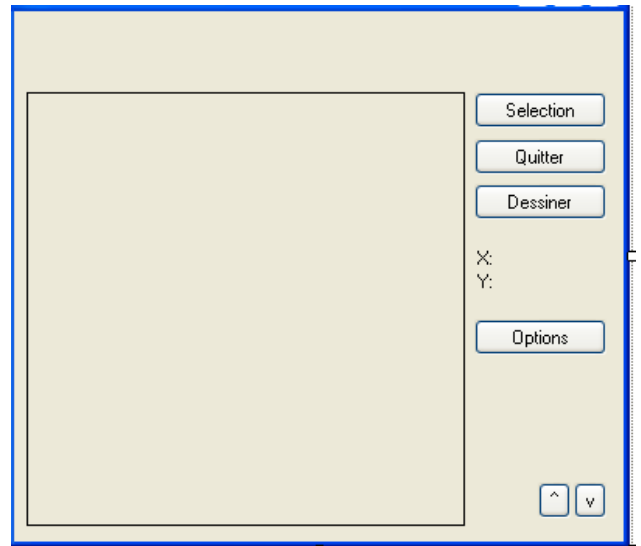


Menus et barre d'outils

I) Préparation de l'interface.

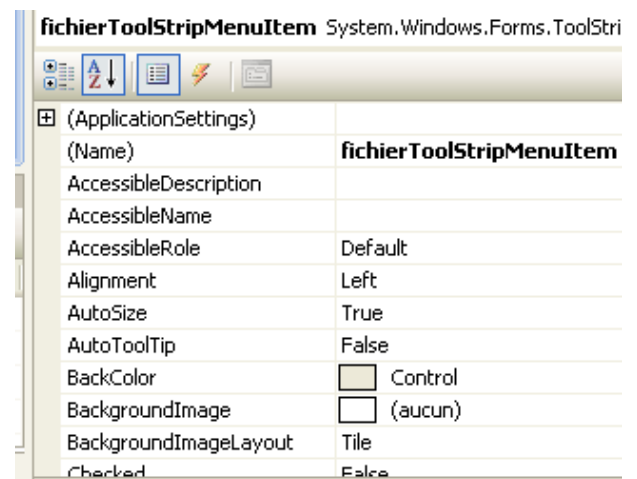
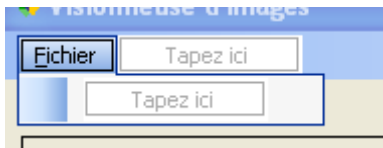
- Ouvrir le projet Visionneuse d'images (avec le bouton Options).
- Afficher le formulaire principal en mode conception.
- Augmenter la hauteur du formulaire: propriété Size.Height à 375.
- Modifier les propriétés de la PictureBox:

Location	8;52
Size	282,279
- Sélection tous les contrôles (clic et touche Alt) sauf la PictureBox et positionner l'ensemble comme ci-contre:



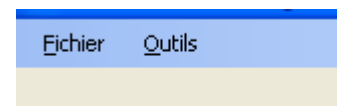
II) Ajout des menus.

- Double cliquer sur le contrôle MenuStrip.
- Remplacer sa propriété Name par **menuFichier**.
- Saisir &Fichier dans à la place de "Tapez ici".



Remarque: Le menu est un nouvel objet avec ses propriétés:

- Saisir **&Outils** dans le menu à droite:
- Renommer sa propriété Name: **menuOutils**



III) Création d'éléments de menus (commandes).

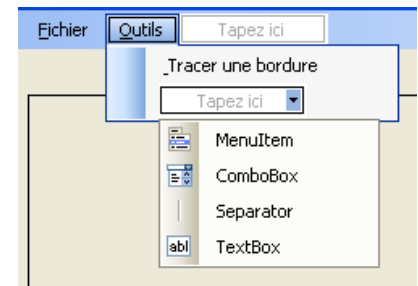
- Cliquer sur l'élément Fichier et saisir en dessous le texte: **&Ouvrir une image...**
- Remplacer sa propriété Name par: **menuItemOuvrirImage**
- sous Ouvrir une image saisir: **&Quitter**
- Remplacer sa propriété Name par: **menuItemQuitter**.



