



Exploration

204 - exploration C# programmation GUI

Rédigé par

David ROUMANET
Professeur BTS SIO

Changement

Date	Révision

Sommaire

A Introduction.....	3
B Objectifs.....	3
C Création d'un projet.....	3
C.1 Environnement Visual Studio pour windows Form.....	4
C.2 Fonctionnement de l'environnement.....	5
C.3 Code sur un événement Clic.....	6
C.4 Le programme maxi.....	7
D Annexes.....	9
D.1 Erreurs potentielles.....	9
D.2 Autres.....	9

Nomenclature :

- **Assimiler** : cours pur. Explication théorique et détaillée (globalement supérieur à 4 pages).
- **Décoder** : fiche de cours, généralement inférieure à 5 pages.
- **Découvrir** : Travaux dirigés. Faisable sans matériel. Faible autonomie (fort guidage)
- **Explorer** : Travaux pratiques. Nécessite du matériel ou des logiciels.
- **Mission** : Projet encadré ou partie d'un projet.
- **Voyager** : Projet en autonomie totale. Environnement ouvert : Vous êtes le capitaine !

A Introduction

La programmation en mode console, bien qu'ayant encore quelques applications pratiques, ne permet pas de créer des interfaces agréables et intuitives pour les utilisateurs. Les interfaces graphiques ont depuis longtemps supplanté le mode texte.

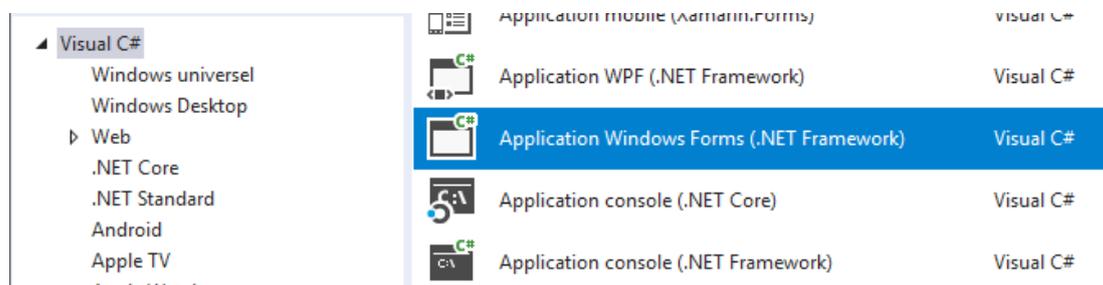
Cette manière de programmer s'appuie sur les mêmes bases : des fonctions (ou méthodes) et des variables, des conditions et des itérations (boucles). Cependant, dans une interface graphique, on utilise la gestion des événements des systèmes d'exploitations, ce qui va générer un fonctionnement moins linéaire...

B Objectifs

Ce TP permet la mise en œuvre des interfaces graphiques en C# et la compréhension du concept de programmation événementielle.

C Création d'un projet

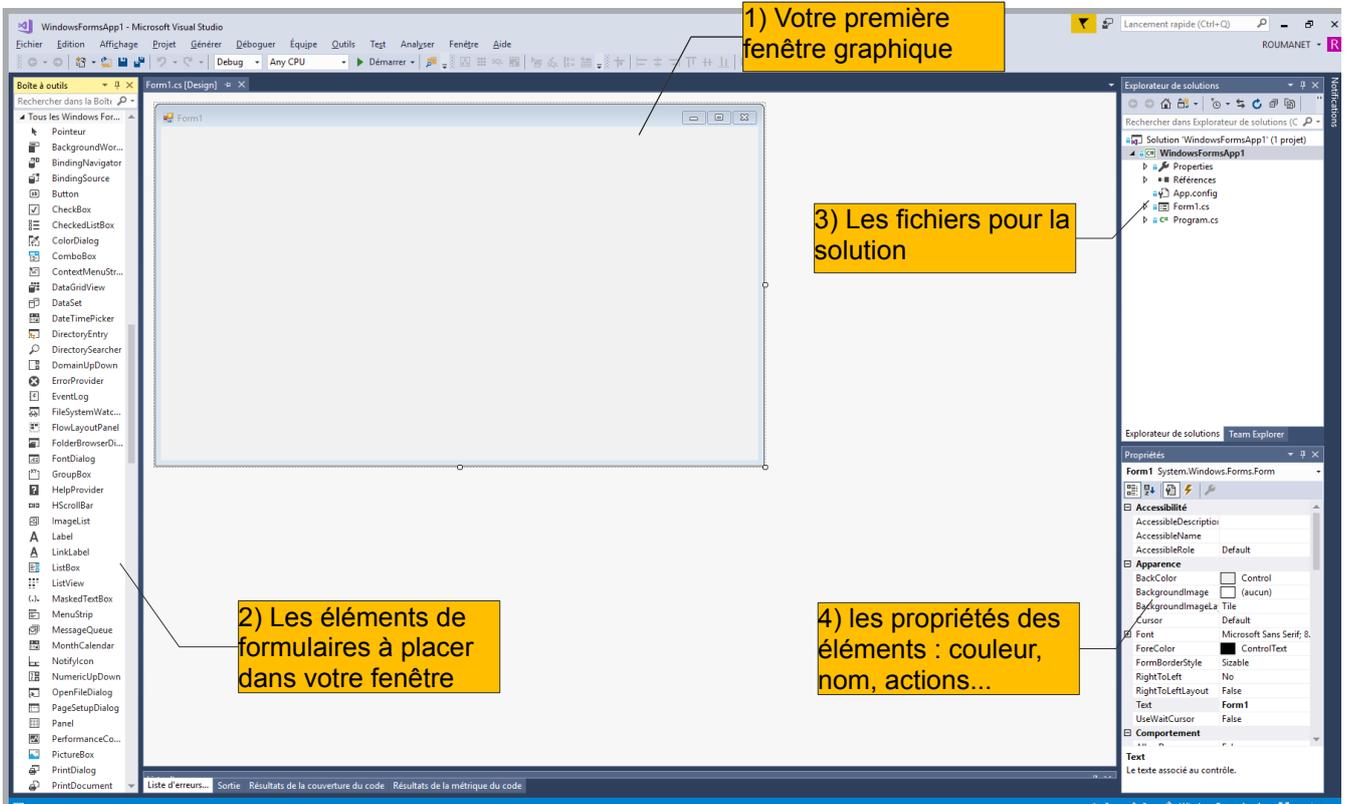
Dans Visual Studio, créer un nouveau projet en utilisant "Application Windows Forms (.NET Framework) :



