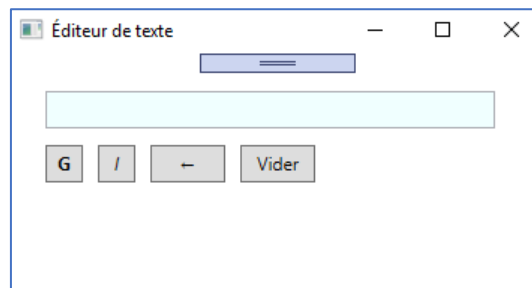


Laboratoire

Contrôles, événements et StackPanel

Exercice 1 :

On vous demande de créer un simple éditeur de texte avec des contrôles de fonctionnement minimaux comme dans la figure ci-dessous :



Contrôles et propriétés :

- La fenêtre est de dimension 200 x 375.
- La zone de texte est de hauteur 25, de largeur 300, sa couleur d'arrière-plan est « Azure » et affiche une infobulle contenant le texte « Veuillez saisir du texte » quand la souris est positionnée au-dessus de cette zone de texte. Nommez-la « txtText ».
- Le texte affiché dans le bouton « G » est en gras. Ce bouton est de hauteur 25 et de largeur 25 et affiche l'infobulle « Mettre le texte en gras ». Nommez-le « btnGras »
- Le texte affiché dans le bouton « I » est en italique. Ce bouton est aussi de hauteur 25 et de largeur 25 et affiche l'infobulle « Mettre le texte en italique ». Nommez-le « btnItalique »
- Le bouton contenant une flèche vers la gauche est nommé « btnEffacerArriere ». Il est désactivé par défaut, de hauteur 25 et de largeur 50, affiche l'infobulle « Effacer le caractère précédent ». La flèche affichée comme contenu de ce bouton ne peut être ajoutée que dans le code behind via le code « \u2190 ».
- Le bouton contenant le texte « Vider » est nommé « btnViderChamp ». Il est désactivé par défaut, de hauteur 25 et de largeur 50 et affiche l'infobulle « Vider la zone de texte ».

Événements :

Ajoutez les événements sur clic pour les 4 boutons.

- Définissez la méthode « *BtnGras_Click* » pour le bouton qui met le texte en gras ou le remet en style normal. Utilisez « *FontWeight* » et « *FontWeights* » pour afficher le texte en « *Bold* » ou « *Normal* » (si déjà en gras). Mettez aussi la couleur d'arrière-plan du bouton en orange quand le texte est en gras.
- Définissez la méthode « *BtnItalique_Click* » pour le bouton qui met le texte en italique ou le remet en style normal. Utilisez « *FontStyle* » et « *FontStyles* » pour afficher le texte en « *Italic* » ou « *Normal* » (si déjà en *Italic*). Mettez aussi la couleur d'arrière-plan du bouton en orange quand le texte est en italique.
- Définissez la méthode « *TxtText_TextChanged* » sur la zone de texte et qui désactive les boutons « précédent » et « Vider » quand le TextBox est vide et les réactive dès qu'ils aient au moins un caractère.
- Définissez la méthode « *BtnEffacerArriere_Click* » pour le bouton qui efface le dernier caractère de la zone de texte (bouton précédent) en utilisant la méthode « *Remove* ». Mais avant, il faut s'assurer que le TextBox n'est pas vide pour qu'il n'y ait pas la levée d'une exception.
- Définissez la méthode « *BtnViderChamp_Click* » pour le bouton « Vider » pour effacer d'un coup tout le texte de la zone de texte.

Exercice 2 :

En utilisant des « *StackPanels* », créez l'interface utilisateur suivante. La dimension de la fenêtre est 250 x 300.

The screenshot shows a WPF application window titled "StackPanel". The window has a light blue background and contains two sections. The first section is titled "Prénom et nom:" and contains two text boxes labeled "Prénom:" and "Nom:". The second section is titled "Adresse:" and contains four text boxes labeled "Rue:", "Ville:", "Province:", and "Code Postal:". The window has standard Windows controls (minimize, maximize, close) in the title bar.