



ライティングを有効に活用するのに最も信頼性の高いガイド

# HD レンダーパイプライン (HDRP)

# 目次

はじめに .....	6
<b>HDRP ライティングの概要 .....</b>	<b>7</b>
<b>インストール .....</b>	<b>8</b>
システム要件 .....	8
Unity Hub .....	9
Package Manager のインストール .....	10
<b>HDRP のサンプルシーン .....</b>	<b>12</b>
その他の HDRP のサンプルコンテンツ .....	14
<b>Project Settings .....</b>	<b>16</b>
「Graphics」設定 .....	16
「Quality」設定 .....	17
HDRP の最適化 .....	19
HDRP 機能の有効化 .....	19
<b>フォワードレンダリングとディファードレンダリング .....</b>	<b>20</b>
レンダリングパスのカスタマイズ .....	21
レンダリングパスに関する詳細 .....	21
フォワードレンダリング .....	21
ディファードシェーディング .....	22
<b>アンチエイリアシング .....</b>	<b>24</b>
マルチサンプルアンチエイリアシング (MSAA) .....	24
ポストプロセスでのアンチエイリアシング .....	25
<b>ボリューム .....</b>	<b>27</b>
ローカルとグローバル .....	28
ボリュームプロファイル .....	29
Volume オーバーライド .....	30

オーバーライドのワークフロー .....	31
ブレンディングと優先度 .....	32
<b>露出.....</b>	<b>33</b>
露出値を理解する .....	33
露出値の計算式.....	34
Exposure オーバーライド .....	35
「Fixed」モード .....	35
「Automatic」モード .....	36
「Metering Mode」のオプション.....	37
Automatic Histogram .....	37
カーブマッピング.....	38
Physical Camera .....	39
「Physical Camera」のその他のパラメーター.....	40
<b>ライト.....</b>	<b>41</b>
ライトのタイプ.....	41
形状.....	42
色と温度.....	43
追加のプロパティ.....	43
ライトレイヤー.....	44
ライトアンカー.....	46
<b>物理ベースのライト単位と強度.....</b>	<b>47</b>
単位.....	47
ライティングと露出の一般的な値 .....	48
IES プロファイルとクッキー .....	48

<b>環境ライティング</b> .....	<b>49</b>
HDRI Sky .....	50
クラウドレイヤーのアニメーション化.....	51
Gradient Sky .....	51
Physically Based Sky .....	52
色のヒント .....	52
<b>フォグと大気散乱</b> .....	<b>53</b>
グローバルフォグ.....	54
ボリュメトリックライティング .....	55
ローカルボリュメトリックフォグ .....	57
<b>雲</b> .....	<b>59</b>
クラウドレイヤー.....	59
ボリュメトリッククラウド.....	61
<b>シャドウ</b> .....	<b>63</b>
シャドウマップ.....	63
シャドウカスケード .....	63
コンタクトシャドウ .....	65
Micro Shadows .....	66
<b>リフレクション</b> .....	<b>67</b>
スクリーンスペースリフレクション .....	67
リフレクションプローブ .....	69
最適化のヒント.....	69
平面リフレクションプローブ.....	70
スカイリフレクション .....	71
リフレクション階層 .....	71
リフレクションプロキシボリューム .....	72
<b>リアルタイムライティングエフェクト</b> .....	<b>73</b>
Screen Space Ambient Occlusion .....	73
スクリーンスペースグローバルイルミネーション.....	74
Screen Space Refraction .....	75

<b>ポストプロセス</b> .....	<b>76</b>
「Post-processing (ポストプロセス)」での オーバーライド.....	77
トーンマッピング.....	77
Shadows、Midtones、Highlights .....	78
Bloom.....	79
Depth of Field .....	80
White Balance .....	81
Color Curves .....	82
Color Adjustments .....	82
Channel Mixer.....	83
Lens Distortion .....	83
Vignette.....	84
Motion Blur .....	84
<b>レンズフレア</b> .....	<b>85</b>
<b>動的解像度</b> .....	<b>87</b>
NVIDIA DLSS (NVIDIA RTX GPU および Windows 向け).....	87
AMD FSR (クロスプラットフォーム).....	87
TAA Upscale (クロスプラットフォーム) .....	88
<b>Rendering Debugger</b> .....	<b>89</b>
<b>レイトレーシング (プレビュー)</b> .....	<b>93</b>
設定.....	93
オーバーライド.....	94
その他のリソース.....	98
<b>さらに先へ</b> .....	<b>99</b>

# はじめに

光を扱うことがすべてのはじまりです。

Unity は HD レンダーパイプライン (HDRP) を開発しましたが、これはクリエイターがハイエンド PC やゲーム機などの性能を発揮し、新境地となるリアリティあふれるグラフィックスを生み出してクリエイターの描く構想を具現化する、ということを目指したものでした。

このガイドでは、Unity をご利用のアーティストやクリエイターの方向けに、HDRP の物理ベースでのライティングの手法を紹介いたします。HDRP は Unity のリアルタイムレンダリングの技術的な進歩を象徴するものであり、現実世界と同じようなライティングを実現することができます。

みずみずしく生い茂る熱帯雨林の風景に自然な太陽光を当ててレンダリングしたり、かと思えば都会のコンクリートジャングルではネオンのエミッシブライティングをふんだんに使って雰囲気を出したり…きっとあなたの作りたい雰囲気に合ったものもあるでしょう。HDRP なら、映画の撮影監督のようにシーンを彩ることが可能です。

見慣れた場所から幻想的な世界へプレイヤーを連れていけるようなゲーム環境を構築できます。

さあ、HDRP と Unity を使ってみましょう！



*Book of the Dead* では、HDRP を使用して雰囲気のあるライティングを作成しています。

# HDRP ライティングの概要

HDRP は、多様な機能で Unity の既存のライティングシステムを拡張し、あなたの作るシーンのレンダリングをより現実世界のライティングに近いものにしてくれます。

- **物理ベースのライト単位と露出**：HDRP では、現実世界の光の強度と単位を使用します。8 既知の光源の基準に合わせて、物理カメラを使用して露出を設定します。
- **高度なライティング**：スポットライトやエリアライト用の新しい形状オプションを使用して、ライトの配置を制御します。ライトレイヤーを使用して、特定のメッシュへのライトの影響を制限します。スクリーンスペースグローバルイルミネーションや Screen Space Refraction などのリアルタイムエフェクトを適用できます。
- **空の景色**：多彩な技術で自然な空を創造することができます。Physically Based Sky システムを使用して、惑星の大気のシミュレーションを手続き型で行ったり、ボリュメトリッククラウドやクラウドレイヤーを追加したり、HDRI を適用して静的な空のシミュレーションを行ったりできます。
- **フォグ**：フォグを使用して、シーンに奥行きや立体感を加えます。ボリュメトリックを有効にして、フォグエフェクトをフォアグラウンドオブジェクトに組み込むことで、シネマティックな光の軌跡をレンダリングできます。ボリュメトリックライトやシャドウをライトごとに管理して、ローカルボリュメトリックフォグコンポーネントを使用し、3D マスクテクスチャーでフォグの密度を微細に調整することが可能です。
- **ボリュームシステム**：HDRP には直観的なシステムが採用されており、カメラの位置や優先度に応じて、さまざまなライティングエフェクトや設定を遮断できます。ボリュームを重ねたりブレンドしたりすることで、シーン内のあらゆる領域を非常に細かく制御できます。
- **ポストプロセッシング**：HDRP のポストプロセッシングは、既存のボリュームシステムからうまく処理を行い、Volume オーバーライドにてまとめて管理されています。アンチエイリアシング、トーンマッピング、カラーグレーディング、被写界深度、その他のさまざまなエフェクトを追加できます。
- **高度なシャドウ**：HDRP では、ビジュアルの美しさとパフォーマンスの兼ね合いを考慮しつつもシャドウを制御することができる高度な機能を使用することができます。色合いやフィルタリング、解像度、メモリバジェット、更新モードを調整できます。コンタクトシャドウや微細なシャドウを利用して、細かいディテールやさらなる深みを際立たせましょう。
- **高度なリフレクション**：さまざまな手法を使用して反射面をレンダリングできます。リフレクションプローブでは、従来からあるリフレクションマッピング手法に加え、フラットな表面向けのみならず高度なオプションが用意された平面リフレクションプローブを使用できます。さらにスクリーンスペースリフレクション (SSR) によって、深度バッファを使用したリアルタイム手法を利用できます。

**拡張性**：HDRP は Unity の [スクリプタブルレンダーパイプライン](#) を基盤にしています。経験豊富なテクニカルアーティストやグラフィックスプログラマーであれば、パイプラインを標準機能以外にも拡張できます。

HDRP に触れるのがまったく初めてという方は、「[Up and Running with HDRP](#)」をご覧ください。

# インストール

Unity 2021 LTS 以降では、常に検証済みの最新のグラフィックスコードを利用しただけのよう、インストールファイルに HDRP パッケージが含まれています。Unity の最新リリースをインストールすると、それに対応するバージョンの HDRP もインストールされます。

HDRP のパッケージバージョン	互換性のある Unity バージョン
13.x	2022.1
12.x	2021 LTS (このガイドで使用)

HDRP グラフィックスパッケージを Unity の特定のリリースに関連付けることで、互換性を確保しやすくなります。ただし、[マニフェストファイル](#) をオーバーライドすることで、HDRP のカスタムバージョンに切り替えることもできます。

## システム要件

現在、HDRP に対応するターゲットプラットフォームは次のとおりです。

- DirectX 11 または DirectX 12 と Shader Model 5.0 対応の Windows および Windows ストア
- 最新のコンソール (Sony PlayStation 4 または Microsoft Xbox One 以上)
- Metal グラフィックス対応の MacOS (バージョン 10.13 以降)
- Vulkan 対応の Linux および Windows プラットフォーム

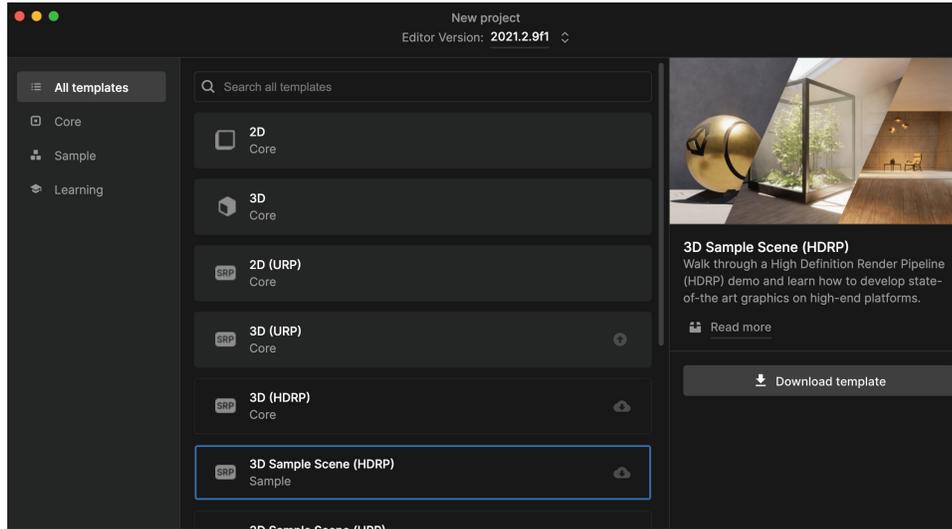
**HDRP は、コンピュータシェーダー対応のコンソールおよびデスクトッププラットフォームでのみ機能します。HDRP は、OpenGL または OpenGL ES デバイスには対応していません。**詳しくは、[要件と互換性に関するドキュメント](#) をご覧ください。

対応している VR プラットフォームおよびデバイスについては、「[HD レンダーパイプラインの VR](#)」をご覧ください。

## Unity Hub

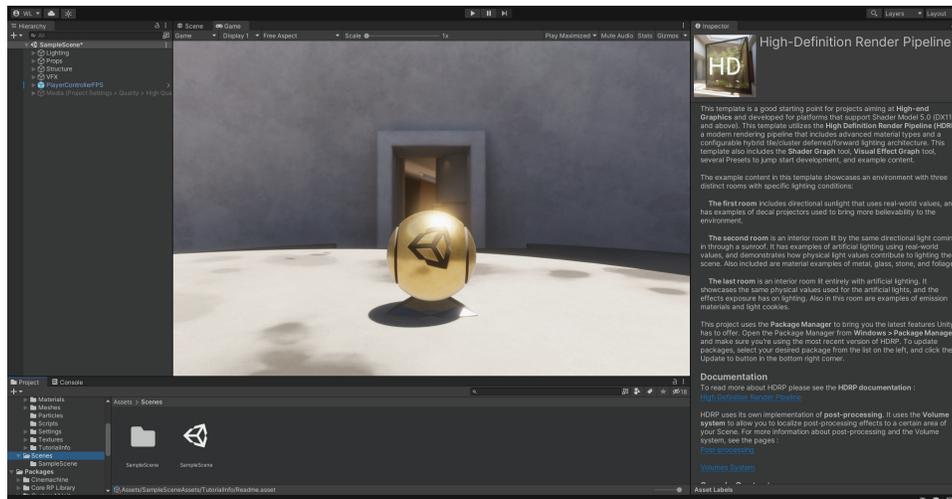
Unity Hub を使用するのが、最も簡単に HDRP プロジェクトを設定する方法です。

まず、新しいプロジェクトを作成します。選択できるテンプレートの中から「3D (HDRP)」という空のテンプレートまたは「3D Sample Scene (HDRP)」を選択します (旧バージョンの Hub での呼称は「High Definition RP」です)。最新のものを選択して、サンプルのプリセットを含む HDRP パッケージをインポートしてください。



「3D Sample Scene (HDRP)」テンプレートを選択します。

「Sample Scene」をロードします。次のようなシーンが表示されます。

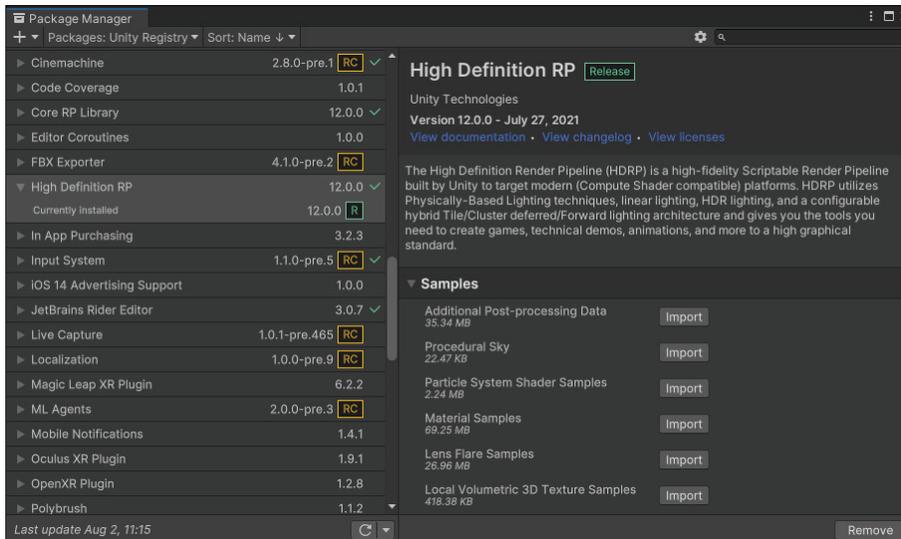


「3D Sample Scene」のプロジェクト設定

## Package Manager のインストール

「**3D Core template**」を使用してプロジェクトを作成した場合は、以前までの**ビルトインレンダーパイプライン**が使用されることになります。**パッケージマネージャー**（「**Window > Package Manager**」）からであれば HDRP に手でプロジェクトを移行することができます。

Unity Registry で「High Definition RP」パッケージを確認（または検索フィールドを使用して確認）し、インストールします。

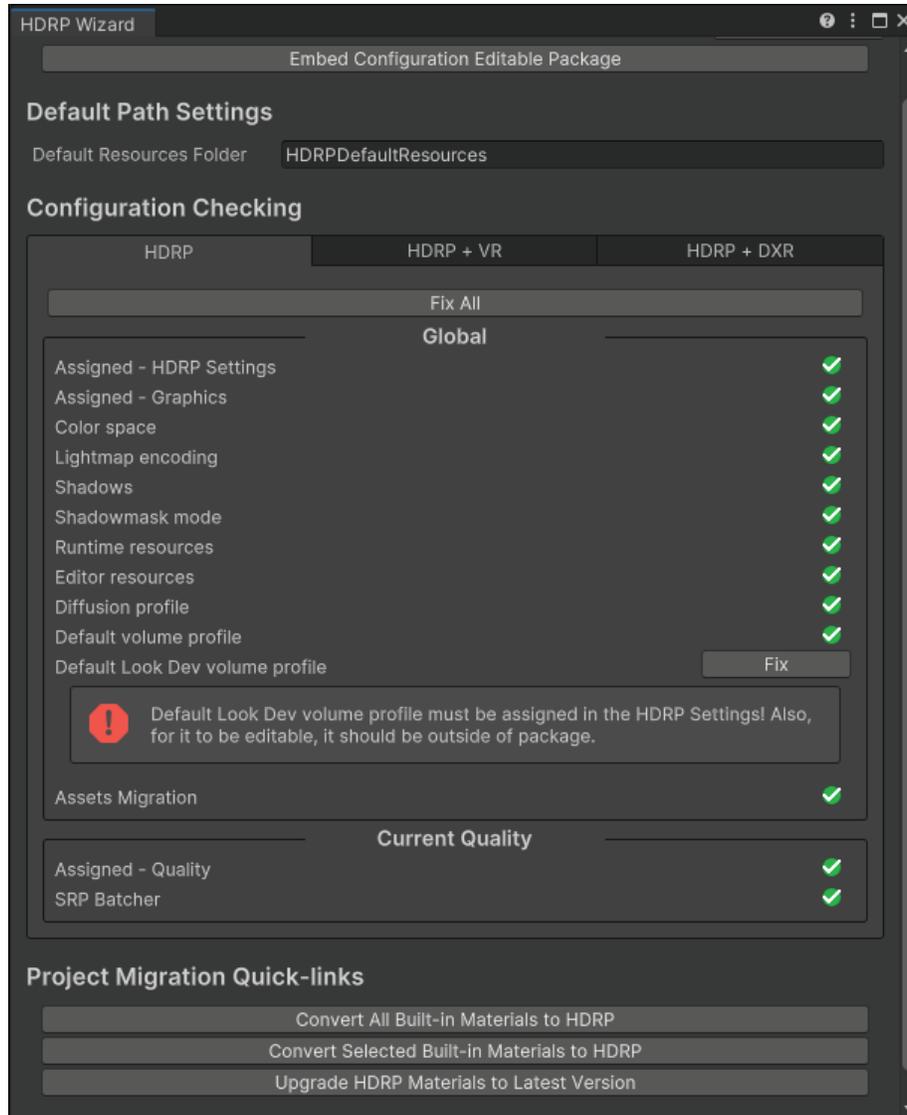


Package Manager のインストール

現行のプロジェクト設定で競合が発生した場合には、トラブルシューティングに対応した「**HDRP Render Pipeline Wizard**」が表示されます（「**Window > Rendering > HDRP Wizard**」で開くこともできます）。<sup>1</sup>

「**Configuration Checking**」で「**Fix All**」をクリックするか、「**Fix**」をクリックしてそれぞれの問題を個別に修正します。このチェックリストは、非 SRP プロジェクトからの移行時にも役立ちます。

<sup>1</sup> Unity 2021.2/HDRP 12 では、「**Window > Rendering > HDRP Wizard**」にあります。



HDRP ウィザード

ウィザードでの操作が完了すると、新規 HDRP パイプラインアセットの作成を促すプロンプトが表示されます。対象のファイルはディスク上の、設定したこの特定のパイプライン設定を保存するためのものになります。「**Create One**」を選択して、新しいレンダerpラインアセットを追加し、ファイルを割り当てます。

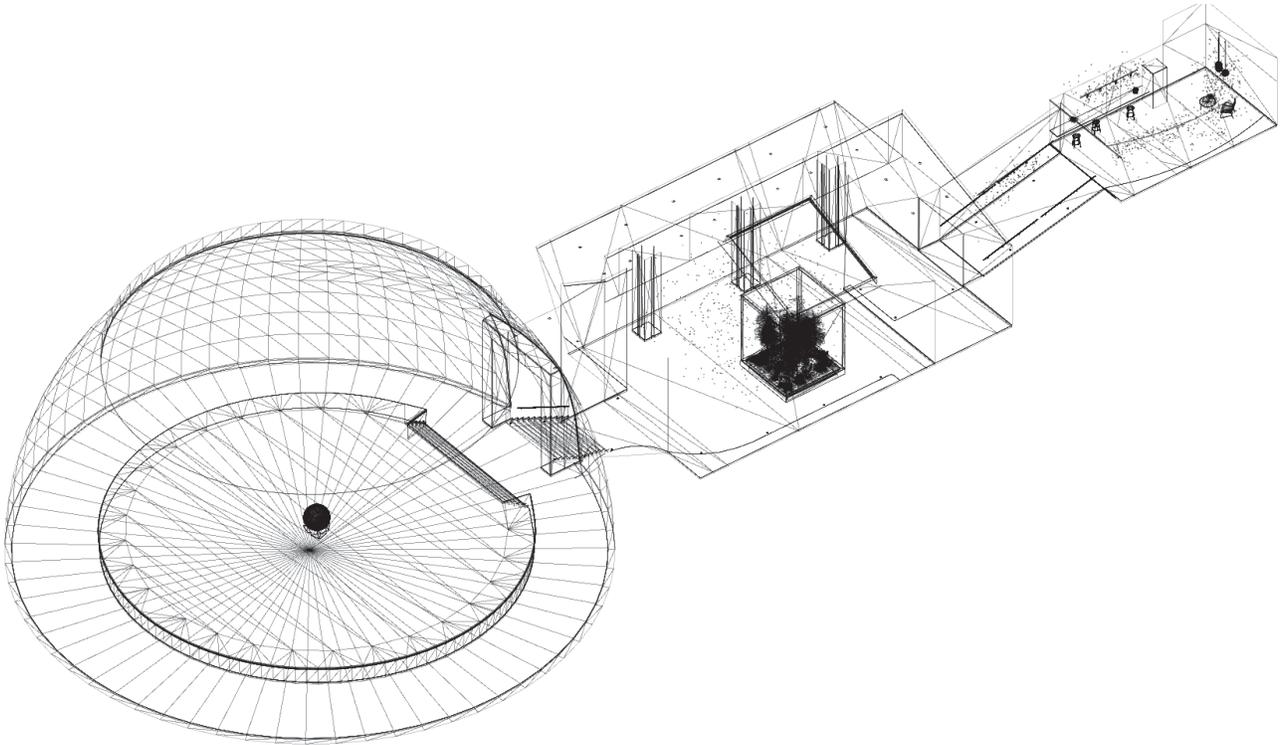
HDRP が適切に機能するようになると、「Configuration Checking」内のチェックボックスがすべて緑色になり、背景環境の色調がはっきりと変わります。

なお、空白のプロジェクトを使って手動でインストールを行った場合は、「**3D Sample Scene (HDRP)**」がインポートされません。このガイドに出てくるサンプルアセットを使用したい場合は、「3D Sample Scene」テンプレートを使用してください。

# HDRP のサンプルシーン

Unity Hub から入手できる「3D Sample Scene」は、HDRP や物理ベースのライティングの入門用テンプレートとしてのプロジェクトです。これは軽量のプロジェクトで容量が 100 メガバイトにも満たないため、いつでも気軽にロードして参照することができる優れた HDRP サンプルです。

このガイドでは、このプロジェクトを使用して、HDRP のさまざまな機能を説明します。



「3D Sample Scene」の環境を表すワイヤーフレーム

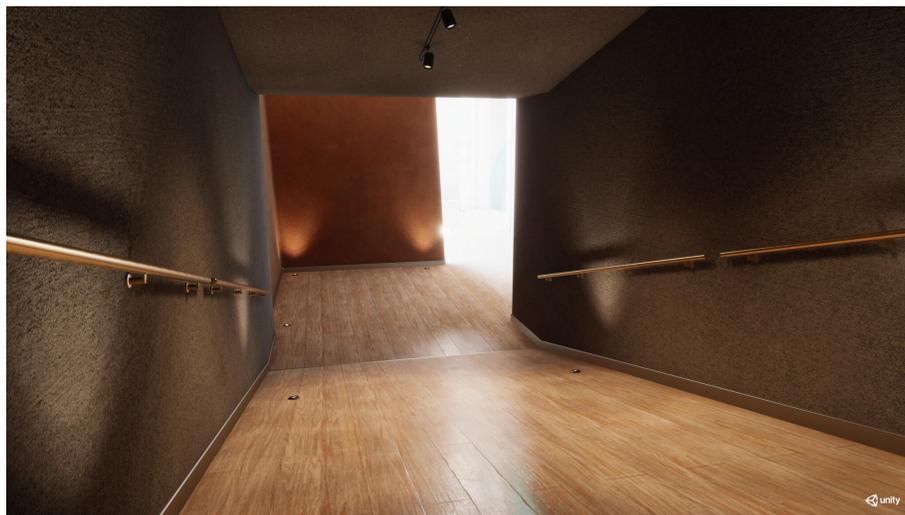
小さな部屋が繋がった構成になっており、ライティングの設定が異なる 3 種類のエリアがあります。日光を表すディレクショナルライトは、現実世界と同じ強度である 100,000 ルクスに設定されており、各部屋のライティング環境に合わせてカメラの露出が補正されています。

WASD キーとマウスを使用して、FPS Controller を移動させてみましょう。



「3D Sample Scene」は3つの部屋で構成されています。

- **Room 1** は、頭上からの日光に照らされた円形の舞台です。デカルによって、コンクリートの床に埃や水たまりが追加されています。
- **Room 2** は、空の光からのボリュームトリックライトシャフトのほか、ガラスケース内の木の高度なマテリアルが追加されています。
- **Room 3** は、室内の人工照明とエミッシブマテリアルのディスプレイです。



このサンプルシーンは、HDRP の機能を説明するための軽量プロジェクトです。

HDRP の「3D Sample Scene」の詳細については、Unity テクニカルアーティストの Pierre Yves Donzallaz が執筆したこちらの[ブログ記事](#)をご覧ください。このテンプレートのシーンの詳細について説明しています。

## その他の HDRP のサンプルコンテンツ

HDRP の「3D Sample Scene」には、参考になるプロジェクトが他にも用意されています。

「**Auto Showroom**」は本来ゲーミング向けのプロジェクトではありませんが、細部まで作り込まれた自動車リアルにライティングされている様子を見ることができます。インタラクティブなデモになっており、ステージのライト、車の塗装マテリアル、背景を変更したりすることができます。このプロジェクトは Unity Hub で入手できます。



「Auto Showroom」 テンプレート

「**Spaceship Demo**」は Visual Effect Graph のデモを目的としたプロジェクトですが、SF 的な環境で多数の HDRP 機能が使用されています。このプロジェクトは、Unity の [GitHub リポジトリ](#) からダウンロードできるほか、[Steam](#) からプレイアブルを入手できます。



「Spaceship」 デモ

VRでHDRPを使用する場合は、「[VR Alchemist Lab](#)」が参考になります。このプロジェクトでは、中世の西洋の小さな研究室の中で、インタラクティブなエフェクトが使用されています。



「VR Alchemist Lab」 デモ

映画のような絵作りやアニメーション映像の制作方法を学びたい場合は、Unity Hubから Cinematic Studio テンプレートをインストールしてください。Mich-L という一風変わったショートムービーを例に、スタイリッシュなレンダリングと写実的なレンダリングを組み合わせたシーン中のショットの設定方法やライティング方法を説明しています。