La solution est créée par Visual Studio : il devient important de comprendre à quoi servent les nombreuses fenêtres présentes.

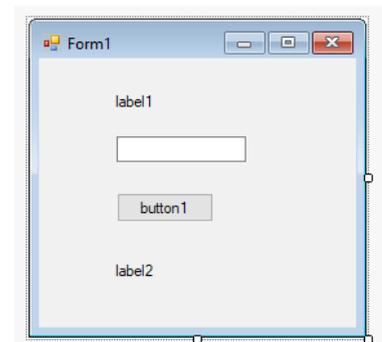
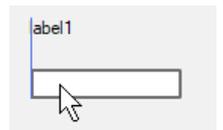
C.1 Environnement Visual Studio pour windows Form

Voici une capture d'écran après la création du projet : il n'y a pas de ligne de code.



Il faut sélectionner les objets de la boîte à outils et les déplacer sur la fenêtre au centre de l'écran.

- Glissez/Déposez un label en haut de la fenêtre
- Glissez/Déposez un TextBox juste en dessous (une ligne bleue s'affiche lorsque les éléments sont alignés).
- Glissez/Déposez un Button en dessous du Textbox
- Glissez/Déposez un label en dessous du bouton
- Réduisez la fenêtre de la forme graphique, Vous devriez avoir ceci :



C.2 Fonctionnement de l'environnement

Vous venez de créer votre première graphique. Vous pouvez exécuter le programme, vous obtiendrez une fenêtre mais pour le moment, il ne se passe rien quand on appuie sur le bouton ou qu'on saisit du texte.

Il s'agit d'une application utilisant des événements mais pour le moment, notre code ne contient aucun événement. Voici une liste de quelques événements que nous pourrions utiliser :

- Double-clic
- Simple-clic
- Changement d'état (saisie de caractères)
- Redimensionnement
- Appuie de touches (Keypress) : en appuie ou en relâchement
- etc.

Pour Microsoft Windows, notre fenêtre n'est qu'une fenêtre parmi d'autres, cependant, lorsque la souris la survole ou qu'une action y est faite, Windows nous envoie le type d'événement et sur quel élément c'est arrivé.

Pour le moment, dans votre programme, chaque élément a reçu un nom par défaut, vous pouvez vérifier ses propriétés dans la fenêtre de VS2017 en bas à droite :

- Cliquez sur le bouton (ne double-cliquez pas!)
- Modifiez le texte du bouton
- Ne changez pas le nom du bouton

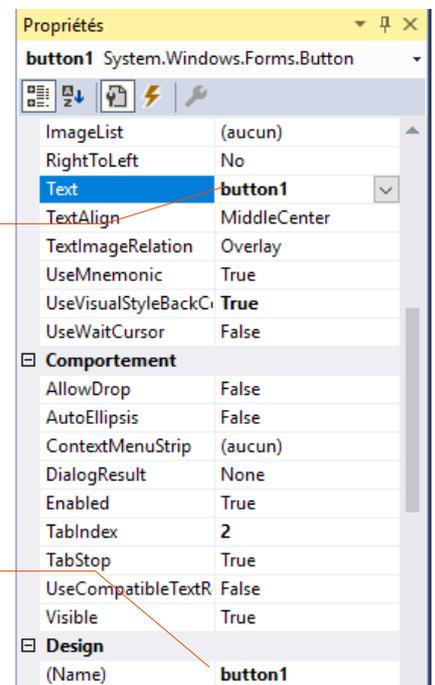
Le texte est ce qui s'affiche dans le bouton.

