

Programmation des Interfaces Interactives Avancées

Travaux pratique numéro 4 - Correction

Ce TP a pour but de vous familiariser avec la manière d'agencer les widgets dans une fenêtre en Visual C#. Pour cela, on utilise classiquement des objets de type `Layout` et `MenuStrip` de `Windows Forms`. Avant de commencer le TP, il est conseillé de créer un nouveau projet dédié que vous appellerez TP4 en suivant les instructions de la dernière fois.

Exercice 1

1) Ajoutez la méthode `SetupUI()` dans le constructeur de la classe TP4 et son code se trouve ci-dessous.

```
namespace TP4
{
    public partial class TP4 : Form
    {
        public TP4()
        {
            InitializeComponent();
            SetupUI();
        }

        private void SetupUI()
        {
            this.Text = "un test";

            this.Size = new System.Drawing.Size(500, 300);

            // Create TableLayoutPanel
            TableLayoutPanel tableLayout = new TableLayoutPanel
            {
                RowCount = 2,
                ColumnCount = 2,
                Dock = DockStyle.Fill,
                CellBorderStyle = TableLayoutPanelCellBorderStyle.Single,
                AutoSize = true
            };

            Label label1 = new Label {Text = "Bonjour!!", AutoSize = true};
            Label label2 = new Label {Text = "Enchanté!!", AutoSize = true};
            Label label3 = new Label {Text = "Comment allez-vous?", AutoSize =
true};

            // Add labels
            tableLayout.Controls.Add(label1, 0, 0);
            tableLayout.Controls.Add(label2, 1, 0);
            tableLayout.Controls.Add(label3, 0, 1);

            this.Controls.Add(tableLayout);
        }
    }
}
```

- 1) Consultez la documentation. Expliquez ce que permet de faire `TableLayoutPanel`.

Correction : C'est un conteneur qui permet d'afficher des widgets dans un tableau avec lignes et colonnes.

- 2) Expliquez la façon de mettre en place des éléments dans le conteneur `TableLayoutPanel` à travers des labels 1, 2, et 3.

Correction : On spécifie la ligne et colonne de chaque widget en appelant la méthode `Add` de la propriété `Controls` du `TableLayoutPanel`.

- 3) Changez les paramètres de `TableLayoutPanel` pour qu'il ait une taille constante (e.g., 200 x 100) et une position fixe (e.g., (50, 50)) par rapport à la fenêtre principale. A noter que cette taille ne peut être garantie à tout moment puisque l'utilisateur a le droit de redimensionner la fenêtre.

Correction : Pour que la taille soit constante, on peut fixer la propriété `Dock` de la manière suivante : `Dock = DockStyle.None`,

Pour définir la taille, on utilise les propriétés `Height` et `Width`.

Pour fixer une position du conteneur `TableLayoutPanel`, on peut utiliser les propriétés `Left` et `Right`. On peut aussi définir la propriété `Padding` de la manière suivante : `Padding = new Padding(50, 50, 50, 50)`.

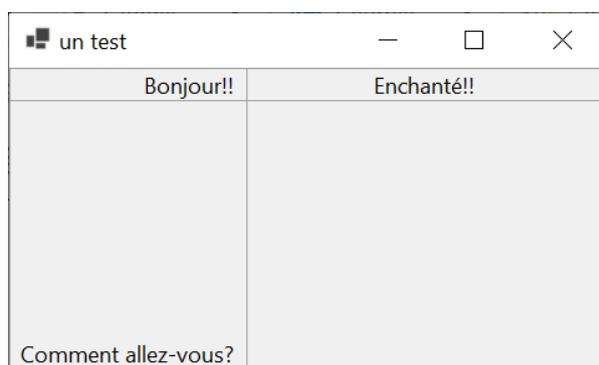
- 4) Changez l'alignement des labels dans le `TableLayoutPanel`. Par exemple, essayez de les aligner à gauche, à droite, en bas, en haut de chaque cellule.

Correction : Il faut utiliser la propriété `Anchor`. Par exemple :

`label1.Anchor = AnchorStyles.Right; //label1 aligné à droite`

`label2.Anchor = AnchorStyles.None; //label2 aligné au milieu`

`label3.Anchor = AnchorStyles.Bottom; //label3 aligné en bas`



- 5) Changez l'espacement entre les labels (suggestion : explorez les propriétés `Padding` et `Margin`).

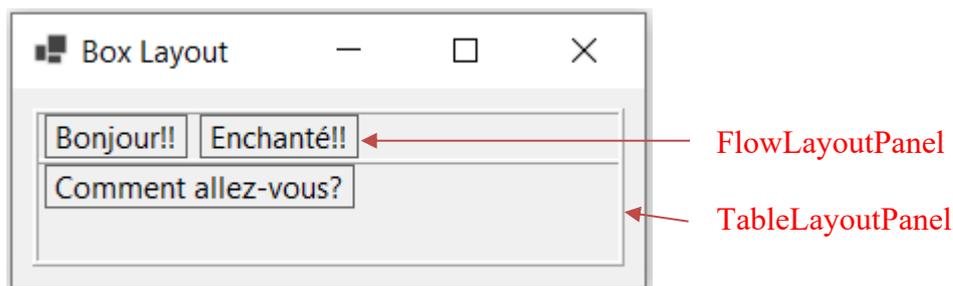
Correction : On peut définir la propriété `Margin` en lui affectant une instance de `Padding` : `label1.Margin = new System.Windows.Forms.Padding(10);`

- 6) Mettez en place le dernier `label3` sur deux colonnes au lieu d'une (le label est placé au milieu des deux colonnes).