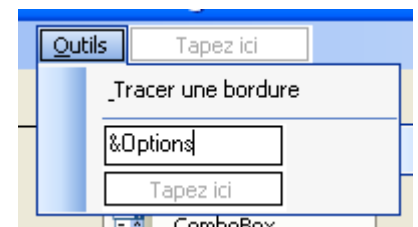
- Ajouter la commande: **&Tracer une bordure** au menu **Outils**
- Remplacer sa propriété Name par **menuItemTracerBodure**.



- survoler la zone sous **Tracer une bordure**, sélectionner la flèche et choisir **Separator** dans la liste déroulante



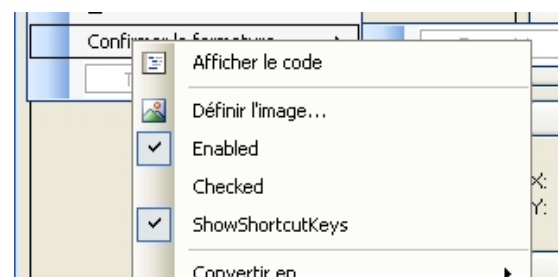
- Saisir dans la nouvelle zone Tapez ici: **&Options**.
- Valider par **Entree** et remplacer sa propriété Name par: **menuItemOptions**



Remarque: Il est très facile de supprimer un menu par clic-droit, supprimer et de déplacer des menus par glisser-déposer

IV) Ajout de coches.

- Ajouter la commande: **Confirmer la fermeture** au menu fichier:
- Remplacer sa propriété Name par **MenuItemConfirmerFermeture**.
- faire un clic-droit sur la commande de menu **Confirmer la fermeture** et sélectionner **Checked**.



- Glisser la commande de menu Confirmer la fermeture sur quitter, afin d'obtenir le résultat suivant:



- sauvegarder et compiler le projet.

V) Programmation des menus

- Ouvrir le menu fichier et double-cliquer sur ouvrir une image, le gestionnaire d'événement clic apparaît à l'écran:
- Compléter ce gestionnaire par le code ci-dessous (ce code est le même que celui du gestionnaire associé au bouton **sélection**).

```
private void MenuItemOuvrirImage_Click(object sender, EventArgs e)
{
    //Affiche la boîte de dialogue Ouvrir un fichier.
    if (ofdSelectionImage.ShowDialog() == DialogResult.OK)
    {
        //charge l'image dans la zone d'image.
        imgAfficherImage.Image = Image.FromFile(ofdSelectionImage.FileName);
        //Présente le nom du fichier dans le titre du formulaire.
        //modification par pg de la propriété texte du formulaire.
        this.Text = string.Concat("Visionneuse Image(" + ofdSelectionImage.FileName + ")");
    }
}
```

- Double-cliquer sur la commande du menu **Confirmer la fermeture** afin d'afficher son gestionnaire d'événement clic, et compléter le code:

```
private void MenuItemConfirmerFermeture_Click(object sender, EventArgs e)
{
    MenuItemConfirmerFermeture.Checked = ! (MenuItemConfirmerFermeture.Checked);
}
```

Checked est un attribut booléen, qui vaut true si la commande du menu est cochée. Ce code "lit" l'état de la coche, inverse son état, et l'affecte à la commande du menu.

- Compiler et tester le programme.
- double cliquer sur la commande de menu **Quitter** afin d'afficher son gestionnaire d'événement clic et compléter le code:

```
private void MenuItemQuitter_Click(object sender, EventArgs e)
{
    this.Close();
}
```

- Double-cliquer sur la commande de menu **Tracer une bordure** du menu **outils** afin d'afficher son gestionnaire d'événement et compléter le code (ce code est le même que celui associé au bouton **Dessiner**):

```
private void tracerUneBordureToolStripMenuItem_Click(object sender, EventArgs e)
{
    Graphics objGraphics = null;
    objGraphics = this.CreateGraphics();
    objGraphics.Clear(SystemColors.Control);
    objGraphics.DrawRectangle(Pens.Blue, imgAfficherImage.Left - 1, imgAfficherImage.Top - 1,
    imgAfficherImage.Width + 1, imgAfficherImage.Height + 1);
    objGraphics.Dispose();
}
```

