ゲームプログラミング

▶　基礎編－第４回　プロジェクトの構成　◀

ゲームプログラミング講座で作成するサンプルゲームのファイル構成について解説します。

サンプルゲームのファイル構成

ゲームプログラミング講座では、最初はＣ言語をベースに、一部Ｃ++を組み込んだプログラム開発を行います。Ｃ言語では、作成するアプリケーションを機能ごとに分割し、分割した機能を「関数」として定義していきます。すべての関数を１つのソースファイルに記述するのではなく、関連のある関数だけをまとめ、複数のファイルに分割して管理します。このようにすると、一部の機能をバージョンアップしたり、差し替えたりすることが簡単に行えます。
ゲームプログラミング講座では、プロジェクトに以下のファイルを作成します。

－ソースファイル－

WinMain.cpp	WinMainをはじめとしたウィンドウ関連の処理を行う関数をまとめたもの
DDUtils.cpp	DirectDrawの機能を呼び出すための関数をまとめたもの
DXAUtils.cpp	DirectX Audioの機能を呼び出すための関数をまとめたもの
DIUtils.cpp	DirectInputの機能を呼び出すための関数をまとめたもの
DSUtils.cpp	DirectShowの機能を呼び出すための関数をまとめたもの
GameFunc.cpp	ゲームの処理を行う関数をまとめたもの
Error.cpp	エラーがおきたときの処理を行う関数をまとめたもの

－ヘッダファイル－

DDUtils.h	DDUtils.cpp で定義した関数を呼び出す時に必要なヘッダファイル
DXAUtils.h	DXAUtils.cppで定義した関数を呼び出す時に必要なヘッダファイル
DIUtils.h	DIUtils.cpp で定義した関数を呼び出す時に必要なヘッダファイル
DSUtils.h	DSUtils.cpp で定義した関数を呼び出す時に必要なヘッダファイル
GameFunc.h	GameFunc.cppで定義した関数を呼び出す時に必要なヘッダファイル
Error.h	Error.cpp　で定義した関数を呼び出す時に必要なヘッダファイル

－リソースファイル－

Resource.rc	作成したリソースを定義されたスクリプトファイル
resource.h	作成したリソースをプログラムで利用するためのヘッダファイル
Icon.ico	アイコンのイメージファイル

ソースファイルの構成

ソースファイル(.cpp)には、実現したい処理を関数に分割ながら定義していきます。ゲームプログラミング講座で作成するソースファイルは、基本的に以下のような構成になります。

```
/*
================================================================================================
                                ゲームプログラミング
    Programmed by Hibikino software. Copyright (c) 2003 Hibikino software. All rights reserved.
================================================================================================

  【対象ＯＳ】
     Microsoft Windows98/2000以降

  【コンパイラ】
     Microsoft VisualC++ 6.0J ServicePack5

  【プログラム】
     Sample.cpp
                サンプルソースファイル

  【履歴】
     * Version    0.00       2003/04/dd  hh:mm:ss  初版

================================================================================================
*/
```

```
/*****************************************************************************************************/
/*                                          インクルードファイル                                     */
/*****************************************************************************************************/
#include "Sample.h"

/*****************************************************************************************************/
/*                                          スタティックライブラリ                                   */
/*****************************************************************************************************/
#pragma comment(lib, "Sample.lib")

/*****************************************************************************************************/
/*                                          定　　数                                                 */
/*****************************************************************************************************/
const double   FRAME_DIRAY = 1000.0 / FPS;  // １フレーム周期

/*****************************************************************************************************/
/*                                          外部参照変数                                             */
/*****************************************************************************************************/
extern int   g_nAppMode;                    // アプリケーションモード

/*****************************************************************************************************/
/*                                          外部公開変数                                             */
/*****************************************************************************************************/
int   g_nSampleState;                       // サンプル関数状態

/*****************************************************************************************************/
/*                                          グローバル変数                                           */
/*****************************************************************************************************/
static int   g_nCnt = 0;                    // カウンタ

/*****************************************************************************************************/
/*                                          プロトタイプ(プライベート)                               */
/*****************************************************************************************************/
static void WorkFunc();

/*****************************************************************************************************/
/*                                          サンプル関数                                             */
/*****************************************************************************************************/
int SampleFunc()
{
    return 0;
}

/*****************************************************************************************************/
/*                                          作業用関数                                               */
/*****************************************************************************************************/
void WorkFunc()
{
}
```

