

ESライブラリ&& ゲームプログラミング

2D編 - 第2回 スプライトの重ね合わせ

スプライトの重ね合わせ

- ・複数のスプライトを表示して画面を構成する
- ・問題になるのが「切り取る」もしくは「くり抜く」部分と描画順
- ・「切り取る」もしくは「くり抜く」(描画したくない)部分は、「アルファ値」または「カラーキー」
- ・描画順は「z値の手前から奥」「z値の奥から手前」「描画した順」の指定ができる

概要

2Dゲームでは、複数の画像を重ねて描画し、画面を構成します。このとき問題になるのが余分な部分を描画しないという透明扱いの色の処理と描画する順番です。

透明扱いの色については、たいていのハードウェアもしくはライブラリで対応しているので、画像に透明となる部分の情報を含めておくか、画像の読み込み時もしくは描画時に、透明となる色を指定することで実現できます。ESライブラリでは、透明度を持つ画像形式に対応しており、また、透明度がない画像も読み込み時に指定色を透明扱いにすることができます。

描画順は、描画時に奥のものから描画して前後関係を正しく表現する「zソート」、画面のピクセルに「深度」という情報(z値)を持たせる「zバッファ」があります。3Dの実行環境では、ポリゴンの前後関係を表現するためにzバッファがサポートされています。Direct3DとESライブラリでは、zソートとzバッファのどちらにも対応しています。

アルファチャンネル

アルファチャンネルとは、ピクセルの赤・緑・青に加え、透明度情報を保存しておく領域のことをいいます。透明度はピクセル毎に、0~255または0.0から1.0の範囲で指定できることが多く、この情報をアルファ()値と呼びます。各ピクセルのアルファ値を保存しておく領域がアルファチャンネルとなります。

アルファ値は、0が完全な透明(色が設定されていたとしても描画されない)、255もしくは1.0が完全な不透明(下にある画像を完全にとささない)という意味になります。

アルファチャンネルは、画像形式やソフトウェアによって、用意されている場合とされていない場合があります。画像の形式としては、.png .tga .dds(.gifも完全透明情報を保存できますが、Direct3Dでは特許の関係上未対応)が対応しています。ソフトウェアでは、PhotoshopやPaint Shop、一部のフリーソフトにも対応しているものがあります。

ESライブラリでは、アルファチャンネルの設定された画像を読み込めば、アルファ値にあわせ、透明/半透明/不透明に描画されます。

カラーキー

DirectXのカラーキーは、描画されない色(透明色)のことをいいます。カラーキーに指定された色は、透明のように振るまい、画像には存在するものの描画時に表示されなくなります。

DirectX9でのカラーキーは、スプライト(テクスチャ)の読み込み時に指定された色を強制的にアルファ値0にすることにより再現しています。

ESライブラリでも、スプライトとなる画像の読み込み時にカラーキーの設定ができます。デフォルトでは、描画時にアルファ値0のピクセルは描画されないようになっているので、カラーキーと同じ動作になり、その色が描画されなくなります。アルファ値を無視し、すべてのピクセルを描画させることもできます。

アルファチャンネルとカラーキーの設定

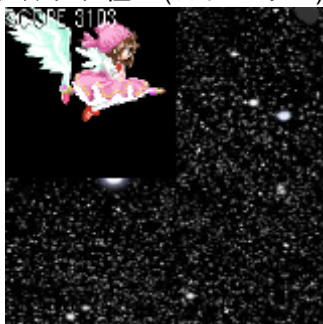
画像にアルファチャンネルがある場合、スプライト描画時に使用されるようになっており、キャラクター周辺の余分な部分は描画されません(もしくは半透明になります)。

画像にアルファチャンネルがない場合で余分な色を描画したくない場合は、スプライトの読み込み時にカラーキーとなる色を指定することにより、強制的にその色のピクセルをすべてアルファ値0に設定させることができます。この動作により、指定色は描画されないようになります。

```
// スプライト型変数の宣言  
SPRITE Chara;
```

```
// カラーキーを指定してスプライト読み込み( RGB(0, 0, 0) = 黒がカラーキー )  
Chara = GraphicsDevice.CreateSpriteFromFile( TEXT("Chara.bmp"), Color(0, 0, 0) );
```

- アルファ値0(カラーキー)使用例 -



アルファ無視またはカラーキーなし



アルファ値もしくはカラーキー指定

zソート

zソートは、一番奥にある画像から順に描画し、画面の前後関係を正しく表現する方法です。たとえば、たいていのゲームでは、背景 奥にあるキャラクターから順に描画 フレーム(飾り枠) ステータスやメッセージなどのテキスト面、といった順番に描画しています。描画した順に画面に出力される環境や、3Dグラフィクスでも複数の半透明の画像を正しく重ね合わせて描画する場合に使用されます(3Dでは、z値で描画順を指定できる場合もあります)。

zバッファ

zバッファは、画面のピクセルごとに前後関係を判定し描画します。前後を判定するための深度(z値)を保存する領域をzバッファと呼びます。ピクセルが描画されるたびに、zバッファを利用して深度を比較することで、最終的に深度が手前のポリゴン(正確にはピクセル)が画面に残ります。

ESライブラリでは、デフォルトでzバッファ有り、スプライトはz値でソートされ、手前のものから描画されるようになっています。最終的にzバッファにより、2Dと3Dが合成されて描画されます。

背景の上にキャラクターを表示しましょう。

(1)キャラクター用の画像を準備しましょう。

(2)キャラクターのスプライトを管理する変数を宣言します。以下のプログラムをヘッダーファイル"GameMain.h"の適切な場所に追加しましょう。

```
SPRITE Chara;    // キャラクタースプライト
```

(3)キャラクターの画像を読み込みます。以下のプログラムをソースファイル"GameMain.cpp"の適切な場所に追加しましょう。

```
// キャラクター読み込み  
Chara = GraphicsDevice.CreateSpriteFromFile( TEXT("キャラクター画像のファイル名") );
```

(4)スプライトを表示します。以下のプログラムを適切な場所に追加しましょう。

```
SpriteBatch.Draw(Chara, Vector3(0.0f, 0.0f, 0.0f));    // キャラクター描画
```

(5)プログラムをビルドして実行し、正しく重なっているか確認しましょう。