- Compiler et tester l'application.
- Double-cliquer sur la commande de menu Options afin d'afficher son gestionnaire d'événement clic et compléter le code (ce code est même que celui associé au bouton **Options**):

```
private void MenuItemOptions_Click(object sender, EventArgs e)
{
    FormOptions ObjetOptions = new FormOptions();
    ObjetOptions.ShowDialog();
}
```

- Compiler et tester tous les menus ainsi que les touches de raccourci.

VI) Amélioration de la commande de menu "Confirmer la fermeture".

- Créer une nouvelle variable Confirm de type bool et l'initialiser à false dans le constructeur.

```
public partial class frmVisionneuse : Form
{
    bool Confirm;

    public frmVisionneuse()
    {
        InitializeComponent();
        Confirm = false;
    }
}
```

- Compléter le gestionnaire d'événement load afin de d'initialiser la coche de la commande de menu à l'ouverture du formulaire:

```
private void Form1_Load(object sender, EventArgs e)
{
    labelx.Text = "";
    labely.Text = "";
    int NbBoutons = SystemInformation.MouseButtons;
    labelx.Text = NbBoutons.ToString();
    MenuItemConfirmerFermeture.Checked = Confirm;
}
|
}
```

- Compléter le gestionnaire d'événement associé à la commande de menu Confirmer la fermeture:

```
private void MenuItemConfirmerFermeture_Click(object sender, EventArgs e)
{
    MenuItemConfirmerFermeture.Checked = !(MenuItemConfirmerFermeture.Checked);
    //récupération de l'état de la coche du menu dans confirm
    Confirm = MenuItemConfirmerFermeture.Checked;
}
```

Le code rajouté permet de stocker dans la variable confirm l'état de la coche après avoir cliquer sur la commande de menu.

- Sélectionner le formulaire et ajouter lui l'événement FormClosing (double clic sur l'événement)

Cette événement est déclenché lors de la demande de fermeture du formulaire.

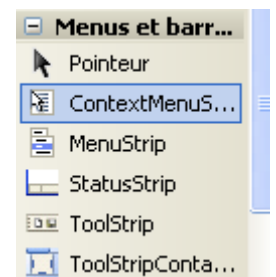
- Compléter le code de ce gestionnaire:

```
private void frmVisionneuse_FormClosing(object sender, FormClosingEventArgs e)
{
    if (Confirm == true) //la commande du menu est coché.
    {
        if (MessageBox.Show("Fermer l'application Visionneuse d'image?", "Confirmer la fermeture",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question)
            == DialogResult.No) //si on a cliqué sur Non
        {
            e.Cancel = true; //abandon de la fermeture
        }
    }
}
```

- Compiler et tester l'application ainsi que la commande de menu **Confirmer la fermeture**.

VII) Menus contextuels

- double-cliquer sur le contrôle **ContextMenuStrip** de la boîte à outils.
- Donner la valeur **ContextMenuImage** à sa propriété **Name**.
- Saisir le texte **Tracer une bordure** à la place de **Tapez ici**, et valider par entrée.
- double cliquer sur ce menu pour accéder à son gestionnaire click et compléter le code (le même que précédemment, il va falloir penser à le placer dans une méthode...)