「ファイル見出し」(コメント)は必ず何らかの形で入れておきましょう。ソースファイルを表示したとき、どのような関数をまとめたものなのか、すぐに確認できます。

「インクルードファイル」は、ソースファイルをコンパイルするのに必要なヘッダファイルを記述します。Sample.cppでは、Sample.hヘッダで必要なものをインクルードしているので、それ以外のヘッダは必要ありませんが、そうでない場合はここに追加します。

「スタティックライブラリ」は、リンクに必要なライブラリ(.lib)をプリプロセッサ「#pragma comment(lib, "ライブラリ名")」を使って指定します。

「定数」は、このソースファイル内だけで使う定数を記述します。ほかのソースファイルでも必要になる定数は、ヘッダファイルに記述します。プロトタイプ宣言や構造体定義などでもそうですが、ほかのソースファイルで使用しないものは、ヘッダファイルに記述する必要はありません。また、定数はC言語の場合は"#define"を使用しますが、C++の"const"を使用するようにします。この方が、型チェックが行われ、カッコで全体を囲まなくても計算順序が狂ったりしないので、こちらを使った方が安全です。

「外部参照変数」は、ほかのソースファイルで宣言されている変数を参照するときに記述します。"extern"指定すると、他のソースファイルで宣言されているグローバル変数を参照することができます。

「外部公開変数」は、ほかのソースファイルで参照しなければならない変数を記述します。

「グローバル変数」は、このソースファイル内だけで有効な変数を定義します。ソースファイル内で(関数を超えて)どうしても共有しなければならない変数を記述します。"static"指定したグローバル変数は、ほかのソースファイルから参照することができません。また、他のソースファイルに同じ名前の

変数が存在しても、別の変数とみなされるので、ほかのソースファイルと変数名が同じかどうかを気にする必要がなくなります。

「プロトタイプ(プライベート)」は、ほかのソースファイルに公開しない関数(このソースファイル内だけで使用する関数)のプロトタイプを記述します。"static"指定で宣言されたプロトタイプは、他のソースファイルから呼び出すことができなくなります。