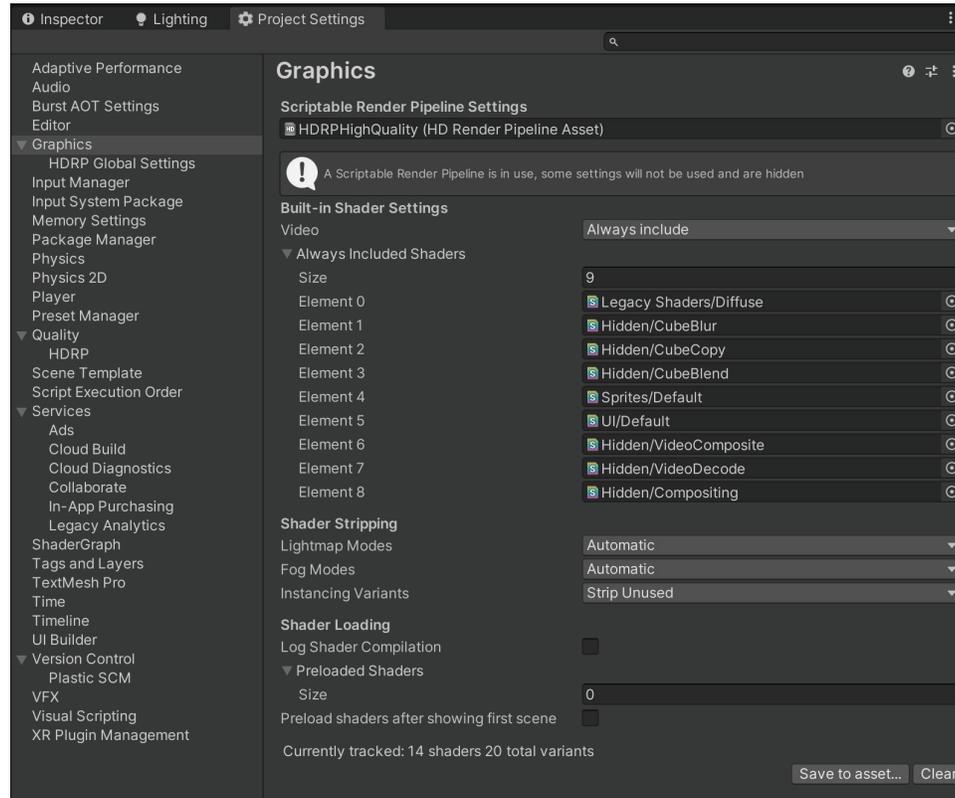
Cinematic Studio Sample

HD レンダーパイプラインについても学べますので、このようなプロジェクトをぜひ参照してください。

# Project Settings

「Project Settings」（「Edit > Project Settings」）の「Graphics」、「HDRP Global Settings」、「Quality」には、欠かせない基本設定が数点あります。

注：「HDRP Default Settings」は、Unity 2021.2/HDRP 12 以降では「HDRP Global Settings」になっています。



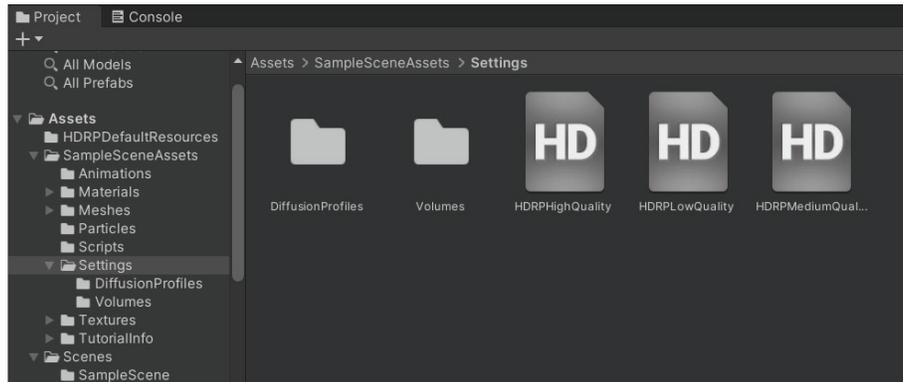
Project Settings

## 「Graphics」設定

最上部の「**Scriptable Render Pipeline Settings**」フィールドは、すべての HDRP 設定が保管されるディスク上のファイルを意味します。

プロジェクトごとに複数の**パイプラインアセット**を使用できます。各アセットが別々の設定ファイルだと考えてください。たとえば、各種ターゲットプラットフォーム（Xbox、PlayStation など）の専用設定を保管するために使用したり、プレイヤーがランタイム時に切り替えることができる個別のビジュアルクオリティーレベルを示すために使用したりできます。

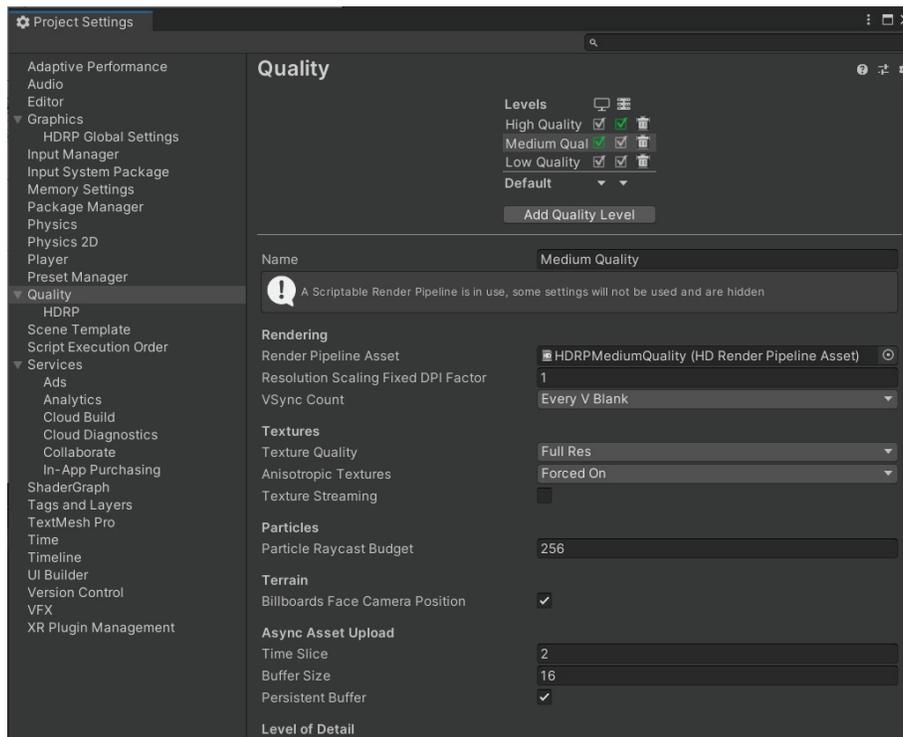
「3D Sample Scene」の場合、まず「Settings」フォルダー内に「HDRPHighQuality」、「HDRPLowQuality」、「HDRPMediumQuality」の各パイプラインアセットがあります。さらに、「HDRPDefaultResources」フォルダーには「DefaultHDRPAsset」があります。



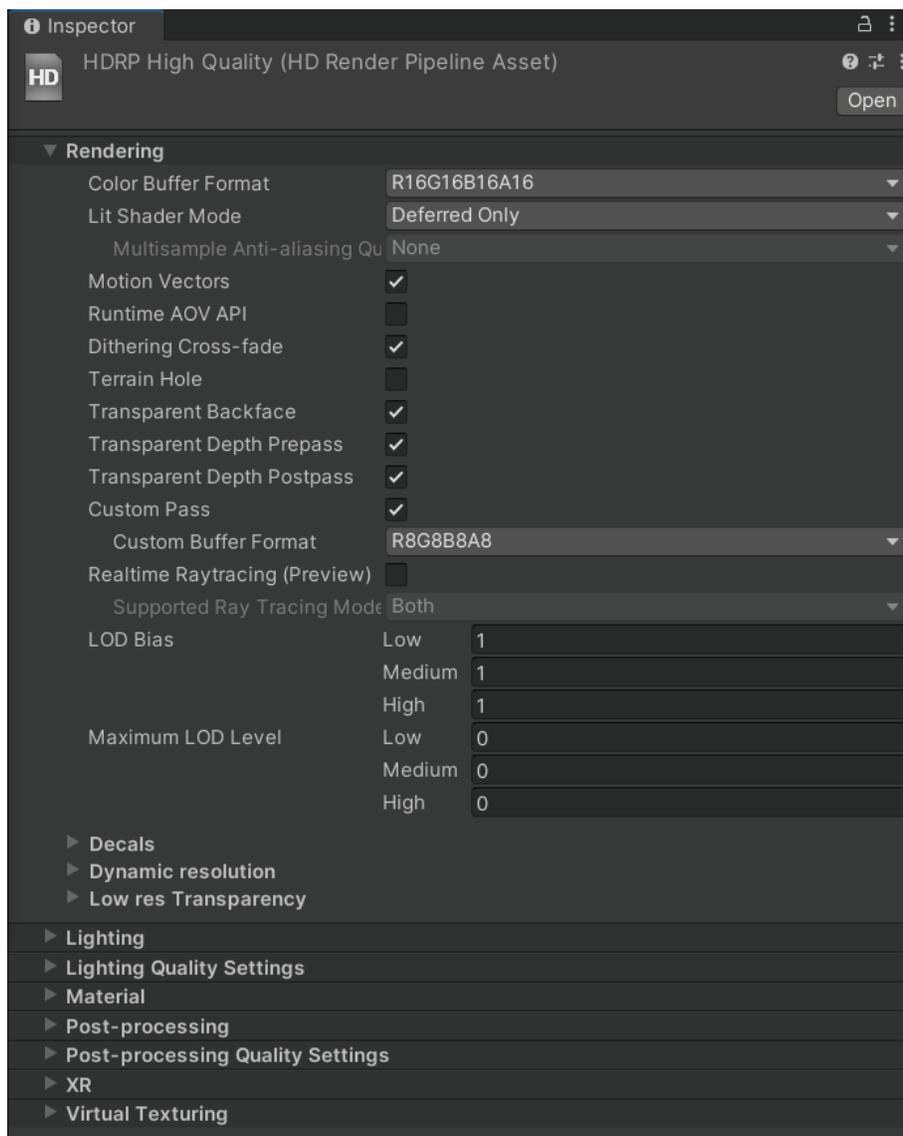
「3D Sample Scene」には、低品質、中品質、高品質のパイプラインアセットが含まれています。

## 「Quality」設定

「Quality」設定では、任意のパイプラインアセットを、事前に定義した品質のレベルへ対応させることができます。最上部の「Level」を選択して、特定のレンダerpラインアセットをアクティベートします。これは「Rendering」オプションに表示されます。



最上部でクオリティレベルを選択し、パイプラインアセットを有効にします。



パイプラインアセットの編集

デフォルト設定をカスタマイズしたり、品質レベルを追加したりした後で、追加のパイプラインアセットとペアリングできます。

品質レベルは、パイプライン内でアクティブ状態になっている特定のビジュアル機能を表します。たとえば、1つのアプリケーションに複数のグラフィックスレベルを作成できます。ランタイム時には、プレイヤーがハードウェアに合わせてアクティブな品質レベルを選択できます。

「Quality/HDRP」のサブセクションで、実際のパイプライン設定を編集できます。Project ウィンドウでパイプラインアセットを選択し、インスペクターで設定を編集することもできます。

## HDRP の最適化

パイプラインアセットで多くの機能を有効にすればするほど、消費リソースも増える点にはご注意ください。一般的には、目標とする効果を実現するために必要なものだけを使用するように、プロジェクトを最適化します。不要な機能がある場合は、オフにすることで、パフォーマンスを高めてリソースを節約できます。

不要な場合は無効にしても問題ない機能の典型例は次のとおりです。

- 「**HDRP Asset**」: デカル、低解像度透明度、透明バックフェース / デプスプリパス / デプスポストパス、SSAO、SSR、コンタクトシャドウ、ポリュメトリック、サブサーフェスキャタリング、ディストーション
- カメラの「**Frame Settings**」(メインカメラ、リフレクションのような統合エフェクトなどに使用するカメラ、またはカスタムしたエフェクトに対して使用している追加カメラ): 屈折、ポストプロセス、ポストプロセス後、トランスミッション、リフレクションプローブ、平面リフレクションプローブ、ビッグタイトルプリパス

HDRP 設定によるパフォーマンス強化について詳しくは、こちらの[ブログ記事](#)でお読みいただけます。

## HDRP グローバル設定

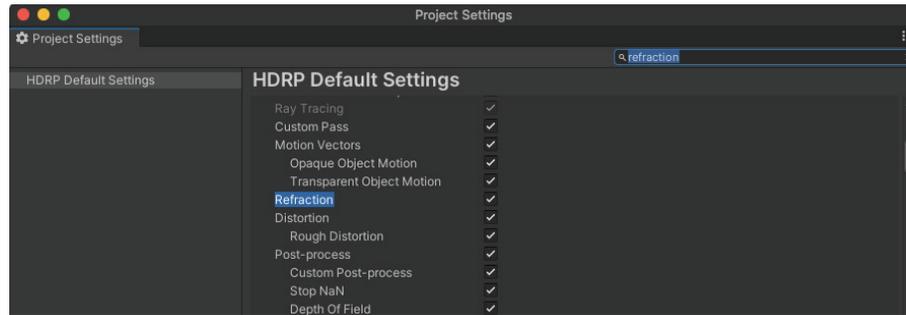
プロジェクトの基本設定は、「**HDRP Global Settings**」セクション (バージョン 12 より前では「**HDRP Default Settings**」セクション) で指定します。これらの設定は、シーン内で、カメラの位置に応じてローカルまたはグローバルボリュームコンポーネントを配置することによりオーバーライドできます (後述の「ボリューム」を参照)。

「Global Settings」は、最上部のフィールドで定義されている個別のパイプラインアセットに保存されます。ここでデフォルトのレンダリングおよびポストプロセスのオプションを設定します。

## HDRP 機能の有効化

プロジェクトを進めている時に、場合によっては、「**Global**」設定へ戻って、特定の機能を切り替えてオンまたはオフに設定することが必要になる場合もあります。一部の機能は、「**HDRP Global Settings**」内の対応するチェックボックスをオンにしておかないとレンダリングされることはありません。有効にするのは、必要な機能に限定してください。レンダリングパフォーマンスやメモリ使用量に悪影響を与える可能性があります。また、用途によって、ある設定は「**Volume Profiles**」に、また別の機能は「**Frame Settings**」に表示されていたりします。

HDRP のさまざまな機能について調べる際は、「Project Settings」の右上にある検索フィールドを活用しましょう。表示対象が関連するパネルのみに絞られるようになり、検索用語が強調表示されます。

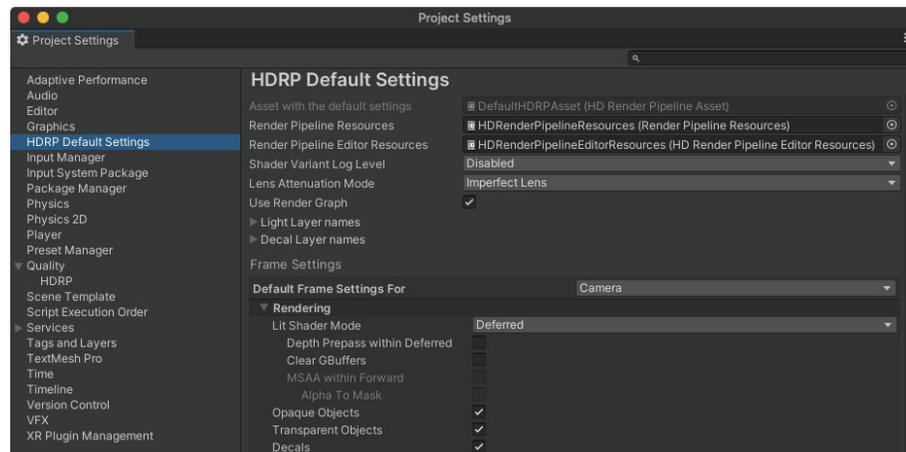


HDRP 機能の検索

「HDRP Global Settings」で機能を有効にしたからといって、どのカメラでもいつでもレンダリングできるわけではありません。「**Project Settings > Quality**」で品質レベルが選択されているレンダーパイプラインアセットについても、その機能をサポートしていることを確認する必要があります。例えば、カメラでボリュメトリッククラウドをレンダリングできるようにするには、「**HDRP Global Settings > Frame Settings > Camera > Lighting**」と、アクティブなレンダーパイプラインアセットの「**Lighting > Volumetrics**」で、それらを切り替える必要があります。

# フォワードレンダリングとディファードレンダリング

パイプラインアセットで HDRP 設定を指定する場合、通常は「**Rendering**」の「**Lit Shader Mode**」を開きます。ここでは、「**Deferred**」、「**Forward**」、または「**Both**」を選択できます。これらは、パイプラインによるジオメトリのレンダリングおよびライティング方法に関連する一連の処理であるレンダリングパスを表します。



デフォルトの HDRP 設定の変更

## レンダリングパスのカスタマイズ

「Lit Shader Mode」で「Forward」または「Deferred」を選択し、デフォルトのレンダリングパスを設定します。

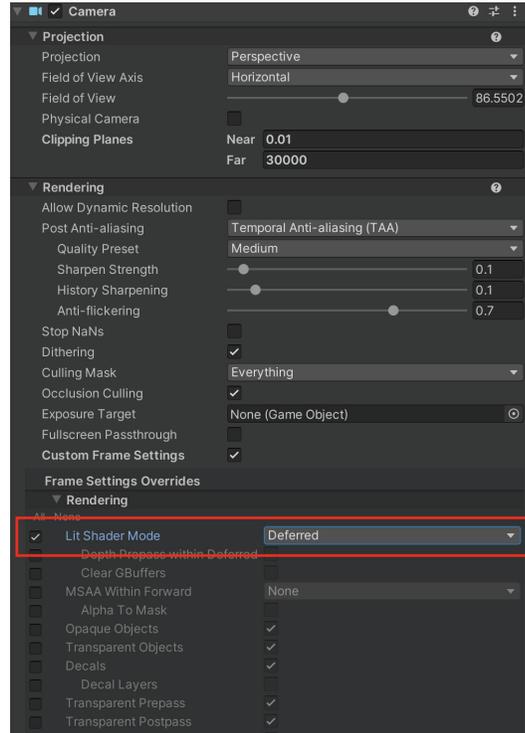
HDRP は柔軟性が高いので「Both」を選択することもできます。このオプションを利用した場合はほとんどのレンダリングにレンダリングパスを 1 つ使用することとなり、カメラごとにそのパスをオーバーライドできるようになります。ただし、この方法では GPU メモリの使用量が多くなってしまいます。ほとんどのケースでは、「Forward」か「Deferred」のいずれかを選択するのが有効です。

- デフォルトですべてのカメラを対象にするには、「HDRP Default Settings」に移動し、「Default Frame Settings」を確認します。ここでは、「Camera」、「Baked or Custom Reflection」、または「Realtime Reflections」を対象に設定できます。

「Rendering」グループで、「Lit Shader Mode」のレンダリングパスを設定します。

- 特定のカメラでデフォルト設定をオーバーライドするには、「Custom Frame Settings」をオンにします。

続いて、「Rendering」グループで、「Lit Shader Mode」のレンダリングパスをオーバーライドして変更します。



カメラのカスタムフレーム設定の変更

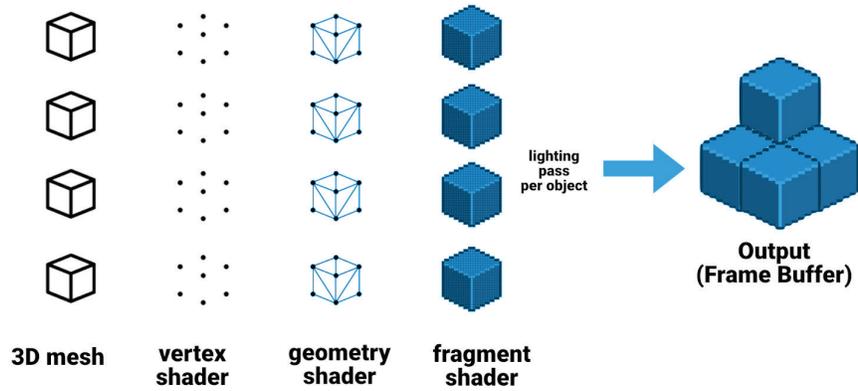
### レンダリングパスに関する詳細

こうしたレンダリングパスの仕組みを理解し、「Lit Shader Mode」がパイプラインのその他の設定にどのように影響するのかを確認することをおすすめします。

#### フォワードレンダリング

フォワードレンダリングでは、グラフィックスカードによって画面上のジオメトリが頂点に分割されます。これらの頂点は、さらにフラグメントやピクセルに分割され、それが画面にレンダリングされて最終的な画像が作成されます。

各オブジェクトは 1 度に 1 つずつグラフィックス API に渡されます。フォワードレンダリングには、各ライトのコストが伴います。シーン内のライトが増えると、レンダリングの所要時間も長くなります。



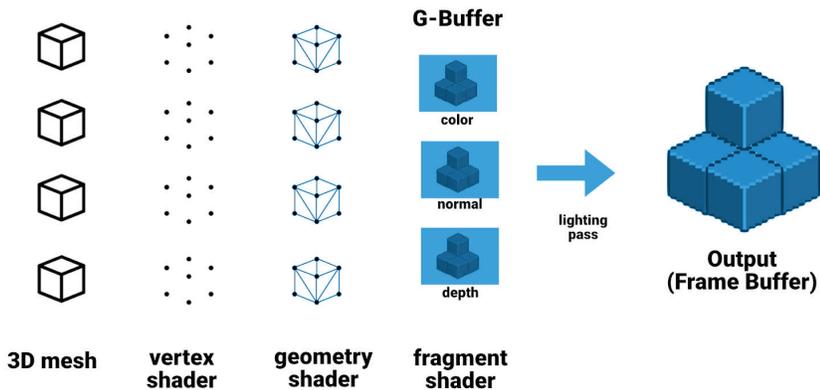
フォワードレンダリングパス

フォワードレンダリングでは、別々のパスでライトを描画します。同一のゲームオブジェクトに複数のライトを当てている場合は、ライトやオブジェクトが多数存在していると、大幅なオーバーフローが生じ、動作が遅くなることがあります。

従来のフォワードレンダリングとは異なり、HDRP では、フォワードレンダラーに効率面の改良が加えられています。たとえば、オブジェクトマテリアルごとに1つのパスで、複数のライトをまとめてカリングし、レンダリングします。ただし、それでもプロセスのコストは比較的大きくなります。パフォーマンスに問題がある場合は、ディファードシェーディングを使用することをおすすめします。

### ディファードシェーディング

HDRP では、オブジェクト単位でのライティング計算が発生しないディファードシェーディングも使用できます。ディファードシェーディングでは負荷の大きいレンダリングを後の段階に遅らせ、2つのパスを使用します。



ディファードシェーディングパス



ディファードシェーディングでは、オブジェクトではなくバッファにライティングを適用します。それぞれのパスが、最終的にレンダリングされる画像に反映されます。

1 つ目のパスである **G-buffer** ジオメトリパスでは、Unity がゲームオブジェクトをレンダリングします。このパスは、複数のタイプのジオメトリプロパティを取得し、テクスチャーのセットに保管します (ディフューズ色およびスペキュラー色、サーフェスのスムーズネス、オクルージョン、法線など)。

2 つ目のパスの **ライティングパス** では、Unity が G-buffer の完了後にシーンのライティングをレンダリングします。つまり、シェーディングを遅らせます。ディファードシェーディングパスはピクセルごとに反復され、個々のオブジェクトではなく、バッファを基準にライティング情報を計算します。

各レンダリングパスの技術的な違いの詳細については、HDRP に関するドキュメントの **フォワードおよびディファードレンダリング** をご覧ください。



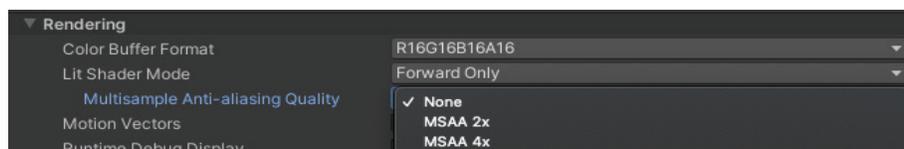
元のシーン

# アンチエイリアシング

「Lit Shader Mode」のレンダリングパスは、アンチエイリアシングを使用してレンダリングからギザギザのエッジを取り除く方法に影響します。HDRP では、制作のニーズに合わせて複数のアンチエイリアシング手法を使用できます。

## マルチサンプルアンチエイリアシング (MSAA)

マルチサンプルアンチエイリアシング (MSAA) は、PC ゲームで一般的なアンチエイリアシング手法です。個々のポリゴンのエッジを滑らかにする高性能ハードウェア向けの手法で、Unity ではフォワードレンダリングを使用する場合のみ機能します。最新の GPU のほとんどは、2x、4x、8x の MSAA サンプルに対応しています。



MSAA の品質設定

アクティブなパイプラインアセットで、「Lit Shader Mode」を「**Forward Only**」に設定します。続いて、「Multisample Anti-aliasing Quality」で、「**MSAA 2x**」、「**MSAA 4x**」、または「**MSAA 8x**」を選択します。値が大きいくほどアンチエイリアシングの質が上がりますが、動作は遅くなります。

これは、カメラビューでズームインしたときにより顕著になります。



MSAA 設定を適用した画像

次の制約事項にご注意ください。

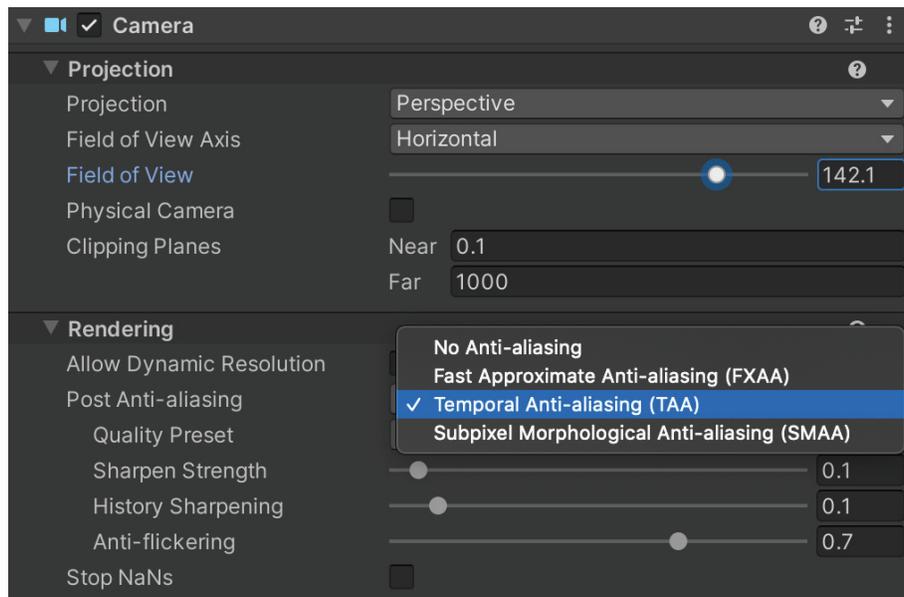
- MSAA は、シーンのジオメトリをテクスチャーに保管するディファードシェーディングの G-buffer では利用できません。そのため、ディファードシェーディングでは、いずれかのポストプロセスでのアンチエイリアシング手法（後述）を使用する必要があります。
- MSAA では、ポリゴンのエッジのエイリアシングのみを処理するため、シャープなスペキュラライトが届く特定のテクスチャーやマテリアルに対するエイリアシングについては防ぐことができません。これが問題になる場合は、MSAA と別で、ポストプロセスでのアンチエイリアシング手法（右側）を組み合わせる必要があります。