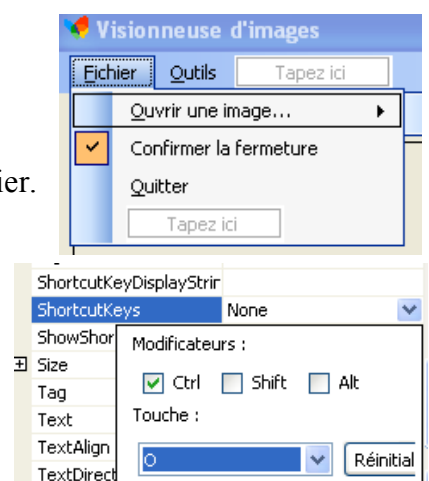
```
private void taceToolStripMenuItem_Click(object sender, EventArgs e)
{
    Graphics objGraphics = null;
    objGraphics = this.CreateGraphics();
    objGraphics.Clear(SystemColors.Control);
    objGraphics.DrawRectangle(Pens.Blue, imgAfficherImage.Left - 1, imgAfficherImage.Top - 1,
    | imgAfficherImage.Width + 1, imgAfficherImage.Height + 1);
    objGraphics.Dispose();
}
```

Il faut lier ce menu contextuel à un élément de l'interface (exemple: la zone image: PictureBox)

- Sur le formulaire en conception, sélectionner la zone image et définir sa propriété **ContextMenuStrip** à la valeur **ContextMenuImage** (qui est le nom du contrôle **ContextMenuStrip**)
- Compiler et tester le menu contextuel.

VIII) Touches de raccourci pour les menus.

- Sélectionner la commande de menu Ouvrir une image du menu Fichier.
- Rechercher la propriété **ShortcutKeys** et cliquer sur la flèche déroulante.
- Sélectionner les valeurs ci-contre.
- Compiler et tester ce raccourci clavier.



IX) Création d'une barre d'outils.

- Double cliquer sur le contrôle **ToolStrip** de la boîte à outils.
- Remplacer son nom (propriété Name) par **toolStripBarrePrincipale**
- Faire un clic droit sur la barre d'outils et choisir: Mettre au premier plan.
(la barre se place sous les menus).

Mise en place des boutons:

le contrôle ToolStrip supporte un collection: la collection Items.

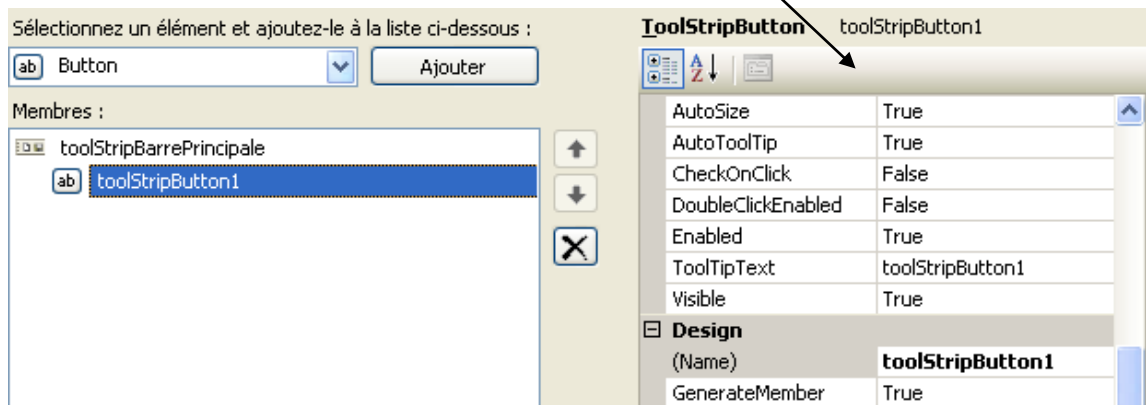
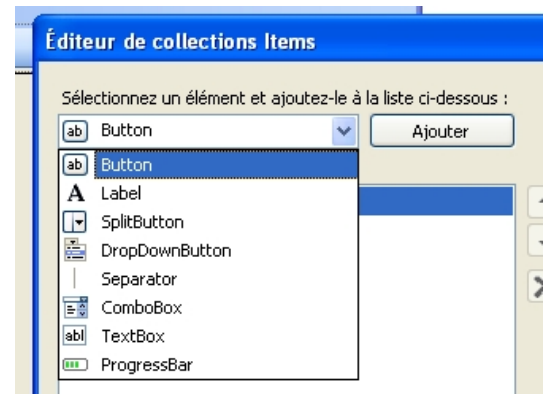
- Cliquer sur la propriété Items, sur le bouton **Générer: ...**

La barre d'outils fait partie des membres, mais aucun bouton n'en fait partie.

