S4 – Développement logiciel

Premiers pas avec C# et Visual Studio

Gestion entrées/sorties

Objectifs : être capable de réaliser une application simple sous Windows en utilisant C# et Visual Studio en suivant un tutoriel, être capable de modifier cette application.

Pré-requis : bases de l'algorithmique, programmation procédurale et bases de programmation orientée objet.

Introduction

L'objectif de ce TP n'est pas de faire de vous des professionnels de développement Windows mais simplement d'être capable de faire une application Windows simple permettant, par exemple, de réaliser une interface homme- machine (IHM) pour piloter un système.

Ce premier TP/tutoriel vous permettra de prendre en main l'outil et d'avoir ainsi un aperçu de ses possibilités.

Si dans le cadre d'un projet particulier vous seriez amené à développer des applications plus conséquentes, il existe sur Internet toute une littérature relative à cet outil.

Voici quelques liens intéressants :

Le site d'un enseignant, illustré de beaucoup d'exemples,

http://pise.info/csharp/00.htm

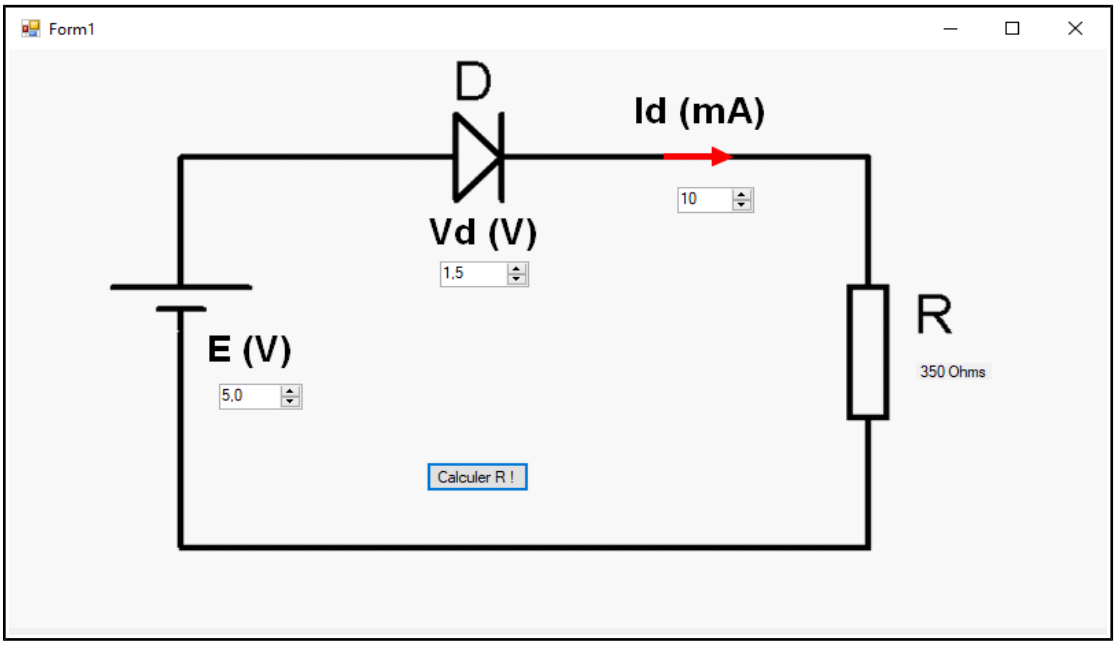
Bien entendu, le site de microsoft :

https://msdn.microsoft.com/fr-fr/library/jj153219.aspx

Les tutoriels sont écrits pour Visual Studio 2015, mais vous devrez vous y retrouver !

Présentation de l'application

L'application que vous allez réaliser doit permettre de calculer la résistance à placer dans un circuit alimentation – diode – résistance à partir de la valeur de la tension d'alimentation, de la tension de seuil de la diode et de la valeur du courant. L'application se présente sous la forme suivante :



Questions préliminaires

Q.1) Rappeler la relation permettant de calculer R en ohms en fonction des paramètres E (V), Vd(V) , Id(mA).

Q.2) Indiquer les conditions de validité de cette expression.

Q.3) Vérifier le calcul effectué par l'application et représenté sur la figure.

Q.4) Calculer la valeur de le résistance à placer pour avoir un courant de 10 mA avec une diode ayant un seuil de 1,5V avec une alimentation de 3,3V.

Travaux pratiques

Q.1) Une version de cette application vous est fournie. Essayez-là. Effectuer la simulation correspondant au cas décrit dans l'activité précédente. Relever les plages de variation des différents paramètres et leur quantum. Que se passe-t-il lorsque les valeurs rentrées ne sont pas cohérentes ?

