

# Aide au tutorat - Prosit 5

---

## Acquis d'apprentissage visés

[Analyse] Les événements {Définir – Discuter – Mettre en œuvre – Choisir}

[Analyse] L'assembly Winforms {Définir – Discuter – Mettre en œuvre – Choisir}

## Philosophie du prosit et fausses pistes

Ce prosit a pour but de faire travailler les étudiants sur des application fenêtrées en utilisant l'assembly Winforms. Ils auront pour objectif d'utiliser et comprendre certains éléments de contrôles communs de Winforms. Le personnage du prosit, indique en fin de texte qu'il souhaite en premier lieu étudier les éléments de code qu'il ne connaît pas. Cette idée est une fausse piste. Il est préférable de comprendre la globalité du programme dans un premier temps avant de se lancer à corps perdu dans des détails de langage.

## Descriptif des ressources

Les ressources principales identifiées doivent permettre aux étudiants d'étudier :

La programmation événementielle en règle générale.

Le paramétrage de Visual Studio pour créer une application graphique avec C++/CLI.

L'intégration de composants graphiques dans une application fenêtrée et leurs paramétrages.

## Notes au tuteur, conseils, trucs et astuces

Le tuteur qui encadre cette activité doit être informaticien développeur. Il doit posséder des connaissances en programmation orientée objet et en modélisation UML a minima.

## Aide au tutorat

### Prendre connaissance de la situation et la clarifier

Mots inconnus :

C++/CLI : C ++ / CLI (C ++ modifié pour Common Language Infrastructure) est une spécification de langage créée par Microsoft qui remplace Managed Extensions pour C ++. Il s'agit d'une révision complète qui simplifie la syntaxe Managed C ++ désormais obsolète et assure l'interopérabilité avec les langages Microsoft .Net tels que C #. C ++ / CLI a été normalisé par Ecma sous le nom ECMA-372. Il est actuellement disponible dans Visual Studio 2005, 2008, 2010, 2012, 2013, 2015, 2017 et 2019, y compris les éditions Express.

Source

Contexte :

Un photographe a développé une application pour gérer sa banque de photos. Il a transmis ce programme à son ami étudiant en informatique pour y jeter un œil et éventuellement corriger quelques bugs.

### Analyser le besoin

Problème : Le programme a été endommagé et aucune sauvegarde n'a été faite. Le programme utilise des techniques de programmation graphique que le personnage principal de l'histoire ne connaît pas.

Contrainte(s) : Les fonctionnalités du programme

Livrable(s) : Le logiciel de gestion des fichiers images.

### Généralisation

La programmation événementielle en environnement graphique.

### Recherche des pistes de solution

Étudier l'architecture globale du code source (Diagramme de classes – Diagrammes de séquences) pour comprendre comment le programme est construit et comment les éléments interagissent les uns avec les autres.

Etudier les éléments graphiques utilisés par le créateur du programme

## Élaboration du plan d'actions

Action 1 : Modéliser le logiciel (UML : Diagramme de classes – Diagrammes d'activités)

Action 2 : Identifier les morceaux de codes qui posent problème.

Action 3 : Étudier les techniques et mécanismes des morceaux de codes identifiés

Action 4 : Corriger le code source

Action 5 : Finaliser et rendre opérationnel le programme (l'ensemble de ses fonctionnalités)

## Prosit Retour

---

### Aide au tutorat

#### Définitions

Programmation événementielle : En informatique, la programmation événementielle est un paradigme de programmation fondé sur les événements. Elle s'oppose à la programmation séquentielle. Le programme sera principalement défini par ses réactions aux différents événements qui peuvent se produire, c'est-à-dire des changements d'état de variable, par exemple l'incrément d'une liste, un mouvement de souris ou de clavier. La programmation événementielle peut également être définie comme une technique d'architecture logicielle où l'application a une boucle principale divisée en deux sections : la première section détecte les événements, la seconde les gère. Elle est particulièrement mise en œuvre dans le domaine des interfaces graphiques. À noter qu'il n'est pas ici question d'interruptions logicielles : le traitement d'un événement ne peut pas être interrompu par un autre, à part en des points précis explicitement prédéterminés du code logiciel (points qui, en fait, créent une seconde boucle événementielle au

sein de la première). La programmation événementielle peut être réalisée dans n'importe quel langage de programmation, bien que la tâche soit plus aisée dans les langages de haut niveau (comme Java). Certains environnements de développement intégrés (par exemple Qt Software) permettent de générer automatiquement le code des tâches récurrentes dans la gestion des événements. Source

Winforms : Windows Forms est le nom de l'interface graphique qui est incluse dans le framework .NET, fournissant l'accès via du Managed code à l'API Windows. Depuis la sortie de .NET 3.0 en 2006, Windows Forms est destiné à être remplacé par un autre système d'interface graphique, Windows Presentation Foundation. Source

## Questions de validation

Les évènements [Définir] : Pouvez-vous définir la notion d'évènement en programmation ?

Les évènements [Discuter] : En quoi les évènements sont utiles en lieu et place d'un appel de fonction ou d'un wrapper de fonctions ?

Les évènements [Mettre en oeuvre] : Pouvez-vous faire un programme au vidéo projecteur qui mette en évidence l'utilisation d'un évènement dans une interface graphique ?

Les évènements [Choisir] : Comment identifier, choisir et utiliser un évènement sur un contrôle graphique ?

Winforms : [Définir] : Pouvez-vous définir l'assembly winforms ?

Winforms: [Discuter] : A quoi servent les arguments dans une méthode qui sera appelées par un évènement ?

Winforms: [Mettre en oeuvre] : Pouvez-vous faire un programme au vidéo projecteur qui utilise deux formulaires graphiques et qui passe des paramètre d'une fenêtre à une autre

Winforms [Choisir] : Quelle méthode principale allez-vous choisir dans votre programme de type graphique ?