Bien sûr ! Je vais expliquer chaque partie du code en détail pour que vous puissiez comprendre comment il fonctionne. Ce code crée une application Windows en C qui permet de gérer une liste d'étudiants et d'enseignants. Vous pouvez ajouter, supprimer et afficher le nombre d'étudiants et d'enseignants.

**1. Inclusion des en-têtes**

c

Copy

#include <windows.h>

#include <tchar.h>

#include <stdio.h>

* **Explication** :
  + windows.h : Fournit les fonctions et types nécessaires pour créer des applications Windows.
  + tchar.h : Permet d'utiliser des chaînes de caractères compatibles avec Unicode et ANSI.
  + stdio.h : Fournit des fonctions standard d'entrée/sortie (utilisé ici pour sprintf).

**2. Constantes pour les ID des contrôles**

c

Copy

#define ID\_STUDENT\_LIST 1

#define ID\_TEACHER\_LIST 2

#define ID\_ADD\_STUDENT 3

#define ID\_ADD\_TEACHER 4

#define ID\_DELETE\_STUDENT 5

#define ID\_DELETE\_TEACHER 6

#define ID\_STUDENT\_COUNT 7

#define ID\_TEACHER\_COUNT 8

#define ID\_FIRST\_NAME 9

#define ID\_LAST\_NAME 10

* **Explication** :
  + Ces constantes sont utilisées pour identifier les contrôles de l'interface (listes, boutons, champs de texte, etc.).
  + Par exemple, ID\_STUDENT\_LIST est l'identifiant de la liste des étudiants.

**3. Structure Person**

c

Copy

typedef struct {

TCHAR firstName[100];

TCHAR lastName[100];

} Person;

* **Explication** :
  + Cette structure stocke les informations d'une personne (étudiant ou enseignant).
  + Elle contient deux champs : firstName (prénom) et lastName (nom).

**4. Variables globales**

c

Copy

Person students[100];

Person teachers[100];

int studentCount = 0;

int teacherCount = 0;

* **Explication** :
  + students et teachers : Tableaux pour stocker les étudiants et les enseignants.
  + studentCount et teacherCount : Compteurs pour le nombre d'étudiants et d'enseignants.

**5. Fonction principale WinMain**

c

Copy

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {

* **Explication** :
  + WinMain est le point d'entrée de l'application Windows.
  + Elle initialise la fenêtre et entre dans une boucle de messages pour gérer les interactions utilisateur.

**6. Initialisation de la classe de la fenêtre**

c

Copy

WNDCLASSEX wc;

wc.cbSize = sizeof(WNDCLASSEX);

wc.style = 0;

wc.lpfnWndProc = WndProc;

wc.hInstance = hInstance;

wc.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

wc.hCursor = LoadCursor(NULL, IDC\_ARROW);

wc.hbrBackground = (HBRUSH)(COLOR\_WINDOW + 1);

wc.lpszClassName = TEXT("MyWindowClass");

* **Explication** :
  + WNDCLASSEX est une structure qui définit les propriétés de la fenêtre.
  + lpfnWndProc : Pointeur vers la fonction WndProc qui gère les messages.
  + lpszClassName : Nom de la classe de la fenêtre.

**7. Création de la fenêtre**

c

Copy

HWND hwnd = CreateWindowEx(0, TEXT("MyWindowClass"), TEXT("Gestion des Étudiants et Enseignants"),

WS\_OVERLAPPEDWINDOW, CW\_USEDEFAULT, CW\_USEDEFAULT, 600, 400, NULL, NULL, hInstance, NULL);

* **Explication** :
  + CreateWindowEx crée la fenêtre avec un titre, une taille et un style spécifiés.
  + La fenêtre est affichée avec ShowWindow et mise à jour avec UpdateWindow.

**8. Boucle de messages**

c

Copy

while (GetMessage(&Msg, NULL, 0, 0) > 0) {

TranslateMessage(&Msg);

DispatchMessage(&Msg);

}

* **Explication** :
  + La boucle de messages récupère les messages (comme les clics de souris ou les touches du clavier) et les envoie à la fonction WndProc pour traitement.

**9. Fonction WndProc**

c

Copy

LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam) {

* **Explication** :
  + Cette fonction gère les messages envoyés à la fenêtre.
  + Par exemple, elle traite les clics sur les boutons et les actions de l'utilisateur.

**10. Création des contrôles**

c

Copy

CreateWindow(TEXT("STATIC"), TEXT("Prénom:"), WS\_VISIBLE | WS\_CHILD, 10, 10, 100, 20, hwnd, NULL, NULL, NULL);

CreateWindow(TEXT("EDIT"), TEXT(""), WS\_VISIBLE | WS\_CHILD | WS\_BORDER, 120, 10, 200, 20, hwnd, (HMENU)ID\_FIRST\_NAME, NULL, NULL);

* **Explication** :
  + CreateWindow crée des contrôles comme des étiquettes (STATIC), des champs de texte (EDIT), des boutons (BUTTON) et des listes (LISTBOX).
  + Chaque contrôle a une position, une taille et un identifiant unique.

**11. Ajout d'une personne**

c

Copy

void AddPerson(Person\* list, int\* count, HWND hwnd, int listID, int countID) {

TCHAR firstName[100], lastName[100];

GetWindowText(GetDlgItem(hwnd, ID\_FIRST\_NAME), firstName, 100);

GetWindowText(GetDlgItem(hwnd, ID\_LAST\_NAME), lastName, 100);

if (\_tcslen(firstName) > 0 && \_tcslen(lastName) > 0) {

\_tcscpy(list[\*count].firstName, firstName);

\_tcscpy(list[\*count].lastName, lastName);

(\*count)++;

TCHAR fullName[200];

\_stprintf(fullName, TEXT("%s %s"), firstName, lastName);

SendDlgItemMessage(hwnd, listID, LB\_ADDSTRING, 0, (LPARAM)fullName);

UpdateCount(hwnd, countID, \*count);

// Effacer les champs de saisie

SetWindowText(GetDlgItem(hwnd, ID\_FIRST\_NAME), TEXT(""));

SetWindowText(GetDlgItem(hwnd, ID\_LAST\_NAME), TEXT(""));

}

}

* **Explication** :
  + Récupère le prénom et le nom des champs de texte.
  + Ajoute la personne à la liste correspondante (étudiants ou enseignants).
  + Affiche le nom complet dans la liste.
  + Met à jour le compteur et efface les champs de saisie.

**12. Suppression d'une personne**

c

Copy

void DeletePerson(Person\* list, int\* count, HWND hwnd, int listID, int countID) {

int index = SendDlgItemMessage(hwnd, listID, LB\_GETCURSEL, 0, 0);

if (index != LB\_ERR) {

SendDlgItemMessage(hwnd, listID, LB\_DELETESTRING, index, 0);

for (int i = index; i < \*count - 1; i++) {

list[i] = list[i + 1];

}

(\*count)--;

UpdateCount(hwnd, countID, \*count);

}

}

* **Explication** :
  + Supprime la personne sélectionnée dans la liste.
  + Met à jour le tableau et le compteur.

**13. Mise à jour du compteur**

c

Copy

void UpdateCount(HWND hwnd, int countID, int count) {

TCHAR text[100];

if (countID == ID\_STUDENT\_COUNT) {

\_stprintf(text, TEXT("Nombre d'étudiants: %d"), count);

} else {

\_stprintf(text, TEXT("Nombre d'enseignants: %d"), count);

}

SetWindowText(GetDlgItem(hwnd, countID), text);

}

* **Explication** :
  + Met à jour le texte affichant le nombre d'étudiants ou d'enseignants.