Démarrer avec Visual Studio

Visual Studio Community, environnement de développement intégré (IDE) est gratuit et totalement fonctionnel pour les étudiants, les développeurs open source et les développeurs particuliers.

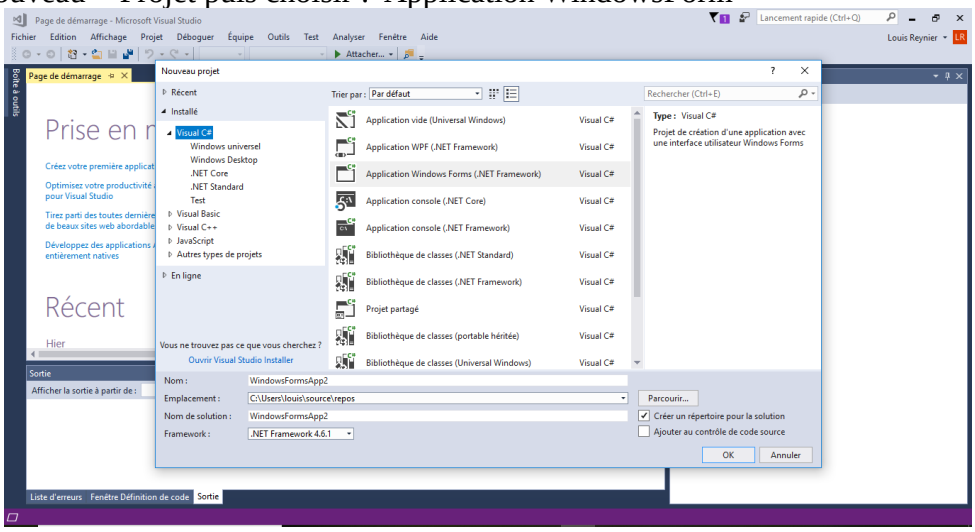
https://www.visualstudio.com/fr/downloads/

Vous pouvez donc le télécharger, l'installer et l'utiliser chez vous. Vous devez disposer d'une bonne connexion Internet, d'assez d'espace sur votre disque dur et d'une bonne dose de patience, l'installation est assez longue !

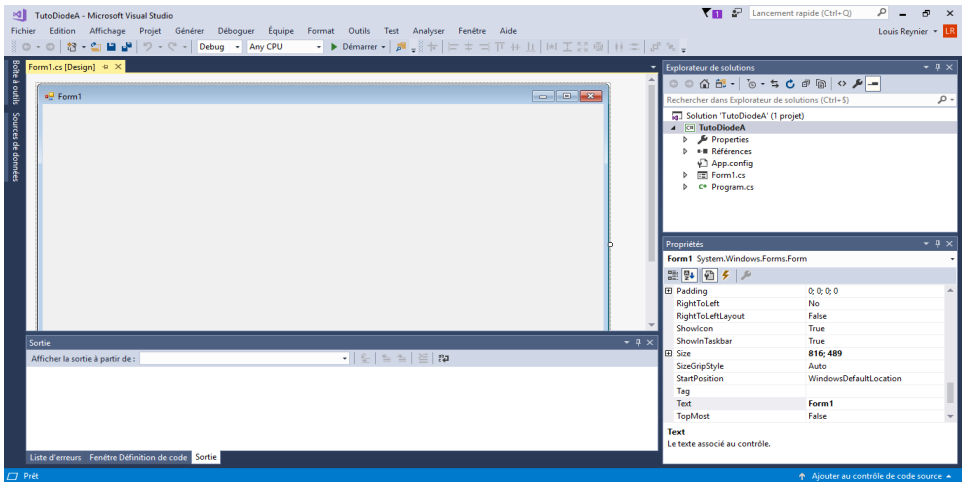
Visual Studio 2017 est normalement installé sur les postes étudiants.

