

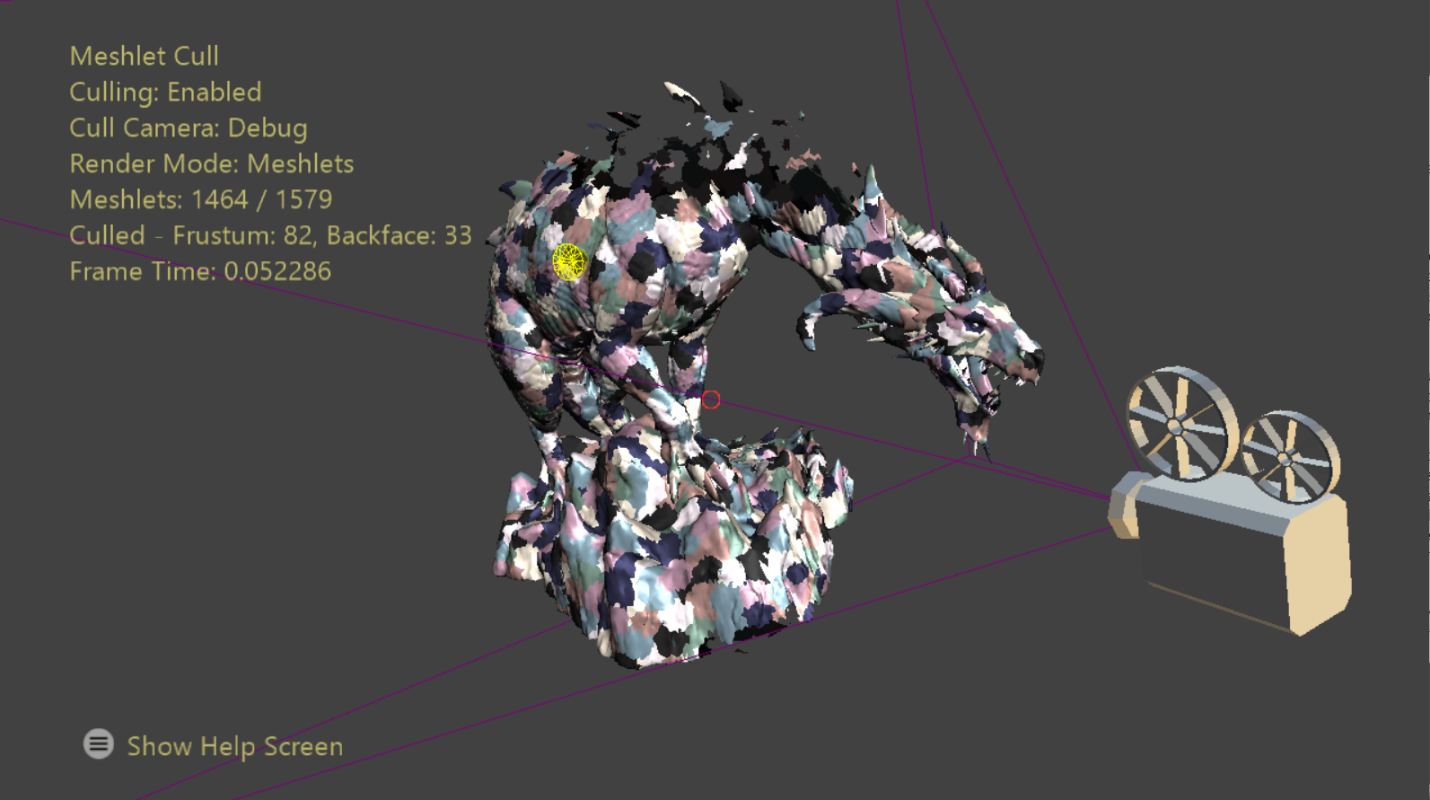
メッシュレット カリングのサンプル

*このサンプルは Microsoft Game Development Kit (2020 年 6 月) および Windows 10 (バージョン 2004) 2020 年 5 月更新プログラムと互換性があります*

# 説明

アンプリフィケーション シェーダーは、メッシュ シェーダー パイプラインのメッシュ シェーダー ステージの前にある省略可能なステージです。その目的は、特定の GPU タスクで必要なメッシュ シェーダー スレッドグループの必要数を決定し、オプションでディスパッチされた MS 子スレッドグループにデータのペイロードを渡すことです。これは、パイプラインのメッシュ シェーダー ステージに到達する前にワークロードを削減または拡張するために使用できます。

このサンプルは、メッシュレットごとのカリング メタデータを使用してカメラに対してメッシュレットをカリングするアンプリフィケーション シェーダーの活用方法を示します。目標は、起動前に表示される可能性があると考えられる部分のみに必要なメッシュ シェーダー スレッドグループの数を最小限に抑えることです。



# サンプルのビルド

Xbox Series X|S 開発キットを使用している場合、アクティブ 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。

適切なハードウェアと Windows 10 リリースで PC 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.Desktop.x64 に設定します。

