

遅延パーティクル

*このサンプルは、Microsoft ゲーム開発キットのプレビュー (2019 年 11 月) に対応しています。*

# 説明

この D3D12 サンプルは、前方または遅延のいずれかの方法で、光を当てた煙のパーティクルをレンダリングする方法を示します。遅延パスを利用すると、各パーティクルの法線、不透明度、および色が遅延バッファに蓄積されます。光を当てた結果は、最終段階で元のシーンに合成されます。



# サンプルのビルド

Xbox One の devkit を使用している場合は、アクティブなソリューション プラットフォームを Gaming.Xbox.XboxOne.x64 に設定します。

Project Scarlett を使用している場合は、アクティブなソリューション プラットフォームを Gaming.Xbox.Scarlett.x64 に設定します。

*詳細については、GDK ドキュメント*の「サンプルの実行」を*参照してください*。

# サンプルの使用

このサンプルでは、次のコントロールが使用されます。

|  |  |
| --- | --- |
| 動作 | コントローラー |
| サンプルを終了する | 選択 |
| カメラを移動する | 左/右スティック |
| 前方/遅延の切り替え | "A" ボタン |
| シミュレーションの一時停止 | "X" ボタン |

# 実装に関する注意事項

パーティクル シミュレーション自体が CPU に対して実行されます。結果として得られるパーティクルの位置は、レンダリング用に GPU メモリにコピーされます。一時的な D3D12 リソース内から、CPU に書き込み可能で GPU で読み取り可能なメモリが割り当てられます。このメモリへの GPU 仮想アドレスは、D3D12\_VERTEX\_BUFFER\_VIEW と ID3D12GraphicsCommandList::IASetVertexBuffers(...) を使用して頂点バッファーとして直接参照されます。

前方および遅延 - 2 つのレンダリング モードを使用できます。前方パスでは、パーティクルはシーンに直接レンダリングされます。遅延パスをたどると、パーティクル データは 2 つの遅延バッファ（法線マップと不透明度/カラー マップ）に蓄積されます。パーティクルのレンダリングが完了したら、これらの遅延バッファーを使用して、パーティクルに光を当て、メイン シーンに合成します。

パーティクルは、シーン ライト（単一の指向性ライト）によって照らされるだけでなく、爆発自体の内部のポイント ライトによっても照らされます。これらのポイント ライトは方向性がないため、パーティクルへのライティングの寄与は、方向ではなく、距離に対する二次減衰に基づきます。

# 更新履歴

2019 年 4 月 – 従来の Xbox サンプル フレームワークから新しいテンプレートに移植。

# プライバシーに関する声明

サンプルをコンパイルして実行すると、サンプルの使用状況を追跡するため、サンプル実行可能ファイルのファイル名が Microsoft に送信されます。このデータ収集を無効にするには、「Sample Usage Telemetry」とラベル付けされた Main.cpp 内のコードのブロックを削除します。

Microsoft のプライバシー方針の詳細については、「[Microsoft プライバシーに関する声明](https://privacy.microsoft.com/en-us/privacystatement/)」を参照してください。