Travaux pratiques

Lancer Visual Studio 2017
Fichier > Nouveau > Projet puis choisir : Application WindowsForm

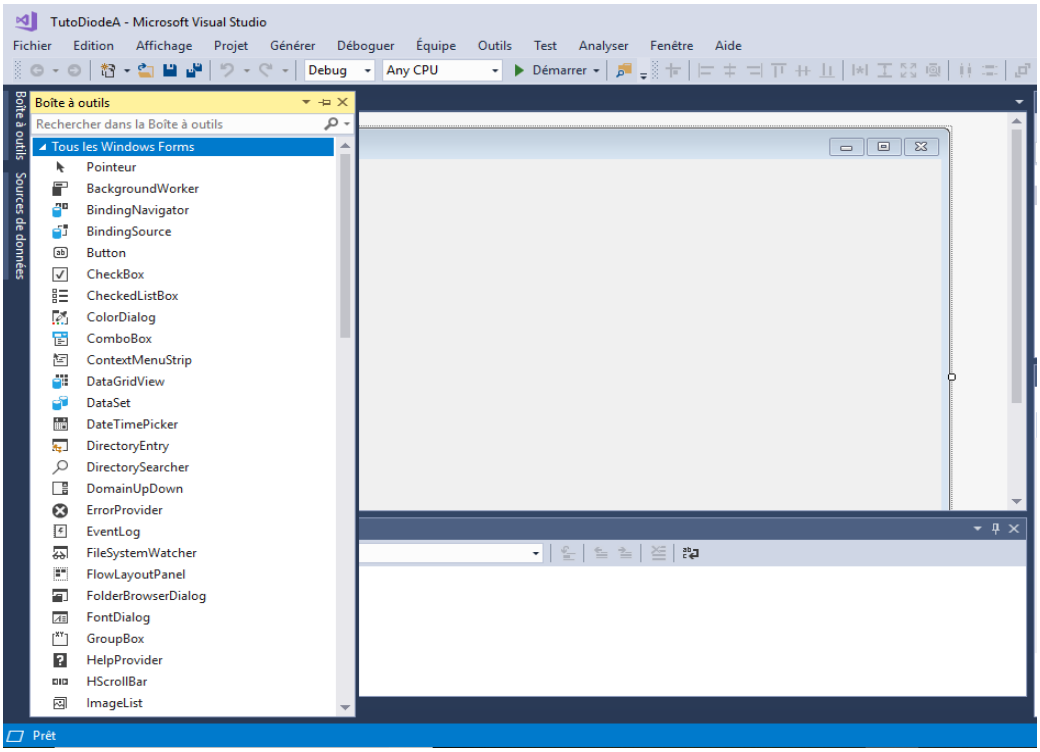


Choisir le répertoire de travail et le nom du projet. Il apparaît alors la fenêtre Form1 : c'est la fenêtre de notre application qui constitue l'interface utilisateur. Cet affichage est appelé Concepteur.

