

シンプルインスタンス作成サンプル（DX12）

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

# 概要

このサンプルは、Direct3D 12 API でインスタンス作成を使用する方法を示します。

# サンプルの使用



|  |  |  |
| --- | --- | --- |
| 作用 | ゲームパッド | キーボード |
| カメラを回転させる | 左サムスティック | WASD / 左マウスおよび左マウスボタン |
| インスタンス数の変更 | LB / RB | Q / E |
| 中央 | n/a | ホーム |
| シミュレーションをリセットする | A | スペース |
| 終了 | 表示ボタン | ESC |

# 実装上の注意

インスタンス化されたジオメトリをレンダリングする際に考慮すべき 4 つの領域があります。

1. **幾何データ**

このサンプルの場合、これには立方体の面を記述する頂点とインデックスを含む頂点とインデックスバッファーが含まれます。また、この立方体を世界にレンダリングするために必要な*パイプライン状態オブジェクト* も含まれます。これらすべてのコンポーネントは、インスタンス化されていないジオメトリと同じ方法で設定および操作されます。（SimpleInstancing12.cpp の *CreateDeviceDependentResources* を参照してください。）

1. **インスタンスデータ**

標準 D3D12 インスタンスレンダリングでは、インスタンスごとのデータが 1 つ以上の頂点バッファーを介して提供されます。これらの頂点バッファーは、その他の頂点バッファーと同じ方法で作成されます。このサンプルは 2 つの頂点バッファーを使用します。1 つは静的でインスタンスごとのカラーデータを含みます（サンプルの有効期間中変更がありません）。その他は動的でインスタンスごとの位置および向きの情報を含みます（フレームごとに変更）。

1. **インスタンス作成レイアウト**

インスタンス化されたジオメトリをレンダリングするには、D3D は提供された頂点データをどのように解釈するかについての情報が必要です。これは、*D3D12\_INPUT\_ELEMENT\_DESC* 構造の配列を使用して行います。標準レンダリングが行われる方法とほぼ同じです。ただし、この構造に他の要素が追加されます。幾何データは、通常通り *InputSlotClass* 要素に対して *D3D12\_INPUT\_CLASSIFICATION\_PER\_VERTEX\_DATA* 値のフラグが付きます。しかし、インスタンスごとのデータは *D3D12\_INPUT\_CLASSIFICATION\_PER\_INSTANCE\_DATA* 値を使用します。*InputSlot* 要素もデータがプルされる頂点ストリームを示すために使用されます。

頂点シェーダーは、幾何およびインスタンスごとのデータがすべてまとまっているように定義された頂点構造を使用します（*D3D12\_INPUT\_ELEMENT\_DESC* 配列に記載されたレイアウトを反映）。

***注:*** *このサンプルはデータを入力レイアウトに自動的に正しく配置するための* AlignedByteOffset *要素用* D3D12\_APPEND\_ALIGNED\_ELEMENT *定数です。これは問題の頂点バッファーの構造に正しく配置されたデータが含まれる場合にのみ機能します。頂点データ内で要素をスキップ（または無視）しようとする場合は、正確な配置オフセットが必要です。*

1. **レンダリング**

インスタンスデータのレンダリングは、前のポイントが確定されれば簡単です。*ID3D12GraphicsCommandList::IASetVertexBuffers* API は、入力（この場合は入力バッファー）として使用される頂点バッファーを設定するために使用され、*ID3D12GraphicsCommandList::DrawIndexedInstanced* API はレンダリングに使用されます。残りのレンダリング設定は、標準的なインスタンス化されていないレンダリングと同様に実行されます。

# 更新履歴

最初のリリース、2018 年 4 月