

Tutorial: Memento d'utilisation de Visual

Ce memento fournit un guide de survie pour l'utilisation restreinte de Visual Studio faite dans le cours d'environnement et programmation en BAT4.

1. Organisation de Visual Studio

Principe général de développement sous Visual Studio

Quand vous ouvrez Visual Studio, vous travaillez sur une **unique** solution. De façon informelle, une solution répond à un besoin client, et fournit différentes fonctionnalités. De façon concrète, une solution (décrite dans un fichier .sln) contient un ou plusieurs *projets*. Par exemple, une solution peut contenir un projet de type console, un projet de type graphique, une dll et un projet de type web. Un *projet* contient tout ce qui est nécessaire au fonctionnement du programme correspondant, comme par exemple des fichiers d'images ou de ressources.

Création d'un nouveau projet

Menu fichier \rightarrow nouveau projet.

Nous utiliserons les 2 types de projets suivants :

 Modèle(s) Visual C# Bureau classique Windows 		Application Windows Forms (.NET Framework)	Visual C#	Proj lign
Web NET Core	C :\	Application console (.NET Framework)	Visual C#	
NET COLUMN	⊑ Ç#			

- Application console : application qui lit/écrit les informations du programmes sur la console (i.e. Fenêtre d'invite de commandes)
- Application Windows Forms : application graphique, les informations sont données dans des composants graphiques (cases à cocher, menus, ...)

Attention à bien choisir le nom de votre application et l'endroit où le projet est stocké sur la machine, créez-vous un répertoire spécifique pour les fichiers du cours de programmation.

	▶ En ligne			
parti	Nom :	HelloWorld		
git	Emplacement :	C:\Users\helen\Documents\BAT4\prog\] [Parcourir
el Gi	Solution :	Créer une nouvelle solution	2	
th a git	Nom de solution :	HelloWorld		Créer un réper
BAL				Créer un référe



Tutorial: Memento d'utilisation de Visual

Importer une solution

Nous vous fournirons des corrigés qui correspondent à des .zip du répertoire associé à la solution. Il faudra **décompresser** l'archive dans votre répertoire de travail pour BAT4. Pour ouvrir la solution, cliquer sur le fichier .sln, cela ouvrira une nouvelle fenêtre Visual Studio. Si vous voulez garder une seule fenêtre Visual Studio, vous pouvez charger la solution en faisant : Fichier \rightarrow nouveau \rightarrow projet à partir de code existant.

2. Edition et exécution d'un programme en mode console

Premier programme de type console

```
// indispensable pour utiliser Console
using System;
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            // affiche "bonjour à tous" dans la fenêtre d'invite de commandes
            Console.WriteLine("bonjour à tous");
            // attend qu'on tape quelque chose au clavier
            // sert à laisser visible la fenêtre où le message s'est affiché.
            Console.ReadLine();
        }
   }
}
```

Ce programme affiche « Bonjour à tous » dans une fenêtre d'invite de commandes windows et attend qu'on tape une touche clavier. Quand le programme est terminé (i.e. on a tapé sur une touche) la fenêtre d'invite de commande se ferme.

Exécution

Pour exécuter, en haut, flèche verte « démarrer ». Cela exécute le main (point d'entrée du programme) du programme HelloWorld.





Tutorial: Memento d'utilisation de Visual

Attention, si votre solution contient plusieurs projets, il y a un seul projet de démarrage. On peut le choisir en faisant : projet \rightarrow définir comme projet de démarrage.

Pour les avancés, si vous voulez passer un paramètre au programme principal, cliquer sur properties \rightarrow debogage et mettez le paramètre dans « arguments de ligne de commandes ».

 Chilling				
	HelloWorld 🗢 🗙 Program.cs	Affichage de classes	-	Explorateur de solutions
gle	Application	Configuration : (Debug) active		© © ☆ # - `o - ≒ ¢ ₽
n 🗐	Build			Rechercher dans Explorateur de solution
Fichi	Événements de build	Platforme : (Any CPU) active		Solution 'HelloWorld' (1 projet)
	Déboguer			A Properties
	Ressources	Démarrer le projet		C* AssemblyInfo.cs
9.	Services	Démarrer le programme externe :		♦ ■ Références
╘┚亅	Paramètres	Démarrer le navigateur avec l'URL :		C# Program.cs
20	Chemins des références			
1	Signature	Options de démarrage		
~	Sécurité	Arguments de ligne de commande : ic	al l	
ċ.				