### ポストプロセスでのアンチエイリアシング

ポストアンチエイリアシングを次のように設定して、ポストプロセスの技術で設定したカメラにアンチエイリアシングを適用することもできます。

- **Temporal Anti-aliasing (TAA)** では、過去と現在のフレームからの情報を組み合わせ、現在のフレーム内の**ジャギー**を取り除きます。この設定を使用するためには、**モーションベクトル**を有効にする必要があります。大半の場合 TAA は良い成果を生み出しますが、状況によって（コントラストのあるサーフェスの前面でゲームオブジェクトが高速で移動した場合など）ゴースト上のアーティファクトが発生することがあります。HDRP10 では、一般的な TAA アーティファクトを削減するための改良が加えられています。Unity の実装ではゴーストの発生が抑制され、シャープネスが改善されたほか、他のソリューションで見られるちらつきが防止されています。
- **Fast Approximate Anti-aliasing (FXAA)** は、ハイコントラストの領域間のピクセルをブレンドする**スクリーンスペースアンチエイリアシングアルゴリズム**です。比較的高速で、大規模な演算能力も不要ですが、画像の全体的なシャープネスは落ちる場合があります。

- **Subpixel Morphological Anti-aliasing (SMAA)** では、画像内の境界を検出してから、ブレンドする特定のパターンを検出します。これによって、FXAA よりシャープな結果が生成されます。フラットスタイルやカートゥーン調のスタイル、クリーンなアートスタイルに適しています。



ディファードシェーディングを使用する際は、カメラのポストアンチエイリアシングを調整します。

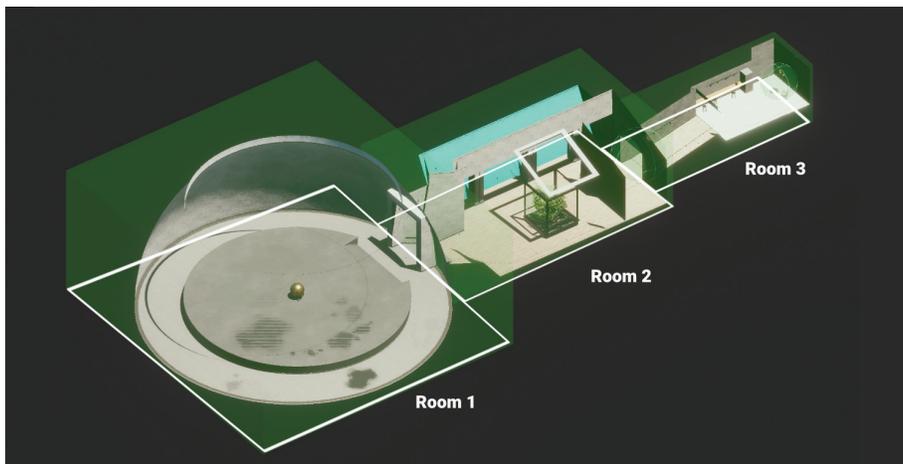


ポストプロセスのアンチエイリアシング：FXAA、SMAA、TAA の各設定の結果を比較。

注：ポストプロセスでのアンチエイリアシングをマルチサンプルアンチエイリアシングと組み合わせる際は、レンダリングコストに注意してください。いつでもビジュアルクオリティとパフォーマンスのバランスがとれるように、プロジェクトは常に最適化を心がけるようにしましょう。

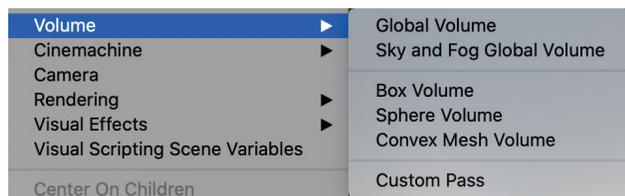
# ボリューム

HDRP では**ボリュームフレームワーク**を使用します。このシステムでは、シーンを分割し、カメラの位置に合わせて特定の設定や機能を有効化できます。たとえば、このHDRP テンプレートレベルは3つに分かれており、それぞれに独自のライティング設定があります。その形で、各部屋も別々のボリュームが内包しています。



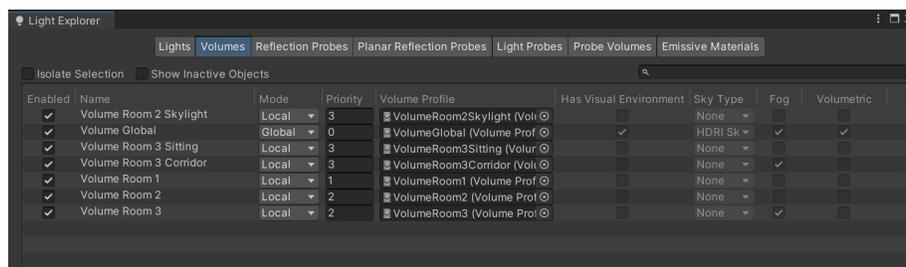
それぞれにライティング条件が異なるスペースをボリュームが覆っています。

ボリュームは、ボリュームコンポーネントが含まれたプレースホルダーオブジェクトに過ぎません。ボリュームは、「**GameObject > Volume**」メニューから、プリセットを選択することで作成できます。または、適切なコンポーネントを含めたゲームオブジェクトを手動で作成します。



プリセットを使用してボリュームオブジェクトを作成

ボリュームコンポーネントは任意のゲームオブジェクトに追加できるため、ヒエラルキーから見つけるのが困難になる場合があります。ライトエクスプローラー（「**Window > Rendering > Light Explorer > Volumes**」）を使用すると、ロードされているシーンのボリュームを見つけやすくなります。このインターフェースを使用して手早く調整することができます。

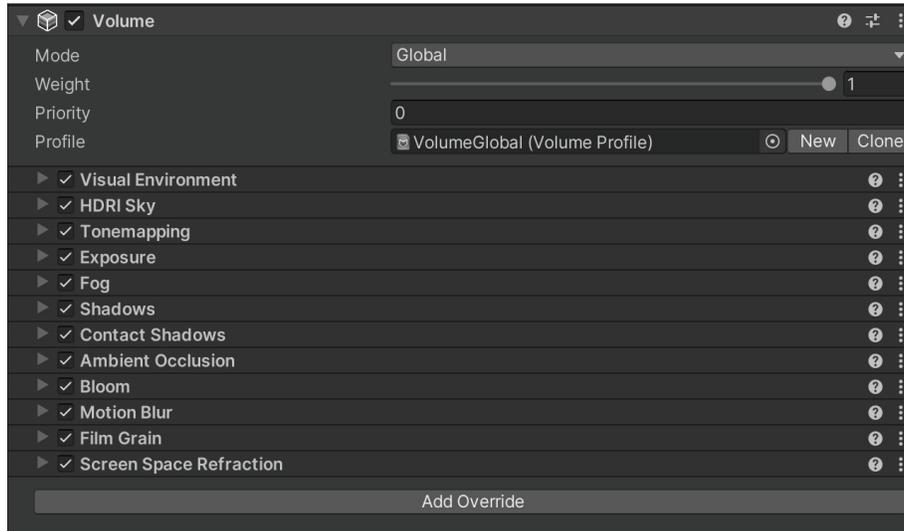


ライトエクスプローラーには、開いているシーン内のすべてのボリュームが一覧表示されます。

## ローカルとグローバル

コンテキストに合わせて、ボリュームコンポーネントの「Mode」を「Global」または「Local」に設定します。

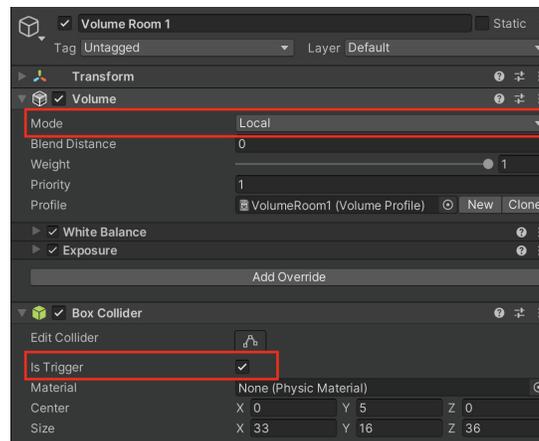
グローバルボリュームは、境界のない「汎用」ボリュームとして使用できます。その場合、シーン内のすべてのカメラに影響します。HDRP テンプレートシーンでは、レベル全体の HDRP 設定のベースラインが VolumeGlobal によって定義されます。



グローバル Volume オーバーライド

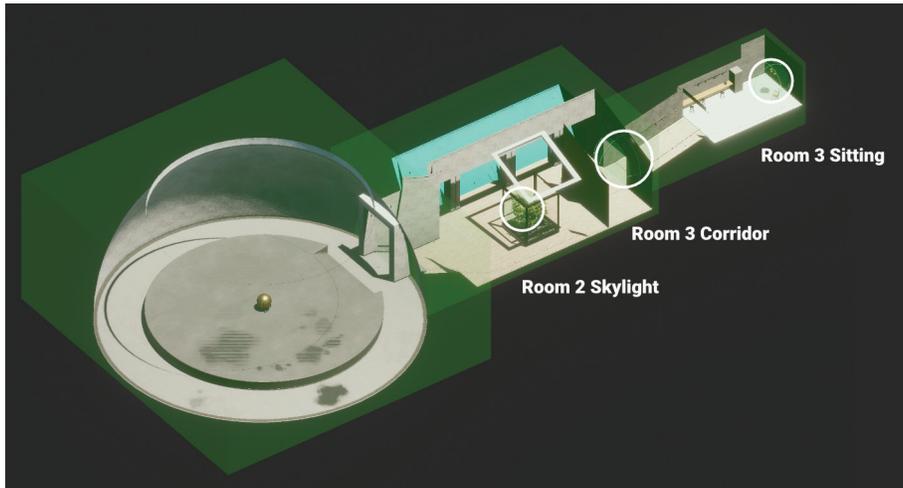
ローカルボリュームでは、設定を適用する限定的なスペースを定義します。コライダーコンポーネントを使用して境界を決定します。コライダーによって FPS プレイヤーコントローラーなどの物理ボディの動きが妨げられないようにするには、「IsTrigger」を有効にします。

このテンプレートシーンでは、各部屋に、グローバル設定をオーバーライドする BoxCollider を含むローカルボリュームがあります。



各部屋に、コライダーが「IsTrigger」に設定されたローカルボリュームがあります。

Room 2 には、ガラスケースの横の中央の明るい場所に小さな球面ボリュームがあります。同様に、Room 3 には、入口の通路と、つり下げの照明の下の椅子があるエリアに、より小さなボリュームがあります。



特殊なライティング条件では小さいボリュームを使用します。

SampleScene では、ローカルボリュームによって「White Balance」、「Exposure」、「Fog」がオーバーライドされます。明示的にオーバーライドされない要素は、すべてグローバルのデフォルト設定にフォールバックされます。

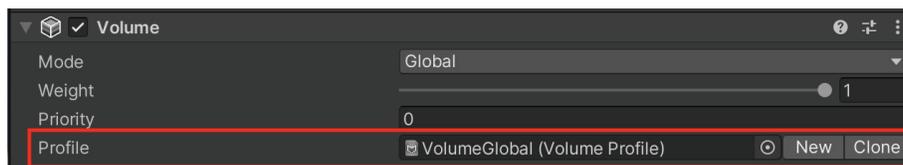
シーン内でカメラが動く際は、グローバル設定をオーバーライドするローカルボリュームにプレイヤーコントローラーが接触するまでは、グローバル設定が適用されます。

### パフォーマンスに関するヒント

ボリュームは大量に使用しないようにしてください。各ボリュームの評価(ブレンディング、立体化、オーバーライドの計算など)には、CPU コストが伴います。

## ボリュームプロファイル

ボリュームコンポーネント自体には、実際のデータは含まれません。その代わりに、ボリュームコンポーネントは**ボリュームプロファイル**を参照します。これは、シーンをレンダリングするための HDRP 設定を含むディスク上の **ScriptableObject** です。新しいボリュームプロファイルを作成するには、「Profile」フィールドで、「New」または「Clone」ボタンを使用します。



「Profile」フィールドでボリュームプロファイルを切り替えるか、新しく作成します。

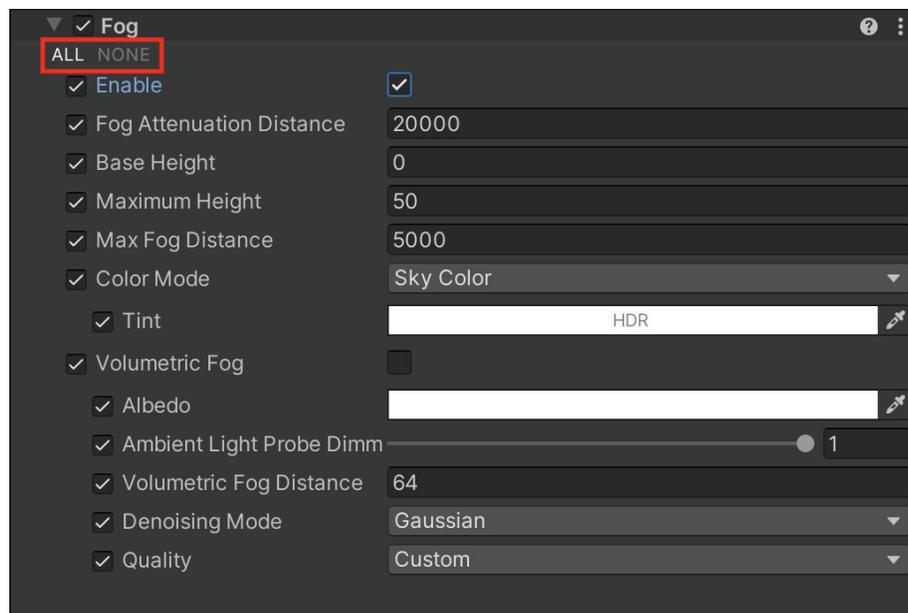
保存済みの別のプロファイルに切り替えることもできます。ボリュームプロファイル  
をファイルとして使用することで、以前の設定を再利用したり、ボリューム間でプロ  
ファイルを共有したりしやすくなります。

再生モード中にボリュームプロファイルに加えられた変更内容は、再生モードを終了  
した後も残ります。

## Volume オーバーライド

各**ボリュームプロファイル**では最初、一連のプロパティがデフォルト値に設定され  
ています。この値を編集するには、**Volume オーバーライド**を使用し、個々の設定を  
カスタマイズします。たとえば、Volume オーバーライドでは、ボリュームのフォグ、  
プロセッシング、露出を変更できます。

**ボリュームプロファイル**を設定したら、「**Add Override**」をクリックし、プロファイ  
ル設定をカスタマイズします。フォグのオーバーライドは次のようになります。



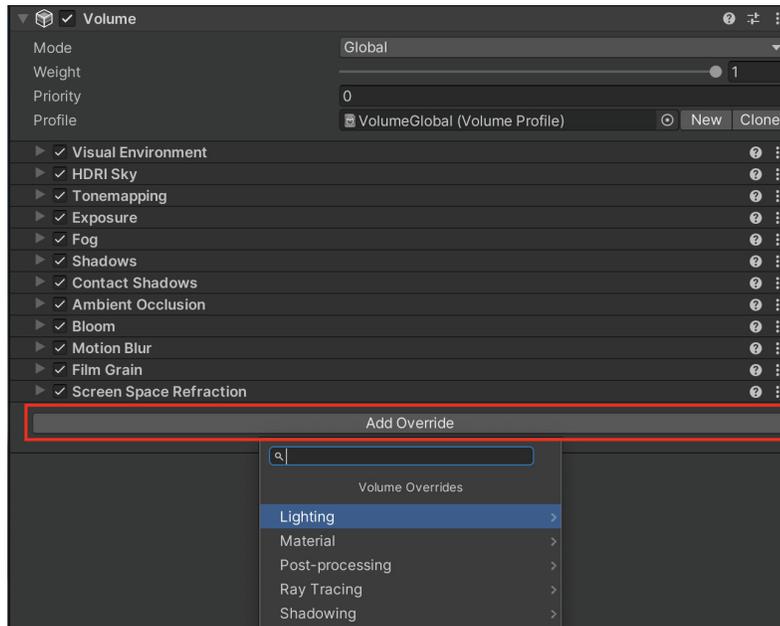
Volume オーバーライドとしてのフォグの例

Volume オーバーライドの各プロパティでは、左側にチェックボックスがあり、そ  
こで対象のプロパティの編集を有効化できます。ボックスを無効のままにすると、  
HDRP では、ボリュームの初期値が使用されます。

各ボリュームオブジェクトには、複数のオーバーライドを設定できます。それぞれの  
オーバーライドで、必要に応じてプロパティを編集できます。左上の「**All**」また  
は「**None**」のショートカットを使用すると、オンとオフを一括で切り替えることが  
できます。

## オーバーライドのワークフロー

オーバーライドの追加は、HDRP の重要なワークフローです。[プログラミングからの継承](#)という概念を理解していれば、Volume オーバーライドも理解しやすくなります。

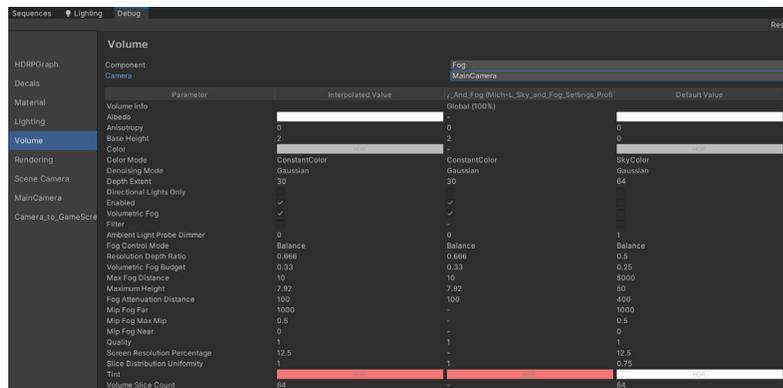


Volume オーバーライドを使用して HDRP 機能を追加

上位レベルのボリューム設定は、下位レベルのボリュームのデフォルト設定として使用されます。ここでは、HDRP のデフォルト設定がグローバルボリュームに渡されます。それがローカルボリュームの「ベース」として使用されます。

グローバルボリュームによって、HDRP のデフォルト設定がオーバーライドされます。次に、ローカルボリュームによってグローバルボリュームがオーバーライドされます。「**Priority**」、「**Weight**」、「**Blend Distance**」（概要は後述）を使用して、ボリュームの重複に起因する競合を解消します。

特定のボリュームコンポーネントの現在の値をデバッグするには、「[Rendering Debugger](#)」の「Volume」タブを使用します。



ボリュームのデバッグ

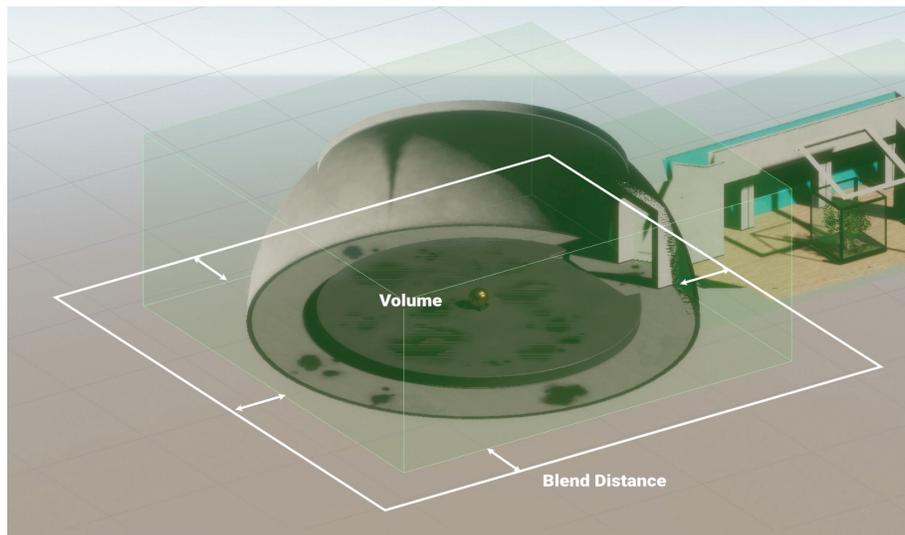
HDRP のドキュメントで、完全な [Volume オーバーライドのリスト](#)を確認できます。

## ブレンディングと優先度

レベルごとに複数のボリュームが必要な場合もあるため、HDRP ではボリューム間のブレンドが可能になっています。これにより、ボリューム間の遷移がスムーズになります。

ランタイム時、HDRP はカメラ位置を基に、最終的な HDRP 設定に影響するボリュームを判断します。

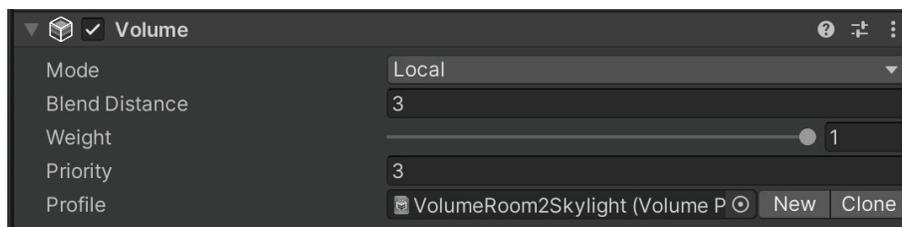
「Blend Distance」によって、ボリュームのコライダー外にどの程度離れたらフェードオンまたはフェードオフを開始するかを指定します。「Blend Distance」の値を「0」に設定すると即座に遷移が開始されます。正の値を設定すると、カメラが指定された範囲内に入ったときに Volume オーバーライドによってブレンディングが開始されます。



「Blend Distance」を使用して、ボリューム周辺の遷移ゾーンを定義します。

ボリュームフレームワークは柔軟性が高く、必要に応じてボリュームとオーバーライドを組み合わせることができます。同一スペース内に複数のボリュームが重複している場合、「Priority」に基づいて、優先されるボリュームが決定されます。この値が大きいほど優先度が高くなります。

通常は、あいまいになるのを防ぐために「Priority」を明示的に設定します。明示的に設定しない場合、「Priority」が同じであれば作成された順番で優先順が決定されますが、それによって予期しない結果が生じることがあります。



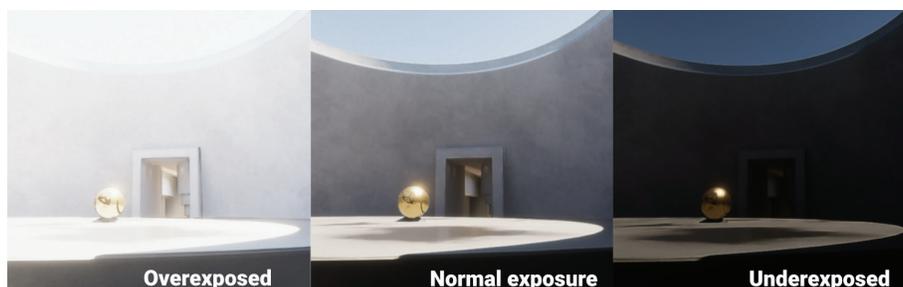
ローカルボリュームが重複する場合は、「Blend Distance」、「Weight」、「Priority」を使用します。

# 露出

HDRP は、現実世界のライティングモデルを使用して各シーンをレンダリングします。そのため、従来の写真術で使われてきたプロパティと類似するプロパティが多数あります。

## 露出値を理解する

**露出値 (EV)** は、カメラのシャッタースピードと F 値（レンズの開口部または絞りのサイズ）の組み合わせによって決まる数値です。明度を最適化し、シャドウとハイライトの両方でディテールを高精度に捉えるためには、露出を適切に設定する必要があります。露出を適切に設定しないと、画像の露出過多や露出不足が生じ、期待する結果が得られなくなります。



露出オーバー、露出アンダー、適正露出の比較。

HDRP での露光範囲は、多くの場合、以下の範囲内に収まります。



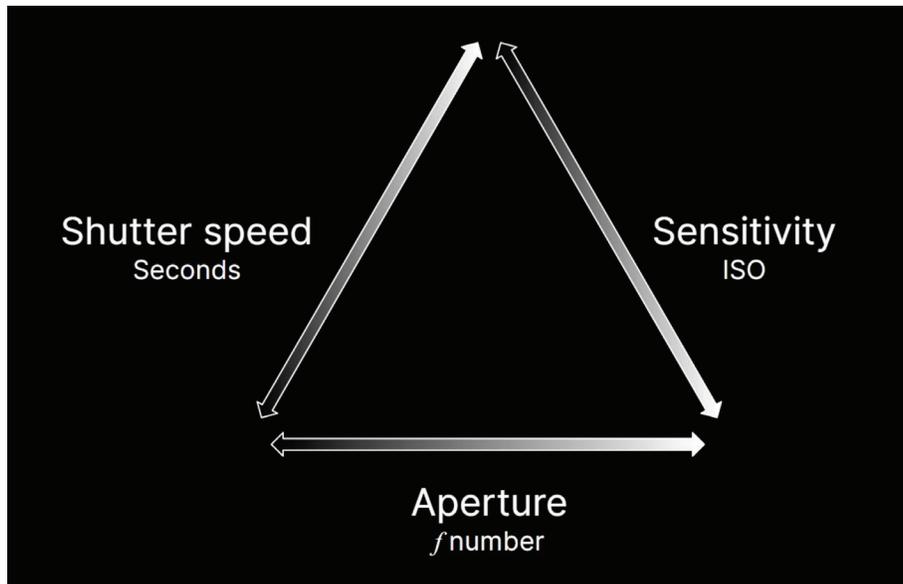
月のない夜から、明るい晴天日までの露出範囲

露出値が大きくなるほどカメラに入る光が少なくなり、より明るい状況に適します。露出値 13 から 16 の範囲は、晴天の日中の屋外に適しています。一方、暗く、月がない夜空には、露出値 -3 から 0 が適しています。

実際のカメラ設定で、次のようなさまざまな要素を変更し、露出値を調整できます。

- シャッタースピード：画像センサーを光にさらす時間
- F 値：絞り / レンズ開口部のサイズ
- ISO：フィルム / センサーの感度

写真家の間では、これは「露出の三角形」と呼ばれています。Unity では、現実のカメラと同様、これらの数値をさまざまな形で組み合わせて同じ露出値を実現できます。



露出の三角形

HDRP では、すべての露出値が  $EV_{100}$  で表現されます。これにより、感度は 100 International Standards Organisation (ISO) フィルムの感度に固定されます。

### **i** 露出値の計算式

以下の計算式によって、実際の露出値が計算されます。

$$EV = \log_2 \left( \frac{f \text{ number}^2 / \text{shutter speed}}{ISO / 100} \right)$$

露出の計算式

これは 2 を底とする対数スケールです。露出値が 1 単位増加すると、レンズに入る光の量が半分に減少します。

HDRP では、実際の画像の露出に合わせることができます。カメラやスマートフォンでデジタル写真を撮影するだけで、その画像からメタデータを取得し、F 値、シャッタースピード、ISO を特定できます。



デジタル写真の Exif データを基に露出を合わせます。

そのうえで、上記の計算式によって露出値を計算します。Exposure オーバーライド（後述）と同じ値を使用すると、レンダリングされる画像は、現実の画像と同じ露出になります。

この方法を使うことで、レベルのライティングを設定する際にデジタル写真を参照情報として使用できます。その画像を完全に再現することが目的とは限りませんが、実際の写真に合わせることによって、勘に頼ることなくライティングを設定できます。

## Exposure オーバーライド

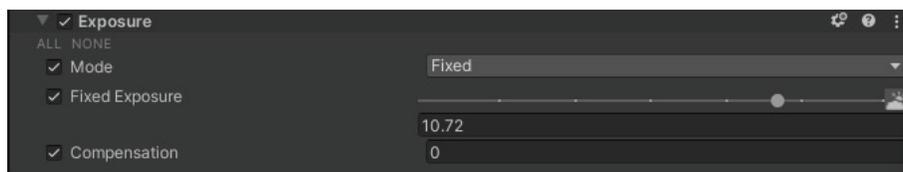
HDRP では、露出は Volume オーバーライドです。ローカルボリュームまたはグローバルボリュームに追加することで利用可能なプロパティを確認できます。

「**Mode**」ドロップダウンで、「**Fixed**」、「**Automatic**」、「**Automatic Histogram**」、「**Curve Mapping**」、「**Physical Camera**」のいずれかを選択できます。

