

Etape 1 : ouvrir un fichier image et afficher l'image

Voir séance précédente

Etape 2 : Visionner les images d'un répertoire

Nouveaux composants :

- prévoir un **FolderBrowserDialog** **ouvreRepertoire**
- un **TreeView** **arbreImages** dans lequel les éléments du répertoire seront stockés.
- ajouter un sous-menu « Répertoire » au **menuStrip** **menu**.
- ajouter un menu « Diaporama ».
- Ajouter un **timer** permettant de régler le délai d'affichage de chaque image durant le diaporama.

2.1 Stockage du répertoire

En réponse à la sélection de l'item **repertoire**, on affichera la boîte de sélection de répertoire (**ouvreRepertoire**) à l'aide de la méthode **ShowDialog**.

A la fermeture par le bouton OK de ce sélecteur, la propriété **SelectedPath** du composant **FolderBrowserDialog** contient le chemin du répertoire sélectionné.

Il faut donc charger le répertoire dans le **TreeView** après avoir réinitialisé ce dernier (méthode **Clear** de la propriété **Nodes** de **arbreImages**), puis fabriquer un sommet racine (**TreeNode** **root**) avec le répertoire sélectionné et ajouter ce sommet à l'arbre par un appel de la forme **arbreImages.Nodes.Add(root)**.

Il reste à parcourir récursivement les sous répertoires et les fichiers (ne garder que les fichiers jpg) pour les ajouter à l'arbre. L'ajout d'un noeud fils à un sommet se fait comme pour la racine par un appel de la forme **node.Nodes.Add(noeud)**.

Notes :

- **Directory.GetDirectories(repertoire)** : fournit un tableau de chaînes contenant les sous répertoires d'un répertoire.
- **Directory.GetFiles(repertoire)** : fournit un tableau contenant les fichiers d'un répertoire.

L'utilisation de la structure de contrôle **foreach** permet de parcourir aisément ces tableaux. On pourra donc écrire du code tel que :

```
string[] fichiers = Directory.GetFiles(repertoire);
foreach (string nomFichier in fichiers)
{
    if (isJpegFile (nomFichier))
    {
        string nomF = nomFichier.Substring(nomFichier.LastIndexOf(@"\" )+1);
        TreeNode noeud = new TreeNode(nomF);
        node.Nodes.Add(noeud);
    }
}
```

Recommandation : on écrira au préalable l'algorithme de parcours de l'arborescence et de construction du **TreeView** en prévoyant un appel récursif pour chaque sous dossier.

Pour réaliser un tel parcours, on peut par exemple écrire une fonction récursive **Charger_Repertoire(TreeNodeCollection, string)**, sachant qu'un **TreeNodeCollection** est une collection de **TreeNode** (eh oui :-); c'est le type de la propriété **Nodes** de **TreeNode**.

2.2 Affichage d'une image sélectionnée à partir du *TreeView*

Lorsque vous travaillez avec le contrôle **TreeView**, il vous arrive fréquemment de devoir identifier le nœud sur lequel l'utilisateur a cliqué et de donner une réponse adéquate. On associera un traitant d'évènement à la propriété **AfterSelect**. Pour identifier le nœud de **TreeView** sur lequel l'utilisateur a cliqué, utilisez l'objet **EventArgs** **e** qui retourne une référence à l'objet de nœud cliqué. **e.Node.FullPath** fournit alors le chemin d'accès du nœud d'arbre en cours.

Notes

*Le chemin d'accès est constitué des étiquettes de tous les nœuds d'arbre qui doivent être explorés pour obtenir ce nœud d'arbre en commençant au nœud d'arbre racine. Les étiquettes de nœud sont séparées par le caractère délimiteur spécifié dans la propriété PathSeparator du contrôle TreeView qui contient le nœud. Par exemple, si le caractère délimiteur du contrôle TreeView appelé « Location » est la barre oblique inverse (\), la valeur de la propriété **FullPath** est « Country\Region\State ».*

2.3 Parcours du *TreeView*, diaporama

Le parcours de l'arbre peut-être réalisé en utilisant les méthodes ou attributs suivants :

- **arbreImages.Nodes[0]** fournit le nœud racine,
- **node.GetNodeCount(true)** fournit le nombre de fils d'un nœud,
- **node.FirstNode** fournit le premier fils ,
- **node.NextNode** fournit le nœud frère ,
- **node.Parent** fournit le nœud père.

Chaque image trouvée sera affichée dans la **PictureBox**, la première dès le lancement du diaporama, les autres à chaque évènement du **timer**.

Recommandation : une première étape consistera à écrire l'algorithme de parcours d'un arbre en utilisant les éléments cités ci-dessus et en se limitant au premier niveau de l'arbre (uniquement les fichiers jpeg de la racine). A partir de cet algorithme, on pourra écrire la partie permettant l'affichage de la première image puis la partie : affichage de l'image suivante. On aura pris soin d'ajouter dans notre classe un attribut de type **TreeNode** permettant de mémoriser le noeud de l'arbre correspondant à l'image affichée. Une deuxième étape pourra consister à généraliser le parcours à l'ensemble de l'arbre.