Compilation incrémentale et mode de saisie par suggestion

Quand vous modifiez un programme dans l'éditeur, VisualStudio effectue une **compilation incrémentale** et signale les erreurs en les soulignant avec des ~~~. Si vous passez la souris sur la zone il suggère des corrections.

<pre>static void Main(string[] args) </pre>		
<pre>// affiche "bonjour à tous" dans la fer Console.WriteLine("bonjour à tous"), // attend qu'on tape quelque chose a</pre>	nêtre d'in	ivite
<pre>// sert à laisser visible la fenêtre ' Console.ReadLine(); }</pre>	attendue	ige s

Vous pouvez opter pour la saisie par suggestion, en sélectionnant le bouton ci-dessous.

	Lancement rapide (Ctri+	
Analyser Fenêtre	Aide	Helene Collavizza 👻 HC
arrer 🔸 🎜 📮 🖔 🛅 🚛	토 2월 📕 % 체 개 개 및	☆ 自 ♥
<u> </u>	Bascule entre le mode de saisie normal et le m	node de saisie par suggestion. (Ctrl+Alt+Espace)
] args) 🔹		
÷	Git failed with a fatal error.	
j args) 👻	Git failed with a fatal error.	

Dans ce cas, quand vous commencez à taper une instruction C# l'éditeur vous proposera des suggestions en adéquation avec le début du mot tapé.



Tutorial: Memento d'utilisation de Visual



Exécution pas à pas (débogage)

Il est souvent difficile de diagnostiquer la raison d'une erreur dans un programme. Il faut pour cela mener une enquête ... Le mode « débogage pas à pas » vous permet de tracer l'exécution de votre programme pas à pas en plaçant des points d'arrêt, ce qui permet de voir en détail le comportement de chaque instruction.

Voici un exemple de programme qui contient un bug. Il calcule (presque) une solution d'une équation du 2d degré mais plante si le discriminant est négatif.

```
using System;
namespace ProgrammeAvecBug
{
    // un exemple de programme avec bug pour montrer l'intérêt
    // d'utiliser le debogueur
   class Program
    {
        static void Main(string[] args)
        {
            // solution d'une équation du 2d degré : a x^2 + b x + c = 0
            // les coefficients
            double a = 1;
            double b = -2; // plante si b=1
            double c = 1;
            // calcul de discriminant
            double delta = b * b - 4 * a* c;
            double racineDelta = Math.Sqrt(delta);
            // calcul et affichage solution
```

Auteur : H. Collavizza



```
double sol = (-D - racineDelta) / (2 * a)
Console.WriteLine("la solution est");
Console.WriteLine(sol);
Console.ReadLine();
}
```

Cas d'exécution sans problème :



Cas avec erreur

}

}

la I No	m (Course d'informa	Fichier Edition Affichage	Projet Generer Deboguer Equipe Data Lake Outilis Test Analyser
	C:\Users\helen\Doc	🗖 🗖 🗶 🔤 Program	meAvecBug.exe 🔹 🐑 Événements du cycle de vie 👻 Thread :
Ē	la solution est Non Numérique		
1	=	Bug	👻 🍫 ProgrammeAvecBug.Program
		{ Oré sta {	férences 0 modification 0 auteur, 0 modification tic void Main(string[] args)
B		-	// solution d'une équation du 2d degré : a $x^2 + b x + c = 0$
	<		<pre>// les coefficients</pre>
			double a = 1;
		18	double b = 1;
		19	double c = 1;
		20	// calcul de discriminant



2017-2018

Tutorial: Memento d'utilisation de Visual

Pour comprendre ce qu'il se passe il faut lancer l'exécution pas à pas pour trouver quelle instruction pose problème. On va donc placer des points d'arrêt dans la zone qui nous semble contenir le problème (calcul du discriminant dans cet exemple). Pour cela, il suffit de cliquer sur la colonne grise en face de la ligne sur laquelle on veut que l'exécution s'arrête.

/ori	Affichage de classes	Program.cs 🛥 🗙
ure	C# ProgrammeAvecBug	🚽 🔩 ProgrammeAvecBug.Program 🚽 🗣 Main(string[] args)
mp	13	double a = 1;
414	14	double b = 1;
eie	15	double c = 1;
	16	// calcul de discriminant
lio	17	double delta = b * b - 4 * a* c;
	10	<pre>double racineDelta = Math.Sqrt(delta);</pre>
pod		<pre>// calcul et affichage solution</pre>
ma	20	double sol = (-b - racineDelta) / (2 * a);
/lus	Emplacement : Program.cs,	ligne 20 caractère 13 ('ProgrammeAvecBug.Program.Main(string[] args)')
lidé	22	Console.WriteLine(sol);



Tutorial: Memento d'utilisation de Visual

On lance maintenant l'exécution qui va s'arrêter sur chaque point d'arrêt en affichant le contenu des variables. Quand on calcule la valeur de sol on voit que la valeur de racineDelta est NaN : Not a Number, c'est donc à cet endroit qu'il y a une erreur.