このサンプルは、Xbox One をサポートしていません。

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

# サンプルの使用方法

サンプルの焦点は、アンプリフィケーション シェーダーベースのメッシュレット カリング手法を示しています。カリングのオン / オフを切り替えることができ、カリングが発生するカメラは、メイン カメラとデバッグ カメラの間で切り替えることができます。デバッグ カメラの位置と向きは、ワールド空間内のカメラ モデルによって表されます。カリング カメラの視錐台は、視錐台カリングの発生場所をより簡単に観察するために視覚化されます。

サンプルには、フラット シェーディングとメッシュレット視覚化という 2 つのレンダリング モードが用意されています。メッシュレット ビュー モードでは、シーンからメッシュレットを選択して、境界球とメッシュレット内の法線の幅を表すコーンを視覚化できます。メッシュレットの法線のコーンが縮退したコーンを形成する (半球よりも広い) 場合、コーンはレンダリングされません。 また、ユーザーは、何らかの種類のメッシュの詳細レベルを 6 段階に切り替えることもできます。

統計情報は、シーン オブジェクト レンダリングに対する GPU 経過時間とフレームからカリングされたメッシュレットの数が画面に表示されます。

# コントロール

|  |  |  |
| --- | --- | --- |
| 操作 | ゲームパッド | キーボード |
| カメラを移動する | 左サムスティック / DPad | WASD キーまたは方向キー |
| カメラを回転する | 右サムスティック | LMB 長押し + マウス |
| カメラのリセット | 右サムスティック (プッシュ) | N/A |
| デバッグ カメラのコントロール | 左ショルダー | 左 Shift キー |
| カリングの切り替え | A ボタン | Tab キー |
| レンダリング モードの変更 | X ボタン | スペース キー |
| カリング カメラの切り替え | B ボタン | Q |
| メッシュレットの選択 | Y ボタン | RMB |
| サイクル LOD | 右ショルダー / トリガー | +/- |
| ヘルプ メニューの表示 | メニュー ボタン | N/A |
| 終了 | ビュー ボタン | Esc キー |

# 実装上の注意

この手法は、メッシュレット生成時のカリング データの生成から始まります。メッシュレット リストが完了すると、各メッシュレットのオブジェクト空間の境界球と法線コーンが計算されます。この一連の手順を実行するアルゴリズムのサンプル コードは、別のサンプル MeshletConverter を介して提供されます。その後、このデータはディスク上のメモリと読み取り帯域幅を節約するため、量子化によって圧縮されます。

通常のコーンは、メッシュレット内の法線の*広がり*を表します - すなわちプリミティブのすべての法線を取り込むコーンです。それは、 4 タプルの正規化された浮動小数点値として格納されています - 平均法線方向を表す 3D 単位ベクトルと、そのベクトルと最も異なる三角形法線の間のドット積を表すスカラーです。

カリング テストのために、実際に格納されている値は -cos(*最大角度* + 90º) です。三角形は、半球全体 (または法線の約 90º) にわたって観察可能であり、反転されたコーンが、この表面法線の背面に面する一連のビュー方向を含み無効になるため、90º が追加されます。これにより、カリング テストが単一ドット積と浮動小数点の比較になります。

アンプリフィケーション シェーダーは、各スレッドで 1 つのメッシュレットを処理するように構成されています。したがって、 *n 個*のメッシュレットで構成されるメッシュをレンダリングするには、 スレッド グループをディスパッチする必要があります。スレッドグループ全体の操作にウェーブ組み込み関数を使用しやすくするために、プラットフォームのウェーブ サイズと等しいスレッドグループ サイズが選択されます。各スレッドは、メッシュレット (ディスパッチ スレッド ID でインデックス付けされている)に対してビュー視錐台と通常のコーン カリング テストを実行する責任があります。プレフィックス合計ウェーブ組み込み関数を使用すると、表示されるメッシュレットのインデックスは、起動済みのメッシュ シェーダー スレッドグループのグループ共有メモリ ルックアップ テーブルに圧縮されます。シェーダーはアンプリフィケーション シェーダー組み込み関数 DispatchMesh の呼び出しで終了し、必要な数のメッシュ シェーダー スレッドグループを起動し、グループ共有参照テーブルをペイロード データとして指定します。

# 更新履歴

2020 年 4 月 20 日 – サンプル作成。

2020 年 4 月 28 日 - メッシュ シェーダー パイプラインの作成に D3DX12 ヘルパーを使用するように更新

# プライバシー ステートメント

サンプルをコンパイルして実行する場合、サンプルの使用状況を追跡するために、サンプル実行ファイルのファイル名が Microsoft に送信されます。このデータ コレクションからオプトアウトするには、Main.cpp の「Sample Usage Telemetry」というラベルの付いたコードのブロックを削除します。

全般的な Microsoft のプライバシー ポリシーの詳細については、「[Microsoft プライバシー ステートメント](https://privacy.microsoft.com/en-us/privacystatement/)」を参照してください。