

Simple Lighting サンプル

*このサンプルは、Windows 10 Anniversary Update SDK（14393）と互換性があります。*

# 概要

このサンプルでは、静的および動的ランバートライティングを使用して照らされたインデックス付きジオメトリを描画するために、静的Direct3D 11頂点、インデックス、および定数バッファを作成する方法を示します。

このサンプルは、立方体としても表される、2つのライト（1つは白で1つは赤）で照らされている大きな立方体をレンダリングします。白色光は静止し、赤色光は中央の立方体の周りを回転します。中央の立方体も回転します。この動きによって、色のついた光の効果をさまざまな角度から観察することができます。



# サンプルの使用

|  |  |  |
| --- | --- | --- |
| 作用 | ゲームパッド | キーボード |
| 終了 | 表示ボタン | ESC |

# 実装上の注意

このサンプルでは、シーンをレンダリングするために3つのシェーダ頂点シェーダ（ "TriangleVS"）と2つのピクセルシェーダ（ "LambertPS"、 "SolidColorPS"）を使用しています。コンパイルされたシェーダーブロブはCreateDeviceDependentResourcesにあり、シェーダーリソースを作成する際に参照されます。 すべてのシェーダは同じHLSLインクルードファイル "SimpleLighting.hlsli"で定義されていて、3つのスタブシェーダがこのファイルをインクルードします。各スタブシェーダは、3つのシェーダブロブを作成するために、異なるエントリポイントに対してコンパイルされます。

シーンのジオメトリは、静的な頂点バッファとインデックスバッファで構成され、それぞれにキューブの6つの四角形を表す24個の頂点のデータが格納されています。これら2つのバッファはCreateDeviceDependentResourcesに作成され、D3D11\_SUBRESOURCE\_DATA構造体の初期化データで提供されたデータですぐに埋められます。これは DirectX 11 の最も効率的な初期化方法です。

この非常に単純なシーンでは、すべてのシェーダ定数は、次のものを含む単一の定数バッファにまとめられます。

* ワールド、ビュー、射影マトリックス
* 光の方向と色
* 単色

より複雑なシーンでは、定数が更新される頻度に応じて、通常、定数を複数のバッファに分割します。

大きな立方体と赤い光がアニメーション化されているので、定数のいくつかはフレームごとに更新されなければなりません。定数バッファは、次の方法で更新されます:

1. D3D11\_MAP\_DISCARD を使用して ID3D11DeviceContext::Map を呼び出し、D3D11\_MAPPED\_SUBRESOURCEを作成します。
2. すべての定数の値をマップされたリソースにコピーします
3. DrawIndexed を呼び出す前にリソースのマッピングを解除

D3D11\_MAP\_DISCARDを使用する場合

* ドライバは基本的に新しいバッファを割り当てるので、シェーダが使用するバッファ内のすべての定数の更新の必要性を想定する必要があります。バッファに前回の更新からの有効なデータが含まれている可能性がある D3D11\_MAP\_WRITE\_NO\_OVERWRITE と比較してください。
* 以前にマップされたバッファがまだGPUによって使用されている場合は、新しいバッファを取得します。D3D11\_MAP\_DISCARD を使い過ぎると、使用可能なメモリーが増えるのを待つ必要があるため、結局失速する可能性があります。

## システム必要条件

このサンプルはWindows 8以降と互換性があります。

このサンプルは、DirectX 11.1 Runtimeがインストールされた（[KB 2670838](https://support.microsoft.com/ja-jp/help/2670838/platform-update-for-windows-7-sp1-and-windows-server-2008-r2-sp1)） Windows 7 Service Pack 1でも動作します。詳しくは、[このブログ記事](https://blogs.msdn.microsoft.com/chuckw/2013/02/26/directx-11-1-and-windows-7-update/)を参照してください。

# 更新履歴

最初のリリース、2018 年 4 月