	ProgrammeAvecBug	(Débogage) - Microsoft Visual Studi	0					
	Fichier Edition Afficha	age Projet Générer Débogu	er Équipe Data Lake Outils	Test				
	🖁 G • O 🖄 • 🚔 🗳	💾 🤊 - 🗘 - Debug -	Any CPU 🕞 🕨 Continuer 🗸	5				
	Processus : [3960] Progra	ammeAvecBug.exe 👻 💽 Événe	ments du cycle de vie + Thread : [6728	B] Th				
	Program.cs 🕫 🗙							
	C# ProgrammeAvecBug	•	🐾 ProgrammeAvecBug.Program					
	13	double a = 1;						
	14	double b = 1;						
	15	double c = 1;						
	16	<pre>// calcul de discrimina</pre>	nt					
	• 17	double delta = b * b - 4	4 * a* c;					
	18	double racineDelta = Ma	th.Sqrt(delta);					
	19	// calcul et affichage	solution	_				
	20	double sol = (-b - raci	neDelta) / (2 * a); <1 ms écoulée	5				
	21	Console.WriteLine("Ia so	olution est");					
	22	Console.WriteLine(sol);						
	23	Console.ReadLine();						
	24	ſ						
	25 5							
	27							
ea	27							
н	100 % -							
I	Automatique		▼ Ț X Pile de	es app				
	Nom	Valeur	Type No	m				
	🤪 a	1	double 🔵 Pro	gram				
	🖌 b	1	double					
.01	🧉 delta	-3	double					
	racineDelta	NaN	double					
	🧉 sol	0	double					

3. Création d'un programme en mode graphique (Application Windows Form)

Pour les programmes en mode graphique, vous allez devoir naviguer entre plusieurs fenêtres et menus. Si vous ne voyez pas les fenêtres, aller dans affichage pour ajouter la fenêtre correspondante. Les fenêtres que vous allez manipuler sont les suivantes :

 la fenêtre de concepteur de vue : vous permet d'ajouter des composants graphiques et de les disposer dans la fenêtre.



Tutorial: Memento d'utilisation de Visual



La boîte à outils qui vous permet de sélectionner un composant graphique (exemple : un bouton) et de placer ce composant dans la fenêtre.

۲	Form1.c	s 7 Form1.cs [Design]	Boîte à outils	+ ×	
ı	Recherc	her dans la Boîte à outils			
ı	▷ Tous	les Windows Forms			
1	▲ Cont	rôles communs			
ı		Pointeur			
8	ab	Button			
d	v	CheckBox			
8	BE	CheckedListBox			
ic	E	ComboBox			
ı	*	DateTimePicker			
	Α	Label			
e	A	LinkLabel			
ı	E#	ListBox			
J		ListView			
1	(.).	MaskedTextBox			
ı	*	MonthCalendar			
1		NotifyIcon			
ı	1	NumericUpDown			
8		PictureBox			
it		ProgressBar			
	o	RadioButton			
		RichTextBox			

- La fenêtre de **propriété du composant graphique**. Vous permet de changer les propriétés telle la couleur, le texte, la taille, ... en cliquant sur la clé à molette



Tutorial: Memento d'utilisation de Visual

-	Propriétés	
	button1 System.Windows.For	ms.Button
	🏥 💱 🖗 🗲 🎾	
I	∃ Accessibilité	
	AccessibleDescription	
	AccessibleName	
	AccessibleRole	Default
l		
	BackColor	Control
	BackgroundImage	(aucun)
	BackgroundImageLayout	Tile
	Cursor	Default
l	E FlatAppearance	
	FlatStyle	Standard
1	∃ Font	Microsoft Sans Seri
	ForeColor	ControlText
	Image	(aucun)
	ImageAlign	MiddleCenter
	ImageIndex	(aucun)
	ImageKey	(aucun)
	ImageList	(aucun)
	RightToLeft	No
	Text	button1

 En sélectionnant l'éclair dans la fenêtre de propriétés du composant, vous pouvez associer des actions aux événements. Par exemple, en double cliquant sur Action → click cela créé une fonction qui sera appelée à chaque clic sur le bouton.

