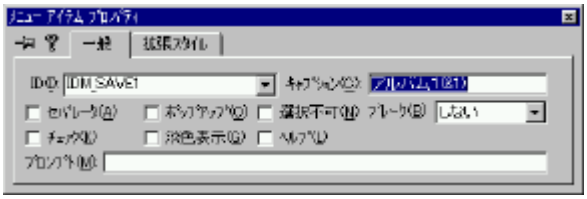
ゲームプログラミング

▶　第１０回　メニュー　◀

ほとんどのアプリケーションやウィンドウモードで動作するゲームについているメニューを作成してみましょう。

メニューの作成

メニューは、VisualC++の「挿入(I)」→「ﾘｿｰｽ(R)」から、「Menu」を選択して「新規作成(N)」ボタンをクリックすると作成することができます。
メニューの項目は、「ﾒﾆｭｰ ｱｲﾃﾑ ﾌﾟﾛﾊﾟﾃｨ」ダイアログで編集することができます。



・ID(I)…メニュー項目のIDです。メニューが選択されると、WM_COMMANDメッセージが発生し、このIDが追加情報WPARAMの下位１６ビットに格納されます。

・ｷｬﾌﾟｼｮﾝ(C)…メニューの項目として表示される文字列です。"&"記号に英数字をつづけて入力し、カッコ"("で括ると、キーボードから、「Altキー→その英数字のキー」でメニューを選択できるようになります。
　たとえば、「ﾛｰﾄﾞ(&L)」と入力すると、Alt→"L"キーと押すことでこの項目を選択できるようになります。

・ｾﾊﾟﾚｰﾀ(A)…ここにチェックを入れると、メニュー項目に区切り線が挿入されます。

・ﾎﾟｯﾌﾟｱｯﾌﾟ(O)…ここにチェックを入れると、メニュー項目の右側に▶が表示され、サブメニューを作成できるようになります。

・選択不可(N)…ここにチェックを入れたメニュー項目は選択できないようになります。

・ﾁｪｯｸ(K)…ここにチェックを入れると、メニュー項目の左側に"√"がつきます。

・淡色表示(G)…ここにチェックを入れたメニュー項目は淡色表示になり、選択できなくなります。

メニューの読み込み

メニューを読み込んでウィンドウに表示させるには、ウィンドウクラスの定義時か、ウィンドウ生成時に指定するかの方法で行うことができます。
ウィンドウクラスの定義時にメニューを指定する場合は、WNDCLASSEX構造体のlpszMenuNameメンバを使います。このメンバに読み込むメニューのリソース名を文字列で指定します。リソース名が整数の場合はMAKEINTRESOURCEマクロを使って変換します。

メニューの名前が文字列の場合：wcex.lpszMenuName = "IDR_MENU1";
メニューの名前が整数の場合：wcex.lpszMenuName = MAKEINTRESOURCE(IDR_MENU1);
※WNDCLASSEX構造体変数名がwcexの場合

ウィンドウの生成時にメニューを指定する場合は、LoadMenu関数でメニューを読み込んでメニューのハンドルを取得し、このハンドルをCreateWindowEx関数の１０番目の引数に渡します。

```
HWND   hWnd = CreateWindowEx(0, MAINWND_CLSNAME, "Windowsプログラミング",
                             WS_OVERLAPPEDWINDOW,
                             CW_USEDEFAULT, CW_USEDEFAULT,
                             CW_USEDEFAULT, CW_USEDEFAULT,
                             HWND_DESKTOP, LoadMenu(hInstance, MAKEINTRESOURCE(IDR_MENU1)),
                             hInstance, NULL);
```

※メニュー名が整数の場合。メニュー名が文字列の場合はLoadMenu(hInstance, "IDR_MENU1") です。

ウィンドウに表示されているメニューは、ウィンドウを破棄するときに自動的に解放されますが、それ以外のメニューが読み込まれている場合は、DestroyMenu関数で解放する必要があります。

メニューの処理

選択されたメニューを処理するには、メニューを選択したときに発生するメッセージを利用します。メニューを選択するとWM_COMMANDメッセージが発生し、追加情報WPARAMの３２ビットのうち下位１６ビットに選択したメニューのIDが格納されます。上位１６ビットにも別な情報が格納されますが通常は使用しません。下位１６ビットを取り出すには、LOWORDマクロを使用します。