「**Compensation**」では、露出を変更、調整できます。通常はこれを利用して小さな調整を加え、レンダリングされた画像の「**ストップ (絞り)**」をわずかに上下させます。

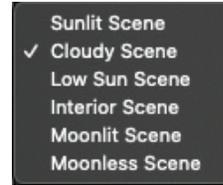
### 「Fixed」モード

「Fixed」モードでは、露出値を手動で設定できます。



「Fixed」モードの露出

「Fixed Exposure」スライダーの目盛りが参考になります。また、右側のアイコンをクリックするとプリセットのドロップダウンが表示されます（「Sunlit Scene」の 13 から「Moonless Scene」の -2.5 まで）。フィールドの値を任意の数値に設定することもできます。



「Fixed」モードの露出のプリセット

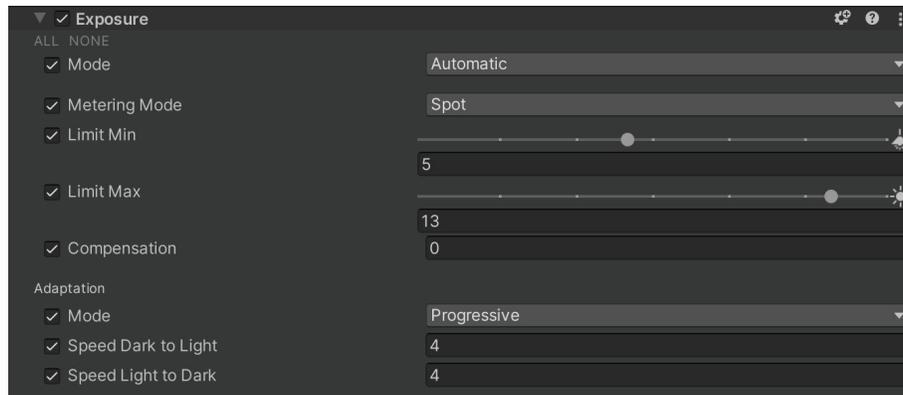
「Fixed」モードはシンプルですが、柔軟性に劣ります。通常は、ライティングが比較的均一で、1つの露出値で対応できるボリュームやシーンを使用する場合に効果的です。

### 「Automatic」モード

「Automatic」モードでは、画面上の明度の範囲に応じて露出が動的に設定されます。この設定は、人間の目が暗さの変化に適応し、黒と認識されるものを再定義する仕組みと非常に似ています。

「Automatic」モードは、さまざまなライティングの状況下で有効ですが、カメラが非常に暗い場所や非常に明るい場所を向いたときに、意図せず画像の露出過多や露出不足が生じる場合もあります。

露出レベルを適切な範囲内に収めるためには、「Limit Min」と「Limit Max」を使用します。テストプレイを行って、レベル全体で上限値が適切な露出の範囲内に収まっているか確認してください。



「Automatic」モードの露出

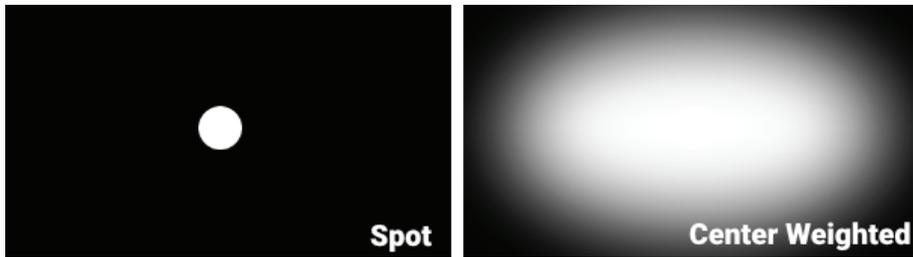
「Metering Mode」をマスクオプションと組み合わせると、フレーム内のどの部分を自動露出に使用するか決定できます。

「Adaptation」モードでは、暗い場所と明るい場所の間をカメラが遷移する際に、自動露出をどのように変化させるかを制御します。スピードを調整するオプションを利用します。人間の目と同じように、非常に暗いエリアから非常に明るいエリアにカメラが移動すると（またはその逆）、一時的にどこを向いているのかわからなくなる場合があります。

## 「Metering Mode」のオプション

「Automatic」、「Automatic Histogram」、「Curve Mapping」モードでは、「Metering Mode」を使用して、フレーム内のどの部分を露出の計算に使用するかを制御します。「Metering Mode」は次のように設定できます。

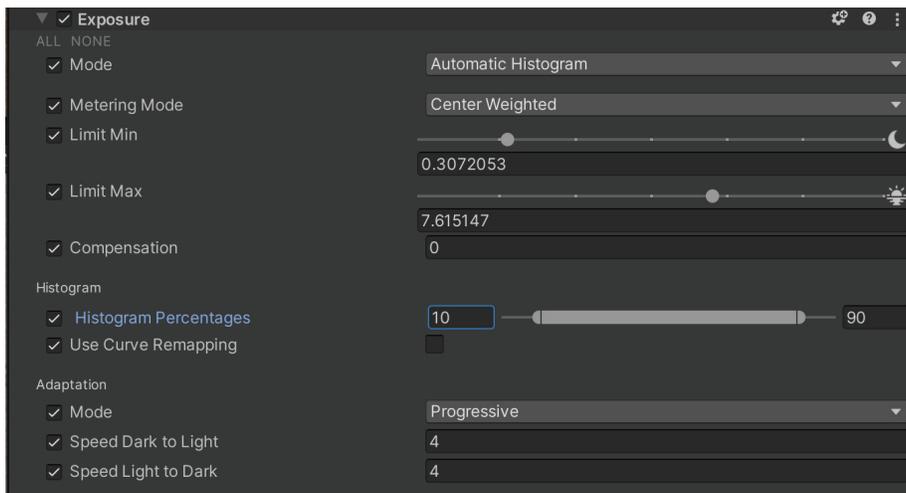
- 「Average」：フレーム全体を使用して露出を測定します。
- 「Spot」：画面の中央部のみを使用して露出を測定します。
- 「Center Weighted」：画面の中央部にあるピクセルを優先的に使用し、フレームの端に向かうにつれて使用率を減らしていきます。
- 「Mask Weighted」：指定した画像（「Weight Texture Mask」）を使用して、露出を測定する際にどのピクセルを最も重視するかを決めます。
- 「Procedural Mask」：手続き型で生成されたテクスチャーを基に露出を評価します。オプションを使用して、中心の位置、半径、ぼかし具合を変更できます。



「Metering Mode」の「Spot」と「Center Weighted」

## Automatic Histogram

「Automatic Histogram」モードは、「Automatic」モードをさらに発展させたものです。露出設定時に画像の**ヒストグラム**を計算し、最も暗いピクセルと最も明るいピクセルを無視します。

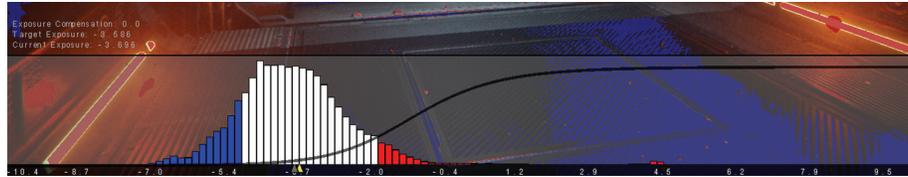


「Automatic Histogram」モード

露出の計算から非常に暗いピクセルと非常に明るいピクセルを排除することで、フレーム内に極端に明るいピクセルや暗いピクセルがある場合でも、安定した露出を得られます。これにより、発光が強いサーフェスや黒い材料があったとしても、レンダリング結果で深刻な露出過多や露出不足が生じることはありません。

「Histogram Percentages」設定を使用すると、ヒストグラム内の指定したパーセンテージ範囲に含まれないデータをすべて無視できます（たとえば、ヒストグラムの左端と右端から、最も明るいピクセルと最も暗いピクセルを除外することができます）。

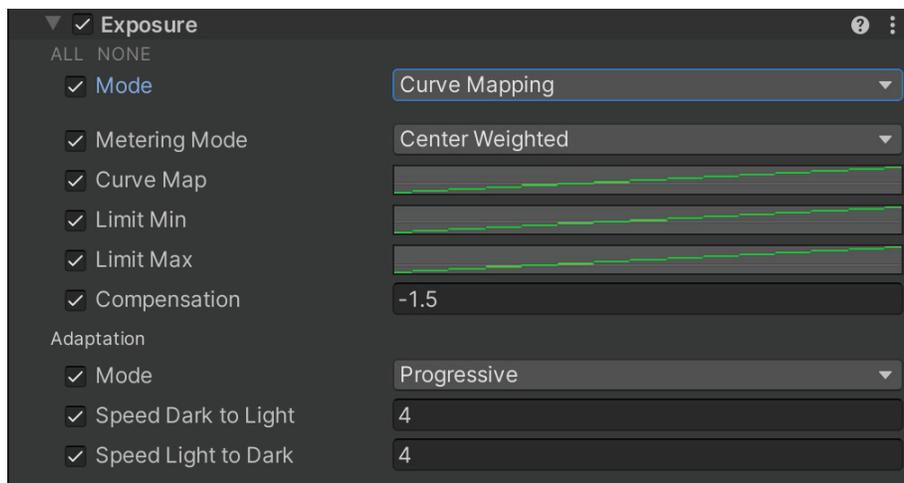
「Curve Remapping」を使用すると、露出曲線もリマップできます（後述の「Curve Mapping」を参照）。



「Automatic Histogram」では、ヒストグラム中央部にあるピクセルを基に露出を計算します。

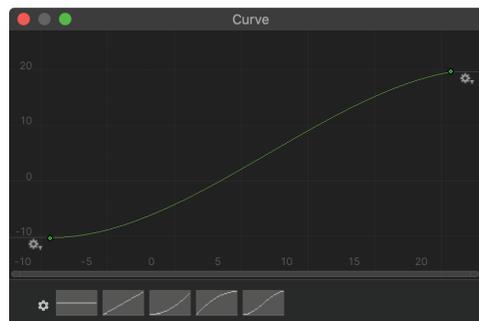
## カーブマッピング

「Curve Mapping」モードは、「Automatic」モードのもう1つのバリエーションです。



「Curve Mapping」モード

この設定では、曲線の X 軸は現在の露出を表し、Y 軸はターゲットとする露出を表します。露出カーブの再マップによって、非常に高精度な結果を生成できます。

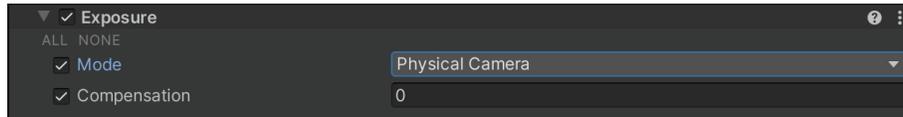


カーブを変えて露出を調整します。

## Physical Camera

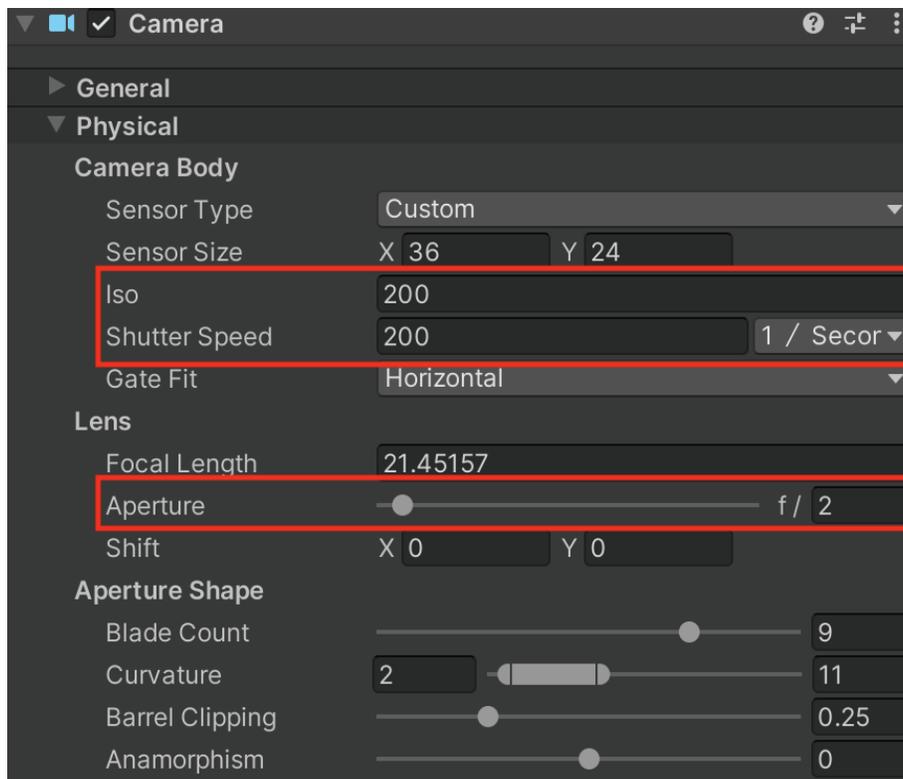
写真に詳しい方は、「Physical Camera」モードを使用すると、カメラのパラメーターを設定しやすいかもしれません。

Exposure オーバーライドの「Mode」を「Physical Camera」に切り替えてから、「Main Camera」を確認します。



「Physical Camera」モード

「Physical Camera」を有効にします。インスペクターに以下のプロパティが表示されます。



カメラの「Physical Camera」プロパティ

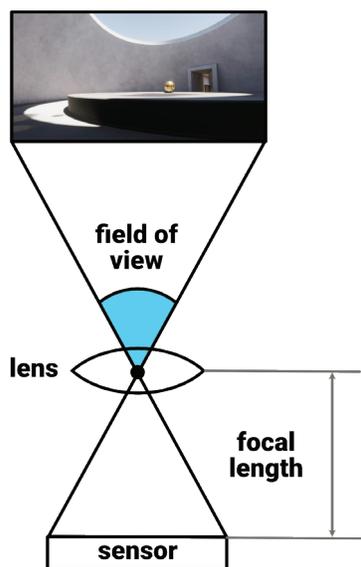
露出関連で重要なのは、「ISO」（感度）、「Aperture」（F 値）、「Shutter Speed」です。参照用の写真に合わせる場合は、画像の Exif データから適切な設定をコピーします。または、[こちらの表](#)を参考にすると、F 値とシャッタースピードを基に露出値を推測できます。

## **i** 「Physical Camera」のその他のパラメーター

露出とは無関係ですが、「Physical Camera」のその他のプロパティも、現実世界のカメラの属性に合わせる助けになります。

たとえば、Unity（および他の多数の 3D アプリケーション）では通常、**有効視野 (FOV)** を使用して、カメラがワールド内を一度に撮影できる範囲を決定します。

しかし、実際のカメラでは、**有効視野**はセンサーのサイズとレンズの焦点距離によって決まります。直接有効視野を設定するのではなく、「Physical Camera」の設定を使用すると、実際のカメラのデータの「**Sensor Type**」、「**Sensor Size**」、「**Focal Length**」を入力できます。データを入力すると、それに対応する有効視野の値が自動的に算出されます。



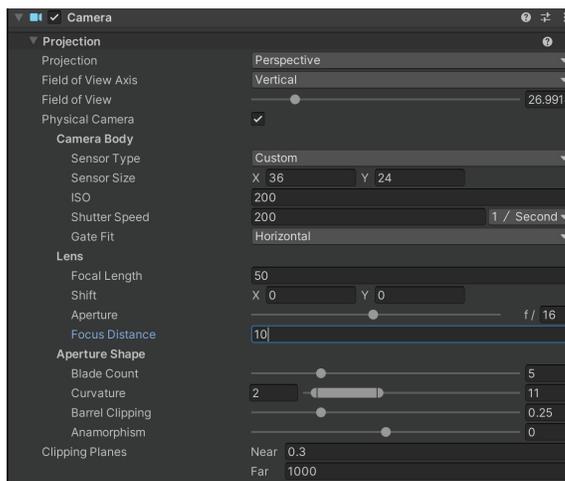
焦点距離、センサーサイズ、有効視野の関係

実際の写真に合わせて露出を設定するには、その画像ファイルに含まれるカメラのメタデータを利用します。Windows と macOS の両方で、デジタル画像から Exif データを読み取ることができます。そのうえで、対象のフィールドをバーチャルカメラにコピーできます。

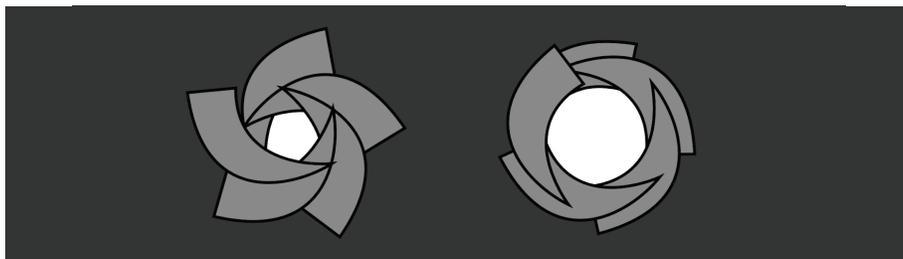
注：場合によっては、メタデータからカメラのメーカーとモデル名を確認し、そのメーカーのウェブサイトですべてのセンサーのサイズを調べる必要があります。[こちらの記事](#)には、一般的な画像センサーのフォーマットの推定が記載されています。

下にあるパラメーターのいくつかは、被写界深度ボリュームに影響します。

Unity 2021 LTS では、カメラのインスペクターから「**Focus Distance**」を制御できます。被写界深度ボリュームコンポーネントで、「**Focus Mode**」と「**Focus Distance Mode**」を「**Physical Camera**」に設定します。



「**Blade Count**」、「**Curvature**」、「**Barrel Clipping**」を使用して、カメラの絞りの形状を変更できます。このパラメーターは、これによって、[被写界深度ボリュームコンポーネント](#)によって生じる「ボケ」の見え方に影響します。



物理カメラのパラメーターを使用して絞りの形状を設定します。5枚羽根絞りの場合、五角形（左）または円形（右）のボケを表現できます。

# ライト

HDRP には、シーン内のイルミネーションの制御に役立つさまざまなタイプのライトや形状が用意されています。

## ライトのタイプ

Unity の他のレンダーパイプラインと同様に、以下のタイプのライトを利用できます。

- **ディレクショナル**：無限遠に位置する光源からの光のような特性を持ちます。強度が落ちない完全な平行光線です。ディレクショナルライトは、多くの場合、日光として使用されます。一般的に、屋外のシーンでは、これがキーとなるライトです。
- **スポット**：現実世界のスポットライトに似ており、円錐、ピラミッド、ボックスといった形状があります。スポットは、前方 Z 軸に沿って、円錐 / ピラミッド形状のエッジに向かって落ちていきます。
- **ポイント**：空間内のある 1 点からすべての方向に光を放つ全方向性ライトです。ランプやキャンドルなどの放射光源に適しています。
- **Area**：特定の形状（矩形、チューブ、またはディスク）のサーフェスから光を投影します。エリアライトは、窓や蛍光灯のような、中央部の強度が均一である面的な光源においての使用に適しています。

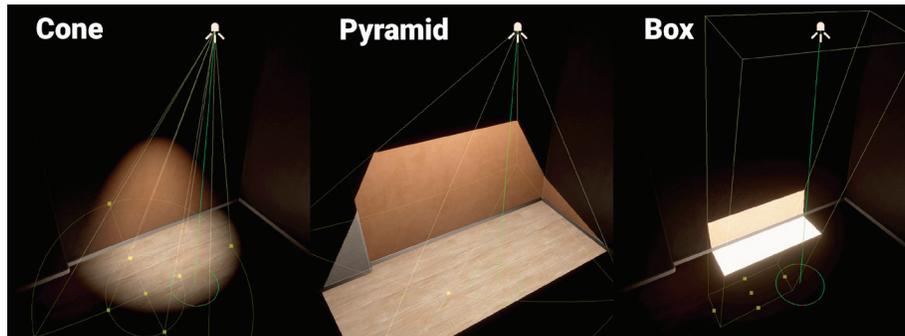
「**Range**」を使用して、スポットライト、ポイントライト、エリアライトをどのように減衰させるかを調整できます。多くの HDRP ライトは、現実世界の光源と同じように、[逆 2 乗の法則](#)に従って減衰します。

## 形状

スポットライトとエリアライトには、各ライトの減衰方法を制御するために独自の形状が追加されています。

HDRP のスポットライトでは、3 つの形状を使用できます。

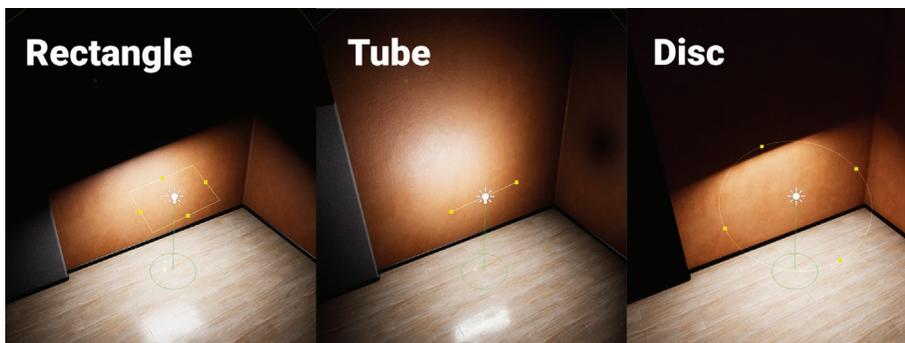
- **Cone (円錐)** : 1 点から円錐形に広がる光を投影します。「**Outer Angle**」(角度)と「**Inner Angle**」(パーセンテージ)を使用して、円錐形の形と角度減衰を調整します。
- **Pyramid (ピラミッド)** : 1 点から四角錐型に広がる光を投影します。「**Spot Angle**」と「**Aspect Ratio**」を使用して四角錐の形状を調整します。
- **Box (ボックス)** : 矩形のボリューム全体へ均一に光を投影します。X サイズと Y サイズによって基部となる長方形を定めて、「**Range**」によって Y 次元を調整します。この光は、「**Range Attenuation**」がオンになっていなければ減衰しないようになっているので、ボックスの範囲内で日光のシミュレーションなども行うことができます。



HDRP のスポットライトの形状

HDRP のエリアライトでは、3 つの形状を使用できます。

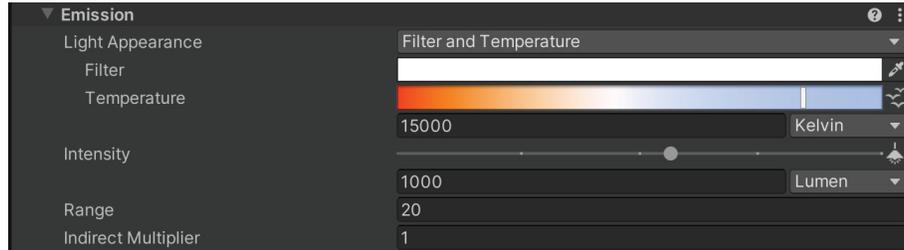
- **Rectangle (矩形)** : 矩形の形状からローカルの正の Z 方向に、定義された「**Range**」まで光を投影します。
- **Tube (チューブ)** : 1 本の線から全方向に、定義された「**Range**」まで光を投影します。このライトは「**Realtime Mode**」でのみ使用できます。
- **Disc (ディスク)** : ディスク型の形状からローカルの正の Z 方向に、定義された「**Range**」まで光を投影します。このライトは「**Baked Mode**」でのみ使用できます。



HDRP のエリアライトの形状

## 色と温度

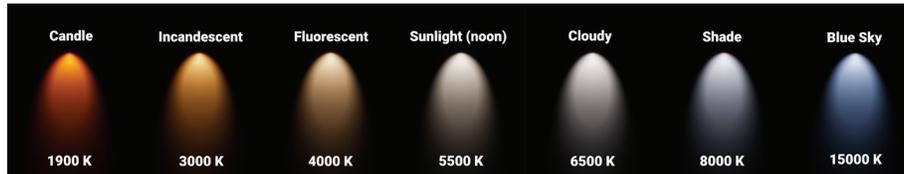
HDRP のすべてのタイプのライトに、ライトの外観を定義する「Emission」プロパティがあります。



「Emission」の「Light Appearance」プロパティを変更

「Light Appearance」を「Color」に切り替えると、RGB カラーを指定できます。または、これを「Filter and Temperature」に変更し、より物理的に正確なデータを入力できます。

色温度では、ケルビン度に基づいて色を設定します。詳しくは、[ライティングと露出のチートシート](#)をご覧ください。



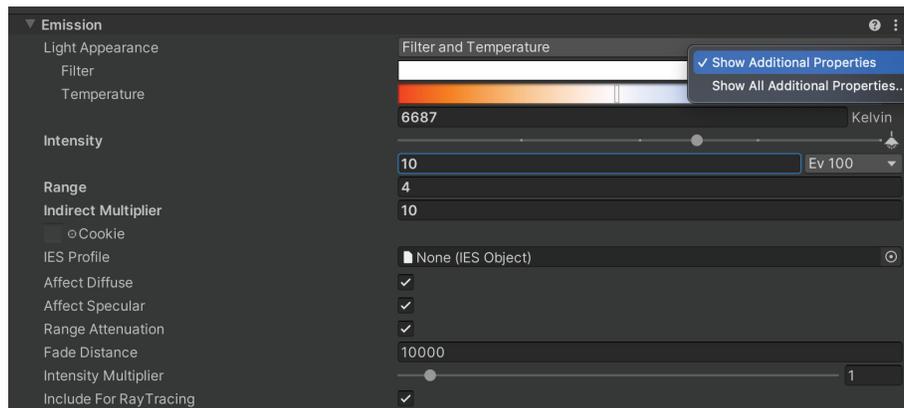
ケルビン温度で表された色温度

別の色を追加してフィルターのように作用させることで、別の色相を使ってライトに色を付けることもできます。これは、現実の写真撮影で[カラージェルフィルター](#)を使用する方法と似ています。

## 追加のプロパティ

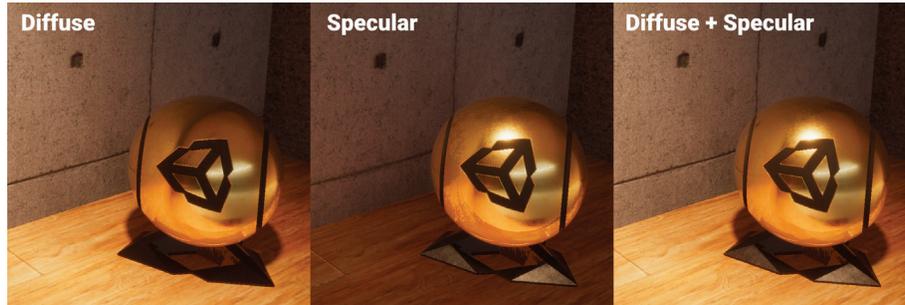
インスペクターのプロパティの右上にある**その他のアイテムメニュー** (:) の下にも、高度なコントロールがいくつか用意されています。「Show Additional Properties」を選択すると、追加のオプションが表示されます。

「Affect Diffuse」、「Affect Specular」などのトグルがあります。たとえば、カットシーンや映画のようなライティングでは、明るいハイライトを制御するライトと、よりソフトなディフューズライトを生成するライトを分離することができます。



各ライトに追加のプロパティがあります。

「**Intensity Multiplier**」を使用して、ライトの元の強度の値を変更せずに全体的なライトの強度を調節することもできます。複数のライトを一括で明るくしたり暗くしたりするのに便利です。



