

XNA Game Studio

ゲームプログラミング

3D編 - 第2回 テクスチャ

テクスチャ

- ・テクスチャは、モデル表面の質感を表現するために、ポリゴンに貼り付ける画像
- ・ポリゴンの組み合わせやマテリアルでは表現できない複雑な模様が表現できる
- ・テクスチャを工夫することにより、少ないポリゴンでもリアリティあるオブジェクトができる

概要

テクスチャは、モデル表面の質感を表現するために、ポリゴンに貼り付ける画像のことをいいます。同じ立方体のモデルでも、金属のテクスチャを貼り付ければ金属片に見え、木目のテクスチャを貼り付ければ木片に見えます。

テクスチャを用いると、ポリゴンの組み合わせやマテリアルでは表現できない複雑な模様を表現することができます。また、テクスチャを貼ることにより、少ないポリゴンでもリアリティあるオブジェクトを作ることができます。

テクスチャの読み込み

テクスチャはスプライト同様、コンテンツに登録しておけば、ContentManagerクラスのLoadメソッドで読み込むことができます。Texture2Dクラスに読み込まれるのも同じです。これをスプライトとして描画するか、テクスチャとして扱うかだけの違いです(両方に使うこともできます)。

XNA(DirectX9)では、テクスチャの最大サイズは2,048×2,048ピクセルとなっています(DirectX10の4.0以降は4,096×4,096)。

テクスチャの設定

ポリゴンにテクスチャを設定するには、頂点データにVertexPositionColorTexture構造体などのように、テクスチャ座標が必要となります。テクスチャ座標とは、「この頂点には、テクスチャのこの部分を貼る」という情報です。また、BasicEffectを用いている場合は、TextureEnabledをtrueにし、テクスチャの適用を有効にします。

この状態でBasicEffectクラスのTextureにTexture2Dクラスの変数を渡すと、以降のBasicEffect.Begin~Endメソッドに囲まれたDrawPrimitives系メソッド呼び出しすべてで、テクスチャが適用されるようになります。

モデルによってテクスチャを変える場合は、その都度BasicEffectのTextureに適用するテクスチャを設定してから、プリミティブの描画を行います。

最終的な色は、ポリゴンの色とテクスチャの色が乗算されたカラー値となります。乗算はRGBの成分ごとに個別に計算されます。色は0.0fから1.0fもしくは0から255で指定しますが、255は1.0fとして扱われます。ポリゴン色が黒[Color(0, 0, 0)]だと、すべての成分が0になるため、オブジェクトにテクスチャが適用されていないように見えます。また、白[Color(255, 255, 255)]は、テクスチャの色×1.0fとなるため、テクスチャの色がそのまま反映されます。

テクスチャ座標(UV座標)

テクスチャ座標とは、UV座標とも呼ばれ、テクスチャに付けられた座標のことです。テクスチャの大きさにかかわらず、左上が(0.0, 0.0)、右下が(1.0, 1.0)となります。どんな大きさでも0.0から1.0に正規化されるので、頂点データでは、テクスチャの具体的なピクセル数の指定ではなく、0.0から1.0を設定すれば良いので、後で仕様変更などがあり、テクスチャのサイズが変わったとしても、画像の変更だけで、テクスチャ座標の修正は不要ということになります。



コーディング例

```
// フィールド変数
VertexDeclaration vtxDec;
BasicEffect basicEffect;

VertexBuffer vertexBuf;
Texture2D objTxtr;
:
protected override void Initialize()
{
    // TODO: Add your initialization logic here
    vtxDec = new VertexDeclaration(GraphicsDevice,
        VertexPositionColorTexture.VertexElements);

    basicEffect = new BasicEffect(GraphicsDevice, null);

    basicEffect.VertexColorEnabled = true;
    basicEffect.TextureEnabled = true;
    :
    // 頂点データの設定
    VertexPositionColorTexture[] vtx = new VertexPositionColorTexture[6];
    vtx[0] = new VertexPositionColorTexture(new Vector3(520.0f, 240.0f, 0.5f),
        new Color(255, 255, 255, 255),
        new Vector2(0.0f, 0.0f)); // UV座標の設定
    vtx[1] = new VertexPositionColorTexture(new Vector3(760.0f, 480.0f, 0.5f),
        new Color(255, 255, 255, 255),
        new Vector2(1.0f, 1.0f));
    vtx[2] = new VertexPositionColorTexture(new Vector3(520.0f, 480.0f, 0.5f),
        new Color(255, 255, 255, 255),
        new Vector2(0.0f, 1.0f));

    vtx[3] = new VertexPositionColorTexture(new Vector3(520.0f, 240.0f, 0.5f),
        new Color(255, 255, 255, 255),
        new Vector2(0.0f, 0.0f));
    vtx[4] = new VertexPositionColorTexture(new Vector3(760.0f, 240.0f, 0.5f),
        new Color(255, 255, 255, 255),
        new Vector2(1.0f, 0.0f));
    vtx[5] = new VertexPositionColorTexture(new Vector3(760.0f, 480.0f, 0.5f),
        new Color(255, 255, 255, 255),
        new Vector2(1.0f, 1.0f));

    vertexBuf = new VertexBuffer(GraphicsDevice, typeof(VertexPositionColorTexture),
        vtx.Length, BufferUsage.None);
```

```

    vertexBuf.SetData<VertexPositionColorTexture>(vrtx);
    ⋮
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be used to draw textures.
    spriteBatch = new SpriteBatch(GraphicsDevice);

    // TODO: use this.Content to load your game content here
    ⋮
    objTxtr = content.Load<Texture2D>("texture");
    ⋮
}

protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    // TODO: Add your drawing code here
    ⋮
    // 3D描画
    GraphicsDevice.Vertices[0].SetSource(vertexBuf, 0,
        VertexPositionColorTexture.SizeInBytes);
    basicEffect.Texture = objTxtr; // BasicEffectに適用するテクスチャの設定

    basicEffect.Begin();
    basicEffect.CurrentTechnique.Passes[0].Begin();

    GraphicsDevice.DrawPrimitives(PrimitiveType.TriangleList, 0, 2);

    basicEffect.CurrentTechnique.Passes[0].End();
    basicEffect.End();
    ⋮
}

```

課 題

四角形を2つ表示し、それぞれ別のテクスチャを設定しましょう。