

## UNIT07:SPRITE – DEPTH BUFFER

---

### 【学習要項】

- ☐Depth testing
- ☐Depth writing

### 【演習手順】

1. framework クラスの render メンバ関数の sprite オブジェクト描画を下記のように変更する  
※最後に描画した **sprite** オブジェクトが、画面に表示されないことを確認する

```
sprites[0]->render(immediate_context, 0, 0, 1280, 720, 1, 1, 1, 1, 0); //画像全体を画面全体に描画する
sprites[1]->render(immediate_context, 500, 200, 200, 200, 1, 1, 1, 1, 0, 0, 0, 140, 240);
```

2. sprite オブジェクトの描画順と画面上の前後関係を修正する  
※最後に描画した **sprite** オブジェクトが、画面の最前面に来るようにする  
①framework クラスに以下のメンバ変数を追加する

```
ID3D11DepthStencilState* depth_stencil_states[4];
```

- ②framework クラスの initialize メンバ関数で深度ステンシルステートオブジェクトを作成する  
※下記コードは深度テスト：オン、深度ライト：オンの設定である  
※深度テスト：オン・オフ、深度ライト：オン・オフの全4通りのオブジェクトを生成する

```
1: D3D11_DEPTH_STENCIL_DESC depth_stencil_desc{};
2: depth_stencil_desc.DepthEnable = TRUE;
3: depth_stencil_desc.DepthWriteMask = D3D11_DEPTH_WRITE_MASK_ALL;
4: depth_stencil_desc.DepthFunc = D3D11_COMPARISON_LESS_EQUAL;
5: hr = device->CreateDepthStencilState(&depth_stencil_desc, &depth_stencil_states[0]);
6: _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深度ステンシルステートオブジェクトを設定する  
※深度テスト：オフ、深度ライト：オフで生成したオブジェクトをセットする

```
immediate_context->OMSetDepthStencilState(depth_stencil_states[?], 1);
```

3. 実行し、後ろに隠れていたキャラクタの画像が表示することを確認する  
※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】

- ☐描画順序