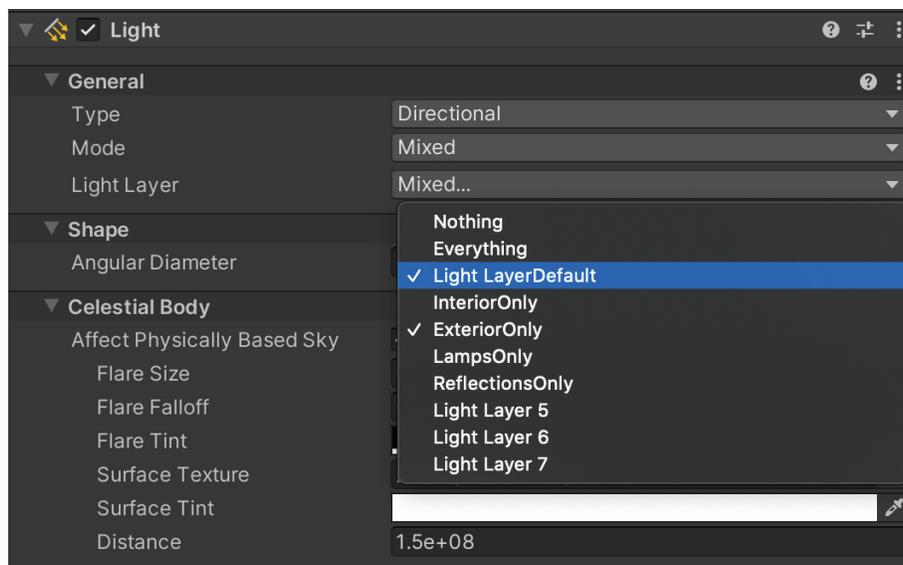
ディフューズライティングとスペキュラーライティングの設定を細かく調整できます。

## ライトレイヤー

HDRP では、**ライトレイヤー**を使用して、ライトの影響範囲をシーン内の特定のメッシュに限定できます。これは、ライトコンポーネントと MeshRenderer に関連付けることができる LayerMask になります。

ライトのプロパティで、「**More Options**」ボタンをクリックします。「**General**」に「**Light Layer**」ドロップダウンが表示されます。ライトに関連付ける LayerMask を選択します。

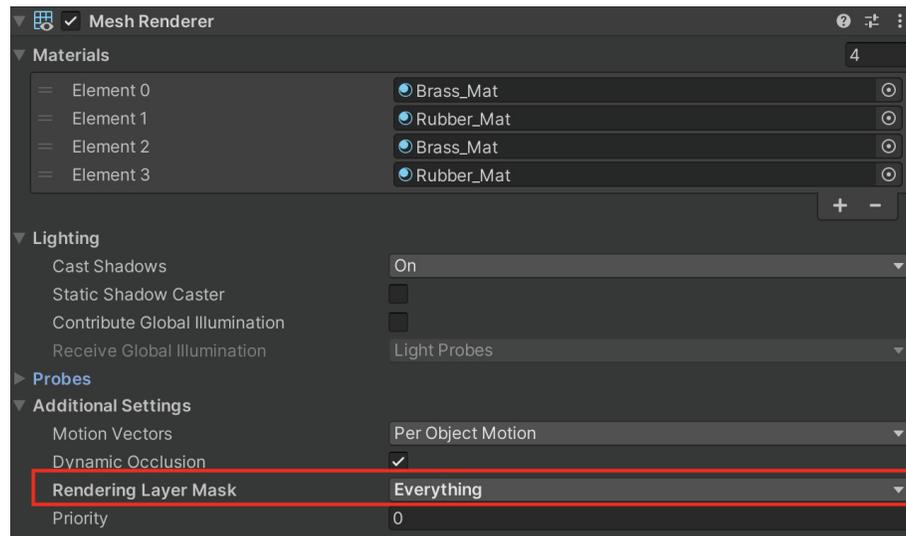
「Light」プロパティで、「**More Items**」メニュー（≡）から「**Show Additional Properties**」を選択します。「**General**」に「**Light Layer**」ドロップダウンが表示されます。ライトに関連付ける LayerMask を選択します。



「Light Layer」の選択

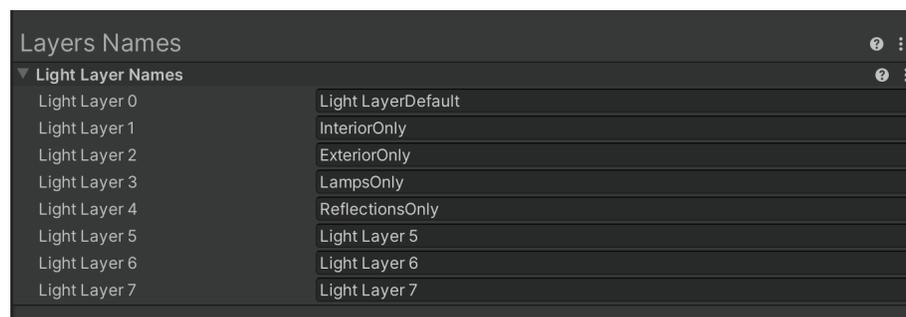
次に、「**Rendering Layer Mask**」を使用して MeshRenderer を設定します。対応する LayerMask 上のライトのみがメッシュに影響します。これは、光漏れを解消し、意図した対象のみに確実にライトを当てるために欠かせない機能です。また、カットシーンのライティングを設定するワークフローにこの機能を組み込むことで、映画的な演出でキャラクターだけに専用のライトが当たるようにすることができます。

たとえば、建物内部のライトが意図せず壁を貫通し、屋外に出るのを防ぎたい場合は、屋内用と屋外用のライトレイヤーを設定します。この機能を使えば、ライトの構成を細かく制御できます。



「Rendering Layer Mask」を設定し、特定のライトのみがメッシュに適用されるようにします。

ライトレイヤーを設定するには、「**HDRP Default Settings**」にアクセスします。「**Layers Names**」セクションで、「**Light Layer 0**」から「**Light Layer 7**」までの名前をそれぞれ文字列で設定できます。



「HDRP Default Settings」の「Layers Names」

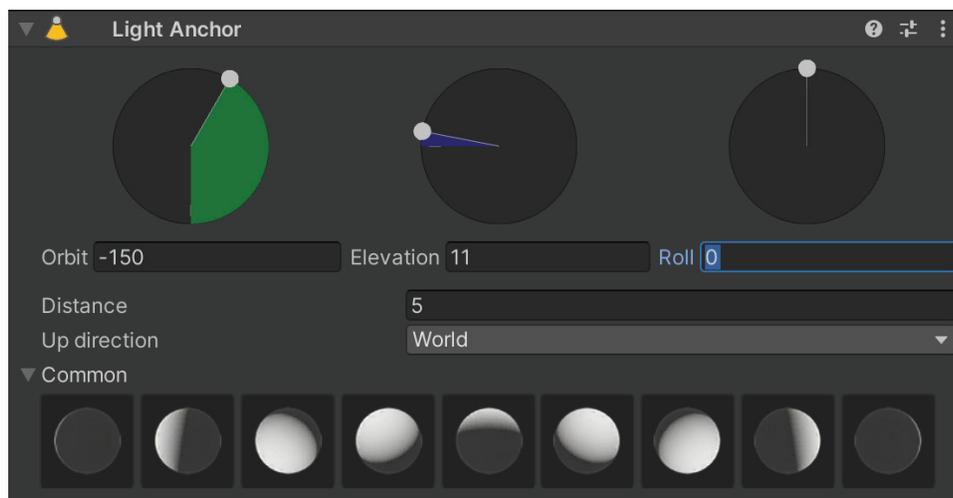
ライトのプロパティに関する全体のリストを含めて、この詳細については、[Light コンポーネントに関するドキュメント](#)をご覧ください。

## ライトアンカー

Unity 2021 LTS 以上には、**ライトアンカー**システムが備わっていて、カメラと被写体の角度と距離を制御することでライトを簡単に設定できます。また、一般的なライティング角度を9つのプリセットから選択できます。

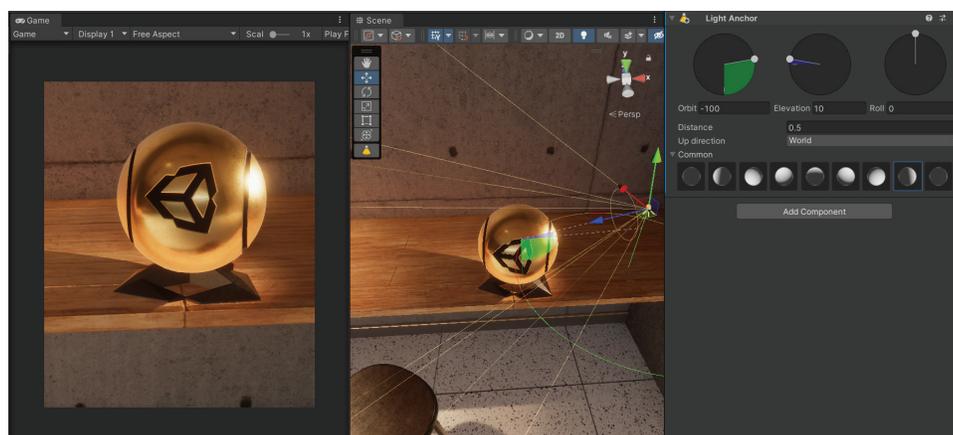
キャラクターや小道具の周りに複数のライトを使用して、映画のショット、シーン、製品を照らす必要がある場合、ライトアンカーコンポーネントで、アンカーターゲットの周りのスクリーンスペースで高速なライト操作を利用できます。

まず、ライトアンカーが動作するためには、カメラがメインカメラとしてタグ付けされている必要があります。そのうえで、制御対象の**スポットライト**に「**Light Anchor**」コンポーネントを追加します。被写体にライトを合わせると、その位置がスポットライトのアンカーポイントになります。アンカーポイントとスポットライトとの間の距離は、「**Distance**」で増やします。後は、ゲームビュー内でライトの「**Orbit**」、「**Elevation**」、「**Roll**」を調整すれば、アンカーポイントの周りでライト位置を調整できます。シーンビューでライトの「**Transform**」を手動で調整する必要はありません。



Light Anchor コンポーネント

詳細については、[ライトアンカー](#)について紹介するこちらのプレゼンテーションを参照してください。



ライトのトランスフォームを直接変更する代わりに、ライトアンカーを使用してライトに狙いを定めます。

# 物理ベースのライト単位と強度

HDRP では、物理ベースのライト単位 (PLU) を光の強度の測定に使用します。これは、カンデラ、ルーメン、ルクス、ニトなど、現実世界での照度の SI 測定単位と一致しています。PLU では、正確性を確保するため、Unity の 1 単位は 1 メートルに等しいことが前提になっていることにご注意ください。

## 単位

物理ベースのライト単位には、**光束**と**照度**の両方の単位が含まれる場合があります。光束とは光源から放出される光の総量であり、照度とはオブジェクトに当たる光の総量を指します (単位面積あたりの光束で表されるのが一般的)。

商用のライティングや写真の分野では、用途によって使用される単位が異なる場合があります。そのため、Unity では互換性を確保するために、次のように複数の物理ベースのライト単位に対応しています。

- **カンデラ** : 1 単位が、1 本のろうそくの光束に相当します。燭 (燭光) などの呼称も一般的です。
- **ルーメン** : SI の光束単位で、1 **ステラジアン** (立体角) に対する 1 カンデラの光束と定義されます。ルーメンは、商用電球の仕様によく見られます。Unity のスポット、ポイント、エリアライトで使用します。
- **ルクス** : 1 平方メートルのエリアに 1 ルーメンの光を放出する光源が、照度 1 ルクスに相当します。現実世界で光を測定する際は、ルクスを読み取るのが一般的です。通常、Unity のディレクショナルライトではこの単位を使用します。
- **ニト** : 1 平方メートルあたり 1 カンデラに相当する照度単位です。ディスプレイデバイスや LED パネル (テレビやモニターなど) では、明るさをニト単位で測定するのが一般的です。
- **EV<sub>100</sub> : EV<sub>100</sub>**、つまり露出値 100 ISO フィルムに相当する強度を使用します (前述の [露出値の計算式](#) を参照)。露出を上げると、対数的な計算により、ライティングが倍増します。

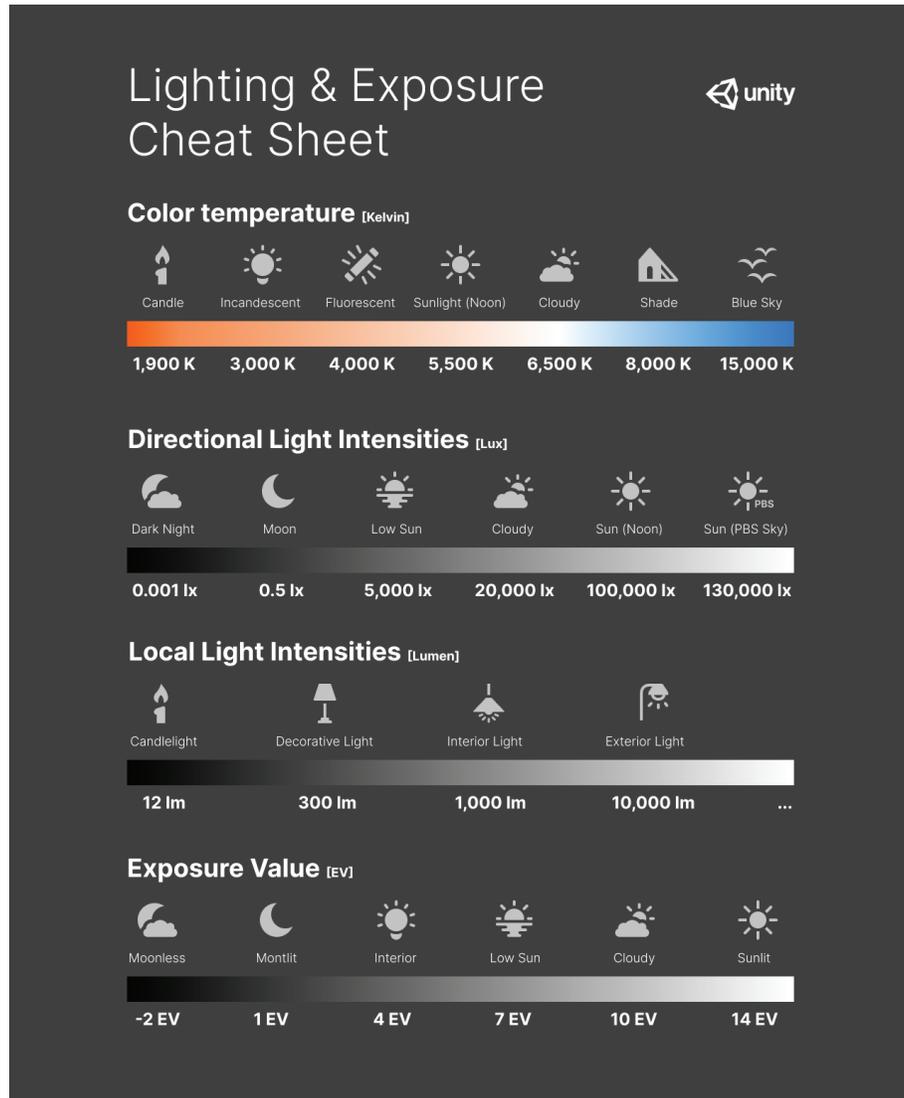
現実世界の光源を再現するには、技術仕様に記載されている単位に切り替え、適切な光束または輝度を設定します。HDRP によって物理ベースのライト単位に合わせることで、勘に頼らずに強度を設定できます。

アイコンをクリックし、「**Exterior**」、「**Interior**」、「**Decorative**」、「**Candle**」のいずれかのプリセットを選択します。特定の値に合わせる必要がない場合は、このような設定から始めることをおすすめします。

## ライティングと露出の一般的な値

以下のチートシートには、現実世界の一般的な光源の色温度の値とライトの強度がまとめられています。また、さまざまなライティングシナリオに合った露出値も紹介されています。

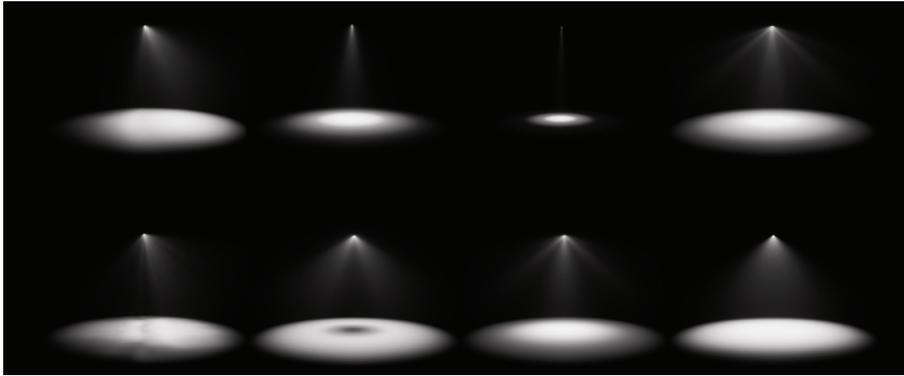
一般的なイルミネーションの値をまとめた詳細な表については、物理ベースのライト単位に関する [ドキュメント](#) をご覧ください。



ライティングと露出のレベルに関するガイダンス ライティングと露出のレベルに関するガイダンス

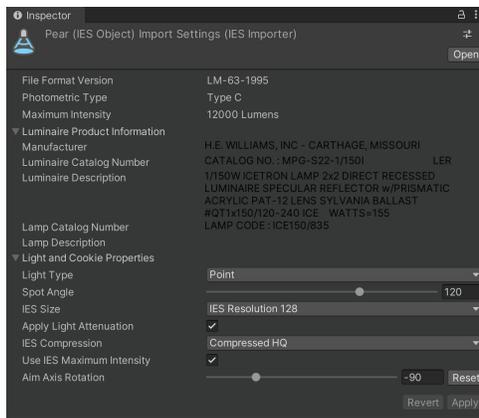
## IES プロファイルとクッキー

**IES プロファイル**を使用すると、ポイント、スポット、エリアライトの減衰を、現実世界の光に近い形で再現できます。これは、特定の製造元の仕様をライトのパターンに適用する **ライトクッキー** のように機能します。IES プロファイルを使用すれば、ライトのリアリティをさらに高めることができます。



さまざまなライトに適用される IES プロファイル

IES プロファイルを「**Assets > Import New Asset**」からインポートします。インポーターによって、適切な強度でライトのプレハブが自動的に作成されます。続いて、プレハブをシーンビューまたはヒエラルキーにドラッグし、色温度を調整します。



IES プロファイルとインポート設定

IES プロファイルのソースの例は次のとおりです。

実際の製造元

- Philips
- Lithonia Lighting
- Efficient Lighting Systems
- Atlas
- Erco
- Lamp
- Osram

アーティストのソース

- Renderman

IES プロファイルのインポーターの詳細については、[ドキュメント](#)をご覧ください。

## 環境ライティング

現実世界では、光は反射して散乱します。空と地面が環境の光へ影響しています。これは大気と地表の間で無差別に跳ね返って散った光子が、最終的に人の目に入っていくからです。

HDRP では、「**Visual Environment**」のオーバーライドを使用して、シーンの空と環境光を定義できます。

「**Ambient Mode: Dynamic**」を使用して、空のライティングを、「Visual Environment」の「**Sky > Type**」に表示される現在のオーバーライドに設定します。または、「**Ambient Mode: Static**」を使用する場合は、「**Lighting**」ウィンドウの「**Environment**」タブの空の設定がデフォルトになります。

他の光源を無効にしている場合でも、SampleScene は「Visual Environment」の環境光に照らされます。



環境ライティングのみ：日光のディレクショナルライトが無効ですが、空には環境光が残っています。 直接的な日光と環境ライティングの組み合わせ

太陽のキーとなるライトを追加することで、シーンの全般的なイルミネーションが決まります。環境光は、シャドウエリアを照らして、不自然に暗くならないようにするのに役立ちます。

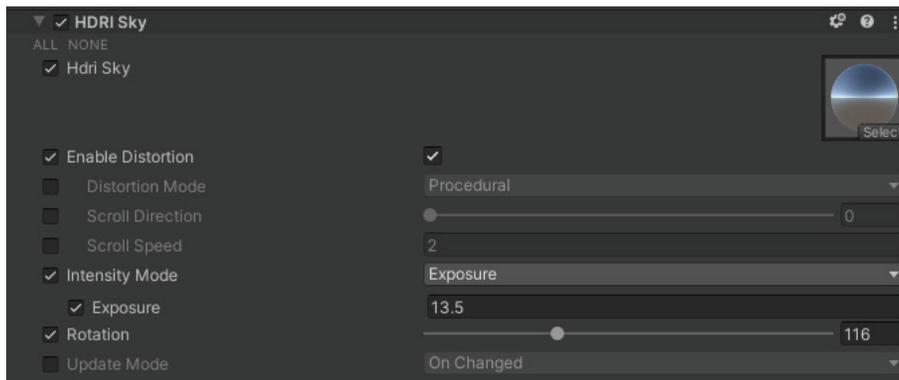
HDRP には、空を生成する手法が 3 種類あります。「Type」を「**HDRI Sky**」、「**Gradient Sky**」、または「**Physically Based Sky**」のいずれかに設定します。続いて、「Sky」メニューから適切なオーバーライドを追加します。

「Visual Environment」の空を適用すると、バーチャルワールド全体が、明るく照らされた巨大な球体で囲まれたような状態になります。この球体の色付きのポリゴンが、空、地平線、地面からの全般的なライトを提供します。

## HDRI Sky

「HDRI Sky」では、[ハイダイナミックレンジの写真](#)から作成したキューブマップを使用して空を表現できます。HDRI については、無料のソースや低コストのソースがオンラインに多数あります。出発点としては、アセットストアにある [Unity HDRI Pack](#) が適しています。

挑戦してみたいと思っている方のために、[自分で HDRI を撮影するためのガイド](#)も用意しています。



HDRI Sky

HDRI アセットをインポートしたら、「**HDRI Sky**」のオーバーライドを追加し、「**HDRI Sky**」アセットをロードします。「**Distortion**」、「**Rotation**」、「**Update Mode**」のオプションを調整することもできます。

空はイルミネーションのソースになるため、「**Intensity Mode**」を指定してから、対応する「**Exposure/Multiplier/Lux**」の値を選択して環境ライティングの強度を制御します。強度と露出の値の例については、前述の[ライティングと露出のチートシート](#)を参照してください。



キューブマップとして球体の内部に適用された HDRI Sky

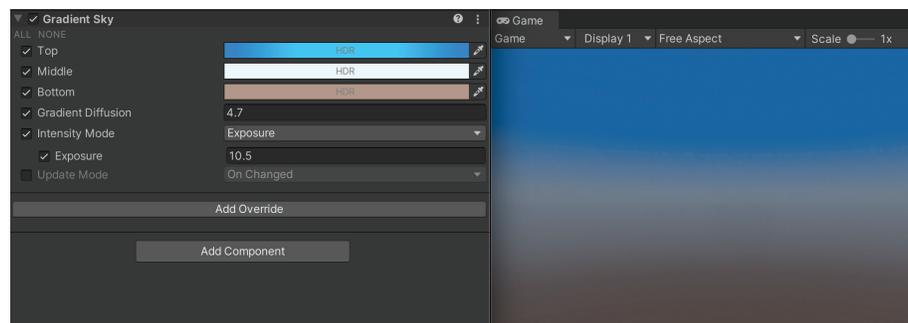
### クラウドレイヤーのアニメーション化

HDRI マップを手続き型またはフローマップを使って歪ませることで、HDRI Sky をアニメーション化することができます。静的 HDRI で風のエフェクトを模したり、より細かい VFX を作成したりすることができます。

## Gradient Sky

「Visual Environment」で「**Gradient Sky**」を選択すると、カラーランプを使用して背景の空を近似します。続いて、「**Gradient Sky**」のオーバーライドを追加します。「**Top**」、「**Middle**」、「**Bottom**」を使用して、グラデーションの色を指定します。

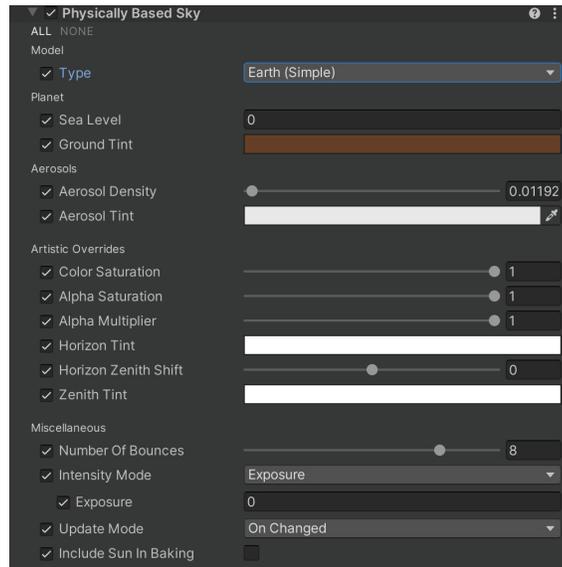
カラーランプを「**Gradient Diffusion**」とブレンドし、ライティングの強さの「**Intensity**」を調整します。



「Gradient Sky」では、「Top」、「Middle」、「Bottom」の色をブレンドします。

## Physically Based Sky

グラデーションより大幅にリアルさを高めたい場合は、「**Physically Based Sky**」のオーバーライドが有効です。これにより、**ミー散乱**や**レイリー散乱**などの現象を再現した空が手続き型で生成されます。この空では、大気を通して分散する光のシミュレーションが行われ、自然な空の配色が再現されます。「Physically Based Sky」では、正確なシミュレーションを行うためにディレクショナルライトが必要です。



「Physically Based Sky」のオーバーライド



手続き型で生成された空（「Fountainbleau」デモより）

### 色のヒント

地面の色は、オブジェクトがリフレクションプロップの影響を受けない実際の地面（地形など）の平均的な色に合わせて選んでください。



「Base Height」と「Maximum Height」を使用して、低い位置に漂うフォグを表現します。

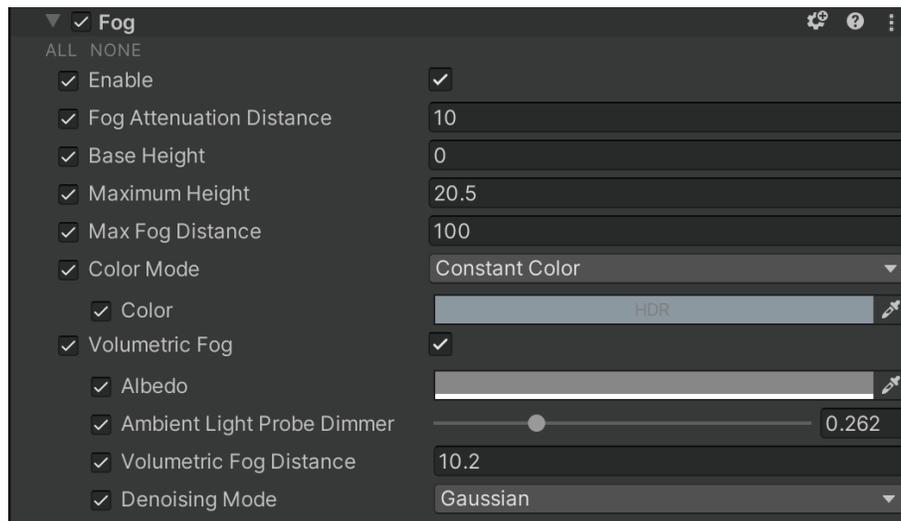
## フォグと大気散乱

スモーク、フォグ、もやは、映画的な画面作りの定番のツールです。奥行きや立体感を加えてライティングを演出したり、趣のある雰囲気を出したりするのに役立ちます。HDRP では、フォグを使用して同様の効果を実現できます。

不透明度は、オブジェクトとカメラの距離に応じます。フォグを利用すると、カメラのファークリップ面を非表示にして、遠くにあるジオメトリをシーンにブレンドすることもできます。

## グローバルフォグ

HDRP では、「Fog」のオーバーライドとしてグローバルフォグを実装しています。この例では、カメラとの距離とワールド空間の高さに合わせて、フォグが急激にフェードしていきます。



「Physically Based Sky」のオーバーライド

シーン内のボリュームで「Fog」のオーバーライドを設定します。「**Base Height**」では、一定の濃いフォグが上方向に向かうにつれ、薄くなり始める境目を指定します。この値を上回ると、「**Maximum Height**」に到達するまで、フォグの密度が急激にフェードしていきます。