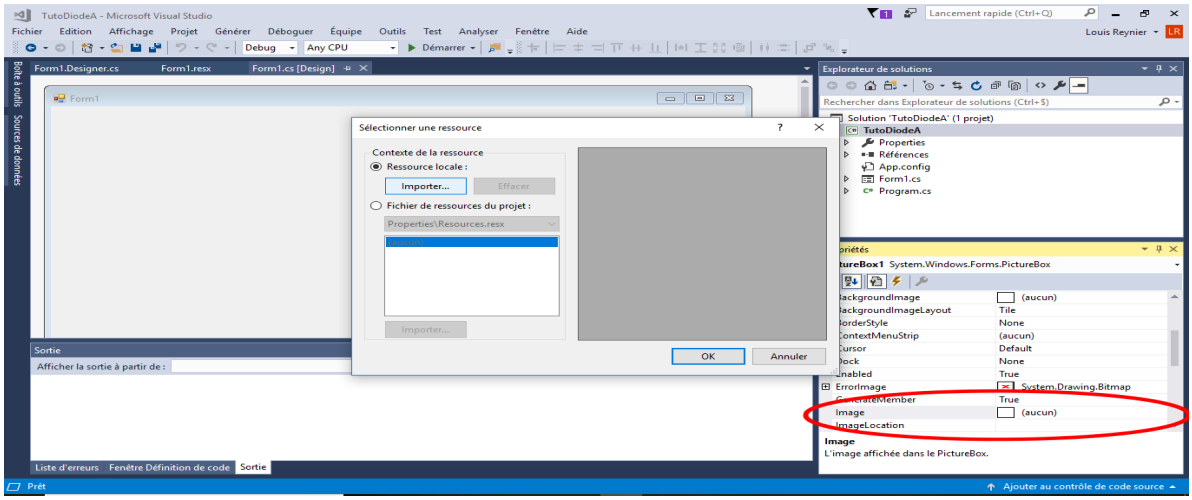


Créer l'interface utilisateur

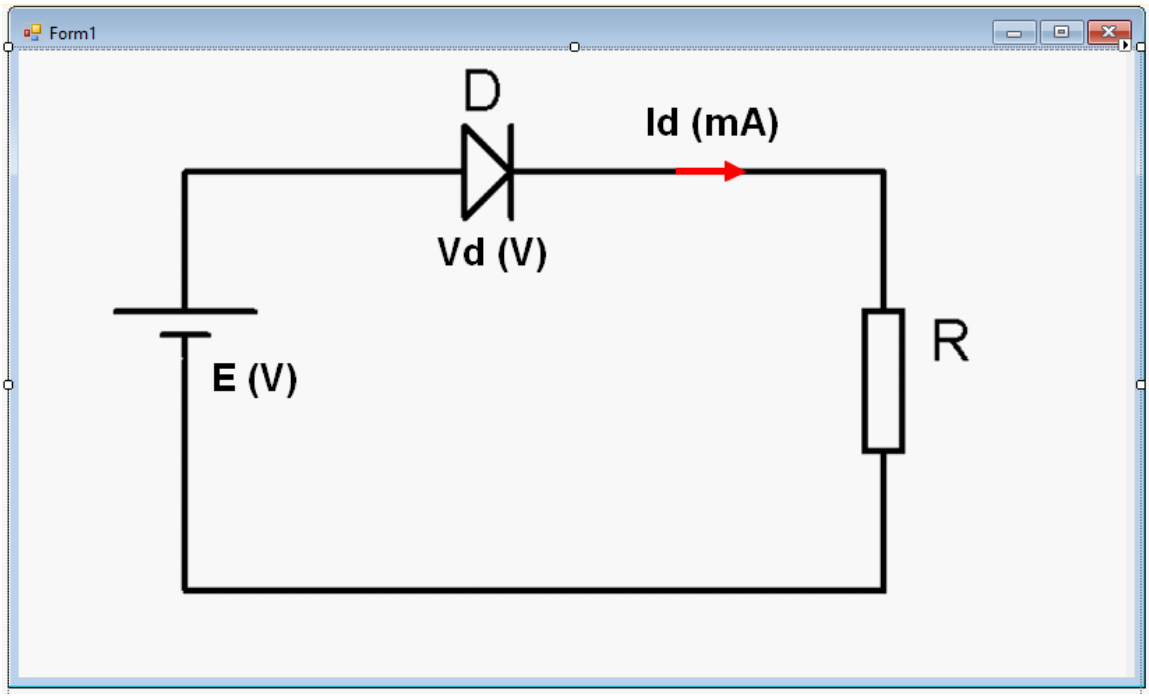
Nous allons commencer par y disposer les éléments de contrôle du programme à savoir l'image principale, les blocs permettant de saisir les valeur numériques, le bouton, etc.
Pou cela, sélectionner l'affichage de la **Boite à outils** (par Affichage>Boite à outils ou onglet à gauche de l'écran)



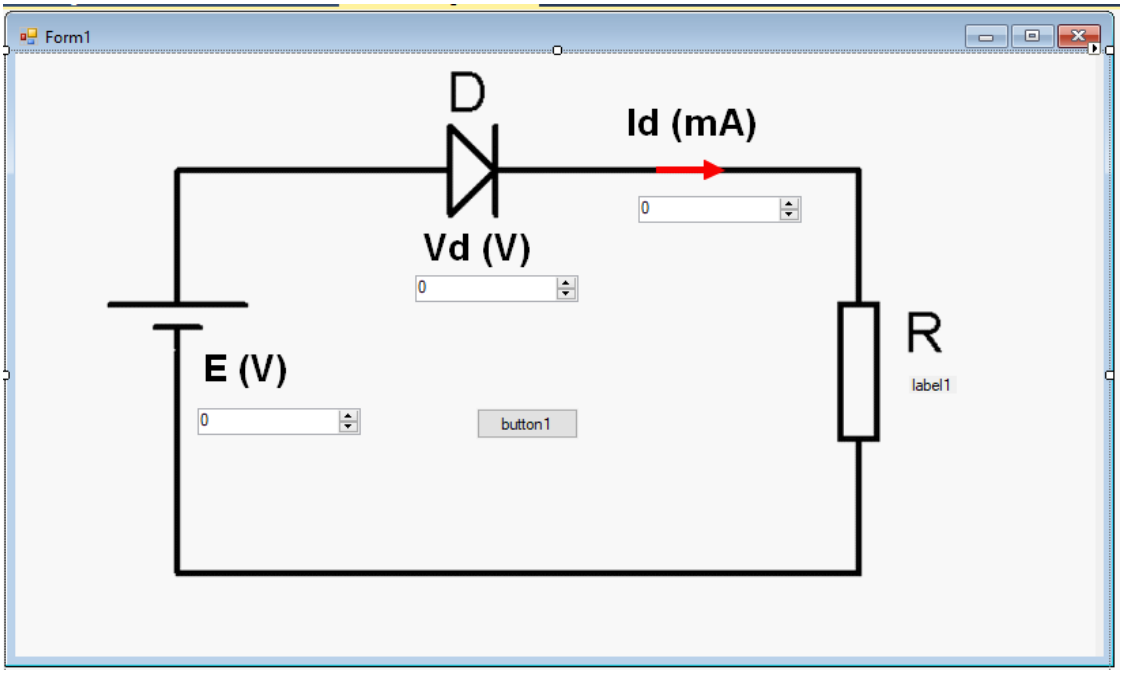
Insérer un outil **PictureBox** par glissé déposé. Éditer les propriétés de cet outil dans la fenêtre en bas à gauche, sélectionner Image puis importer l'image fournie.