Correction :

```
tableLayout.SetColumnSpan(label3,2); //Place label3 au milieu des deux colonnes
```

- 7) Modifiez la méthode `SetupUI()` pour ajouter un `FlowLayoutPanel` qui gère l'emplacement horizontal et `TableLayoutPanel` l'emplacement vertical. Utilisez ces éléments graphiques pour reproduire le résultat suivant :



Correction : Il faut créer un FlowLayoutPanel pour la ligne du haut et y mettre les label. Voir code.

- 8) Changez la « largeur préférée » du `FlowLayoutPanel`, son espacement, son alignement et l'espacement entre ses éléments.

Correction : Il faut manipuler les propriétés Anchor, Width, Height, Margin des éléments dans les panels. Voir code.

Exercice 2

Vous devez développer les menus d'une application d'édition de texte. Selon le cahier des charges, cette application doit inclure les fonctionnalités suivantes :

- Ouvrir, fermer et enregistrer un fichier de texte
- Créer un nouveau fichier ex-nihilo
- Imprimer le fichier
- Insérer des numéros de page
- Compter le nombre de mots ou de caractères dans le texte
- Modifier la police de caractères, leur taille et style (italique, gras, souligné)
- Insérer une note de bas de page
- Copier/Couper/Coller du texte
- Chercher un segment de texte
- Remplacer un segment de texte
- Insérer une image ou dessin
- Insérer un tableau
- Insérer une table des matières
- Quitter l'application
- Vérifier l'orthographe
- Mettre en majuscule ou en minuscule
- Insertion de caractères spéciaux

- Zoomer

1) Pour respecter les heuristiques vues en cours, faut-il rajouter une/des fonctionnalité(s) ?

Correction : Il faut rajouter l'aide (heuristique Nielsen 9) et permettre d'annuler/recommencer pour accroître la tolérance à l'erreur (heuristique Nielsen 3).

2) Groupez ces opérations dans des menus et donnez un intitulé à chaque menu. Ordonnez les items dans chaque menu. Quels sont les principes qui vous ont guidés pour construire cette structure de menus ? Justifiez vos choix d'organisation des items pour les grouper dans les menus.

Correction :

Proposition de menus : Fichier, Edition, Format des caractères, Insertion, Outils

Proposition de regroupement :

- *Fichier : Nouveau, Ouvrir, Fermer, Enregistrer sous ..., Enregistrer, Imprimer, Quitter*
- *Edition : Annuler, Recommencer, Copier, Couper, Coller, Chercher, Remplacer,*
- *Format des caractères : Police, Taille, Style, Majuscule/Minuscule*
- *Insertion : Numéros de pages, note de bas de page, Image, Dessin, Tableau, Table des matières, Caractères spéciaux*
- *Outils : Compter mots, Compter caractères, Vérifier l'orthographe, Zoomer, Aide*

Il faut regrouper les items selon leurs fonctions. Les items dans les menus doivent être spécifique et différencié. Il faut présenter toutes les alternatives possibles d'un choix dans un même menu. Le nombre optimum d'items par menu devraient idéalement être compris entre 4 et 8, il doit rester inférieur à 10. Il est également conseillé de se conformer aux standards, par exemple en s'inspirant d'éditeurs de texte courant (heuristique Nielsen 4).

3) Modifier le contenu de la méthode SetupUI () dans la classe TP4.cs.

```
namespace TP4
{
    public partial class TP4 : Form
    {
        public TP4()
        {
            InitializeComponent();
            SetupUI();
        }

        private void SetupUI()
        {
            this.Text = "Menu Example";
            this.Size = new Size(400, 300);

            // Create MenuStrip
            MenuStrip menuBar = new MenuStrip();

            // Create Menus
```

```

ToolStripMenuItem menu1 = new ToolStripMenuItem("Menu 1");
ToolStripMenuItem menu2 = new ToolStripMenuItem("Menu 2");

// Create Submenu
ToolStripMenuItem subMenu = new ToolStripMenuItem("Sub menu");

// Create Menu Items
ToolStripMenuItem menuItem1 = new ToolStripMenuItem("Item 1");
ToolStripMenuItem menuItem2 = new ToolStripMenuItem("Item 2");

// Menu Item Click Events
menuItem1.Click += (sender, e) => MessageBox.Show("Menu Item1: " +
DateTimeOffset.UtcNow.ToUnixTimeMilliseconds());
menuItem2.Click += (sender, e) => MessageBox.Show("Menu Item2: " +
DateTimeOffset.UtcNow.ToUnixTimeMilliseconds());

// Create Submenu Items
ToolStripMenuItem subMenuItem1 = new ToolStripMenuItem("Sub Menu
Item 1");
ToolStripMenuItem subMenuItem2 = new ToolStripMenuItem("Sub Menu
Item 2");

// Add submenu items
subMenu.DropDownItems.Add(subMenuItem1);
subMenu.DropDownItems.Add(subMenuItem2);

// Add items to menu1
menu1.DropDownItems.Add(menuItem1);
menu1.DropDownItems.Add(menuItem2);
menu1.DropDownItems.Add(subMenu);

// Add menus to menuBar
menuBar.Items.Add(menu1);
menuBar.Items.Add(menu2);

// Create Label and TextBox
Label label1 = new Label { Text = "Bonjour!!", AutoSize = true,
Location = new Point(10, 50) };
TextBox txtArea = new TextBox { Multiline = true, Size = new
Size(360, 150), Location = new Point(10, 80) };

// Add controls to the form
this.Controls.Add(menuBar);
this.Controls.Add(label1);
this.Controls.Add(txtArea);
this.MainMenuStrip = menuBar; // Assign the menu bar as the main
menu
    }
}
}

