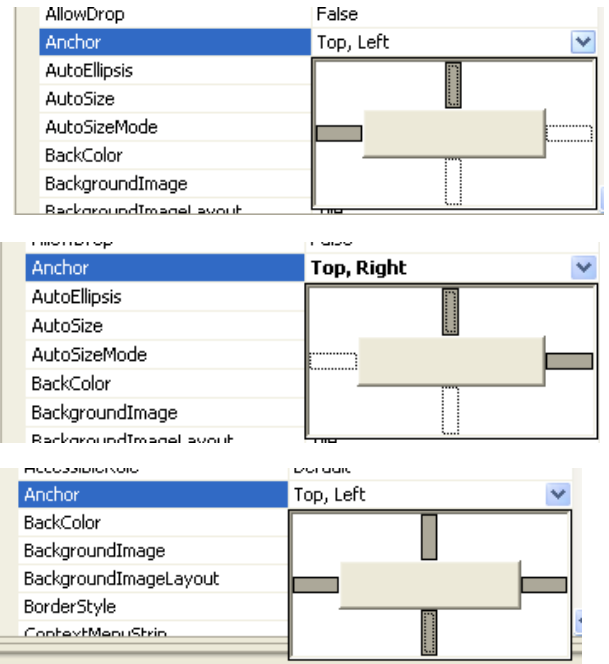


TP Contrôles classiques

I) Ancrage, Opacité, barre de défilement...

Ouvrir le projet Visionneuse avec le formulaire options

- Compiler le projet, et agrandir le formulaire, les boutons restent à la même place.
- Sélectionner tous les boutons (touche MAJ + clic).
- Sélectionner la propriété Anchor, et cliquer sur la liste déroulante. ici les contrôles sont ancrés au bords gauche et haut du formulaire.
- Modifier la propriété Anchor de la façon suivante:
- Tester le projet.
- Refaire la même chose pour les labels X, Y
- Pour la zone image, donner à la propriété Anchor la valeur suivante:
- Sélectionner la zone image (PictureBox), et essayer les différentes valeur de la propriété **SizeMode**.
- Tester la valeur de la propriété **Opacity** sur le formulaire Options.
- Tester la propriété **AutoScroll** en lui donnant la valeur **True**.



II) Mise en œuvre de divers contrôles de base

Nous allons travailler sur le formulaire secondaire Options

- Ajouter un nouveau label, avec les propriétés suivantes:

Name	LabelNomUtilisateur
Location	40,41
Text	Nom d'utilisateur

- Ajouter une nouvelle zone de texte (contrôle TextBox) avec les propriétés suivantes:

Name	textBoxNomUtilisateur
Location	135,39
Size	139;20
Text	Voici un exemple de texte.

- Tester les différentes valeurs de sa propriété **TextAlign** et donner lui la valeur **Left**.
 - Définir ses propriétés de la façon suivante:
- | | |
|-----------|--|
| multiline | true |
| Size | 139;60 |
| Text | Voici un exemple de texte, une zone de texte à plusieurs lignes s'adapte à la taille du texte. |

- Donner la valeur **False** à sa propriété **Enabled** et tester le projet.
- Tester les différentes valeurs de sa propriété **ScrollBars**.
(essayer la valeur **horizontal** avec la propriété **WordWrap** à **True** et à **False**)

Remarque:

La propriété **AcceptReturn** permet à l'utilisateur d'utiliser la touche "entrée" pour créer une nouvelle ligne.
La propriété **AcceptTabs** permet à l'utilisateur d'utiliser la touche "Tab" pour créer des colonnes.

- Définir ses propriétés aux valeur suivantes, et tester le projet.

Text
MaxLength **11**
Multiline False
ScrollBars None

Remarque: Si **MaxLength** est définie à 0, il n'y a pas de limite maximale.

- Tester la propriété **PasswordChar** avec le caractère *
- Remarque: Cette propriété ne fonctionne que si la propriété **Multiline** est sur **False**

Evénements les plus courants utilisés avec un contrôle TextBox

Evénement	Description
TextChanged	Se déclenche chaque fois que l'utilisateur appuie sur une touche. Utilisez cet événement pour gérer l'enfoncement de touches spécifiques (pour réaliser des captures, par exemple) ou pour accomplir une action dès que le contenu change.
Click	Se déclenche dès que l'utilisateur clique sur la zone de texte. Utilisez cet événement pour capturer des clics lorsque les coordonnées du pointeur de la souris ne sont pas importantes.
MouseDown	Se déclenche lorsque l'utilisateur appuie pour la première fois sur un bouton de la souris dans la zone de texte. On l'utilise souvent en combinaison avec l'événement MouseUp.
MouseUp	Se déclenche lorsque l'utilisateur relâche le bouton de la souris dans la zone de texte. Utilisez MouseDown et MouseUp lorsque vous avez besoin de plus de fonctionnalités que celles fournies par l'événement Click.
MouseMove	Se déclenche lorsque l'utilisateur place la souris sur la zone de texte. Utilisez cet événement pour accomplir des actions basées sur le mouvement du curseur.
KeyDown	Se déclenche lorsque l'on appuie sur une touche alors que le focus se trouve dans le contrôle.
KeyUp	Se déclenche lorsque l'on relâche une touche alors que le focus se trouve dans le contrôle.
KeyPress	Se déclenche lorsque l'on appuie sur une touche alors que le focus se trouve dans le contrôle. Cet événement se produit entre les événements KeyDown et KeyUp.

