

ESライブラリ & & ゲームプログラミング

導入編 - 第1回 ESライブラリの概要

ES Game Library

ES Game Libraryは、WindowsとDirectXを使ったゲームプログラミングをできるだけ簡単に行えるようにするためのゲーム向けライブラリです。Windowsとゲームの基本的な処理を行うフレームワーク群、DirectX Graphics, DirectX Audio, DirectX Inputをクラス化したライブラリに、ゲームループをコールバック式にするためのGameMain.hとGameMain.cpp、さらにXNA風の関数名とクラスを定義したヘッダファイルXNA.hから構成されています(ゲームループは無限ループ風にすることもできます)。

DirectX9.0cの基本的な機能を簡単な関数呼び出しで使うことができるようになっています。また、Win32APIも使用でき、今となっては低速ですが豊富な描画命令を備えるGDIも使用できます。さらに、DirectX Graphicsにおいては、シェーダーにも対応できるので、GPUの性能を大きく引き出すこともできます。

本来、WindowsとDirectXでゲームを作成するには、ウィンドウの管理やDirectXの制御など、かなりの量のコードを書く必要があります。特にデバイスの管理は複雑になる傾向があり、合わせて1,000行以上のプログラムを書かなければなりません。

ES Game Libraryでは、そういった面倒な部分をクラスに閉じ込めてあります。また、ソースコードも公開していますので、自由に変更・拡張して使うことができます。配布も自由です。

GameMainクラスの機能

- | | |
|-------------------------------|--------------------------|
| ・ Initialize関数 | グラフィクス、サウンド以外の初期化 |
| ・ LoadContent/UnloadContent関数 | グラフィクス、サウンドなどリソースの初期化・解放 |
| ・ Update/Draw関数 | ゲームのメインループ |

概要

ESライブラリでプログラミングの核となるのがGameMainクラスです。Windowsで動作するゲームの最小の機能のみ定義してあります。このクラスには、Initialize, LoadContent, UnloadContent, Update, Drawの5つの関数のひな形が用意されています。

Initialize関数

変数の初期化、コンフィグやユーザデータといった画像や音声以外のデータの設定を行います。

LoadContent関数

グラフィクスやサウンドといったマルチメディアデータの読み込みを行います。

UnloadContent関数

LoadContent関数で読み込んだグラフィクスやサウンドといったマルチメディアデータの解放を行います。

CやC++では、生成したリソースは解放しないとメモリに残ったままになるケースが多々あります(これをメモリリークやリソースリークと呼びます)。この関数には、リークを防ぐため、リソースを解放するプログラムを記述します。