Agrandir la taille de l'image et la positionner afin qu'elle occupe toute la fenêtre. Redimensionner la fenêtre du formulaire si nécessaire !



En piochant toujours dans la boite à outils, placer 3 blocs « numeric up down », un bouton qui permettra de calculer la résistance du circuit et enfin un label : une étiquette où sera écrite la valeur de la résistance.



Configurer les contrôles

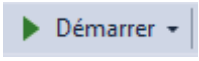
Vous allez maintenant configurer les contrôles (les outils que nous avons disposés dans notre fenêtre). Pour cela, cliquer sur la premier bloc NumericUpDown et configurer ses propriétés.

Le champ Design>Name sera appelé saisieE
Modifier les champs Données>DataBinding>
DecimalPlaces : 1 (on affiche une décimale)
Increment : 0,1 (on incrémente/décrémente de 0,1 à chaque appui sur le bouton)
Maximum : 50 (valeur maximum = 50V)

Vous remarquerez que pour chaque champ figure une explication en français et en clair.
Redimensionner enfin la taille du bloc à une largeur plus optimale.

Configurer ensuite de manière pertinente les deux autres blocs NumericUpDown.
On appellera ces deux autres blocs saisieVd, saisieIdmA.

Vous pouvez observer le résultat de cette opération en lançant la commande Démarrer. Cette commande permet d'effectuer le débogage et donc exécute le programme.

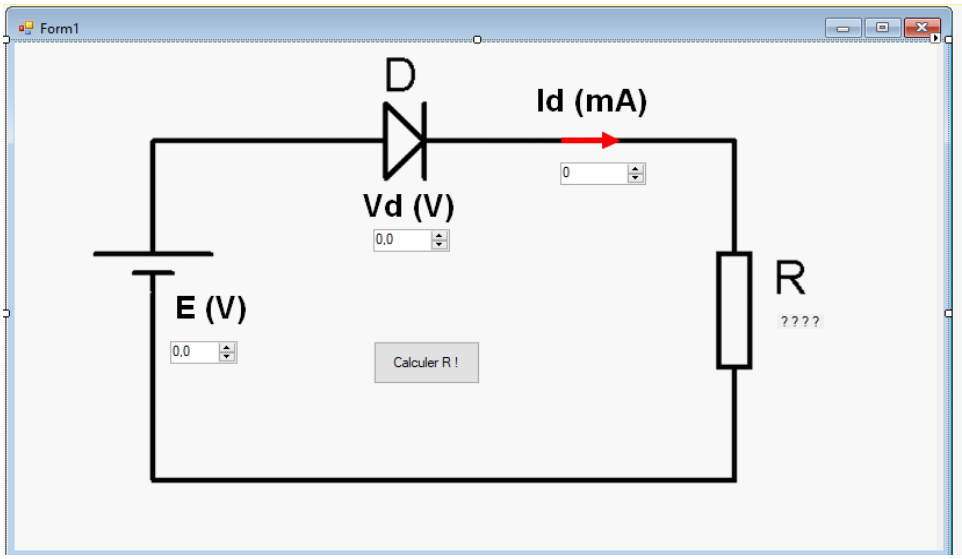


On revient au mode édition avec la commande Arrêter (carré rouge).

Bien entendu à ce stade, vous ne pouvez que faire varier les éléments, le code na pas encore été écrit !

Configurer ensuite le bouton en éditant ses propriétés. Nommer l'élément boutonCalcul. Saisir pour la propriété Text : « Calculer R ! ». Vous pouvez agrandir aussi sa taille.

Configurer le label en éditant ses propriétés. Saisir labelR pour la propriété Name. Saisir pour la propriété Text : « ? ? ? ? ». La fenêtre concepteur doit alors avoir l'allure suivante.



Maintenant que notre interface graphique et terminée, il va falloir **coder notre application**.