同様に、「**Fog Attenuation Distance**」と「**Max Fog Distance**」では、カメラから距離が離れるにつれ、フォグをどのようにフェードさせるのかを設定します。「**Color Mode**」は、「**Constant Color**」と既存の「**Sky Color**」のどちらかに設定します。



Fog Attenuation Distance

「**Volumetric Fog**」を有効にすると、大気散乱のシミュレーションが行われます。「**Lighting**」の「**Frame Settings**」（カメラの下か、「**HDRP Default Settings**」内）で、「**Fog**」と「**Volumetrics**」がオンになっていることを確認します。また、「**HDRP Pipeline Asset**」で「**Volumetric Fog**」も有効にします。

「**Volumetric Fog Distance**」では、カメラのニアクリップ面からボリュメトリックライティングバッファの背部までの距離（メートル単位）を設定します。これにより、大気中に浮遊物質が満たされ、範囲内のゲームオブジェクトが部分的に隠れます。

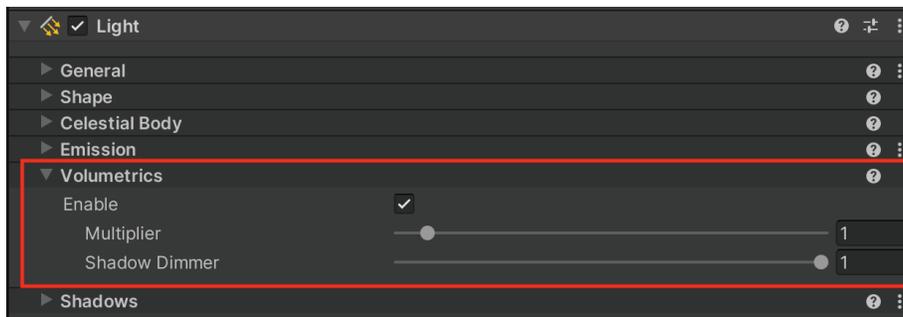


ボリュメトリックフォグは、より正確に前景のジオメトリを保持します。

## ボリュメトリックライティング

[ボリュメトリックライティング](#)によって、日没時の雲の背後や植生の間から射す**光芒**など、印象的な太陽光線をシミュレーションでレンダリングできます。

それぞれの Light コンポーネント（エアライト以外）に、「**Volumetrics**」グループがあります。「**Enable**」をオンにしてから、「**Multiplier**」と「**Shadow Dimmer**」を設定します。「**Real-time**」または「**Mixed Mode**」のライトを使用すると、ボリュメトリックフォグ内に「薄明光線」が生成されます。「Multiplier」で強度を調整し、「Shadow Dimmer」を使用して、シャドウを投影するサーフェスがライトを遮る具合を設定します。



「Light」コンポーネントの「Volumetrics」

このサンプルシーンの Room 2 では、空の光とボリュームトリックフォグが使用されています。ガラスケースのフレームによって、天井からの太陽光線に伴うボリュームトリックシャドウが形成されています。「Shadow Dimmer」と「Multiplier」の値を高めると、エフェクトが強くなります。



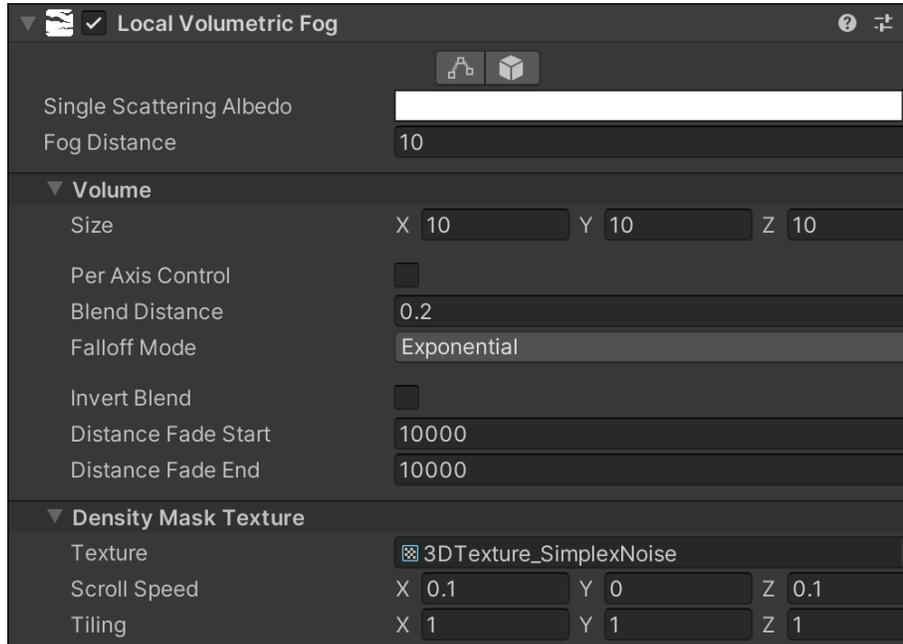
ボリュームトリックライティングが、ボリュームトリックフォグのエリアを印象的に照らしています。



天井のスポットライトに適用されたボリュームトリックライティング

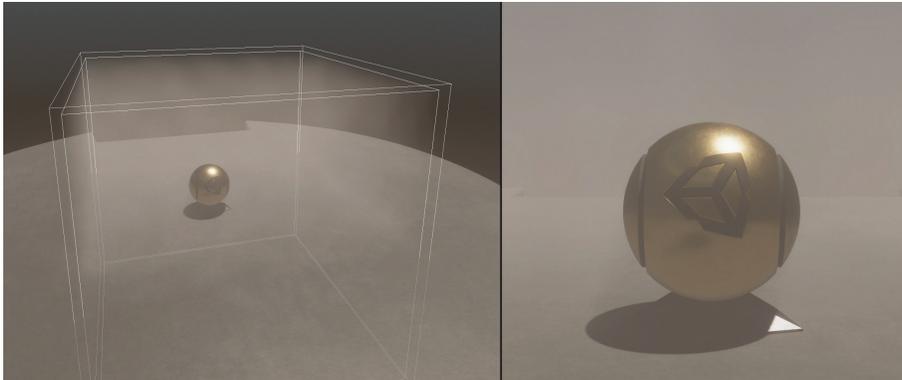
## ローカルボリュメトリックフォグ

HDRP では、「Fog」のオーバーライドよりも細かいフォグエフェクトが必要な場合に、「[Local Volumetric Fog](#)」（HDRP 12 より前のバージョンでの呼称は「Density Volume」コンポーネント）を利用できます。



「Local Volumetric Fog」コンポーネント

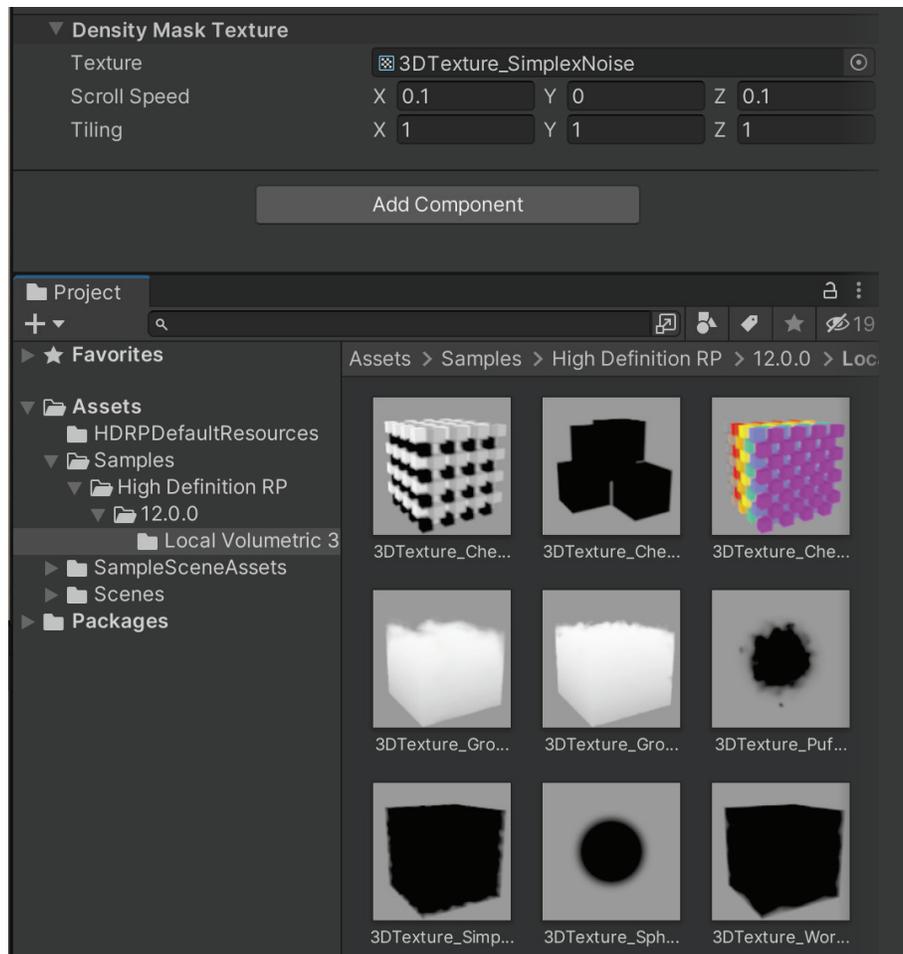
これはボリュームシステムの外にある別個のコンポーネントです。メニューから「**Local Volumetric Fog**」のゲームオブジェクトを作成（「**GameObject > Rendering > Local Volumetric Fog**」）するか、「Hierarchy」を右クリックします（「**Rendering > Local Volumetric Fog**」）。



ローカルボリュメトリックフォグはバウンディングボックスに表示されます

これにより、フォグが満ちたバウンディングボックスが生成されます。サイズ、軸のコントロール、ブレンディング / フェーディングのオプションを調整します。

デフォルトではフォグは一樣ですが、3D テクスチャーを「Density Mask Texture」サブセクションの「Texture」フィールドに適用できます。これにより、ユーザーはより柔軟にフォグの外観をコントロールできます。Package Manager の「Local Volumetric 3D Texture Samples」からサンプルをダウンロードするか、[ドキュメントの手順](#)に即して密度マスクを作成します。

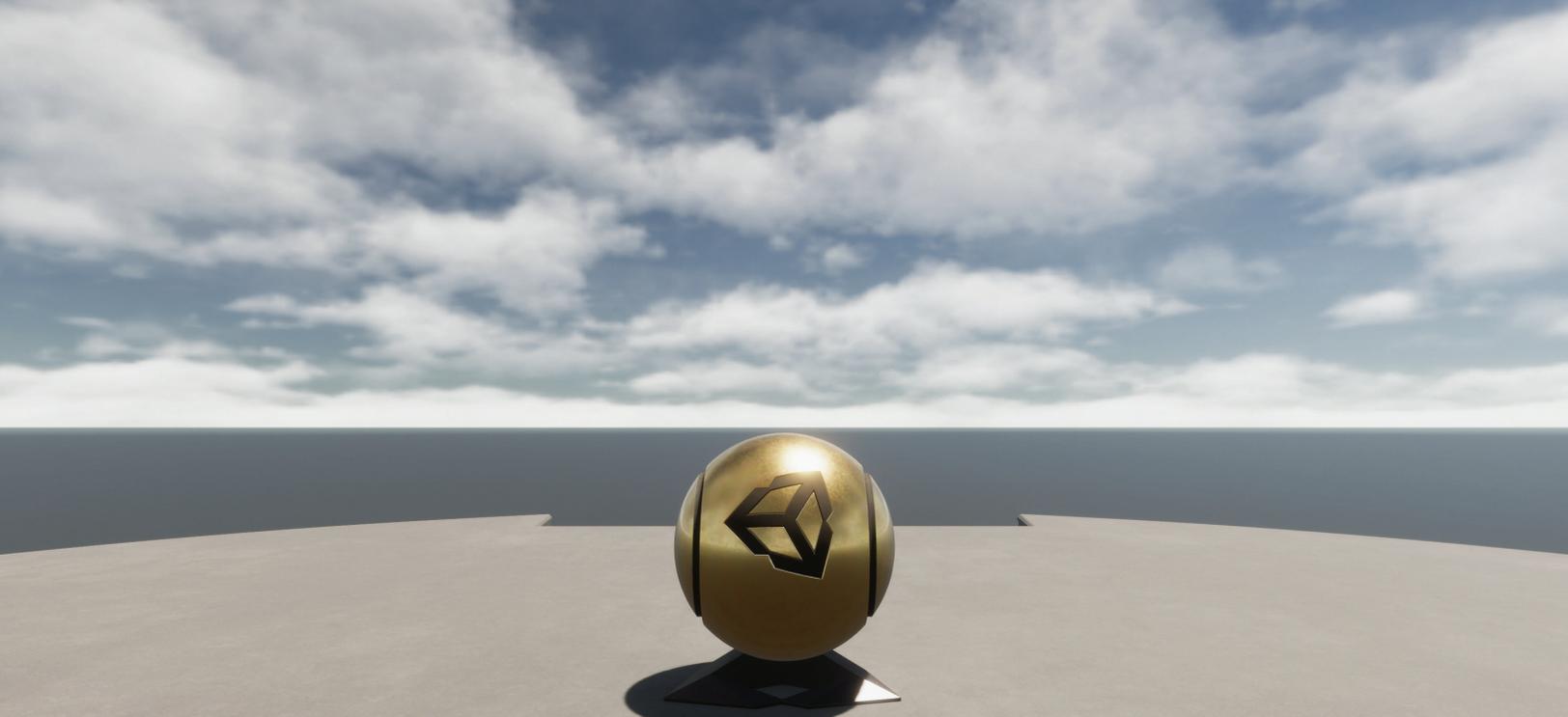


「Local Volumetric 3D Texture Samples」の「Density Mask Texture」

アニメーションの「Scroll Speed」を追加し、「Tiling」を調整します。以上で、ボリュームメトリックフォグを徐々にシーン内に描画できます。

注：HDRP では、ローカルボリュームメトリックフォグをボクセル化してパフォーマンスを高めます。ただし、ボクセル化によって外観が非常に粗くなる場合があります。エイリアシングを低減するためには、「Density Mask Texture」を使用し、「Blend Distance」を上げてフォグのエッジを滑らかにします。

HDRP バージョン 12 以降では、「HDRP Pipeline Asset」で最大 256×256×256 のローカルボリュームメトリック解像度を有効にできるため、より正確で大規模なエフェクトが可能となります。



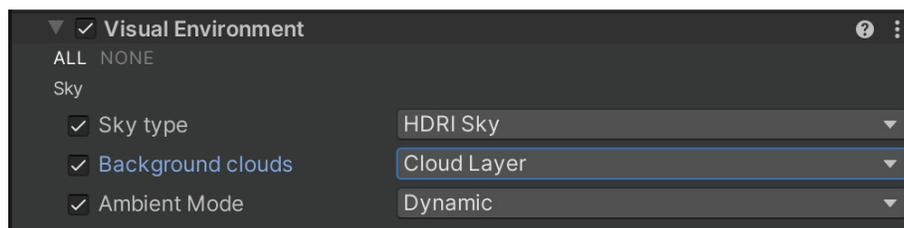
クラウドレイヤーが HDRI Sky の前面に表示されます。

# 雲

空は雲がなければ完全とはいえません。HDRP 12 以降では、[クラウドレイヤー](#)システムで自然な外観の雲を生成することにより、**Sky** と **Visual Environment** のオーバーライドを補間することができます。「[Volumetric Clouds](#)」は、ライティングや風に反応して、実際の厚みを持ったリアルな雲を作り出します。

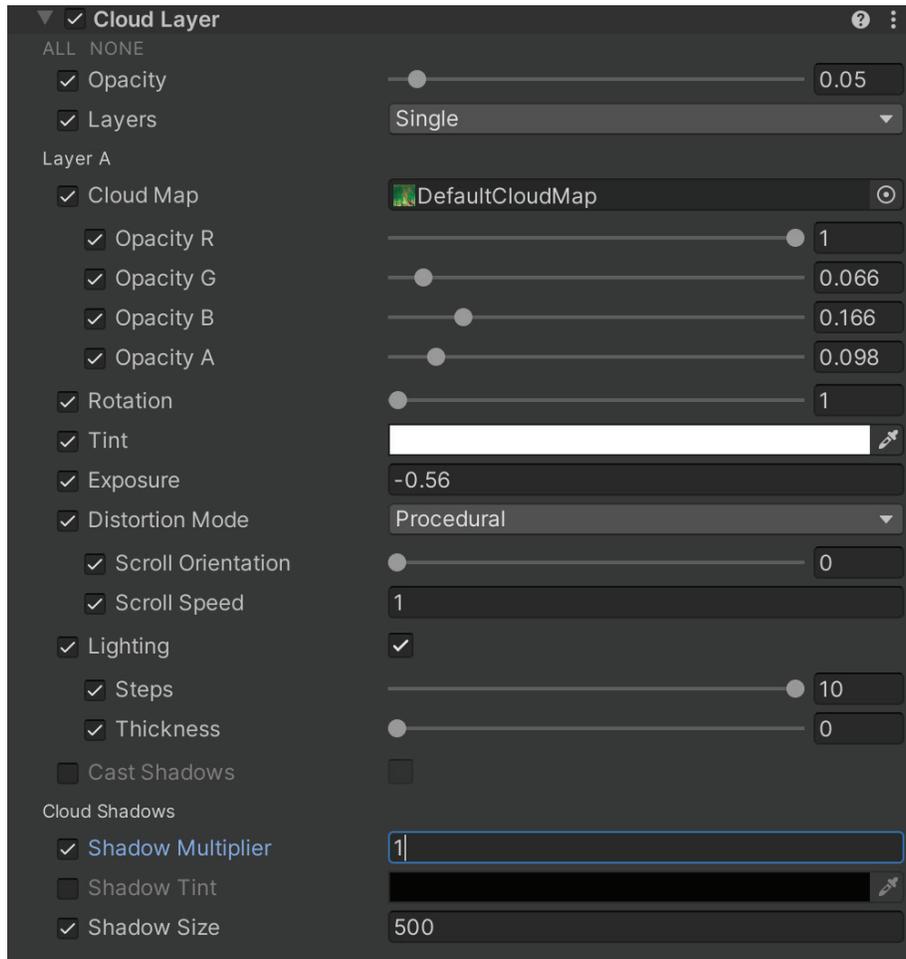
## クラウドレイヤー

[クラウドレイヤー](#)は、[フローマップ](#)を使ってアニメーション化できる 2D テクスチャです。ベクトルのディスプレイメントが、赤色と緑色のチャンネルを使用して制御されます。再生モードで空にわずかな動きを加えることで、背景をより動的にすることができます。クラウドレイヤーは空の手前に配置され、オプションで地面に影を付けることができます。



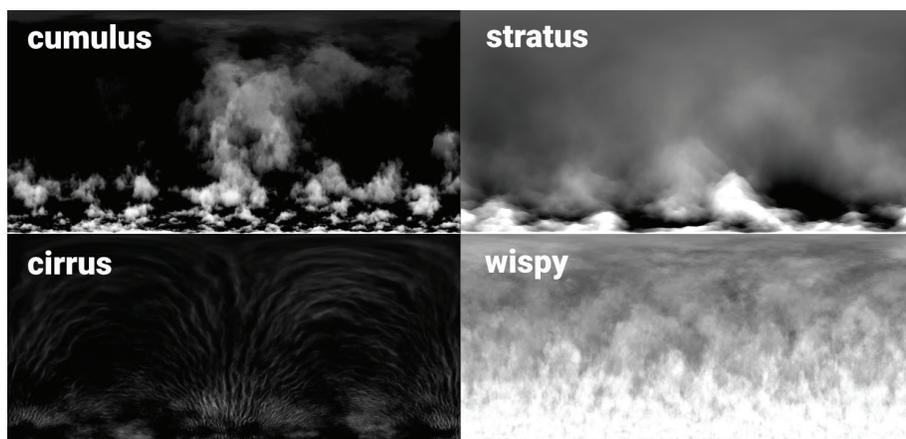
「Visual Environment」でクラウドレイヤーを有効にする

ローカルまたはグローバルボリュームで、「Visual Environment」の「**Background clouds**」を有効にし、「**Cloud Layer**」オーバーライドを追加します。



クラウドレイヤーのオーバーライド

クラウドマップ自体は、[円柱投影](#)を使用したテクスチャで、RGBA チャンネルにはそれぞれ異なる雲のテクスチャ（積雲、層雲、巻雲、薄雲）が含まれています。そのうえで、クラウドレイヤーコントロールで各チャンネルをブレンドし、好みの雲形状を作り上げることができます。4 つのチャンネルを持つ 2 つのレイヤーで、最大 8 つの雲景のシミュレーションとブレンドを行うことができます。



DefaultCloudMap の 4 つのチャンネル

雲のアニメーション、ライティング、色、影などを修正しながら、好みの空の景色に仕上げましょう。

## ボリュメトリッククラウド

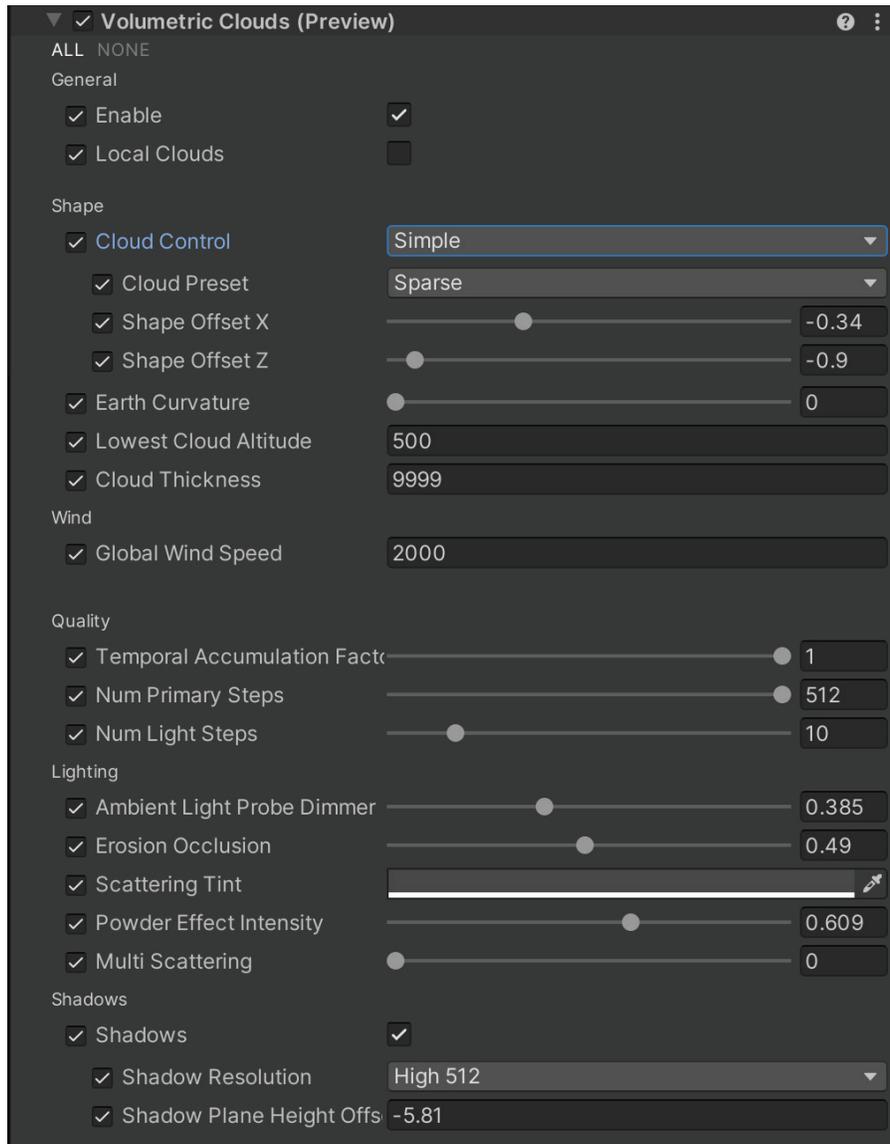
雲と光を相互に作用させる必要がある場合は、[ボリュメトリッククラウド](#)を使用します。影をレンダリングしたり、フォグを活用したり、ボリュメトリックライトシャフトを作成したりすることができます。これらをクラウドレイヤーの雲と組み合わせたり、別々に追加したりします。

ボリュメトリッククラウドを有効にするには、以下のようになります。

- HDRP Asset で、「**Lighting > Volumetric Clouds > Volumetric Clouds**」を有効にします。
- ローカルボリュームまたはグローバルボリュームで、「**Volumetric Clouds**」オーバーライドを追加します。



ボリュメトリッククラウドは、メインのディレクショナルライトに反応できます。



「Volumetric Clouds」のオーバーライド

各種の雲のマップは、「Advanced」と「Manual」の「**Cloud Control**」オプションで定義できます。

クラウドレイヤーとボリュメトリッククラウドのオーバーライドについては、[HDRPのクラウドに関するドキュメント](#)を参照してください。ボリュメトリッククラウドの詳細については、「[Unity 2021.2の新しいライティング機能](#)」を参照してください。

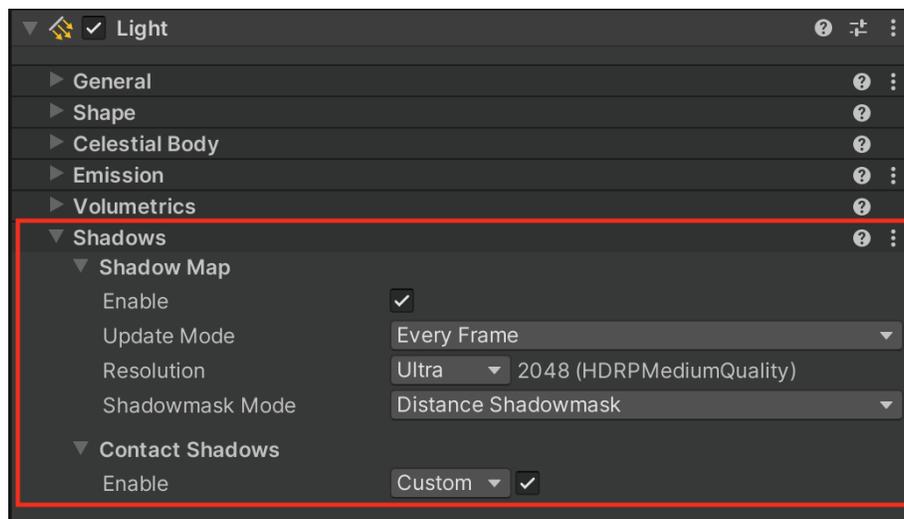
# シャドウ

暗闇がなければ光を認識することはできません。シーン内に適切に配置されたシャドウは、ライティングそのものと同じくらい趣を生み出し、シーンに奥行きや立体感を加えることができます。HDRP には、シャドウを微調整し、レンダリングが単調になるのを防ぐための機能が多数あります。

## シャドウマップ

シャドウは、**シャドウマッピング**という手法でレンダリングされます。この手法では、ライトの視点からの深度の情報がテクスチャーに保管されます。

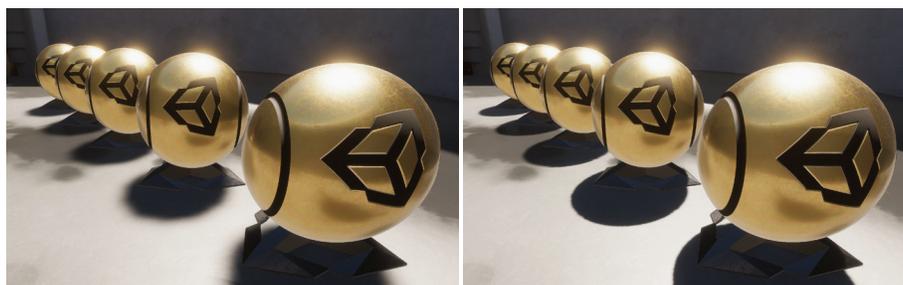
「Light」コンポーネントの「Shadows」サブセクションで、シャドウマッピングの「Update Mode」と「Resolution」を変更できます。解像度と更新頻度の設定値が高くなるほど、リソースの消費が大きくなります。



