

TP3 : Création d'un bloc note

But du TP

Le but du TP d'aujourd'hui est de créer une fenêtre qui permet d'afficher un texte contenu dans un fichier.

Tout d'abord, il faut organiser un peu mieux l'affichage de notre fenêtre. Pour cela, créez une classe **Fenetre** qui permette de bien gérer la fenêtre. Vous intégrerez alors la fonction **MainProc()** à cette classe pour qu'elle gère les événements par défaut (en gros, vous recopiez le contenu de la fonction **MainProc** qui est sur le site...). Vous devez avoir une classe **Fenetre** de ce type :

```
class Fenetre{
public:
    Fenetre();
    void afficher();
    void quitter();
    LRESULT CALLBACK MainProc(HWND,UINT,WPARAM,LPARAM);

protected:
    HWND identifiantFenetre;
}
```

Une fois le code des fonctions créé, définissez une autre classe qui hérite de cette classe et qui s'appellera **FenetreNotePad**. Cette classe ne redéfinira que les fonctions nécessaires. Par exemple, pour la fonction **MainProc**, on ne traitera que les nouveaux événements. Pour les autres, on appellera la fonction de la classe mère (en faisant `Fenetre::MainProc(...)`).

Partie 1 : ouverture du fichier...

En vous rappelant de vos cours de l'année dernière, créez une fonction qui permet de charger un fichier dans un attribut de type string (vous aurez donc une variable de type string dans la classe **FenetreNotePad**). Vous nommerez la fonction : `void openFic(string nomFichier)`.

Rappel : vous aurez besoin de deux includes en plus :

```
<fstream.h>
```

```
<iostream.h>
```

Vous pourrez ensuite ouvrir un fichier avec : `ifstream fluxE("C:\\MesDocs\\Cpp\\test.txt");`

Partie 2 : choix du fichier...

Heureusement pour vous, vous n'aurez pas à créer entièrement une fenêtre de choix... Le code pour ouvrir une fenêtre qui propose de choisir un fichier est sur la page suivante, mais pour plus d'information, aller voir sur <http://msdn2.microsoft.com/en-us/library/ms646960.aspx>.

```

OPENFILENAME ofn;          // boite de recherche de fichier
char fich[260];           // nom du fichier
string nomfichier;        // nom du fichier (mais de type string...)

// Initialisation de la fenetre de dialogue
ZeroMemory(&ofn, sizeof(ofn));
ofn.lStructSize = sizeof(ofn);
ofn.lpstrFile = fich;     // ici, on dit que c'est cette chaine qui
// contiendra le chemin vers le fichier

// on met lpstrFile[0] à '\0' pour que GetOpenFileName ne cherche pas
//ce fichier
// On voit donc que si on met un fichier qui existe, on démarrera sur
//le dossier qui le contient !
ofn.lpstrFile[0] = '\0';

//ici, il faut mettre la taille max du chemin...
ofn.nMaxFile = 260;

//ici, c'est un filtre. On peut afficher tout les fichiers,
// mais aussi que ceux qui finissent par .txt
ofn.lpstrFilter = "Tous\0*.*\0texte\0*.txt\0";
ofn.nFilterIndex = 2;

//ici, on dit qu'il faut que le fichier existe...
ofn.Flags = OFN_FILEMUSTEXIST;

// affiche la dialog box Open.

if (GetOpenFileName(&ofn)){//si on n'a pas choisi de fichier
    nomfichier.clear();
    nomfichier=fich;
}

```

Avec la variable `nomfichier`, il est alors possible d'ouvrir le fichier que l'utilisateur a spécifié en utilisant la fonction prédéfinie.

Vous créez alors un bouton qui permettra de choisir un fichier, de le charger, puis de mettre le contenu dans un edit box. Pensez à autoriser l'écriture sur plusieurs lignes et la scroll bar verticale (dans les options du edit box...).

Partie 3 : l'enregistrement

Vous rajouterez un bouton permettant d'enregistrer le contenu dans un fichier. Vous utiliserez le même code que pour ouvrir un fichier, mais vous modifierez cette ligne :

```
ofn.Flags = OFN_FILEMUSTEXIST;
```

```
en mettant ofn.Flags = OFN_OVERWRITEPROMPT;
```

Ensuite, la fonction à appeler devient `GetSaveFileName`.