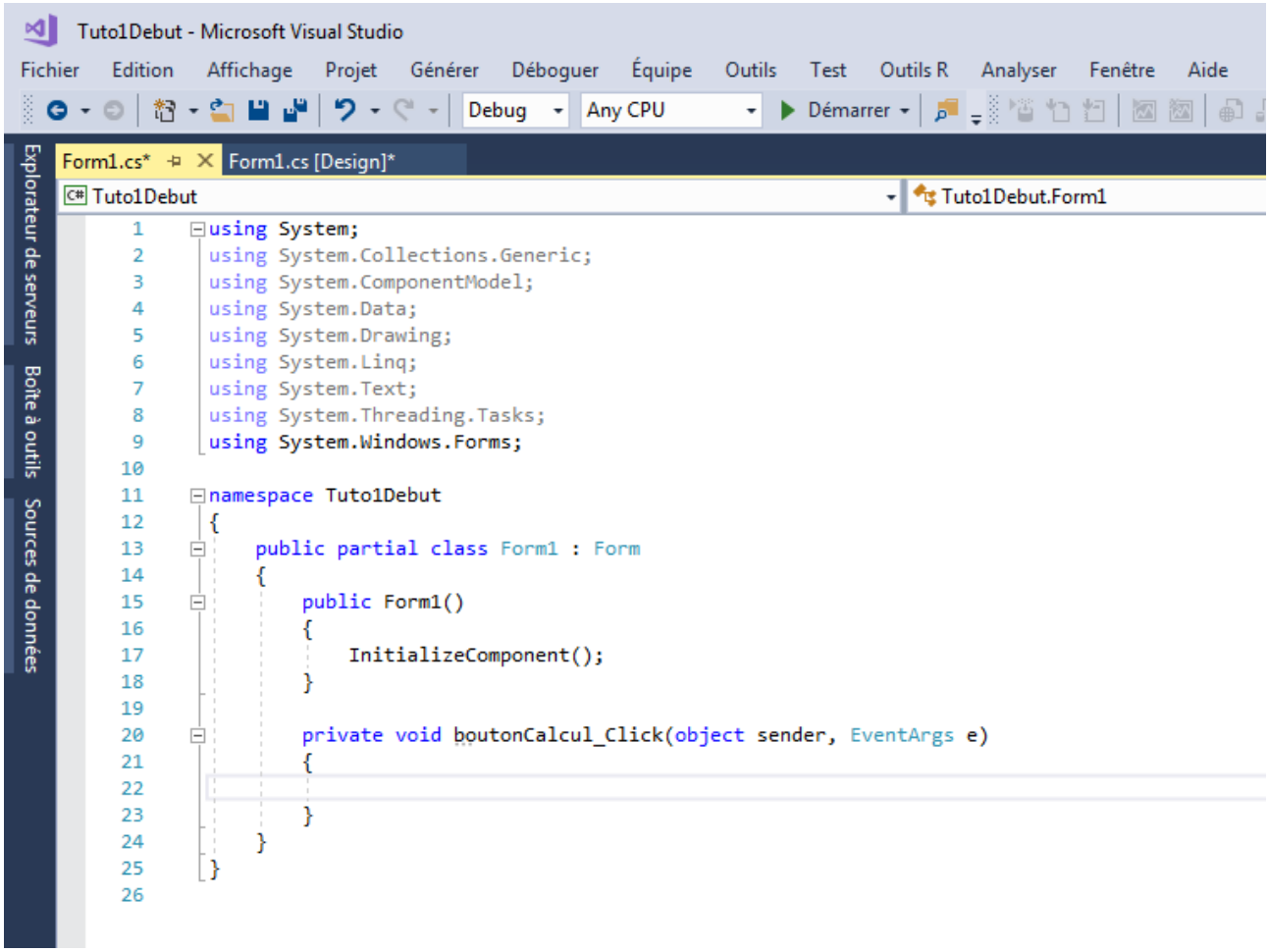
Coder

Nous allons utiliser ici de la programmation événementielle.
Les programmes C en mode console développés en première année étaient représentatifs d'une programmation séquentielle : les instructions s'enchaînaient les unes après les autres, la fin d'une instruction entraînant le début d'une autre.
En programmation événementielle, les instructions vont se produire en fonction de l'apparition d'événements. On retrouvera un peu cette démarche dans un microcontrôleur fonctionnant en interruptions, mais c'est un autre cours...

Dans notre application, c'est l'appui sur le bouton « Calculer R ! » qui va entraîner le calcul de la résistance en fonction des paramètres lus à l'instant où le bouton est apuyé.