```

- 4) Exécutez l'application. Qu'est-ce qui se passe quand on sélectionne un élément d'un menu ? Repérez dans le code les appels de méthodes responsable de ce comportement.

Correction : Quand on sélectionne un élément dans le menu, une fenêtre apparaît avec le temps en millisecondes. Les méthodes responsables de ce comportement sont exprimées en lambda expression comme suit :

```
// Menu Item Click Events
```

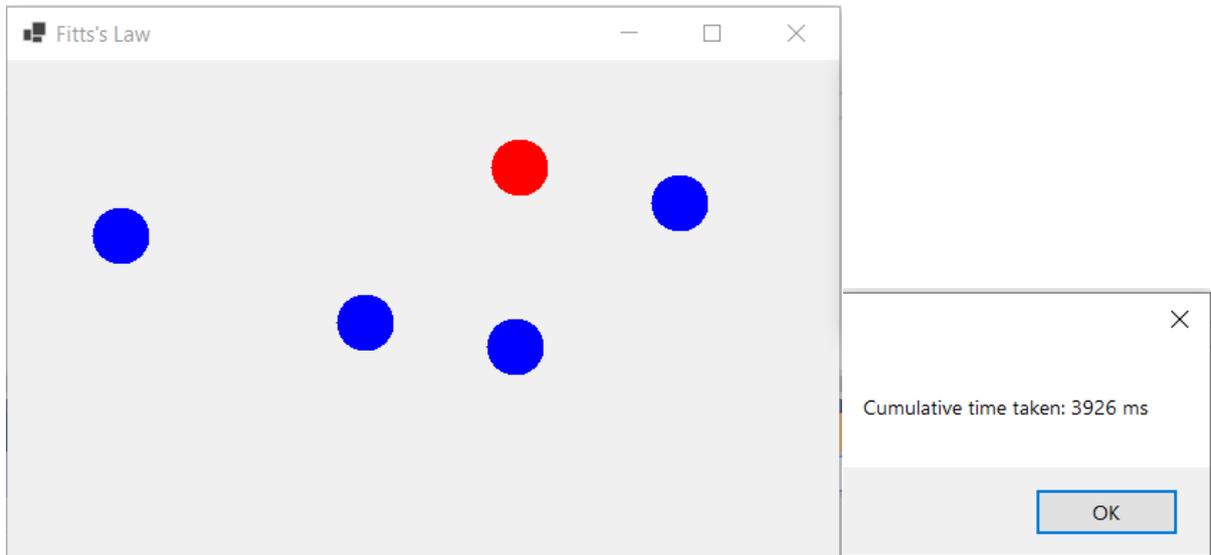
```
menuItem1.Click += (sender, e) => MessageBox.Show("Menu Item1: " +  
DateTimeOffset.UtcNow.ToUnixTimeMilliseconds());  
menuItem2.Click += (sender, e) => MessageBox.Show("Menu Item2: " +  
DateTimeOffset.UtcNow.ToUnixTimeMilliseconds());
```

Lisez attentivement le code et inspirez-vous en pour créer les menus que vous avez définis ci-dessus. Ces menus doivent apparaître sous forme d'une barre de menus et de menus déroulants.

Exercice 3 (facultatif)

Vous devez créer une fenêtre contenant des formes et qui permet de calculer le temps de pointage pour aller d'un point courant à un point rouge.

Exemple de fenêtre :



Pour créer des formes, vous pouvez utiliser des classes telles que `Graphics` et `Brush` avec la méthode `FillEllipse`. La mise à jour de la fenêtre peut être faite par la propriété `Paint` de l'objet de type `Form` à travers un contrôle d'événement `PaintEventHandler()`. De la même manière, le mouvement de la souris peut être suivi par sa propriété `MouseMove` avec un contrôle d'événement `MouseEventHandler()`. Pour le temps, vous pouvez utiliser la classe `Stopwatch`.

Utiliser cette application pour calculer les coefficients de la loi de Fitts et vérifiez-la.