

# Compte rendu TP2

## Dessiner un beau dessin

### Développement en C++ Semestre 2

De ce TP, nous pouvons retenir qu'un programmeur, ce doit d'être vigilant dans la rédaction de son travail. En effet il ne faut pas se limiter au fait que le programme marche, mais soigner son code, pour qu'il soit compréhensible par les autres, mais également pour nous même ultérieurement.

En effet il peut arriver qu'après coup on ne comprenne plus ce qu'on a voulu faire.

Ainsi, voici différentes conventions que nous avons essayé de corriger, cela rend le code plus propre, et plus lisible.

## 1 Variable

Les variables sont très importantes dans un programme, ainsi, il faut faire attention à leurs utilités, leurs appellations, et leurs types.

### 1.1 Noms

Les noms de variables sont extrêmement important.

Dans le code fournit par l'IUT, tous les noms de variables s'appelaient *vlx*, c'était catastrophique pour la compréhension du code :

il était impossible de comprendre à quoi servait une variable.

### 1.2 Unicité d'utilité

On ne doit pas avoir une variable qui fait plusieurs choses différentes, cela permet d'améliorer la compréhension.

### 1.3 Variables globales

Les variables globales sont à proscrire, en effet, elles rendent le code sale, et utilisent beaucoup de mémoire, étant donné qu'elles sont disponibles dans tous les sous-programmes, même si on ne s'en sert pas.

## 2 Boucles

Il est préférable d'utiliser une boucle `for` lorsque que l'on doit répéter un nombre d'actions connu à l'avance, plutôt que d'utiliser une boucle `while`. La boucle `for` est plus lisible, elle permet de déclarer une variable uniquement dans la boucle, et les informations sont condensées, donc plus claires, on saura tout de suite qu'une boucle `for` veut dire que l'on va répéter des actions un certain nombre de fois.

## 3 Sous-programmes

*« Diviser pour mieux régner »*

Il faut diviser le code avec le plus de sous-programmes possible : un sous-programme ne doit pas dépasser 40 lignes.

En effet il est plus simple de comprendre ce que fait le sous-programme, que de devoir comprendre tout un code, situé dans le programme principal.

Il faut également penser à donner un nom clair à son sous-programme, quitte à ce qu'il soit un peu long à écrire.

## 4 Indentation

L'indentation d'un code est indispensable : en effet un code bien indenté est beaucoup plus facilement compréhensible, on voit tout de suite quand commence une boucle ou une condition et quand elle finit.

Sans indentation, un code est tout simplement illisible.

## 5 Commentaires

*« Si après avoir lu uniquement les commentaires d'un programme vous n'en comprenez pas l'utilité, jetez-le tout ! » IBM*

Les commentaires d'un programmes sont eux aussi utiles, ils servent à aider les programmeurs dans le cas où il relit son code quelques temps plus tard, ou lorsqu'un autre programmeur lit le code, grâce aux commentaires, il comprendra le code.

Cependant, il n'est pas utile de commenter toutes les lignes, en effet certaines lignes sont évidentes, de plus si les fonctions ont des noms explicites, elles parleront toutes seules !