Pour coder l'application, il faut double-cliquer sur le bouton « Calculer R ! ».

la fenêtre suivante s'ouvre :



Nous ne sommes plus en mode **Concepteur** mais en mode **Code**.
Vus pouvez passer à tout moment d'une fenetre à l'autre par le menu Affichage>Code ou Affichage>Concepteur.

Visual Studio a généré une méthode boutonCalcul_Click().

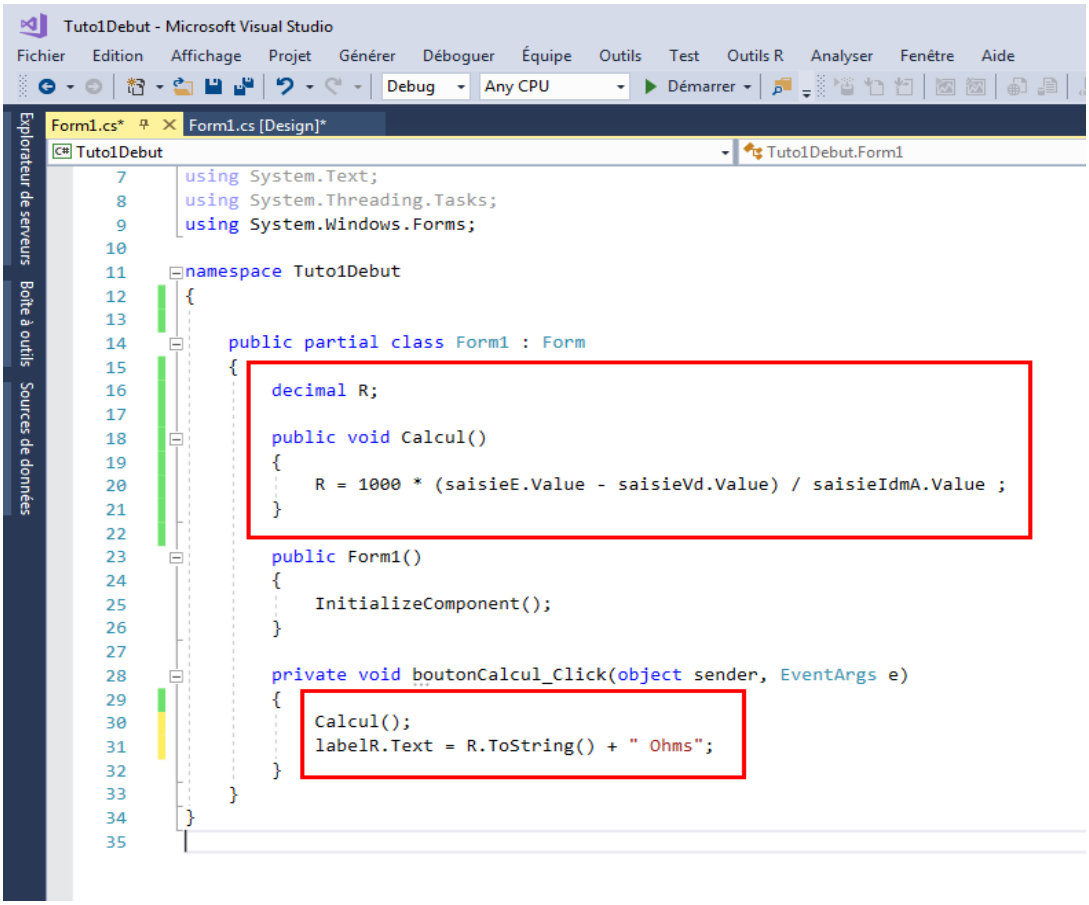
```
private void boutonCalcul_Click(object sender, EventArgs e)
      {

      }
```

Cette méthode s'exécutera lorsque l'utilisateur cliquera sur le « boutonCalcul ».

Nous allons maintenant déclarer une variable réelle R et créer une méthode appelée Calcul() qui calculera la valeur de la résistance en fonction des paramètres.

Puis dans la méthode boutonCalcul_Click() nous allons appeler la méthode Calcul() et afficher dans le labelR la valeur calculée.

Voir figure suivante, les lignes à rajouter sont indiquées en rouge !



Vous remarquerez :

- l'utilisation de decimal pour déclarer R. Par rapport à d'autres types virgule flottante (float ou double, , le type decimal fournit une plus grande précision et une plage de valeurs plus réduite.

- la sélection des valeurs indiquées se fait par la propriété Value

- la modification du texte du label se fait par la propriété Text

- la conversion float en chaîne de caractères par la méthode ToString()

- l'opération de concaténation de chaînes de caractères avec + « Ohms »

Vous pouvez maintenant tester votre application en cliquant sur la commande Démarrer.

