



Le Développement Rapide d'Interfaces (RAD) sous Visual Studio 2005



RAD (Rapid Application Development) :

Le RAD consiste à développer rapidement une application en construisant l'interface avec la souris. Dans un deuxième temps, le fonctionnement de l'application est mis en place par programmation des gestionnaires d'évènements.

1- Le Développement Rapide d'Applications (RAD) pour faire des Interfaces Graphiques Utilisateur (GUI)

Le RAD consiste à développer des interfaces graphiques avec la souris et, donc, à réaliser des applications GUI :

1. Visual Studio permet, via les applications WinForms, de développer rapidement des applications avec interfaces graphiques sous Windows. C'est un outil graphique puissant pour mieux construire les interfaces Homme/Machine : les fenêtres Windows sont construites très facilement. Ces applications avec interfaces graphiques utilisateur sont appelées applications G.U.I. : (*Graphical User Interface*).
2. Les composants sont déposés sur l'interface et leurs propriétés sont configurées ; ensuite, les traitements nécessaires sont codés dans les gestionnaires d'évènements. Il est alors question de programmation événementielle.

Différents environnements de développement intégrés (EDI) permettent de faire du RAD :

1. Parmi les logiciels de développement graphique en C++ sous Windows, il existe : Visual C++ (inclus dans Visual Studio), C++ Builder...
2. Des EDI existent pour faire du RAD dans d'autres langages, notamment : Delphi, Visual Basic (inclus dans Visual Studio), Windev...

La réalisation d'interfaces est rapide car la souris permet de construire l'interface et l'EDI écrit automatiquement une partie du code :

1. L'interface graphique est construite très simplement, par simple manipulation de souris. La juxtaposition des composants permet de configurer l'aspect graphique de l'interface. Le programmeur doit :
 - ✚ placer des composants sur l'interface,
 - ✚ puis modifier leurs propriétés dans une fenêtre spéciale (la fenêtre des propriétés).
2. Une partie des instructions n'est pas à écrire car une partie du code en C++ est générée automatiquement par l'EDI.



2- Le développement Rapide d'Interfaces (RAD) utilise la Programmation événementielle

L'exécution séquentielle en C consiste à aller du début à la fin du programme :

1. L'exécutable d'un programme écrit en langage C s'exécute sous DOS, avec pour point d'entrée dans le programme : la fonction `main()`, qui correspond au début de l'exécution de l'application.
2. Principe de l'exécution séquentielle : les instructions du programme sont exécutées, par le processeur, une par une dans l'ordre logique où elles se présentent. L'exécution se fait dans un ordre préétabli par le programmeur.

L'exécution événementielle en C++ consiste à réagir à des événements lorsqu'ils se produisent :

1. L'exécutable d'un programme écrit en langage C++ s'exécute sous WINDOWS, avec pour point d'entrée dans le programme : la fonction `Winmain()`, qui correspond au début de l'exécution de l'application. L'exécution est globalement de type événementiel et localement (dans une fonction donnée) de type séquentiel.
2. Principe de l'exécution événementielle : les instructions sont exécutées dans un ordre qui dépend (notamment) des actions de l'utilisateur. Par exemple, le programme peut attendre que l'utilisateur utilise l'interface de l'application en cliquant sur un élément de menu. Quand ces actions sur l'interface (ou événements) sont détectées par le programme, des fonctions particulières s'exécutent pour répondre à l'ordre de l'utilisateur : ce sont les gestionnaires d'événements. Un événement n'est pas forcément lié à l'utilisateur (ex. : une fin de temporisation).

La démarche de RAD avec programmation événementielle se résume aux deux grandes étapes : construire l'interface avec la souris et écrire le code des gestionnaires d'événements :

1. La première étape du développement d'une application GUI revient donc à choisir les composants utiles et leurs propriétés afin de réaliser l'interface désirée.
2. La deuxième étape consiste à déterminer les événements possibles et les réactions nécessaires de l'application à ces événements. Le programmeur doit alors écrire le code des gestionnaires d'événements.