

XNA Game Studio ゲームプログラミング

2D編 - 第4回 全角文字の描画

全角文字の描画

- ・ XNAでは、文字を描画するときは、フォントの描画画像をXNB形式で生成しておく必要がある
- ・ スプライトフォント追加時に、半角英数字はデフォルトで生成されるようになっている (.spritefontファイルの<CharacterRegions>で指定されている)
- ・ 半角英数字以外の文字も、描画画像を作成すれば描画することができる
- ・ 描画画像を作成するには、以下の方法がある
 - 1 . .spritefontファイルの<CharacterRegions>の範囲を広げる(Unicode)
 - 2 . FontDescriptionProcessorクラスを使うと描画画像を作成することができる
 - 2 - 1 . ひらがな、カタカナ、記号、JIS第1、2水準漢字などあらゆる文字を生成する (JIS第1・2水準漢字で約6300字)
 - 2 - 2 . ひらがな、カタカナ、記号、常用漢字、人名用漢字などから、よく使う文字を生成する (常用・人名用漢字で約3000字)
 - 2 - 3 . ゲームのテキストやソースファイルを読み取り、必要な文字だけを抽出してを生成する
- ・ フォントデータがあれば、どんな文字でもSpriteBatchクラスのDrawStringメソッドで描画できる

概要

XNAでは、使用する文字をスプライトフォントとして作成しておく必要があります。スプライトフォントは画像として文字を保持しているので、実行環境にフォントがインストールされていなくても問題なく描画できます。また、画像なのでビデオカードの支援を受け、3Dの機能を用いて高速に描画することができます。

しかし、フォントの形や大きさを変えたい場合、それ専用のスプライトフォントが必要となります。描画するすべての文字の形と大きさのデータを作成しなければなりません。

XNAでは、必要な文字のスプライトフォントを簡単に作成するためにFontDescriptionProcessorクラスを提供しています。このクラスを使えば、簡単なコードを記述するだけでスプライトフォントを生成することができます。ただし、文字数が多くなると、比例してビルド時間とXNBファイルのサイズが大きくなってしまいます。

- 5 . ".NET"タブにある"Microsoft.Xna.Framework.Content.Pipeline"を選択し、追加してください。
- 6 . "FontProcessors.cs"を以下のように編集してください。

- FontProcessors.cs -

```
using System;
using System.IO;
using System.Collections.Generic;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Storage;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.Content.Pipeline;
using Microsoft.Xna.Framework.Content.Pipeline.Graphics;
using Microsoft.Xna.Framework.Content.Pipeline.Processors;

[ContentProcessor]
public class FontProcessor : FontDescriptionProcessor
{
    public override SpriteFontContent Process(FontDescription input,
        ContentProcessorContext context)
    {
        // ファイル読み込み
        string fullPath = Path.GetFullPath("kanji.txt");
        context.AddDependency(fullPath);
        string letters = File.ReadAllText(fullPath, System.Text.Encoding.UTF8);

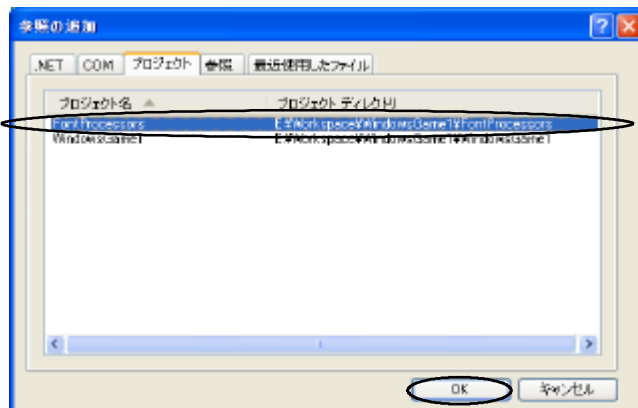
        // フォントの追加
        foreach (char c in letters)
        {
            input.Characters.Add(c);
        }

        return base.Process(input, context);
    }
}
```

- 7 . ソリューションエクスプローラのプロジェクト名"FontProcessors"で右クリックし、ビルドを選択、プロジェクトをビルドしてください。

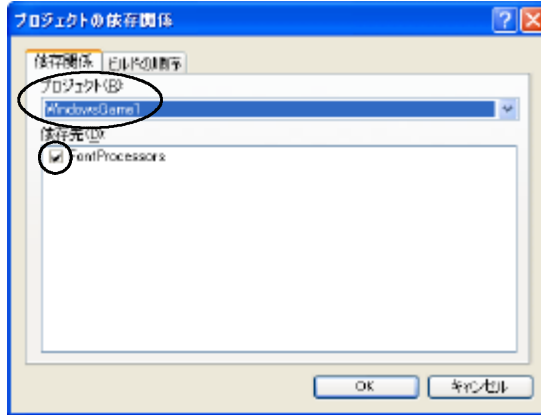
(3)プロジェクトの関連づけを行います。

- 1 . ソリューションエクスプローラのプロジェクト名"WindowsGame1", "Content"の「参照設定」で右クリックし、「参照の追加(R)...」を選択します。
- 2 . 「プロジェクト」タブからプロジェクト名"FontProcessors"を選択し"OK"ボタンをクリックします。

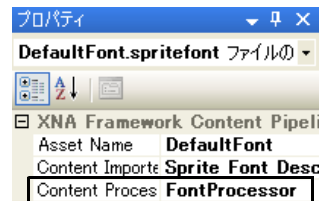


- 3 . ソリューションエクスプローラのソリューション名を右クリックし、「プロジェクト依存関係(S)...」を選択します。

4. 「プロジェクト(R)」で"WindowsGame1"を選択、依存先(D)の"FontProcessors"にチェックを入れます。こうすると、WindowsGame1プロジェクトにFontProcessorsプロジェクトが関連づけられ、ビルドの順序などが適切に設定されます。



5. ソリューションエクスプローラのプロジェクト名"WindowsGame1", "Content"の"DefaultFont.spritefont"が選択(反転)された状態で、プロパティの"Content Processor"を"FontProcessors"に変更します。



(4) "WindowsGame1"をビルドします。"kanji.txt"の文字数が多いと時間がかかります。エラーがなければ成功です。

(5) 半角英数字と"kanji.txt"の文字が描画できるようになりましたので、描画してみましょう。