

オブジェクト指向と ゲームプログラミング

フレームワーク編 - 第11回 ゲーム処理クラスの作成

課 題

ゲーム処理をカプセル化したCGameProcクラスを作成しましょう。

(1) CGameProcクラスのヘッダファイルを作成しましょう。

CGameProcクラスに最低限必要な機能は、ゲーム処理の初期化を行う機能、使用していた資源を解放する機能、そしてゲームを実行する機能です。ここで、初期化と解放は、ゲームのシーンごとに行うのが一般的なので、今回は必要ありません。また、属性(変数)も同様にシーンごとに定義するので、必要ありません。

以上をふまえると、CGameProcクラスのヘッダファイル(GameProc.hpp)は以下のようになります。

- GameProc.hpp -

```
/*
=====
                          オブジェクト指向ゲームプログラミング
                          Programmed by Hibikino software. Copyright (c) 2005 Hibikino software. All rights reserved.
=====
【対象OS】
  Microsoft Windows2000/XP
【コンパイラ】
  Microsoft Visual C++ 2005
【プログラム】
  GameProc.hpp          ゲーム処理クラスヘッダ
【履歴】
  * Version    1.00      2004/04/dd  hh:mm:ss
=====
*/
#pragma ???

/*****
/*                          インクルードファイル                          */
*****/

/*****
/*                          ゲーム処理クラス定義                          */
*****/
class CGameProc {
public:
  CGameProc();
  ??????? -CGameProc();

  bool Run();

private:
  CGameProc(const CGameProc&);
  CGameProc& operator=(const CGameProc&);
};
```

(2) CGameProcクラスのメンバは、以下のとおりです。

CGameProc コンストラクタ

CGameProcオブジェクトを構築します。

~CGameProc デストラクタ

CGameProcオブジェクトを解放します。

Run 処理

ゲームを実行します。ゲームの終了など、アプリケーションを終了する必要があるときはfalseを返します。それ以外の場合はtrueを返します。

bool Run();

Return アプリケーション続行 : true アプリケーション終了 : false

(3) CGameProcクラスのソースファイル(GameProc.cpp)を以下のように作成しましょう。

- GameProc.cpp -

```
/*
=====
                          オブジェクト指向ゲームプログラミング
Programmed by Hibikino software. Copyright (c) 2005 Hibikino software. All rights reserved.
=====
【対象OS】
  Microsoft Windows2000/XP
【コンパイラ】
  Microsoft Visual C++ 2005
【プログラム】
  GameProc.cpp
                      ゲーム処理クラス
【履歴】
  * Version   1.00      2004/04/dd hh:mm:ss
=====
*/
/*****
/*                          インクルードファイル                          */
/*****
  ここは各自考えましょう
/*****
/*                          コンストラクタ                          */
/*****
CGameProc::CGameProc()
{
}
/*****
/*                          デストラクタ                          */
/*****
CGameProc::~CGameProc()
{
}
/*****
/*                          ゲーム実行                          */
/*****
bool CGameProc::Run()
{
  return true;
}

```

(4) ゲーム処理は、ゲームアプリケーションの一部と考えられます。CGameAppクラスとCGameProcクラスの関係は、CGameAppクラスがCGameProcクラスを「持っている」関係になります。

以下のプログラムを適切な場所に追加し、CGameProcオブジェクトを適切なクラスに集約させましょう。

```
CGameProc m_GameProc; // ゲーム処理
```

(5) (4)を追加したクラスのコンストラクタ初期化子に、以下のプログラムを追加し、CGameProcオブジェクトが適切に初期化されるようにしましょう。

```
m_GameProc();
```

(6) アプリケーションのメイン処理にゲーム処理を追加します。CGameAppクラスのMessageLoop関数を以下のように変更しましょう。

```
int CGameApp::Main()
{
    while(true) {
        // メッセージ処理
        if(MessageLoop() == false)
            break;
        // ゲーム処理
        if(m_GameProc.???() ==  // ここは各自考えましょう)
            Release(); // アプリケーション終了
    }

    return 0;
}
```