- Placer un nouveau bouton avec les propriétés suivantes:

Name	BoutonAnnuler
Location	304;38
Text	Annuler

- On va définir un bouton par défaut (bouton d'acceptation), c'est-à-dire que son événement click se déclenchera lors de l'appui sur la touche "Entrée":
- Sélectionner le formulaire, sélectionner la propriété **AcceptButton**.
- Dans la liste déroulante choisir **BoutonOK**
- Vérifier que le gestionnaire d'événement clic du Bouton OK contient le code **this.Close()**
- Compiler et tester le projet de la façon suivante: saisir du texte dans la textBox et valider par "Entrée"

Exceptions: si le contrôle actif est une TextBox à plusieurs lignes avec sa propriété **AcceptsReturns** définie à **True**, l'appui sur "Entrée" ne déclenche pas l'événement click du bouton d'acceptation mais crée une nouvelle ligne de texte dans la TextBox.

- définir la propriété **CancelButton** à la valeur **BoutonAnnuler**, ainsi l'appui sur "Echap" va lancer le code du gestionnaire d'événement clic associé au bouton **Annuler**.
- Ajouter un case à cocher (contrôle checkBox) avec les propriétés suivantes:


Name	checkBoxConfirmer
Location	140;79
Text	Confirmer pour quitter

- Tester les valeurs de la propriété CheckState (elle permet de cocher ou pas la case), et laisser la valeur Unchecked pour finir.

III) Les contrôles "conteneur".

Les contrôles "conteneur" peuvent contenir d'autres contrôles.

Il existe le contrôle **GroupeBox** et le contrôle **Panel**. Le contrôle **Panel** est une version allégée du contrôle **GroupeBox**.

- Ajouter un contrôle **GroupeBox** (catégorie conteneur)
Remarque: Pour le déplacer, il faut utiliser 
- Définir les propriétés de la zone de groupe de la façon suivante:

Name	groupeBoxArrierePlanParDefaut
Location	140;112
Size	200;72
Text	Couleur d'arriere plan par défaut.

Nous allons ajouter deux contrôles dans ce conteneur

- Par "Glisser Déposer", placer un bouton radio (contrôle **RadioButton**) sur le conteneur.

- Définir les propriétés de ce bouton de la façon suivante:

Name	radioButtonArrierePlanParDefaut
Checked	True
Location	6;19
Text	Gris par défaut

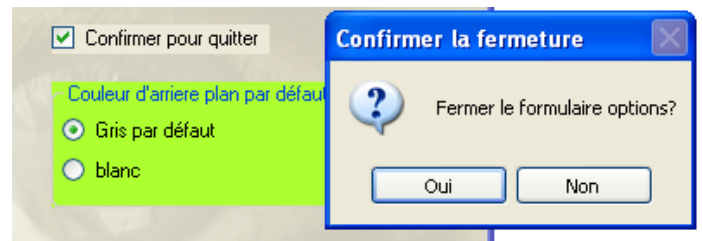
- Placer un deuxième Bouton radio avec les propriétés suivantes:

Name	radioButtonArrierePlanBlanc
Checked	False
Location	6:42
Text	Blanc

- Compiler et tester le projet

IV) Travail: Compléter le code du formulaire Options afin de satisfaire aux point suivants:

- Si la case "Confirmer pour quitter" est cochée une MessageBox s'affiche
- Lorsqu'on clique sur un des boutons radio, celui-ci se coche et l'arrière plan du formulaire prend la valeur choisie.



Aide:

- Pour tester si la case à cocher (qui s'appelle checkBoxConfirmer dans l'exemple) est cochée ou pas:

```
if (checkBoxConfirmer.CheckState == CheckState.Checked)
```

- Pour tester si un bouton radio est coché ou pas, on peut mettre en place l'événement "CheckedChanged" du bouton radio (appelé radioButtonArrierePlanParDefaut dans l'exemple):

```
private void radioButtonArrierePlanParDefaut_CheckedChanged(object sender, EventArgs e)
{
    Color ArrierePlan1 = System.Drawing.Color.Gray;
    this.BackColor = ArrierePlan1;
}
```

Remarque: On a déclaré une variable (ArrierePlan1) de type "Color" afin de définir la couleur de fond du formulaire secondaire.