最後は「関数本体の定義」です。ソースファイルでもっとも重要な部分で、関数本体を記述します。

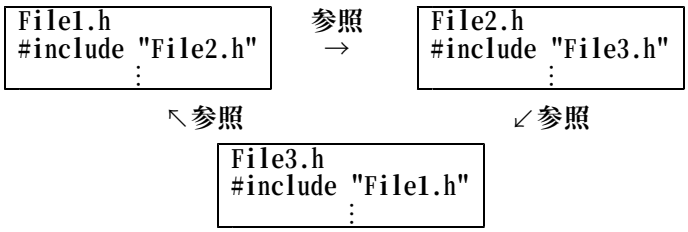
ヘッダファイルの構成

ヘッダファイル(.h)は、さまざまなソースファイルから呼び出される関数のプロトタイプ宣言、構造体や定数を定義します。ゲームプログラミング講座で作成するヘッダファイルは、基本的に以下のような構成になります。

```
/*
================================================================================================
                                        ゲームプログラミング
    Programmed by Hibikino software. Copyright (c) 2003 Hibikino software. All rights reserved.
================================================================================================

  【対象ＯＳ】
     Microsoft Windows98/2000以降

  【コンパイラ】
     Microsoft VisualC++ 6.0J ServicePack5

  【プログラム】
     Sample.h
                サンプルヘッダファイル

  【履歴】
     * Version    0.00       2003/04/dd  hh:mm:ss  初版

================================================================================================
*/

#pragma once

/************************************************************************************************/
/*                                        インクルードファイル                                  */
/************************************************************************************************/
#include <windows.h>

/************************************************************************************************/
/*                                          定　数                                              */
/************************************************************************************************/
const DWORD   FPS = 60;      // フレームレート

/************************************************************************************************/
/*                                        列挙体定義                                            */
/************************************************************************************************/
enum SURFACE {
    PRIMARY,                 // プライマリサーフェイス
    BACKBUF                  // バックバッファ
};

/************************************************************************************************/
/*                                        構造体定義                                            */
/************************************************************************************************/
// 移動方向情報
struct MOVEINFO {
    int   nLeft;     // 左入力回数
    int   nRight;    // 右入力回数
    int   nUp;       // 上入力回数
    int   nDown;     // 下入力回数
};

/************************************************************************************************/
/*                                        プロトタイプ                                          */
/************************************************************************************************/
int SampleFunc();
```

ヘッダファイルの基本は、いろいろなファイルで使用する情報(対応するソースファイルで定義されている関数のプロトタイプ宣言、ほかのファイルでも使わなければならない定数、構造体、列挙体などの定義)を記述することです。

「#pragma once」は、ヘッダファイルを作成するときには必ず記述しておきます。こうすると、同じヘッダファイルは１回しかインクルードされなくなります。間違って２回以上インクルードしたときの「すでに定義されています」といったエラーを防げるだけでなく、以下のような互いにヘッダファイルを参照しあって無限にインクルードしてしまうという現象を防ぐことができます。

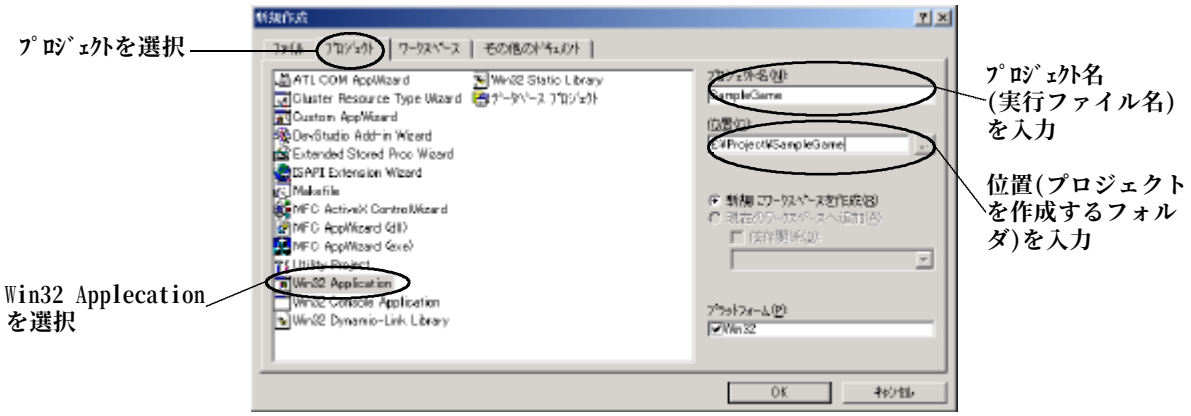


「インクルードファイル」には、このヘッダファイルをコンパイルするのに必要なものだけ記述すればよいのですが、対応するソースファイルをコンパイルするのに必要なすべてのヘッダファイルを記述してもかまいません。

○ 課題 ○

サンプルゲームのプロジェクトを作成しましょう。

メニューから「ﾌｧｲﾙ(F)→新規作成(N)」を選択し、新規作成ダイアログを開きます。"Win32 Application"を選択し、プロジェクト名とプロジェクトを作成する位置を入力して"OK"をクリックします。



作成するアプリケーションの種類を選択するダイアログが開くので、「空のﾌﾟﾛｼﾞｪｸﾄ(E)」を選択して"終了(E)"をクリックします。作成されるファイルの確認ダイアログが表示されるので、"OK"をクリックすれば空のプロジェクト作成されます。