

ESライブラリ&& ゲームプログラミング

3D編 - 第9回 カリング

カリング

- ・カリングは、描画する必要のないポリゴンを破棄(描画しないように)すること
- ・背面カリング、視錐台カリング、オクルージョンカリング、早期zカリングなどがある

概要

カリングとは、描画する必要のないポリゴンやモデルを事前に描画候補から外すことです。カリングを行えば、描画処理の負担を軽減することができます。

カリングには、背面カリング、視錐台カリング、オクルージョンカリング、早期zカリングなどがあります。

背面カリング

背面カリングとは、後ろ向きのポリゴンを削除することです。後ろ向きのポリゴンは基本的にモデルの背面にあり隠れているので描画する必要がない、という考え方です。

ポリゴンは、頂点の並びが時計回りと反時計回りがあります。どちらを背面とみなすかを指定し、該当するポリゴンを描画しないようにできます。

半透明のモデルや隙間からあえて背面ポリゴンを見せているモデル、裾やスカートのあるモデルなど背面ポリゴンを活用している場合もあります。その場合は、カリングをせずに描画します。

Direct3Dでは、背面カリングをサポートしており、レンダリング状態でカリング方法を変更できるようになっています。

```
// カリングなし
```

```
GraphicsDevice.SetRenderState(CullMode, CullMode_None);
```

```
// 反時計回りをカリング
```

```
GraphicsDevice.SetRenderState(CullMode, CullMode_CullCounterClockwiseFace);
```

```
// 時計回りをカリング
```

```
GraphicsDevice.SetRenderState(CullMode, CullMode_CullClockwiseFace);
```



背面カリングなし



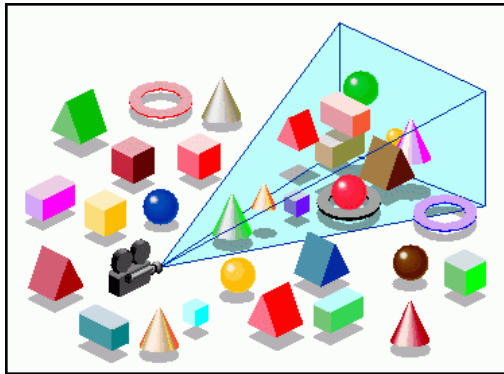
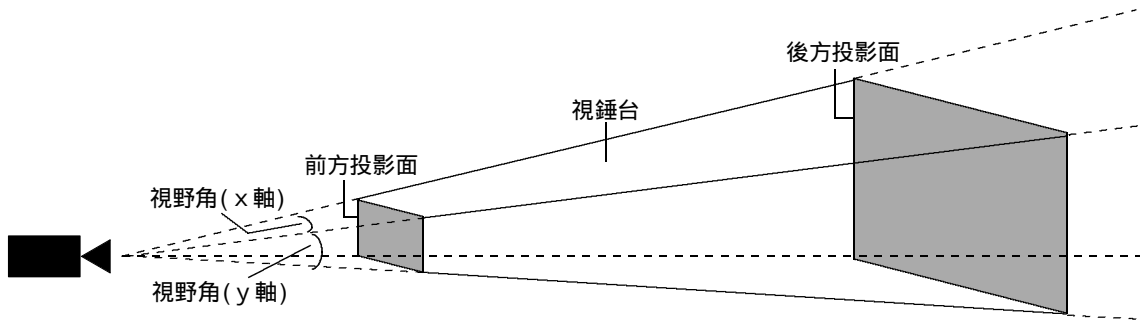
反時計回りのポリゴンをかリング



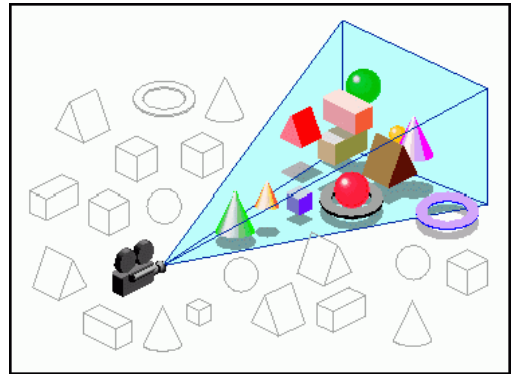
時計回りのポリゴンをかリング

視錐台カリング

視錐台カリングは、カメラの視錐台内に存在しないモデルを描画しないようにすることです。Direct3Dではサポートされていませんが、ESライブラリには、ワールド空間上の視錐台範囲を取得できるようになっています。視錐台とモデルのバウンディングスフィア(境界球)との衝突判定も行えるようになっているので、視錐台カリングを簡単に行うことができます。



視錐台カリングなし



視錐台カリングを行うと、描画する必要があるものだけ処理することができる

オクルージョンカリング

オクルージョンカリングは、モデルが不透明のモデルによって隠される(見えない位置にある)場合、描画を行わないというものです。通常は、zバッファによって最終的に前後判定が行われますが、あらかじめ見えないものを処理しないことによって、負担を軽減する方法です。

ほとんどの環境では、モデルを実際に描画してみないと隠れるかどうか判定できないので、簡易モデルを描画して前後関係を判断しています。Direct3Dでは、クエリという機能を用いることによって実現できます。

早期 z カリング

早期 z カリングは、頂点変換後のラスタライズ処理時に、深度値(zバッファ)を調べ、明らかに後ろに隠れるピクセルは描画しないというものです。近年のグラフィックカードに標準搭載されており、無駄なピクセル描画を省くことによってピクセルシェーダーや最終的なピクセル描画の負担を軽減しています。

LOD (Level of Detail)

LODは始点からの距離によってモデルの頂点数やポリゴン数を削除し、遠くのものほど少ないポリゴン数で描画するという方法です。遠くのモデルは、ポリゴン数が多い詳細なモデルであっても、そのほとんどは描画されないことが多いので、頂点やポリゴンの数を減らした簡易的なモデルを描画しても問題がないという考え方です。

Direct3Dでは、視点からの距離によって頂点やポリゴン数を変更する仕組みは用意されていませんが、プログレッシブメッシュという機能があり、いつでも自由に頂点やポリゴン数を変更することができるようになっています。ただし、モデルがもともともっていた数を超えてより詳細なものにすることはできません。また、最小数もモデルごとに決まっており、それを越えて少なくすることもできません。モデルによっては、もともとの数に対して8~7割程度であっても見かけ上まったく問題がないものもあるので、プログレッシブメッシュを用いることにより、描画負担をより減らすことができます。

カリングを行ってみましょう。

(1)レンダリング状態を変更し、背面カリングを確認しましょう。

(2)プログレッシブメッシュの機能を確認しましょう。

1. プログレッシブモードはデフォルトで無効になっています。有効にするには、モデルまたはアニメーションモデルのEnableProgressive関数を呼び出します。

```
// プログレッシブモード  
model->EnableProgressive();
```

2. プログレッシブモードに移行すると、SetLOD関数により、ポリゴン数を自由に調整できます。SetLOD関数の引数は「もともとのポリゴン数に対する割合」で、1.0fから0.0fの数値を指定します。1.0fはもともとモデルが持っていたポリゴン数、0.0fはモデルがサポートする最小のポリゴン数になります。

```
// ポリゴン数を7割にする  
model->SetLOD(0.7f);
```

(3)キーボードの入力によって、LODレベルを1.0から0.0まで自由に増減できるプログラムを作成しましょう。

応用問題1：カメラからの距離によってLODレベルが自動的に変わるプログラムを作成しましょう。
応用問題2：視錐台カリングを実装してみましょう。