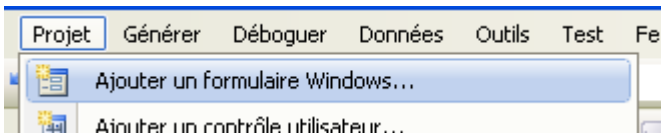
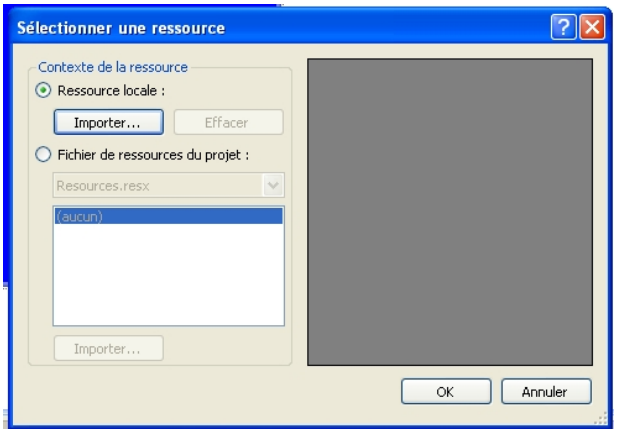
Ajout de formulaire à une application.

- A partir de l'application visionneuse d'images, ajouter un nouveau formulaire de la façon suivante:



- Choisir WindowsForm
- Dans la zone de texte, saisir le nom du nouveau formulaire: formOptions.cs.
- Cliquer sur Ajouter.
- Remplacer la propriété text du formulaire par: Options visionneuse d'images.
- Fixer la propriété Size à 400;300
- Remplacer la propriété BackColor qui a la valeur control par la valeur 0; 0; 255
  (La valeur Control correspond au modèle de couleurs système défini dans Windows).
- Cliquer sur la propriété BackgroundImage, cliquer sur les …



- Cliquer sur Importer, sélectionner le fichier Options.bmp (préalablement copié dans votre projet), valider par OK.
- Choisir un icône avec la propriété Icon.
- Supprimer les icones "Agrandir" et "réduire" en modifiant les propriétés suivantes:

Propriété	Valeur
MinimizeBox	False
MaximizeBox	False

Remarque: La propriété ControlBox permet d'afficher ou pas le bouton de fermeture.

- La propriété FormBorderStyle contrôle l'apparence et le comportement de la bordure d'un formulaire.

Fixer cette propriété à la valeur FixedToolWindow.

- Ajouter un bouton au formulaire principal avec les propriétés suivantes:

Propriété	Valeur
Name	BoutonOptions
Location	301;156
Size	85;23
Text	Options

- Double-cliquer sur le bouton pour accéder au code du gestionnaire d'événement et compléter le code:

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

- Compiler et tester le projet: Comme l'indique le code ci-dessus, chaque appui sur le bouton **Options** crée une nouvelle instance de la classe FormOptions (classe associée au formulaire FormOptions).

  Remarque: En mode conception, le code de cette classe est visible par un clic droit sur le formulaire For mOptions.

  La méthode Show() permet d'afficher le formulaire sous la forme "**non modal**".
  Le formulaire affiché et "non modal" car on peut continuer à cliquer sur le formulaire principal, même si le formulaire FormOptions n'est pas fermé.

  Pour afficher le formulaire FormOptions sous la forme "**modal**", modifier le code du gestionnaire com me ci-dessous (utilisation de la méthode ShowDialog()):

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

Position initiale d'un formulaire.

La position initiale d'un formulaire est précisée par la propriété **StartPosition.**

- donner les valeurs suivantes à la propriété **StartPosition** du formulaire FormOptions et observer le fonc- tionnement:

Valeurs à tester: CenterScreen, WindowsDefaultLocation, WindowsDefaultBounds, CenterParent

- Remarque:

La valeur **Manual** permet de spécifier la position par programmation à l'aide de la propriété **Location**.

Etat d'un formulaire.

La propriété WindowsState permet de définir si le formulaire s'affiche:

- Réduit dans la barre des tâches Minimized
- Agrandit Maximized
- Normal Normal

Tester ces différentes possibilités

Il est bien sur possible de changer cette propriété par le code:

- Placer la ligne de code suivante dans le code du constructeur du formulaire

```
this.WindowState = FormWindowState.Minimized;
```

La propriété **BackgroundImageeLayout** permet d'adapter le format de l'image de fond au formulaire.

- Tester les différentes valeurs de cette propriété.

La propriété **ShowInTaskbar** avec la valeur **False** permet d'empêcher un formulaire d'apparaître dans la barre des tâches. Une fois réduit, il faut utiliser ALT TAB pour le faire réapparaître.

Décharger un formulaire

Pour masquer un formulaire (c'est-à-dire qu'il n'est plus visible, mais il est toujours en mémoire, on peut donc toujours le manipuler par du code):

Définir la Propriété **Visible** sur **False**
Utiliser la méthode **Hide** du formulaire.

Pour le fermer définitivement, utiliser la méthode Close();

- Placer un nouveau bouton sur le formulaire FormOptions, avec les propriétés suivantes:

Propriété	Valeur
Name	BoutonOK
Location	305;12
Text	OK

Compléter le gestionnaire d'événement click du bouton:

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