3		• • •	
Règle de d	cible riétés		p
	button1 System.Windows.Forms	Button	
	🔡 👷 yl 🗲 🎾		
	Action		
	Click	button1_Click	
	MouseCaptureChanged		
	MouseClick		

 La fenêtre de code où vous pourrez associer une action à un événement sur un composant graphique (exemple, une action à effectuer quand on clique sur un bouton). Par exemple, on affiche « coucou » quand on clique sur le bouton.



Tutorial: Memento d'utilisation de Visual

atiqu	Form1.cs	Form1.cs [Design]		
	🖙 Graphiq	ue simple 🔹	🕈 🏘 Graphique_simple.Form1	 [©]_a button1_Click(object
•	10		on strong s	
	11	📃 🔤 namespace Graphiq	ue_simple	
	12	{		
		3 références 0 mo	dification 0 auteur, 0 modification	
	13	🛛 📄 👘 public partia	l class Form1 : Form	
CAC	14	{		
SAC		1 référence	0 modification 0 auteur, 0 modification	
	15	😑 🛛 public Fo	rm1()	
ptic	16	{		
/is	17	Initi	alizeComponent();	
	18	}		
ová	19			
oye		1 référence	0 modification 0 auteur, 0 modification	
)	20	private v	oid button1_Click(object sen	der, EventArgs e)
	21			
(2)	22	Conso	<pre>le.WriteLine("coucou");</pre>	
(~/	23	1	· · · ·	
	24			
	25	•		
	25	L J		
-	20			

Nota 1 : Form1.cs est le programme C#, Form1.cs[Design] est le concepteur de vue de ce programme, il vous permet de modifier le programme en déplaçant et en ajoutant des composants.

Nota	2 : le code	e C# complet peut être édité en allant sélecti	ionner les méthodes « cachées » de la classe		
.//11	i 😋 🕶 🕤	🎦 - 😩 💾 🧬 🤊 - 🤍 - 🛛 Debug 🕞 Any CPU	🔹 🕨 Démarrer 👻 🎜 📮		
atiqu	Form1.cs # × Form1.cs [Design]				
	C# Graphiqu	e simple 🔹 🔹 🔩 Graphique_simple.Form1			
-	10	Tarrie alacturing and a second	Sutton1		
	11	<pre>Part = Part = Part</pre>	♀ _a button1_Click(object sender, EventArgs e)		
	12	3 références 0 modification 0 auteur, 0 modification	● components		
	13	public partial class Form1 : Form	♥ ₊ Dispose(bool disposing)		
SAC	14	{ 1 référence 0 modification 0 auteur, 0 modification	© Form1()		
	15	<pre>public Form1()</pre>	P _a InitializeComponent()		
ptic	16	{	3		
/is	17	<pre>InitializeComponent();</pre>			

Cela ouvre une troisième fenêtre Form1.Designer.cs qui contient le code C# complet qui a été généré. Dans certains cas, vous aurez besoin (ou envie) de modifier directement ce code.



Tutorial: Memento d'utilisation de Visual

	😋 🕶 🔊 📸 🖬 💾 🗳 🎔 – 🦿 – 🛛 Debug 🕒 Any CPU 🛛 – 🕨 Démarrer – 🎜 🚽			
tiqu	Form1 cs # Form	1 Designer csR. X. Form1 cs.[Design] -R. X.	•	Drop
			_	FIOP
	Graphique simple Graphique_simpl C:\Users\helen\Documents\BAT4\prog\visualStudio\Grap			
	20	/// le contenu de catte méthode sues l'éditeur de code	÷	
	27	/// le contenu de cette methode avec l'editeur de code.		
	20 -	///		
	29	ncivate void InitializeComponent()		
	30			
	31	l this hutton1 - new System Windows Forms Button():		
SAC	32	this Suspend avout():		
	22	//		
otic	24	//		
s	24		_	
	20	//	_	
	50	this.buttoni.Location = new System.Drawing.Point(62, 39);		
ye	3/	this.buttoni.Name = buttoni;		
	38	this.button1.Size = new System.Drawing.Size(75, 23);		
21	39	this.button1.TabIndex = 0;		
	40	<pre>this.button1.Text = "coucou";</pre>		
~/	41	<pre>this.button1.UseVisualStyleBackColor = true;</pre>		
	42	<pre>this.button1.Click += new System.EventHandler(this.button1_Click)</pre>		
	43			
	44	// Form1		
•	45	//		