Vérifier que celui ci fonctionne correctement en comparant les résultats obtenus avec ceux déterminés théoriquement (première partie du TP).

Faites tourner l'application en saisissant les valeurs suivantes :
E = 1V, Vd = 1,5V, I = 10 mA
Que constatez-vous ?

Même questions avec E=5V, Vd = 1,5V et I = 0

Amélioration du code

Gestion des erreurs d'entrée

Vous allez améliorer le code fourni. En particulier, vous afficherez un message d'erreur si la valeur de Vd est supérieure à la valeur de E.

Pour cela, vous placerez un label que vous nommerez labelErreur contenant le texte « ! Attention, il faut E > Vd ». Vous choisirez une taille et une couleur de police appropriée. La propriété Visible sera configurée à False. Vous ferez passer cette propriété à True en cas d'erreur.
Vous modifierez aussi la valeur de l'affichage de la valeur de R en cas d'erreur.

Effectuer les modifications, testez et validez votre programme.

Si vous ne l'avez pas déjà fait, proposer une modification simple permettant d'éviter la division par 0 si Id = 0.
Testez et validez votre programme.

Amélioration de l'affichage

Tester le programme dans la configuration suivante :
E = 2,5V, Vd=1,5V, I = 3mA

Que constatez-vous au niveau de l'affichage du résultat ?

Pour résoudre ce problème, vous allez créer une variable locale à la méthode boutonCalcul_Click() nommée R_arrondi de type décimal et vous utiliserez la méthode Math.Round(). Un arrondi à deux chiffres après la virgule est largement suffisant. Utiliser l'aide en ligne, vous pouvez aussi interroger un moteur de recherche.

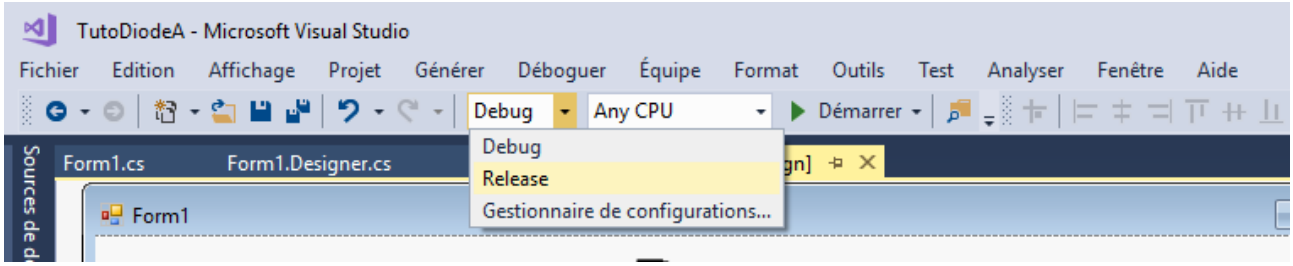
Faire les « finitions »

Récupérer le fichier exécutable

Vous venez de terminer votre première application Windows, mais pour l'instant vous ne l'avez vu tourner qu'avec Visual Studio. Vous préfériez sûrement avoir une application que vous pourriez utiliser sur un autre ordinateur, sans avoir installé Visual Studio.

Pour cela, vous choisirez le mode Release à la place du mode Debug.

Puis : Générez> Générez la solution



Vous trouverez votre application prête à être utilisée dans votre répertoire de travail>bin>Release>fichier.exe

Vérifiez que le programme fonctionne en dehors de l'environnement Visual Studio.

Améliorer l'esthétique

Après avoir sélectionné la fenêtre principale (Form1), vous effectuerez les modifications suivantes.

Vous modifierez la propriété Text du formulaire Form1, au lieu de Form1 : Calcul résistance diodes

Vous modifierez la propriété Icon. Il s'agit de l'icône présente en haut à gauche de la fenêtre lorsque celle-ci est ouverte. Celle ci doit être au format .ico et de dimensions 16x16 ou 32x32.

Pour générer une icône au format .ico, faites vous aider par Google...

Pareillement, vous modifierez l'icône de l'application. Cette modification est accessible par Projet>Propriétés de… Cette icône doit être au format .ico et de dimensions 256x256.

Les bonus

Pour ceux qui ont terminé en avance, voici quelques pistes d'améliorations en questions bonus.

Ne plus utiliser le bouton mais afficher la valeur dès que l'utilisateur modifie une des entrées !

Utiliser un comboBox(s) et un boutonRadio avec des valeurs prédéterminées de Vd ou de E.