

## UNIT08:SPRITE – BLENDING

---

### 【学習要項】

- ☐Blend state
- ☐Alpha Channel
- ☐Blend mode formulas

### 【演習手順】

#### 1. 画像の背景色を透過させる

- ①framework クラスに以下のメンバ変数を追加する

```
ID3D11BlendState* blend_states[4];
```

- ②framework クラスのコンストラクタでブレンディングステートオブジェクトを作成する

```
1: D3D11_BLEND_DESC blend_desc{};
2: blend_desc.AlphaToCoverageEnable = FALSE;
3: blend_desc.IndependentBlendEnable = FALSE;
4: blend_desc.RenderTarget[0].BlendEnable = TRUE;
5: blend_desc.RenderTarget[0].SrcBlend = D3D11_BLEND_SRC_ALPHA;
6: blend_desc.RenderTarget[0].DestBlend = D3D11_BLEND_INV_SRC_ALPHA;
7: blend_desc.RenderTarget[0].BlendOp = D3D11_BLEND_OP_ADD;
8: blend_desc.RenderTarget[0].SrcBlendAlpha = D3D11_BLEND_ONE;
9: blend_desc.RenderTarget[0].DestBlendAlpha = D3D11_BLEND_ZERO;
10: blend_desc.RenderTarget[0].BlendOpAlpha = D3D11_BLEND_OP_ADD;
11: blend_desc.RenderTarget[0].RenderTargetWriteMask = D3D11_COLOR_WRITE_ENABLE_ALL;
12: hr = device->CreateBlendState(&blend_desc, &blend_states[0]);
13: _ASSERT_EXPR(SUCCEEDED(hr), hr_trace(hr));
```

- ③framework クラスの render メンバ関数の sprite オブジェクト描画の直前でブレンディングステートオブジェクトを設定する

```
immediate_context->OMSetBlendState(blend_states[0], nullptr, 0xFFFFFFFF);
```

#### 2. sprite クラスの render メンバ関数の色指定の引数が機能するように、ピクセルシェーダーに変更を加える

※色調の変更や、アルファ値を変化させフェードアウト等の効果を表現できるようにする

```
1: #include "sprite.hlsl"
2: Texture2D color_map : register(t0);
3: SamplerState point_sampler_state : register(s0);
4: SamplerState linear_sampler_state : register(s1);
5: SamplerState anisotropic_sampler_state : register(s2);
6: float4 main(VS_OUT pin) : SV_TARGET
7: {
*8:     return color_map.Sample(point_sampler_state, pin.texcoord) * pin.color;
9: }
```

#### 3. 実行し、キャラクタ画像の背景色が透過していることを確認する

#### 4. 各種ブレンドモード（無効・加算・減算・乗算など）を作成する

※Blend mode formulas.docx 等を参考に設定する

#### 5. 実行し、各種ブレンドモードの違いを確認する

※出力ウィンドウに COM オブジェクト未開放の警告が出るので、適切な場所で解放する

### 【評価項目】

- ☐透過処理
- ☐頂点カラー
- ☐ブレンドモード