Le nom du bouton est son étiquette (comme pour une variable).

J'ai choisi de mettre "Cliquez-moi" dans le texte du bouton, ainsi la propriété 'Text' du bouton 'button1' contiendra "Cliquez-moi".

En langage objet, cela s'écrit :

```
button1.Text = "Cliquez-moi" ;
```



Nous allons maintenant programmer une action sur ce bouton.

C.3 Code sur un événement Clic

Double-cliquez sur le bouton : la fenêtre principale affiche maintenant le code suivant :

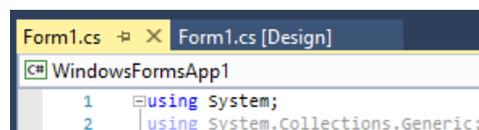
```
11 namespace WindowsFormsApp1 {
12     public partial class Form1 : Form {
13         public Form1() {
14             InitializeComponent();
15         }
16
17         private void button1_Click(object sender, EventArgs e) {
18
19         }
20     }
21 }
```

Vous pouvez lire qu'il s'agit d'une fonction (une méthode !) qui s'appelle `button1_Click` et qu'elle reçoit deux arguments : `sender` (de type objet) et `e` (de type EventArgs).

Peu importe pour le moment, vous allez seulement rajouter l'instruction suivante à l'intérieur de cette méthode, puis compiler votre programme :

```
MessageBox.Show("Vous avez cliqué le bouton " + button1.Text);
```

Pour revenir à la fenêtre et modifier votre fenêtre, il suffit d'afficher l'onglet correspondant : `Form1.cs [Design]`



Nous verrons qu'il y a également un code caché derrière cette forme (en appuyant sur la touche [F7]) mais pour le moment, il ne faut absolument pas modifier ce code, sous peine de planter votre code.

Vous allez maintenant modifier le deuxième label qui prendra la valeur du champ TextBox : comment ?

Sous la ligne `MessageBox.Show...` écrivez ceci :

```
label2.Text = "Vous avez écrit " + textBox1.Text;
```

Relancez votre programme, tapez un mot dans votre champ Textbox puis cliquez sur le bouton.

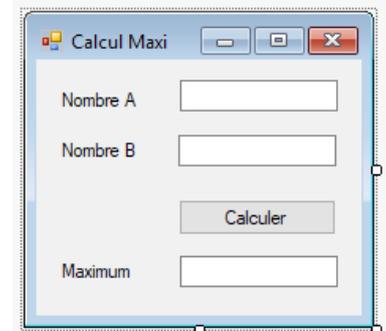
Notez que le message s'affiche mais que le label n'est pas modifié, puis lorsque vous validez le message, le label affiche "Vous avez écrit " suivi de votre mot.

La boîte de message **MessageBox** est dite *modale* (c'est-à-dire que le programme arrête de s'exécuter).

C.4 Le programme maxi

À partir de maintenant, vous pouvez transformer le programme qui saisissait 2 valeurs et donnait le maximum.

Créez un nouveau projet "Windows Form (.Net)"

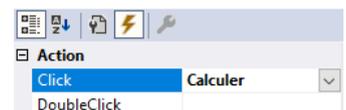


Dessinez la fenêtre suivante en donnant les noms suivants :

- TextBox1 devient txtNombreA
- TextBox2 devient txtNombreB
- TextBox3 reste TextBox3
- button1 devient btCalculer (Text devient 'Calculer')

Pour créer une action, deux méthodes (résultat presque identique)

- la méthode précédente que nous n'allons pas choisir : Double-cliquez sur le bouton Calculer
- la nouvelle méthode : cliquez sur l'éclair jaune dans les propriétés et dans la propriété Clic, tapez "Calculer" puis [Entrée]



Puis éditez le code de Form1.cs comme suit :

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Max {
12     public partial class CalculMax : Form {
13         int nbA, nbB;
14
15         public int leMax(int leNombre1, int leNombre2) {
16             if (leNombre1 > leNombre2) return leNombre1;
17             return leNombre2;
18         }
19         public CalculMax() {
20             InitializeComponent();
21         }
22
23         private void Calculer(object sender, EventArgs e) {
24             nbA = Convert.ToInt32(txtNombreA.Text);
25             nbB = Convert.ToInt32(txtNombreB.Text);
26             textBox3.Text = Convert.ToString(leMax(nbA, nbB));
27         }
28     }
29 }
```

Variables dans la classe

Notre fonction Max

L'équivalent de la fonction Main()

Compilez votre programme et ne saisissez que des nombres.

D Annexes

D.1 Erreurs potentielles

D.2 Autres