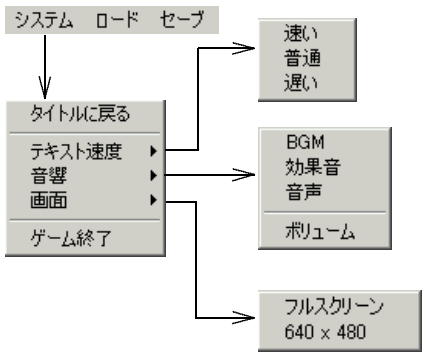
メニューを操作するAPI

メニューはAPIで必要に応じていろいろな変更を加えることができます。メニューを変更したり、メニューの項目に√を入れたり、項目を選択不可にしたりといったことができます。

API名	動　　作
AppendMenu	新しい項目をメニューに付加します
CheckMenuItem	メニュー項目のチェックマークの状態を変更します
DestroyMenu	メニューを破棄してメモリを解放します
DrawMenuBar	メニューバーを再描画します
EnableMenuItem	メニュー項目を使用可能、使用不可、灰色表示にします
GetMenu	使用しているメニューのハンドルを取得します
GetMenuItemCount	メニュー内の項目数を取得します
GetMenuState	指定されたメニュー項目の状態を取得します
LoadMenu	メニューリソースを読み込みます
ModifyMenu	メニュー項目を変更します
SetMenu	メニューバーを変更します

○ 課題 ○

(１)以下のメニューを作成し、メインウィンドウに付加しましょう。メニューのリソース名は"IDR_MAIN_MENU"とします。



※メニューのＩＤは以下のように設定しましょう

ロード…IDM_LOAD　　ボリューム…
セーブ…IDM_SAVE　　IDM_AD_VOLUME

タイトルに戻る…IDM_TITLE　　フルスクリーン…
ゲーム終了…IDM_EXIT　　IDM_SCR_FULL
640 x 480…IDM_SCR_WND

速い…IDM_TXT_FAST
普通…IDM_TXT_NORMAL
遅い…IDM_TXT_SLOW

ＢＧＭ…IDM_AD_BGM
効果音…IDM_AD_SE
音声…IDM_AD_VOICE

※メニューを付加したウィンドウにAdjustWindowRectEx関数を使用する場合、３つ目の引数をTRUEにしないと正しいサイズが得られません。

(２)「テキスト速度」メニューの「速い」「普通」「遅い」のいずれかが選択されたら、√がつくようにしましょう。ただし、√は選択された項目だけにつくものとします。

(３)「音響」メニューの「ＢＧＭ」「効果音」「音声」のそれぞれについて、選択された項目に√がつくようにしましょう。すでに√がついている場合は、√を消すようにしましょう。

(４)「画面」メニューの「フルスクリーン」「640 x 480」のいずれかが選択されたら、√がつくようにしましょう。ただし、√は選択された項目だけにつくものとします。

(５)メインウィンドウが生成されたとき、「テキスト速度」メニューの「普通」、音響メニューの「ＢＧＭ」「効果音」「音声」、「画面」メニューの「640 x 480」に√がついている状態にAPIを使って設定しましょう。また、「フルスクリーン」を灰色表示にして選択できないようにしましょう。

・ヒント

```
// メニューアイテム処理
case WM_COMMAND:
   HMENU   hMenu;
   hMenu   = ???????(????);   // メニューハンドル取得
   UINT    uMenuID;
   uMenuID = ?????????????;  // メニューID取得

  switch(uMenuID) {
    // テキスト速度設定
    case IDM_TXT_FAST:
    case IDM_TXT_NORMAL:
    case IDM_TXT_SLOW:
       ※ひみつ１(チェックをはずす１)
       ※ひみつ２(チェックをはずす２)
       ※ひみつ３(チェックをはずす３)
       ※ひみつ４(チェックをする)
       break;

    case IDM_AD_BGM:
    case IDM_AD_SE:
    case IDM_AD_VOICE:
      // 音響設定
       ※ひみつ５(メニューがチェックされているかを調べる)
            ※ひみつ６(チェックをはずす)
       else
            ※ひみつ７(チェックをする)
       break;

    case IDM_SCR_FULL:
    case IDM_SCR_WND:
       // スクリーン設定
       ※ひみつ８(テキスト速度と同パターン)
       ※ひみつ９
       ※ひみつ10
       break;
  }   // switch(uMenuID)
  return 0;
```