Fiche savoirs : Création des objets graphiques avec C#

Présentation

Les objets graphiques peuvent être créés directement dans le code.

Exemple de l'intérêt de créer des objets :

- groupes d'objets similaires dont la création directe et la gestion seraient fastidieuses,
- objets dont la création dépend d'informations non connues avant l'exécution ,
- etc.

Les objets isolés nommés

Déclarer et créer :

Tout d'abord l'objet doit être déclaré en haut de la classe, en précisant son type.

Il doit être créé avec **new** pour réserver l'espace mémoire nécessaire.

Cette seconde étape concerne tous les objets, graphiques ou non, car ce sont des variables complexes. La création peut être faite dans un module.

Exemple :

```
// déclaration du bouton
Button btnUnique;
// création du bouton
btnUnique = new Button();
```

Intégrer dans une zone d'affichage :

Pour que l'objet soit visible, il faut l'intégrer :

- soit directement dans le formulaire,
- soit dans un conteneur (un groupe de contrôles, un panel, etc.).

```
// ajout du bouton dans la zone d'affichage
Controls.Add(btnUnique);
```

Gérer la taille et la position :

Par défaut l'objet aura une taille standard et sera positionné dans le coin haut gauche de son conteneur.

Il est possible de modifier ces caractéristiques.

```
// position et dimension
btnUnique.Location = new Point(12, 12);
btnUnique.Size = new Size(75, 23);
```

Manipuler l'objet

Comme n'importe quel objet graphique créé directement dans l'interface, il est possible de manipuler l'objet dans le code : gérer les propriétés et les méthodes.

```
// autres propriétés
btnUnique.TabIndex = 1;
btnUnique.Text = "Unique";
```

Créer des écouteurs sur les événements

Contrairement aux objets graphiques créés directement dans l'interface, où les écouteurs d'événements sont générés automatiquement dans le code du Designer, il faut ici écrire le code directement dans la classe de travail du formulaire. N'oubliez pas que les 2 classes sont par "partielles" et qu'elles ne représentent qu'une seule classe.

```
// écouteur sur l'événement clic
btnUnique.Click += new EventHandler(btnUnique_Click);
```

==== Créer les procédures événementielles ====

Une fois l'écouteur créé, il est ensuite possible d'écrire les procédures événementielles correspondant aux écouteurs. Il faut bien sûr les écrire soi-même, en respectant la syntaxe officielle, en particulier au niveau des paramètres, car elles ne sont pas générées automatiquement.

```
private void btnUnique_Click(object sender, EventArgs e)
{
  txtMessage.Text = "Coucou";
}
```

Les objets groupés non nommés

Les objets ne sont pas nommés et seront repérés par rapport au groupe dans lequel ils se trouvent. Comme il est tout de même obligatoire de donner un nom au moment de la création, le même nom sera utilisé pour tous les objets, créés dans une même boucle.

Comme le montre l'exemple ci-dessous, tout est fait dans la boucle : déclaration, création, ajout dans le groupe, positionnement et taille, paramétrage et ajout d'écoute d'événement.

Déclarer, créer et configurer

Voici un exemple de module qui va générer plusieurs boutons radios, insérés dans un groupe, et comportant comme texte, les 10 premières lettres de l'alphabet.

```
private void CreerPlusieursObjets()
{
   for (int k = 0 ; k < 10 ; k++)
   {
      // déclaration et création d'un bouton radio
      RadioButton btr = new RadioButton();
      // ajout du bouton radio dans le groupe
      grbChiox.Controls.Add(btr);
      // position
      btr.Location = new Point(10, 19 + 23*k);
      // texte
      btr.Text = ((char)('A' + k)).ToString();
      // écouteur sur l'événement clic
      btr.Click += new EventHandler(btr_Click);
   }
}
```

Créer les procédures événementielles

La procédure événementielle est unique pour tous les objets précédemment créés. Pour récupérer des caractéristiques d'un objet en particulier, il faut utiliser le paramètre sender qui contient l'objet.

```
private void btr_Click(object sender, EventArgs e)
{
   // récupération de l'objet concerné par le clic
   RadioButton chbLettre = (RadioButton)sender;
   // Affiche la lettre dans la zone de texte
   txtMessage.Text = chbLettre.Text;
}
```

En dehors de la boucle, il sera toujours possible d'accéder à l'objet en connaissant son numéro d'ordre dans le groupe. Il est même possible d'avoir une autre approche de création, suivant les besoins. Par exemple, on peut décider d'ajouter un objet au groupe, à chaque clic sur un

bouton. Dans tous les cas, le groupe fonctionne un peu comme un tableau.

From:
/ - **Les cours du BTS SIO**

Permanent link:
/doku.php/bloc1/codeobjetgraphique?rev=1639139799

Last update: **2021/12/10 13:36**