ライトごとのシャドウ設定

## シャドウカスケード

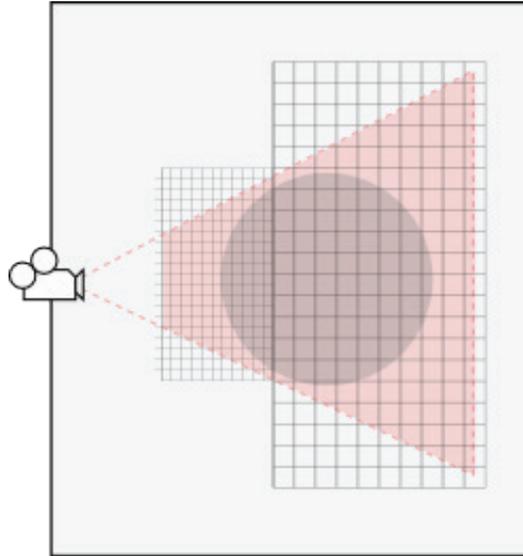
ディレクショナルライトの場合は、シャドウマップがシーンの大部分を覆うため、**透視エイリアシング**と呼ばれる問題が生じることがあります。カメラから近い位置にあるシャドウマップのピクセルは、遠くにあるものよりも、ギザギザで粗くなります。



透視エイリアシングと粗いシャドウ

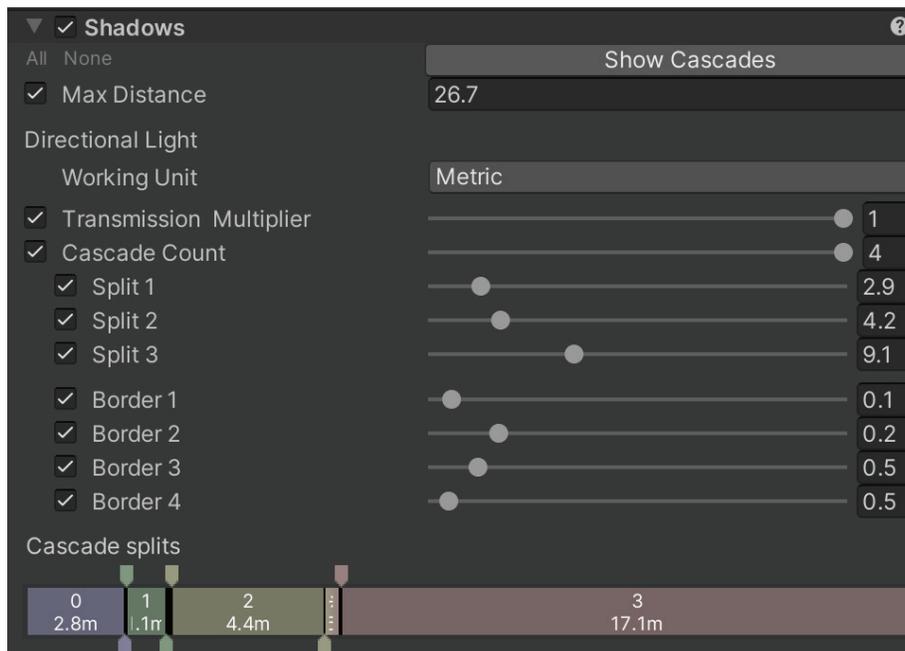
シャドウカスケードによって透視エイリアシングが低減されます

Unity では、[カスケードシャドウマップ](#)を使用してこの問題を解決しています。この手法では、カメラの錐台を、それぞれにシャドウマップがある複数のゾーンに分割します。これにより、透視エイリアシングのエフェクトが低減されます。



Shadow cascadesでは、カメラの錐台を複数のゾーンに分割します。各ゾーンにそれぞれShadowマップがあります。

HDRP では、「[Shadows](#)」のオーバーライドで、Shadow cascadesをより詳細に制御できます。ボリュームごとのカスケード設定を使用して、それぞれのカスケードの始点と終点を微調整できます。



「Shadows」のオーバーライド

「**Show Cascades**」 ボタンを切り替えて、カスケードの分割をより簡単に可視化できます。ある程度の調整を加えることで、透視エイリアシングを最小限に抑えることができます。



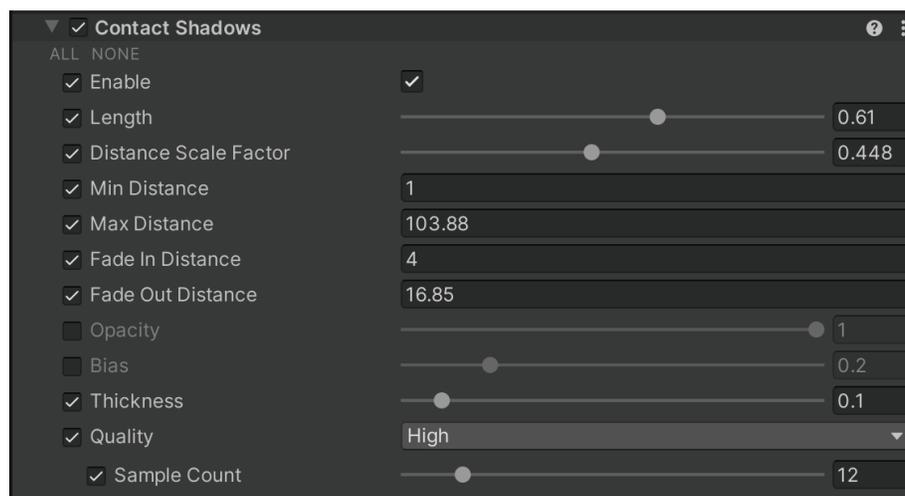
「Show Cascades」をクリックすると、カスケードの分割を可視化できます。

## コンタクトシャドウ

シャドウマップは、特に 2 つのメッシュサーフェスが結合する目立ちやすいエッジで、小さなディテールを取り込めないことがよくあります。HDRP では、「**Contact Shadows**」のオーバーライドを使用して、こうしたコンタクトシャドウを生成できます。

コンタクトシャドウはスクリーンスペースエフェクトの一種で、フレーム内の情報を基に計算を行います。フレーム外のオブジェクトは、コンタクトシャドウには影響しません。コンタクトシャドウは、画面上のフットプリントが小さいシャドウのディテールに使用します。

「**Frame Settings**」で「**Contact Shadows**」を有効にします。「**Lighting Quality Settings**」の「**Pipeline Asset**」で、「**Sample Count**」を調整することもできます。



「Contact Shadows」のオーバーライド



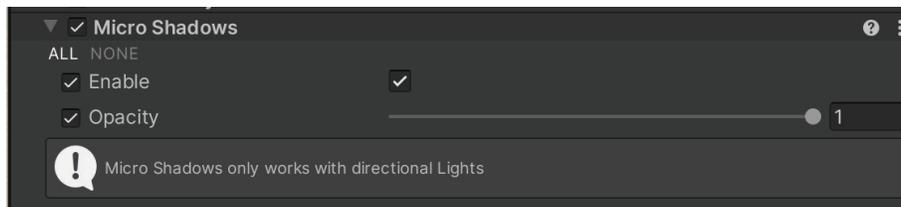
コンタクトシャドウ

シャドウの厚み、品質、フェードの設定を調整します。

この機能は、2021 LTS 以降のバージョンで、Terrain（地形）や SpeedTree との連携が改善されました。[ブログで詳細をご覧ください。](#)

## Micro Shadows

HDRP では、小さなシャドウの詳細をマテリアルに広げることができます。「**Micro Shadows**」は、法線マップとアンビエントオクルージョンマップを使用して、メッシュのジオメトリそのものを使用せずに、非常に細かいサーフェスのシャドウをレンダリングします。



「Micro Shadows」のオーバーライド

「**Micro Shadows**」のオーバーライドをシーン内のボリュームに追加し、「Opacity」を調整します。「Micro Shadows」を利用できるのはディレクショナルライトのみです。



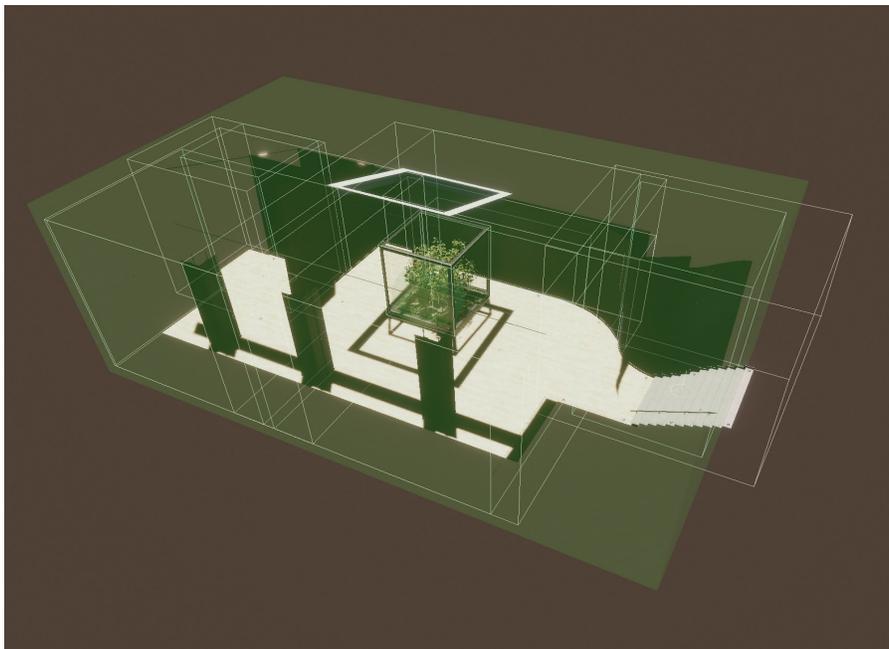
マイクロシャドウによって、植え込みのコントラストが強められています。

# リフレクション

リフレクションは、ゲームオブジェクトと周囲の環境を統合するのに役立ちます。通常、リフレクションは、滑らかで光沢のあるサーフェスに関連付けますが、滑らかではないマテリアルの場合でも、PBR ワークフローで適切なリフレクションを受ける必要があります。HDRP では、次のようなさまざまな手法でリフレクションを生成できます。

- スクリーンスペースリフレクション
- リフレクションプローブ
- スカイリフレクション

各リフレクションタイプは、リソースの消費が大きくなる場合があるため、ユースケースに合わせて最適な手法を選択してください。複数のリフレクション手法がピクセルに適用される場合、各リフレクションタイプの効果がブレンドされます。「Influence Volume」という名前のバウンディングサーフェスを使用して 3D スペースを分割し、リフレクションを受け取るオブジェクトを指定してください。

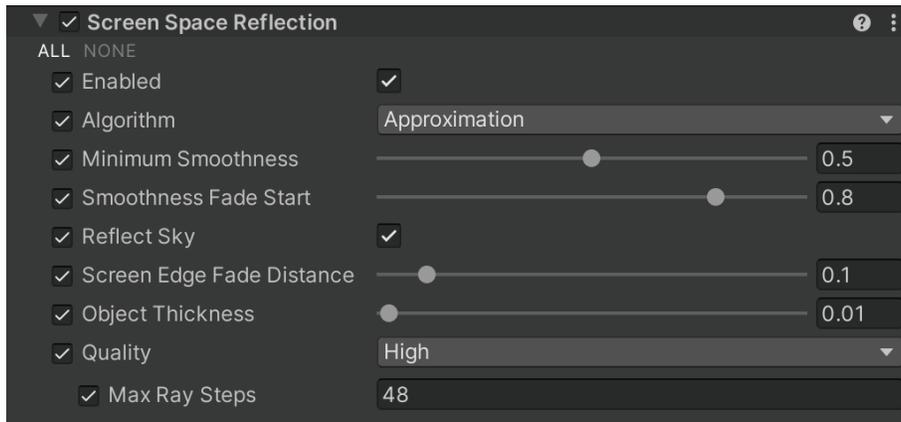


「Influence Volume」では、リフレクションプローブによってリフレクションを生成する場所を指定します。

## スクリーンスペースリフレクション

スクリーンスペースリフレクションでは、深度とカラーバッファを使用してリフレクションを計算します。そのため、反射するのはカメラビューに入っているオブジェクトのみになり、画面上の位置によっては適切にレンダリングされない場合があります。光沢のある床や濡れた平面サーフェスが、スクリーンスペースリフレクションを受け取る対象として適しています。

スクリーンスペースリフレクションでは、フレーム外のオブジェクトはすべて無視されるため、それがエフェクトの制限となる場合があります。



「Screen Space Reflection」のオーバーライド



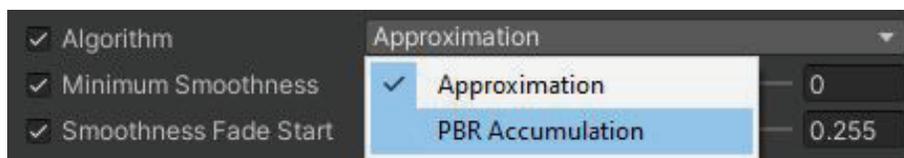
スクリーンスペースリフレクション

「Lighting」の「Frame Settings」（「HDRP Default Settings」またはカメラの「Custom Frame Settings」）で、「Screen Space Reflection」が有効になっていることを確認してください。続いて、「Screen Space Reflection」のオーバーライドをボリュームオブジェクトに追加します。

スクリーンスペースリフレクションを表示するには、マテリアルのサーフェスを「Minimum Smoothness」の値以上にする必要があります。比較的粗いマテリアルに SSR を適用したい場合は、この値を引き下げます。ただし、「Minimum Smoothness」のしきい値を下げると、計算コストが大きくなる場合があります。スクリーンスペースリフレクションの効果をピクセルに適用できなかった場合は、HDRP はリフレクションプロップを使用する方法にフォールバックします。

「Quality」ドロップダウンを使用して、既定数の「Max Ray Steps」を選択します。「Max Ray Steps」が多くなると、クオリティは上がりますが、コストは大きくなります。すべてのエフェクトと同様、パフォーマンスとビジュアルクオリティのバランスをとることが大切です。

注：HDRP バージョン 12 以降では、アキュムレーションを用いた物理ベースのアルゴリズム（新）と、それより精度の低い近似アルゴリズム（デフォルト）のいずれかを選択することができます。



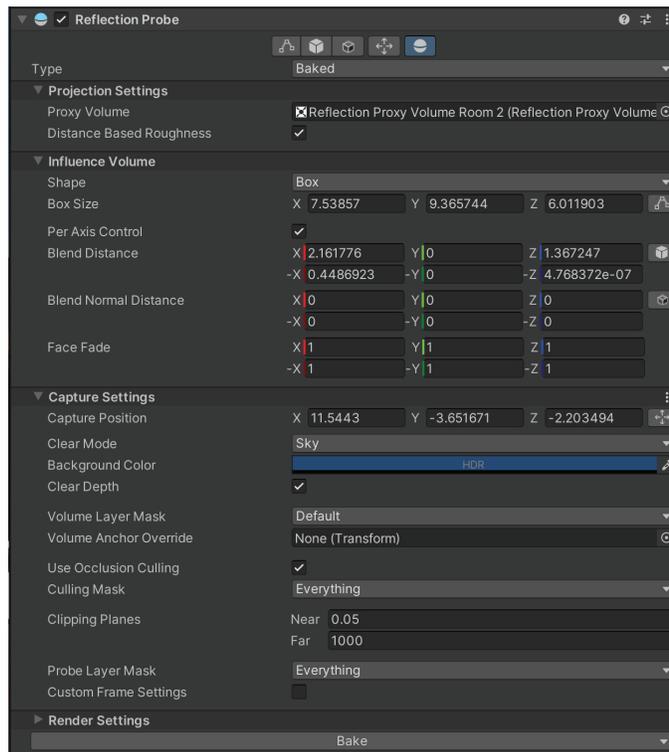
## リフレクションプローブ

リフレクションプローブを使用すると、画像ベースの手法でリフレクションが生成されます。プローブは、周囲全方向の球面ビューを取り込み、その結果をキューブマップのテクスチャーに保管します。シェーダーは、そのキューブマップを使用してリフレクションを再現します。

各シーンに複数のプローブを適用し、結果をブレンドできます。以降、ローカライズされたリフレクションが環境内のカメラの動きに合わせて変化します。

各プローブの「Type」を「Baked」または「Real-time」に設定します。

- 「Baked」のプローブは、静的環境でキューブマップのテクスチャーを1回だけ処理します。
- 「Real-time」プローブは、ランタイム時にエディターではなくプレイヤー内にキューブマップを作成します。そのためリフレクションが静的オブジェクトに限定されなくなりますが、リアルタイム更新によってリソース消費が大きくなる場合があることに注意してください。



「Reflection Probe」コンポーネント

### 最適化のヒント

リアルタイムリフレクションプローブを最適化するには、全般的なカメラ設定またはリフレクションプローブごとのカメラ設定をオーバーライドして、リフレクションのビジュアルクオリティに大きく寄与しないレンダリング機能をすべて無効にします。また、スクリプトを作成して更新をタイムスライスすることもできます。

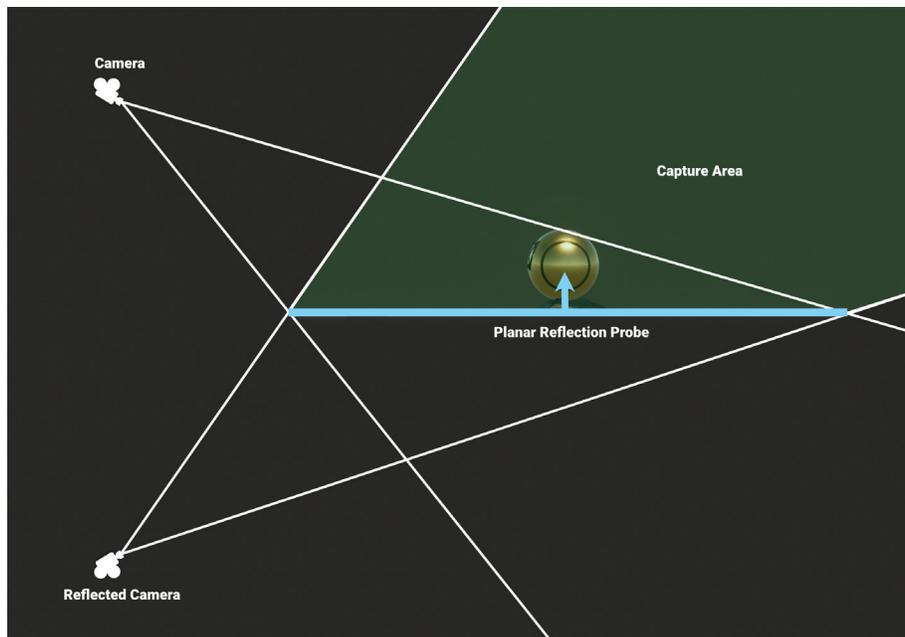
「**Influence Volume**」によって、ゲームオブジェクトがリフレクションを受け取る 3D の境界を指定します。一方「**Capture Settings**」では、リフレクションプローブによってキューブマップのスナップショットを取得する方法をカスタマイズできます。

## 平面リフレクションプローブ

平面リフレクションプローブでは、サーフェスのスムーズネスを考慮して、反射する平面サーフェスを再現できます。鏡や光沢のある床を表現するには最適です。

平面リフレクションプローブは、通常のリフレクションプローブとの共通点が多いですが、処理が若干異なります。環境をキューブマップとして取り込むのではなく、プローブの鏡面で反射されたカメラのビューを再現します。

その後、生成された反射像が 2D RenderTexture として保存されます。この画像が矩形のプローブの境界に描画され、平面リフレクションを生み出します。



平面リフレクションプローブでは、平面にカメラを反射させることで、鏡面画像を取り込みます。



3 種類のマテリアルに適用された平面リフレクションプローブ

## スカイリフレクション

近傍のリフレクションプローブの影響を受けないオブジェクトは、スカイリフレクションにフォールバックします。



リフレクションプローブは周囲の部屋を映し、スカイリフレクションはグラデーションのある空を映します。

### リフレクション階層

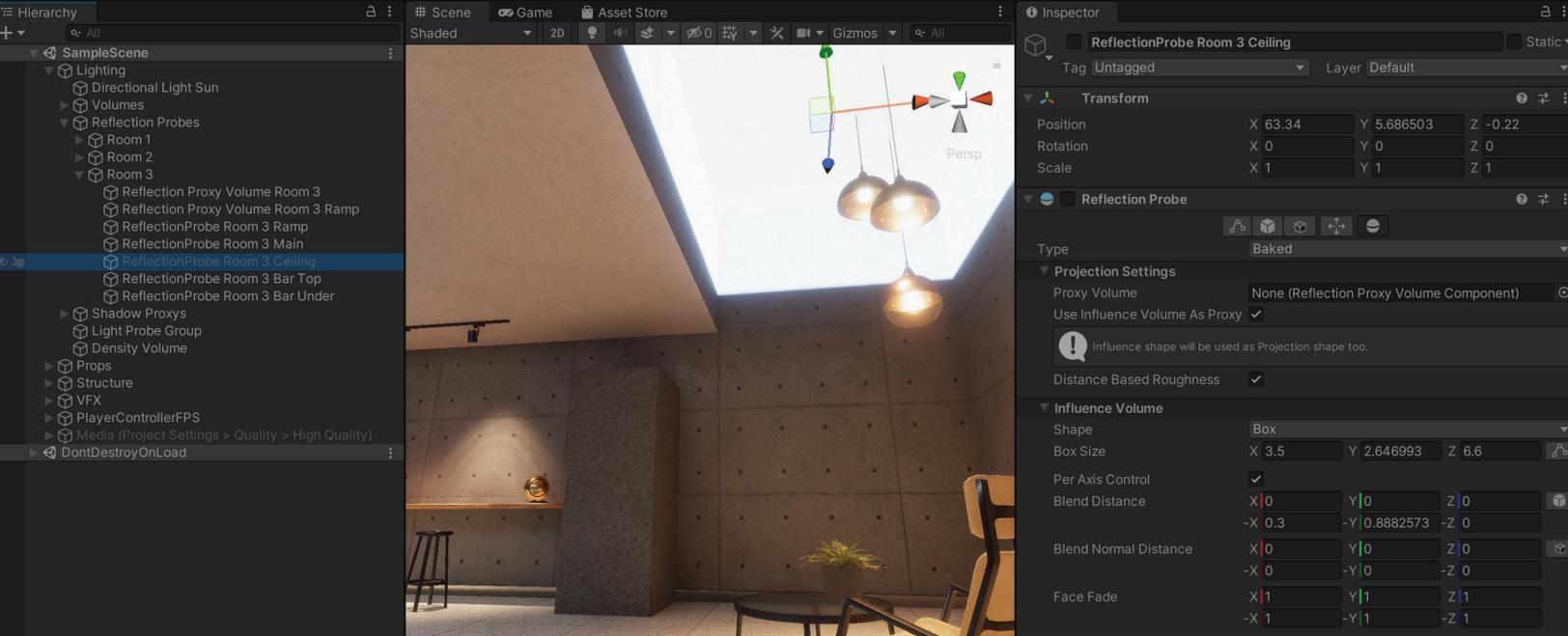
HDRP では質の高いリフレクションが得られるよう、各ピクセルの精度を最大限に高め、他の手法とブレンドすることができる手法を使用しています。HDRP では、加重優先度を基に、3つのリフレクション方法（SSR、リフレクションプローブ、空）をチェックします。リフレクションを評価する際のこのシーケンスは、[リフレクション階層](#)と呼ばれます。

1つのピクセルで、ある手法によってリフレクションを特定できなかった場合は、その次の手法にフォールバックします。つまり、スクリーンスペースリフレクションはリフレクションプローブにフォールバックし、その後スカイリフレクションにフォールバックします。

「Reflection Probe」プロパティで、「Influence Volume」を設定することが大事です。これを設定しないと、不適切なスカイリフレクションからライトが漏れる場合があります。

これは、SampleScene の Room 3 で顕著です。いずれかの「Reflection Probe」を無効にするか、「Influence Volume」を変更すると、リフレクションが強制的に空にフォールバックされます。これにより、HDRI の明るい空が、強力なリフレクションによってシーンよりも過剰に強調されます。

リフレクション階層の確認の詳細については、ドキュメントの [HDRP のリフレクション](#) に関するページをご覧ください。

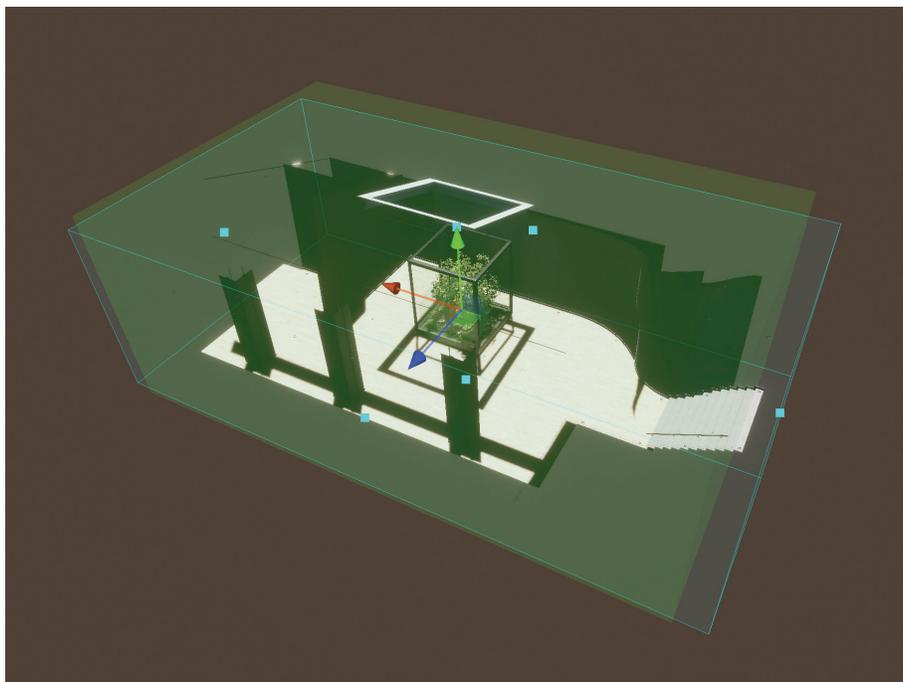


Room 3 の天井でリフレクションプローブを無効にすると、不要なライトの漏れが生じます。

## リフレクションプロキシボリューム

リフレクションプローブのキャプチャポイントは固定されており、リフレクションプローブの近くのカメラ位置と一致することはほとんどないため、生成されたリフレクションで、透視のシフトが目立つ場合があります。それにより、リフレクションが環境になじんでいないように見える場合があります。

リフレクションプロキシボリュームは、これを部分的に修正するのに役立ちます。カメラ位置を基に、プロキシボリューム内でより正確にリフレクションを再投影します。



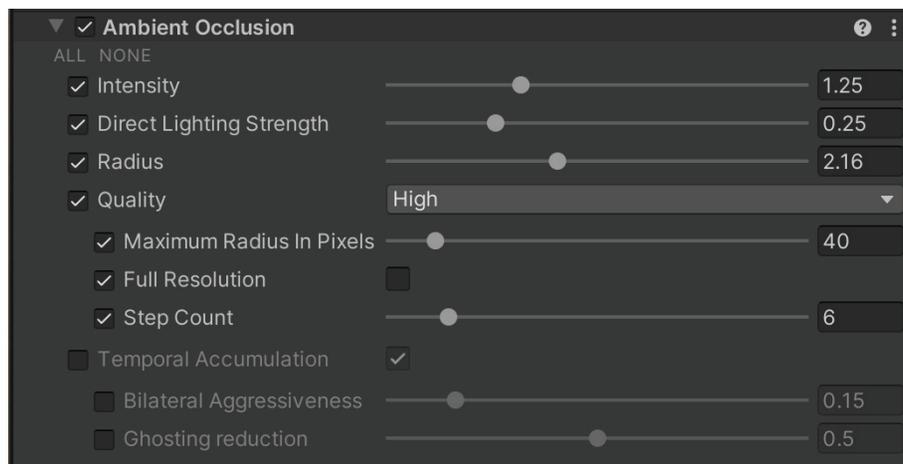
リフレクションプロキシボリュームでは、部屋のワールド空間の位置に合わせてキューブマップを再投影します。

