Chapitre 1 : Introduction à la programmation événementielle

1.1 Introduction

Dans les années 80, les micros ordinateurs ont obtenu les capacités nécessaires pour supporter des interfaces graphiques. La complexité des applications utilisant des interfaces graphiques est importante car il faut gérer un nombre important d'actions de bas niveau (par exemple, les fenêtres sur l'écran). Il fallait donc décharger le concepteur d'applications de toutes ces difficultés et ainsi des outils de développement pour des applications basées sur ces interfaces graphiques se sont donc imposés. La programmation événementielle et les langages de programmation associés sont donc apparus à cette époque.

1.2 La programmation événementielle

Il existe un ensemble de langages de programmation, chacun est spécialisé dans un domaine d'application donné et chacun possède un type spécifique. On distingue :

- *Programmation structuré ou modulaire* : le programme est vu comme un ensemble d'unités structurés hiérarchiquement. Il existe une interaction entre les modules et on fait généralement distinction entre les données et les traitements.

- *Programmation orientée objet* : le programme n'est une autre qu'une collection d'objet qui communique. Un objet et par définition une instance d'une classe dont les composantes peuvent être de deux types : propriétés et méthodes.

- *Programmation événementielle* : Les composants d'une application événementielle c'est à dire les objets interagissent entre eux et avec l'environnement. Ils communiquent en réponse à des événements. Ces événements peuvent correspondre à une action de l'utilisateur : un click sur un bouton de commande, une écriture dans une zone de texte, un choix dans une case d'option ou une case à cocher, le déplacement d'un objet, ... Ils peuvent aussi être déclenchés par le système : chargement d'une feuille, un top déclenché par l'horloge.

Les événements sont captés par le système d'exploitation, le traitement consiste en l'exécution des procédures événements associées à celui-ci s'il en existe. C'est le programmeur qui doit prévoir la procédure à exécuter en réponse à un événement donné. Par exemple, le déclenchement de l'événement click sur un bouton quitter doit terminer l'exécution, le choix d'un élément dans un menu doit déclencher certaines opérations, un top d'horloge doit modifier le contenu d'une zone d'image.

Un objet

Propriétés
Méthodes
Evénements

Important : Le programmeur doit toujours avoir à l'esprit le schéma suivant :

Objet : O Evénement : E Traitement : T

Si l'événement E se produit sur l'objet O alors le traitement T est exécuté.

Exemple : un bouton dans une interface graphique

- Attributs : une image, un label, une couleur de fond, une police de caractères.
- Méthodes : se dessiner,
- Evénement : réagir quand on clique dessus.

1.3 Outils de travail

Un environnement de développement développement intégrés (IDE : Integrated Develompent Environment) est un ensemble d'outils qui en plus des tâches classiques d'un langage de programmation offre des fonctionnalités étendues permettant de couvrir une plus large partie du cycle de création d'un logiciel.
On peut distinguer des tâches telles que :

- Compilation et déboguage
- Déploiement de l'application.
- Intégrer des outils de test et de vérification.
- Outil de création graphique (icône).
- Interface avec les SGBD.
- Générateur de menus.
- Un ensemble d'assistants.
-

Dans ce qui suit, des exemples d'IDE chez des éditeurs différents et basés sur des langages de programmation différents :

- Borland Delphi
- Borland Jbuilder
- Oracle JDeveloper
- Microsoft Visual Basic et C#
- Microsoft Visual C++

1.4 Présentation de Microsoft Visual C

C# est un langage orienté objets dont la syntaxe est très proche de java. Il est disponible en version béta depuis 2000 et est devenu disponible officiellement depuis 2002 avec la plate forme .Net. Il fonctionne avec l'environnement d'exécution .Net sous Windows.

1.4.1 Principe de fonctionnement

- Conception de l'interface.
- Mise en place des contrôles.
- Ecriture du code
- Test et exécution.