- Dérouler la liste déroulante pour afficher les éléments qu'il est possible de rajouter à la barre d'outils:

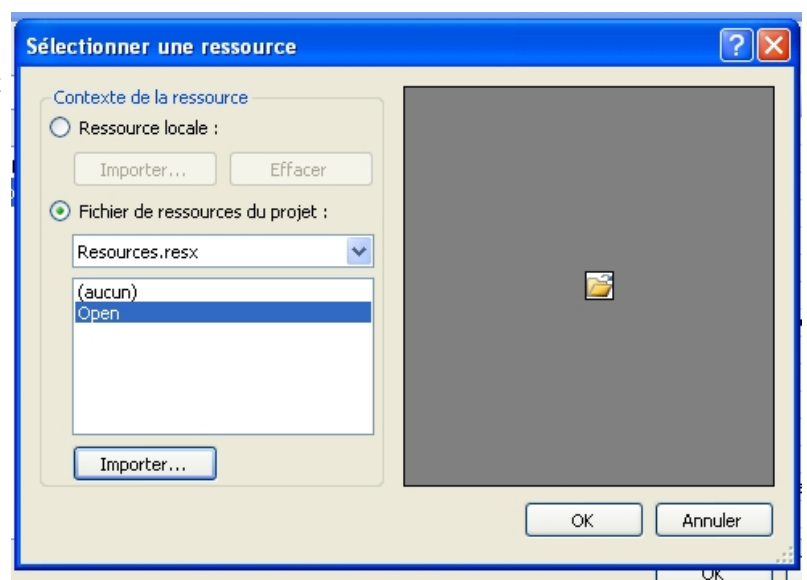
Nous allons créer trois boutons (associés à trois image):

- Dans la liste déroulante, sélectionner Button et cliquer sur Ajouter.
- Sélectionner toolStripButton1 et modifier ses propriétés:



Name ToolStripButtonOuvrirImage
Text Ouvrir une image
ToolTipText Ouvrir une image

- Cliquer sur la propriété Image du bouton et sur le bouton Générer ...
- Importer un fichier image dans le projet (exemple: Open.ico).
Ce fichier sera ajouter au projet et fera partie des ressources du projet.
- Valider par OK

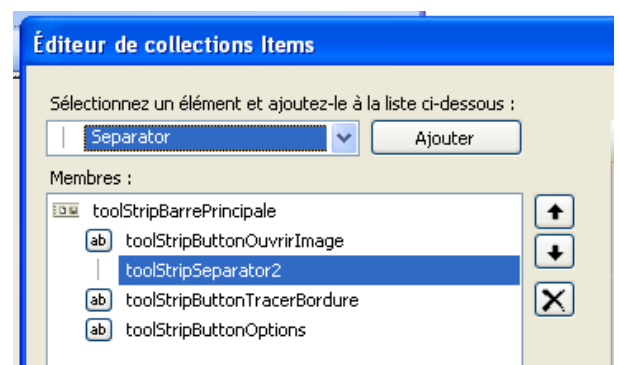


- Refaire le même travail avec deux autres boutons associés à deux autres images:

Propriété	Valeur
Name	ToolStripButtonTracerBordure
Text	Tracer une bordure
ToolTipText	Tracer une bordure
image	DrawBorder.ico
Name	ToolStripButtonOptions
Text	Options
ToolTipText	Options
image	Options.ico

Mise en place de séparateurs.

