

XNA Game Studio ゲームプログラミング

第3回 アプリケーションモデル

XNAアプリケーションモデルの機能

- ・ウィンドウの作成
- ・グラフィクスデバイスの生成
- ・ゲームのメインループ

概要

アプリケーションモデルとは、アプリケーションの元となる、土台を構成する基本的な部分のことです。WindowsでDirectXの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。

XNAでは、そういった面倒な部分を容易にするために、アプリケーションモデルが準備されています。XNAのアプリケーションモデルは100行程度のコードで上記の機能を実装することができます。また、テンプレートが用意されているため、Visual C#でプロジェクトを作成すれば、自動的にコードが生成されるようになっています。

Gameクラスの機能

- | | |
|---------------------------------|---------------------|
| ・ Initializeメソッド | グラフィクス、オーディオ以外の初期化 |
| ・ LoadContent/UnloadContentメソッド | グラフィクス、オーディオの初期化・破棄 |
| ・ Update/Drawメソッド | ゲームのメインループ |

概要

XNAアプリケーションモデルの核となるのはGameクラスです。Gameクラスには、XNA Framework上で動作するゲームの最小の機能のみ定義されています。テンプレートを用いると、Gameクラスを継承したクラスが定義されます。このクラスには、Initialize、LoadContent、UnloadContent、Update、Drawの5つのメソッドのひな形が定義されています。これらは、通常の間数やメソッドと異なり、システムが必要に応じて自動的に呼び出すようになっています。

これらのメソッドにコードを追加するのが、XNAゲームプログラミングの第一歩になります。

Initializeメソッド

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読み込みを行います。

LoadContentメソッド

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。このメソッドは、デバイスが作成されたり、リセット、ロストしたときに呼び出されます。

Xbox 360では起こりませんが、Windowsでは他のアプリケーションに切り替わった場合、グラフィックデバイス(ビデオカード)やサウンドデバイスが取られてしまう(自アプリから見るとデバイスのロスト状態)場合があります。また、ウィンドウモードとフルスクリーンモードの切り替え時にも、解像度などが変更になるため、デバイスにリセットがかかり、このメソッドが呼び出されます。

そのような場合、デバイスが保持していた画像や音声は消失してしまいます。この状態のまま自アプリケーションに戻っても、デバイス内のデータは使用できませんので、描画などに支障が出てしまいます。そういった場合にこのメソッドが呼び出されますので、再度読み込み直しデータの復旧を行います。

UnloadContentメソッド

デバイスのリセットやロストの前に呼ばれます。基本的に何も記述する必要はありませんが、デバイスの復元時に特別な処理を行う必要がある場合に、コードを記述する場合があります。

Updateメソッド

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。この間隔はGame.TargetElapsedTimeを変更することで変えることができます。

Drawメソッド

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

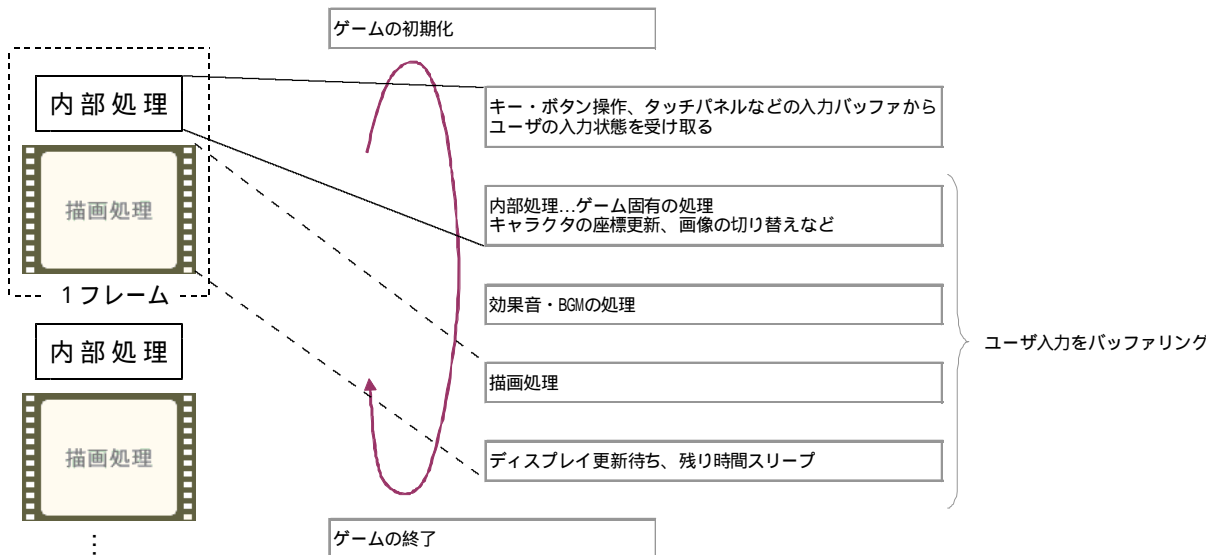
概要

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メソッド)
- ・「もの」を画面に描画(描画処理 = Drawメソッド)

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。

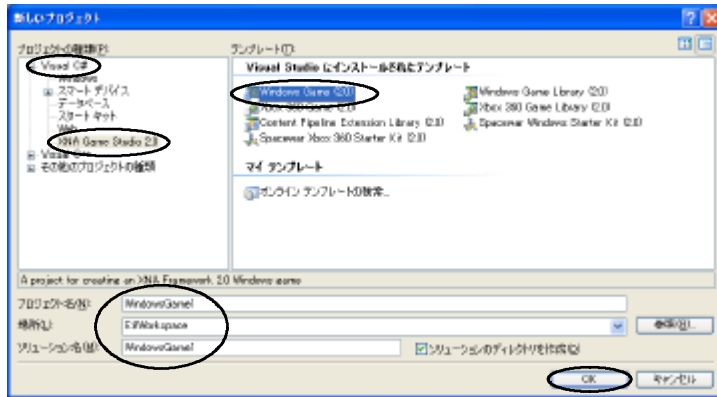
簡易図



ゲームプログラムの基本的な部分を作成しましょう。

(1) Windows用のプロジェクトを作成します。

1. Visual C#のメニューから「ファイル 新規作成 プロジェクト」と選んでください。
2. プロジェクトの種類は"Visual C#"の"XNA Game Studio"、テンプレートは"Windows Game"を選んでください。
3. プロジェクト名、場所、ソリューション名は適切に設定してください。"OK"をクリックすると、プロジェクトが作成されます。



(2) ファイル名を変更します。

1. ソリューション エクスプローラの"Game1.cs"で右クリックし、「名前の変更(M)」を選びます。
2. ファイル名を"GameMain.cs"に変更してください。

(3) プロジェクトをビルドし、実行してみましょう。ウィンドウが表示されれば成功です。

(4) ウィンドウをXbox 360の16:9サイズにします。GameMainクラスのコンストラクタを以下のように変更しましょう。

```
public GameMain()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";

    // 画面をHDサイズに変更
    graphics.PreferredBackBufferWidth = 1280;
    graphics.PreferredBackBufferHeight = 720;
}
```