquettes :*
   S'inspirer de la gestion des rebonds de la balle. Ici il faut non seulement vérifier si la balle va atteindre le bas ou le haut de l'écran mais aussi si elle est assez proche en abscisse de la raquette correspondante.
5. *Comptage et affichage du score :*
   Modifier la fonction BougeBalle afin de comptabiliser le score des deux joueurs et l'afficher dans la console.
6. *Pour ceux qui ont fini :*
   Lors d'un rebond sur la raquette, modifier l'inclinaison de la trajectoire de la balle en fonction de la vitesse de la raquette ou de l'endroit de frappe.

Vous devriez avoir obtenu un programme ressemblant à celui de la figure 1.

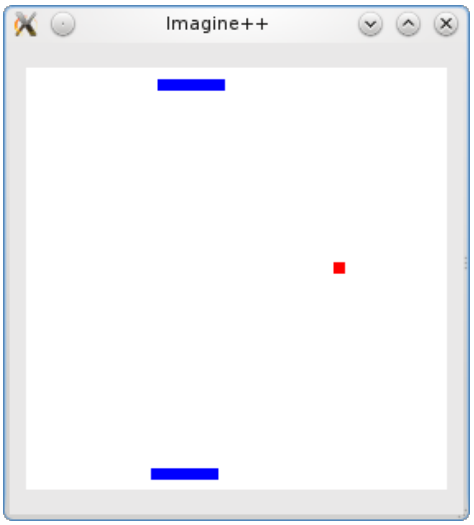

FIGURE 1 – Mini tennis...