ESライブラリでは、C++のデストラクタを使い、リークを防ぐための工夫をしていますが、必要に応じてリソースを解放するプログラムを記述しておいた方が無難です。

Update関数

ゲームの状態変更や背景、キャラクタの座標更新、衝突判定など描画以外の処理をします。デフォルトでは16.6ms毎、秒間60回呼ばれるようになっています。

Draw関数

画面に描画するための処理をします。

メインループ

- ・ゲームの仕組みはセルアニメーションと同じ
- ・1秒間に数十回程度のループを行う
- ・ループ内でゲーム固有の処理を行い、画面に描画する

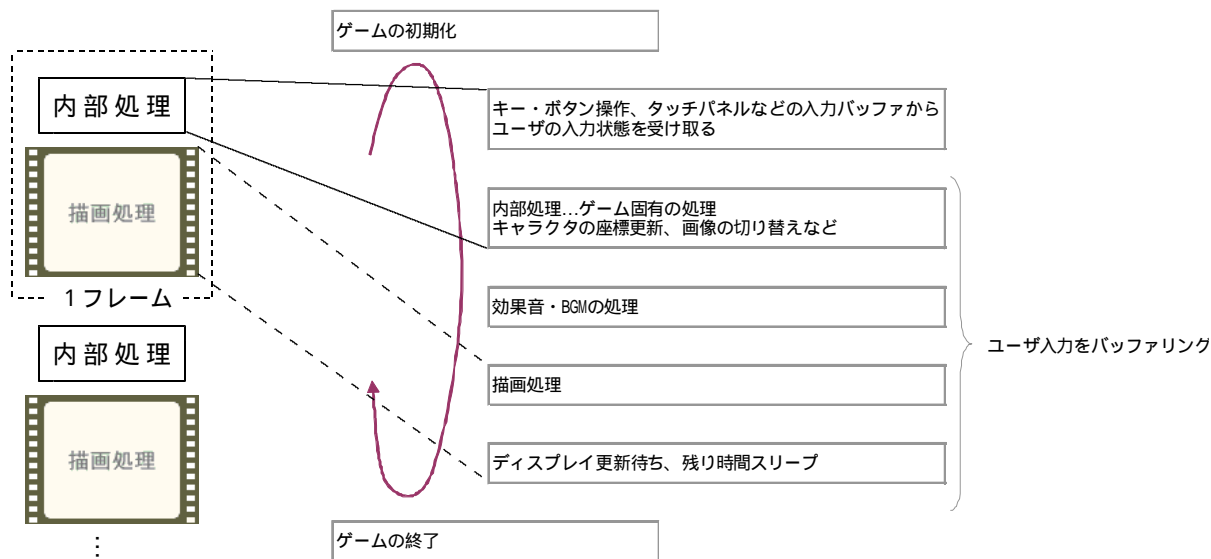
概要

ゲームプログラムは、キャラクタや背景など、いろいろな「もの」の状態を時間にそって更新し、画面に描画するプログラムと考えることができます。これを非常に単純化すると、

- ・「もの」の状態を更新(内部処理 = Update関数)
- ・「もの」を画面に描画(描画処理 = Draw関数)

の2つにまとめることができます。この処理の繰り返しゲームのメインループになります。ループ1回ぶんを「1フレーム」と呼びます。1フレームごとに、キャラクタの位置や状態、背景やそのほかの表示状態などを更新し、その状態を反映して描画を行う、という処理を繰り返しています。

簡易図



ゲームループの構成

ゲームループの構成方法には、内部処理と描画処理を無限ループ内で実行する方法と、OSなどのシステムにより、必要に応じて自動的に実行されるコールバック式の2つが主に使われています。

無限ループ式は、古くからある方法で、OSなどが搭載されていない純粋なゲーム機で使われる方法です。現在もNintendoDSで使われています。名前のおり、メインとなる関数で無限ループを構成し、そのなかでゲームの処理を実行し続けます。仕組みが単純で開発しやすいのが利点です。

コールバック式は、近年主流となっている方法です。(目に見える形での)ゲームループは構成せず、内部処理や描画処理を行う関数は、システムによって必要なときに呼び出されて実行されます。OSを搭載したPCや携帯機、ワークステーションに近い構成のゲーム機などで採用されています。無限ループ式に比べると、構成が少しだけ複雑になりますが、あらゆるプラットフォームに対応することができるというのが利点です。

ES Game Library必要環境

- VisualC++2005, 2008, 2010(Express可)
- DirectX SDK 2007以降(ID3DXFont::PreloadCharacters等使用のため)
- oggファイル対応のため、ogg vorbis SDKも必要

概要

ES Game Libraryでゲーム開発を行うには、VisualC++とDirectX SDKの導入が必要になります。VisualC++は、仕様を満たすDirectX SDKがビルドできるバージョンであればどれでもかまいません。

DirectX SDKは、9.0cが必要ですが、年月によって引数が増えたり、機能追加されたり(もしくは削除されたり)しているため、ID3DXFontインタフェースに機能追加があった以降のバージョン(2007以降)が必要です。それらの機能を使わないように、もしくは引数の変更を行えば、ビルド自体はできるようになっています。また、August2007に削除されたDirectMusicを使用しているため、DirectMusic関連のヘッダーファイルとライブラリファイルが必要です。さらにサウンド関連で、ogg形式に対応するため、ogg vorbis SDKも必要となっています。

これらは、SDKをインストールまたはコピーしただけでは認識されません。VC2010版では、プロジェクトごとにフォルダを設定するようになっていたため、プロジェクトプロパティでES Game LibraryのDirectX/SDKフォルダにパスを設定済みとなっていますが、そのほかのバージョンでは、インクルードフォルダとライブラリフォルダの追加設定が必要です。

ES Game Libraryでの開発

SDKのインストールと各フォルダの設定が終わったら、ES Game Libraryでの開発が行えます。.slnファイルをダブルクリックしてVisualC++を起動させてビルドすれば、空のウィンドウが表示されます。ESCキーを押すと終了です。これが、ES Game Libraryのフレームワークの仕事です。

これ以降のゲームプログラムをGameMain.hとGameMain.cppに記述していくことが、ゲーム開発の第一歩となります。

課題

ES Game Libraryが使えるように開発環境を整備し、ソリューションをビルドしてプログラムを実行しましょう。成功すると、横長のウィンドウが表示されます。