- De la même façon, ajouter un séparateur.
- A l'aide de la flèche verticale positionner le séparateur entre le 1^o et 2^o bouton.
- Ajouter et placer de la même façon un autre séparateur entre le 2^o et 3^o bouton.
- Valider par OK

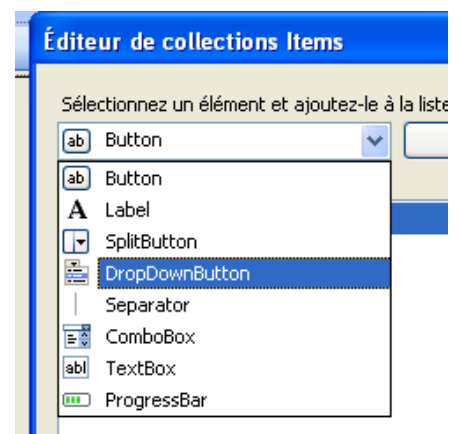


- Compiler et tester le projet, il ne reste plus qu'à associer du code à chaque bouton.
- Double-cliquer respectivement sur chaque bouton afin d'accéder à son gestionnaire clic, et placer le code associé dans chaque gestionnaire (même code que les menus).
- Compiler et tester les boutons de la barre d'outils du projet.

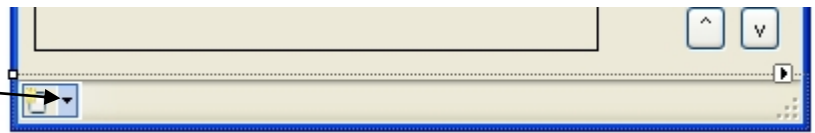
Remarque: Il est possible de créer des menus déroulants pour les boutons, il suffit de choisir `DropDownButton` à la place de `Button`.

Mise en place d'une barre d'état.

- Double cliquer sur le contrôle **StatusStrip** dans la boîte à outils. (la barre s'affiche en bas du formulaire)
- Renommer ce contrôle en **statusStripBarreEtat**
- Modifier la position des contrôles qui sont recouverts par la barre d'état.

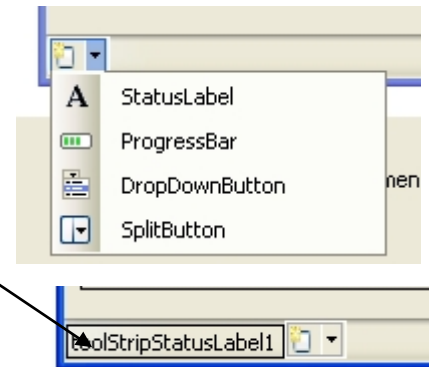


- Sélectionner la barre d'état
- Dérouler la liste déroulante.
- Choisir **StatusLabel**



- sélectionner l'étiquette.
- Modifier ses propriétés comme suit:

Propriété	Valeur
Name	toolStripStatusLabel
Text	Aucune image chargée.



- Compiler et tester le projet

X) Création de méthodes.

Certaines parties de codes sont répétées trois fois, il est donc intéressant de placer ce code dans une méthode et d'appeler cette méthode à partir de multiples endroits.

- Créer la méthode **OuvrirImage()**, et compléter là avec le code utilisé dans les gestionnaires d'événements du menu Fichier, Ouvrir une image, bouton sélection, bouton de barre d'état:



```
public partial class frmVisionneuse : Form
{
    bool Confirm;

    private void OuvrirImage()
    {
        //Affiche la boîte de dialogue Ouvrir un fichier.
        if (ofdSelectionImage.ShowDialog() == DialogResult.OK)
        {
            //charge l'image dans la zone d'image.
            imgAfficherImage.Image = Image.FromFile(ofdSelectionImage.FileName);
            //Présente le nom du fichier dans le titre du formulaire.
            //modification par pg de la propriété texte du formulaire.
            this.Text = string.Concat("Visionneuse Image(" + ofdSelectionImage.FileName + ")")
        }
    }
}
```

- Dans ces gestionnaires, remplacer le code par l'appel de la méthode **OuvrirImage()**.

```
private void btnSelectionImage_Click(object sender, EventArgs e)
{
    OuvrirImage();
}
```

- Compléter les deux autres gestionnaires.
- Refaire le même travail pour les autres parties de code redondantes.

- Afin d'afficher le nom du fichier chargé, compléter la méthode OuvrirImage() avec le code suivant:

```
//Affiche le nom du fichier dans la barre d'état  
statusStripBarreEtat.Items[0].Text = ofdSelectionImage.FileName;
```