1.4.2 Organisation d'une application Visual C#

C# possède une grille d'écran principal (forme principale) qui va contenir des contrôles, un menu, etc.
La forme principale appelle d'autres formes (formes filles) qui sont appelées des boites de dialogue. Les boites de dialogue peuvent être :

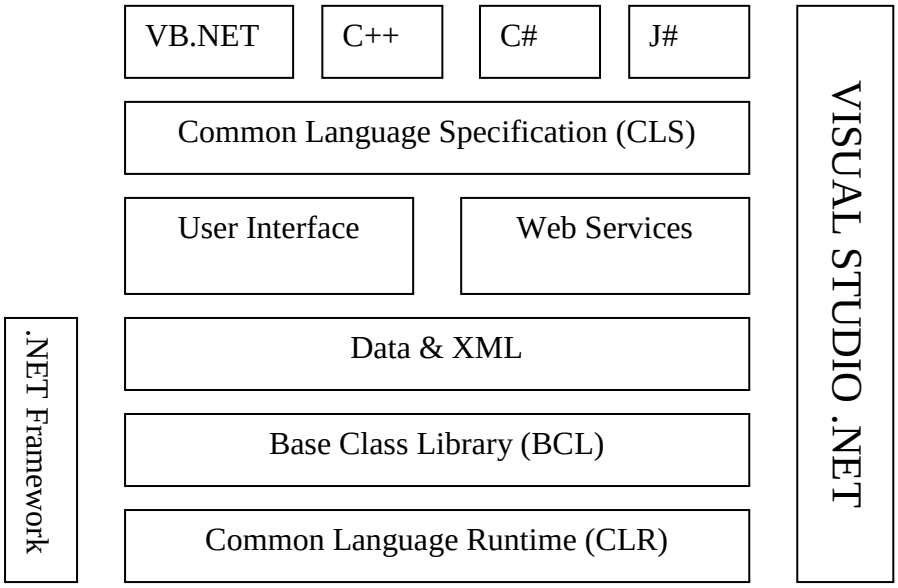
- Boite de dialogue modale : si elle est ouverte, elle recevra tous les événements et aucune autre forme ne peut prendre le focus tant que cette dernière est ouverte.

- Boite de dialogue non modale : on peut ouvrir plusieurs qui seront chargées en mémoire centrale et on peut passer d'une fenêtre à une autre.

1.4.3 Microsoft Visual Studio.NET

La plate-forme .NET est la base sur laquelle sera développée la prochaine génération de logiciels permettant de relier des systèmes, des données, des périphériques et des utilisateurs au moyen d'une méthode unifiée et personnalisée.

Visual Studio .NET est un environnement de développement qui permet le design, le développement, le débogage et le déploiement rapide de solutions basées sur .NET Framework. Il est possible d'accéder à tout un ensemble communs d'outils à partir de n'importe quel langage de programmation (VB, C#, J#, C++).



1.4.4 .NET Framework

.NET Framework est un ensemble de services de programmation conçu pour simplifier le développement des applications distribuées. Il comprend deux composants principaux qui sont les suivants : le CLR (Common Language Runtime) et la bibliothèque de classes.
Lorsque vous créez une application avec C#, vous disposer d'un code de démarrage qui inclut l'espace de noms System.Windows.Forms et la classe Form qui permet la création de fenêtres et des contrôles.

- Common Language Runtime : Quelque soit le langage de programmation utilisé, l'application ,une fois compilée, est convertie en Common Language Runtime ou code MSIL (MicroSoft Intermediate Language) que le runtime gère et exécute.

- Bibliothèque de classes : C'est une collection de classes réutilisables organisées en fonction de leurs fonctionnalités dans des espaces de noms hiérarchiques.

- Espace de noms : Collection de classes

1.4.5 Les principaux fichiers d'une application C#

(.sln) : *Visual Studio Solution* : Organise les projets, les éléments de projet et les éléments de solution dans la solution (conteneur de projets)
(.suo) : *Solution User Options* : Contient toutes les options associées à la solution
(.cs) : *Fichiers source c sharp* qui appartiennent au projet.
(.csproj) : *Projet c sharp* : regroupe plusieurs formulaires et fichiers de modules