# リアルタイムライティング グエフェクト

HDRP は、ボリュームシステムを通じて、いくつかのリアルタイムライティングエフェクトを利用することもできます。ローカルまたはグローバルボリュームを選択してから、「**Add Override > Lighting**」で適切なエフェクトを追加します。

## Screen Space Ambient Occlusion

アンビエントオクルージョンでは、近接した溝や穴、サーフェスの影のシミュレーションを行います。アンビエントライトが遮断されるエリアは、陰影が付いているように見えます。



「Ambient Occlusion」のオーバーライド



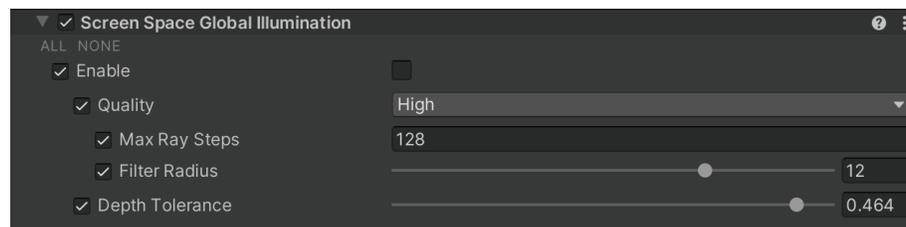
「Auto Showroom」プロジェクトでのアンビエントオクルージョンの可視化

Unity のライトマッパーを使用して静的ジオメトリのアンビエントオクルージョンをベイクできますが、HDRP によって、リアルタイム処理を行う **Screen Space Ambient Occlusion** を追加で利用できます。これはスクリーンスペースエフェクトなので、フレーム内の情報のみが生成されるエフェクトに影響します。SSAO では、カメラの有効視野外のオブジェクトはすべて無視されます。

「Lighting」の「Frame Settings」で、「Screen Space Ambient Occlusion」を有効にします。続いて、ローカルまたはグローバルボリュームで「Add Override」をクリックし、「Lighting > Ambient Occlusion」を選択します。

### スクリーンスペースグローバルイルミネーション

スクリーンスペースグローバルイルミネーション (SSGI) では、画面の深度とカラーバッファーを使用して、反射するディフューズライトを計算します。ライトマッピングで間接光を静的レベルのジオメトリのサーフェスにベイクする方法と似ていますが、SSGI では光子がサーフェスに衝突して反射する際に色やシェーディングが変化する仕組みのシミュレーションをさらに正確に行うことができます。



「Screen Space Global Illumination」のオーバーライド

このサンプルシーンの Room 2 では、SSGI を有効にすることで、揺れる木の葉の緑が反射光として壁面に映っていることがわかります。



スクリーンスペースグローバルイルミネーションによって、植物の葉の反射光をリアルタイムで描画しています。

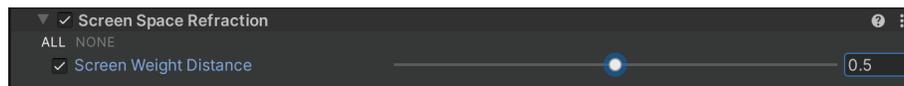
カメラの有効視野外のオブジェクトはグローバルイルミネーションには影響しないため、フレームバッファに依存する他のエフェクトと同様、画面端が問題になります。このジレンマは、フレームバッファの外でレイマーチングを行う際にフォールバックを提供するリフレクションプローブを使用することで部分的に改善することができます。

SSGI は、「**Lighting**」の「**Frame Settings**」で有効にします。パイプラインアセットの「**Lighting**」セクションでも有効にする必要があります。

注：スクリーンスペースグローバルイルミネーションを含む Unity 2021.2 以降を使用することをおすすめします。HDRP 12 では、SSGI のクオリティが大幅に改善されています。

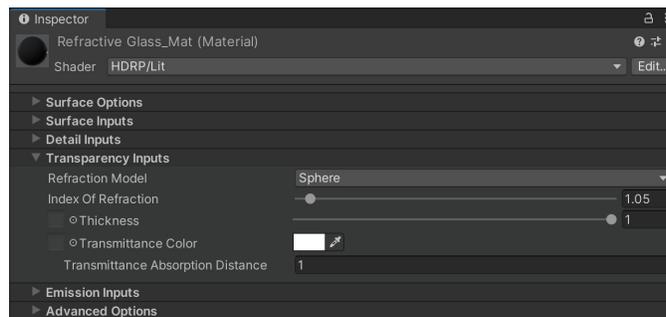
## Screen Space Refraction

「**Screen Space Refraction**」オーバーライドは、大気よりも密度が高いメディアを通過する光の動きのシミュレーションを行う際に役立ちます。HDRP の Screen Space Refraction では、深度とカラーバッファを使用して、ガラスなどの透明なマテリアルを透過する光の屈折を計算できます。



「Screen Space Refraction」のオーバーライド

**HDRP/Lit** シェーダーでこのエフェクトを有効にするには、マテリアルの「**Surface Type**」が「**Transparent**」になっていることを確認します。

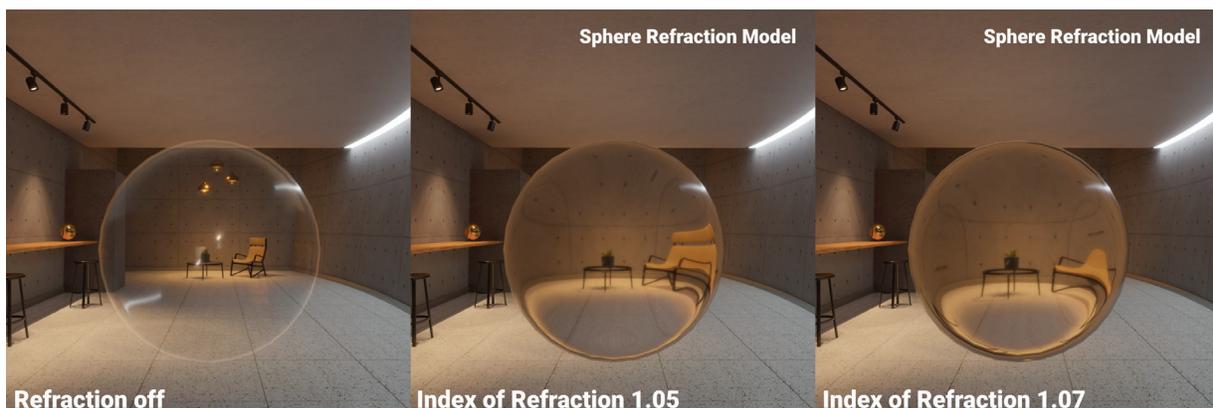


屈折を制御する「Transparency Inputs」

続いて、

「**Transparency Inputs**」で

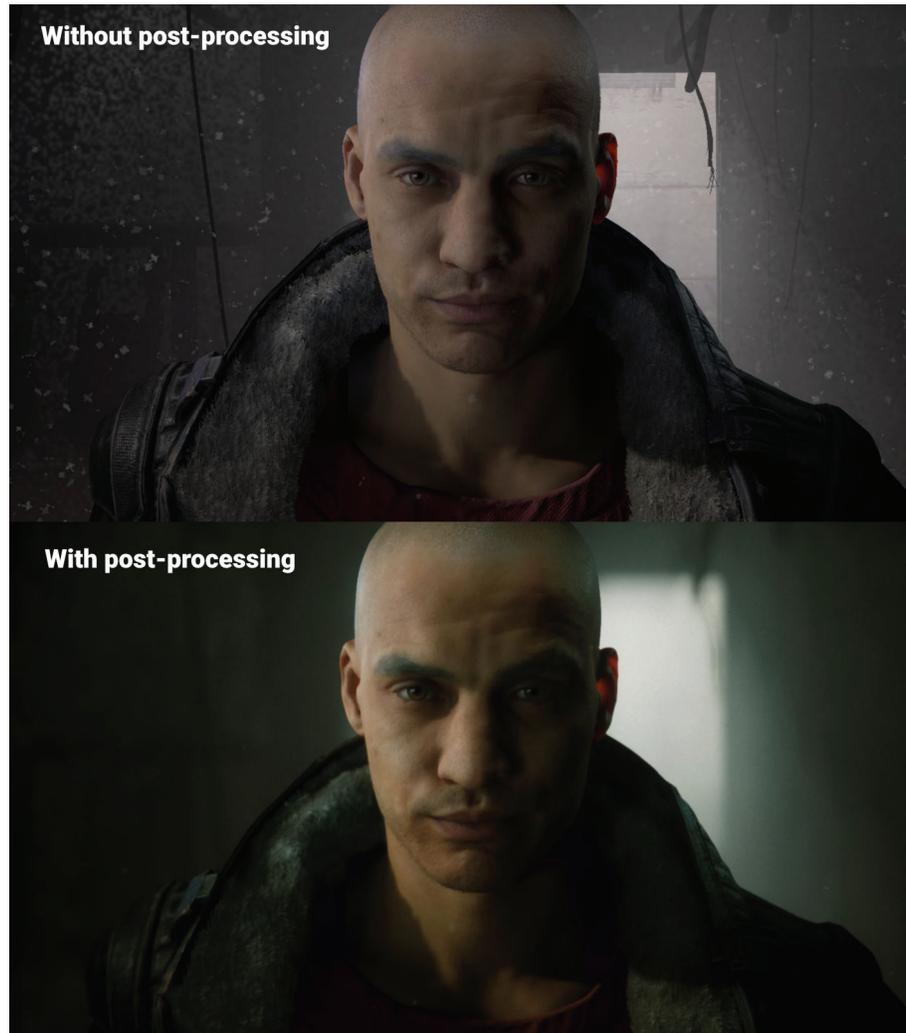
「**Refraction Model**」と「**Index of Refraction**」を選択します。中空でないオブジェクトの場合、「**Refraction Model**」には「**Sphere**」を使用します。中空のオブジェクトについては「**Thin**」（泡など）または「**Box**」（わずかに厚みがあるもの）を選択します。



Screen Space Refraction

# ポストプロセス

最新のハイエンドグラフィックスにおいては、ポストプロセスを行わずして完成することはないとも言えます。必ずしもプロセッシングで問題を修正できるわけではありませんが、シネマティックな効果を高めるフィルターやフルスクリーンのイメージエフェクトを使用せずに画像をレンダリングすることは、ほとんど考えられません。そのため、HDRP には組み込みのポストプロセスエフェクトがバンドルされています。



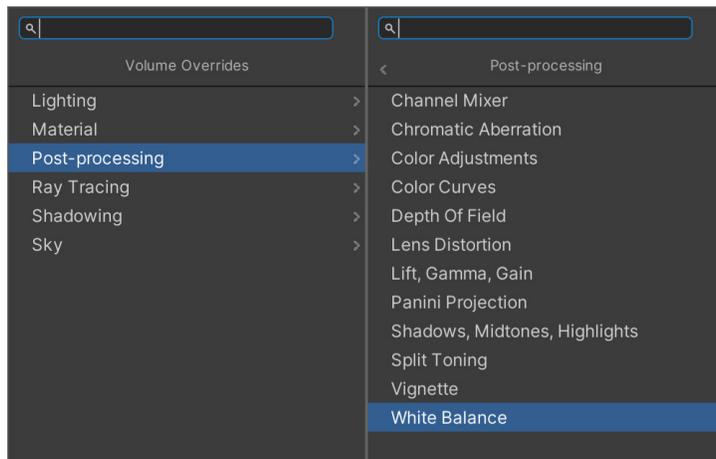
ポストプロセスエフェクトを使用すると、よりシネマティックなレンダリングが可能になります。

HDRP のプロセッシングでは、ボリュームシステムを使用してイメージエフェクトをカメラに適用します。オーバーライドを追加する方法を理解すれば、ポストエフェクトを適用するプロセスも理解しやすくなるはずです。

## 「Post-processing (ポストプロセス)」でのオーバーライド

色やコントラストを制御するこうした「Post-processing」のオーバーライドの多くは、機能が重複しています。適切な組み合わせを見つけるためには、ある程度の試行錯誤が必要です。

すべてのエフェクトを利用する必要はありません。目標とするビジュアルを生み出すために必要なオーバーライドのみを追加し、その他は無視します。



「Post-processing (ポストプロセス)」のオーバーライド

使用例については、SampleScene のボリュームをご覧ください。

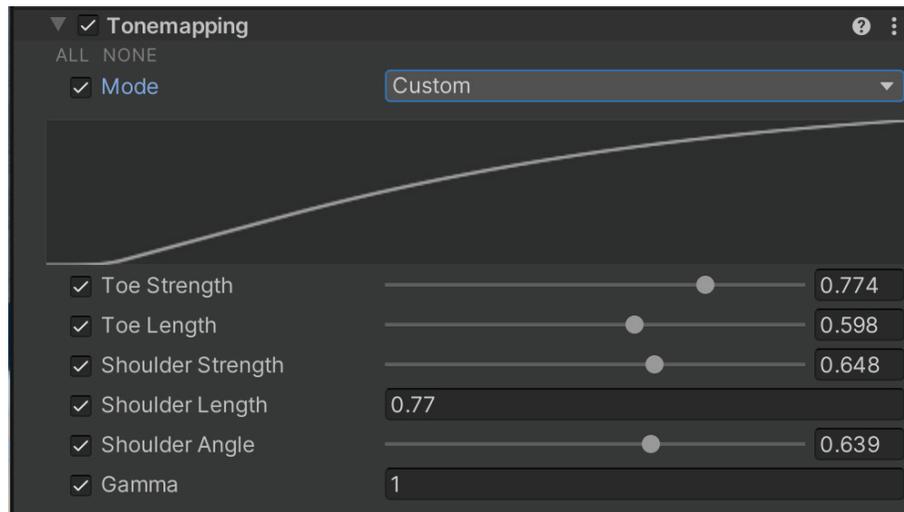
## トーンマッピング

トーンマッピングとは、**ハイダイナミックレンジ**の色を、使用する画面の狭い**ダイナミックレンジ**にマッピングする手法です。レンダリングでのコントラストやディテールを改善できます。



「ACES」と「Neutral」のトーンマッピングの比較

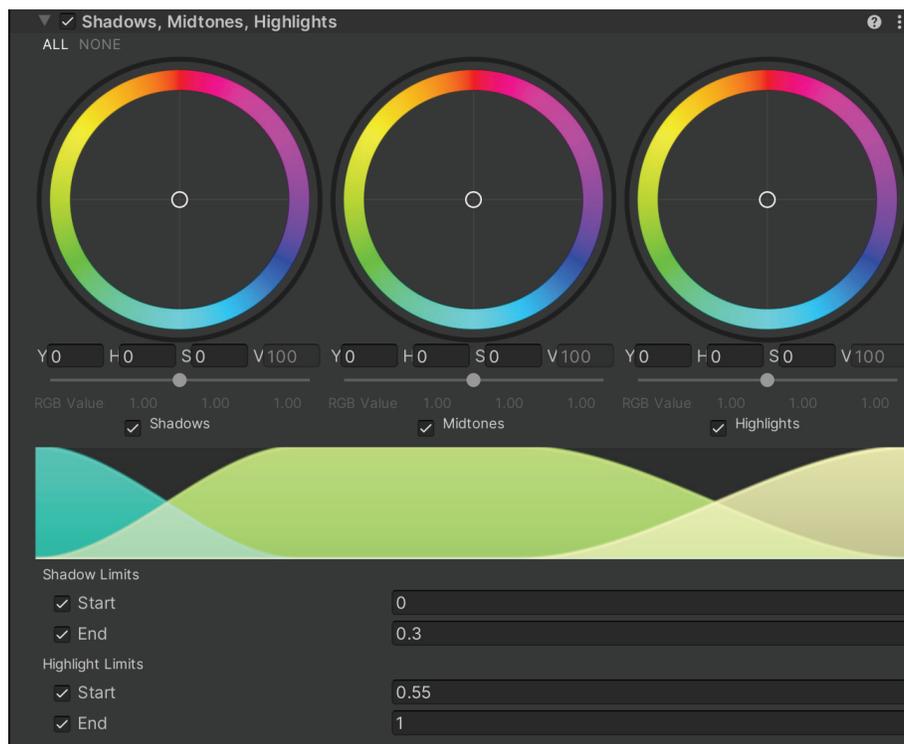
ビジュアルをフィルム調にしたい場合は、「Mode」を業界標準の「ACES」(Academy Color Encoding System) に設定します。彩度やコントラストを落とす場合は、「Neutral」を選択します。経験豊富なユーザーであれば、「Custom」を選択し、トーンマッピング曲線を自分で定義することもできます。



カスタム曲線を使用したトーンマッピング

## Shadows、Midtones、Highlights

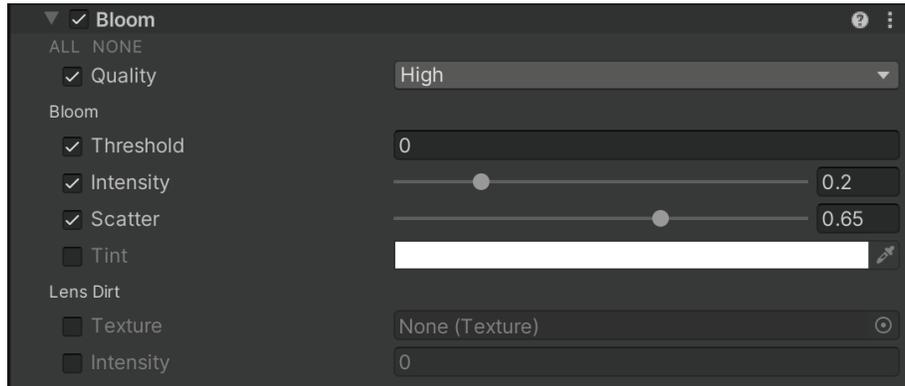
「Shadows、Midtones、Highlights」のオーバーライドでは、レンダリング対象のシャドウ、中間調、ハイライトそれぞれの色調と色の範囲を制御できます。それぞれのトラックボールを使用して、画像内のそれぞれの特性を調整できます。その後、色補正をクリップまたはプッシュしすぎないように、「Shadow」と「Highlight Limits」を使用します。



Shadows, Midtones, Highlights

## Bloom

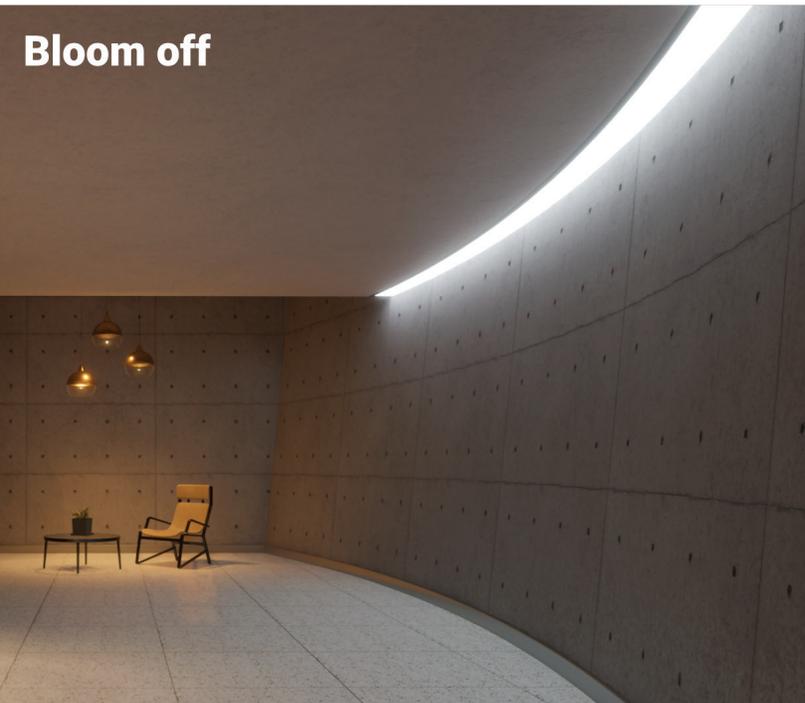
「Bloom」では、光源の周囲に発生する光のにじみのエフェクトを生成できます。これにより、強烈に輝く、まばゆい光の印象を生み出すことができます。



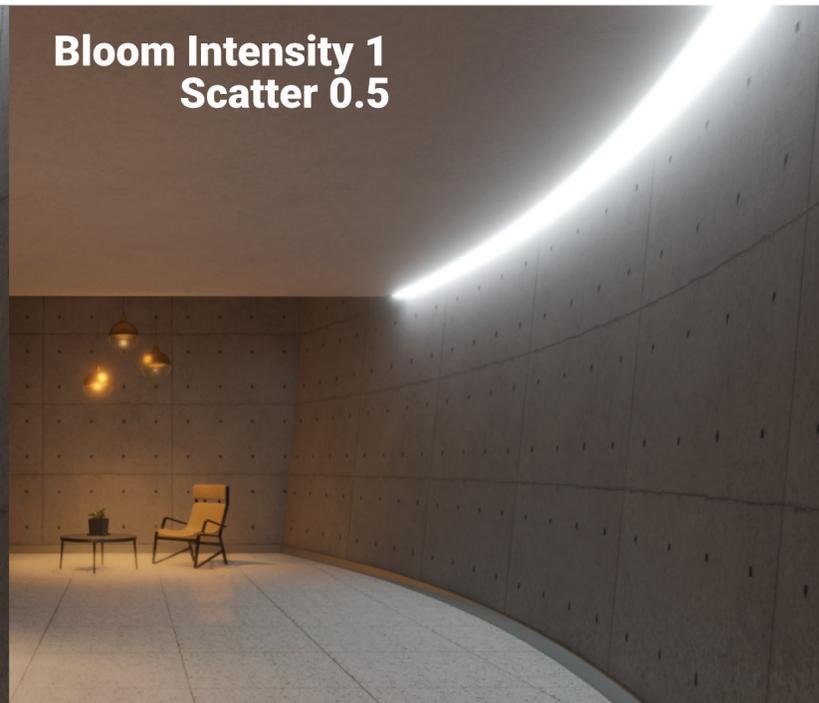
「Bloom」のオーバーライド

「Intensity」と「Scatter」を設定することで、ブルームのサイズと明度を調整します。「Lens Dirt」は、ブルームエフェクトを散乱させる汚れやちりのテクスチャを適用します。明度の低いピクセルでシャープネスを維持するには、「Threshold」を使用します。

Bloom off



Bloom Intensity 1  
Scatter 0.5



ブルームのエフェクト

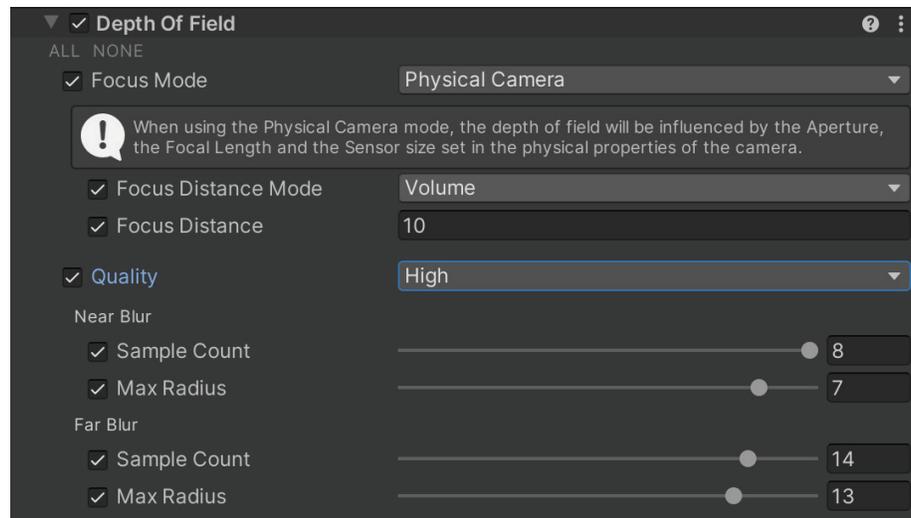
## Depth of Field

「Depth of Field」は、実際のカメラレンズの集束特性のシミュレーションを行います。カメラの焦点距離より近い、または遠いオブジェクトは、ぼやけて見えます。

焦点距離は、以下の場所から設定できます。

- 「**Manual Ranges**」フォーカスモードを使用した Volume オーバーライド：焦点距離は、Volume 自体によって制御されます。例えば、水中のシーンなど、場所によって意図的にカメラをぼやけさせるといった使い方ができます。
- 「[Volume Settings Extension](#)」での Cinemachine カメラの使用：ターゲットを追従してオートフォーカスすることができます。
- 「**Physical Camera**」フォーカスモードを使用した「**Physical Camera**」プロパティ:Camera コンポーネントの「**Focus Distance**」パラメーターをアニメーション化できます。

「Depth of Field」が有効な場合は、**ボケ**と呼ばれるフォーカス外のぼかしエフェクトが、画像の明るいエリアを中心に現れます。カメラの開口形状を調整し、ボケの見え方を変えることができます（前述の「[Physical Camera](#)」の[その他のパラメーター](#)を参照）。

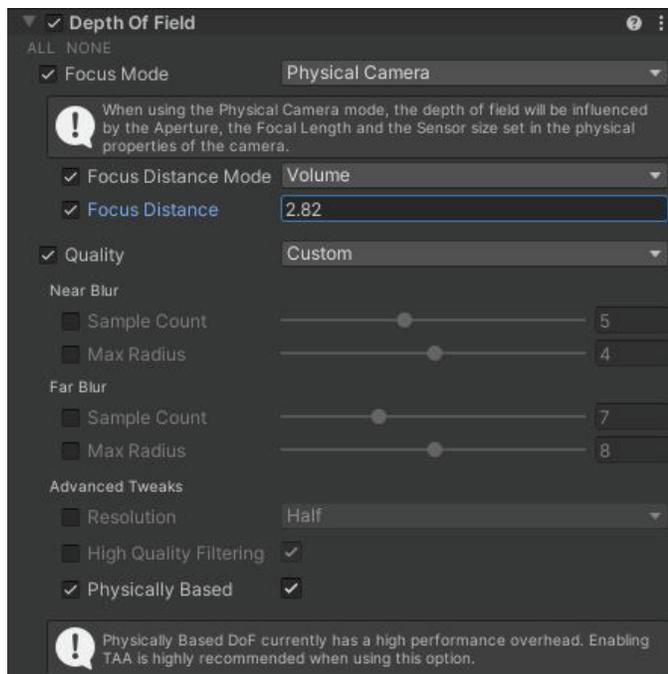


「Depth of Field」のオーバーライド



「Depth of Field」では、実際のカメラの被写界深度のシミュレーションを行います。

シネマティクスやオフラインレンダリングでは、追加の設定とカスタム品質を有効にすることで、負荷は増しますが「**Physically Based**」の被写界深度を試すことができます。

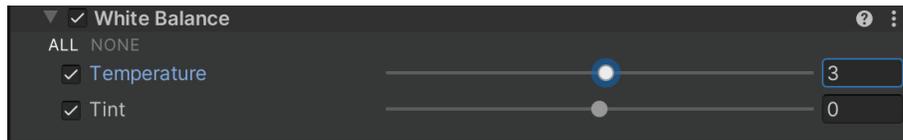


「Depth of Field」では、実際のカメラの被写界深度のシミュレーションを行います。

## White Balance

「White Balance」のオーバーライドを使用すると、最終的な画像で白色が適切にレンダリングされるようにシーンの色を調整できます。「**Temperature**」で、黄色（暖色）と青（寒色）の間を調整することもできます。「**Tint**」では、緑とマゼンタの間で色かぶりを調整します。

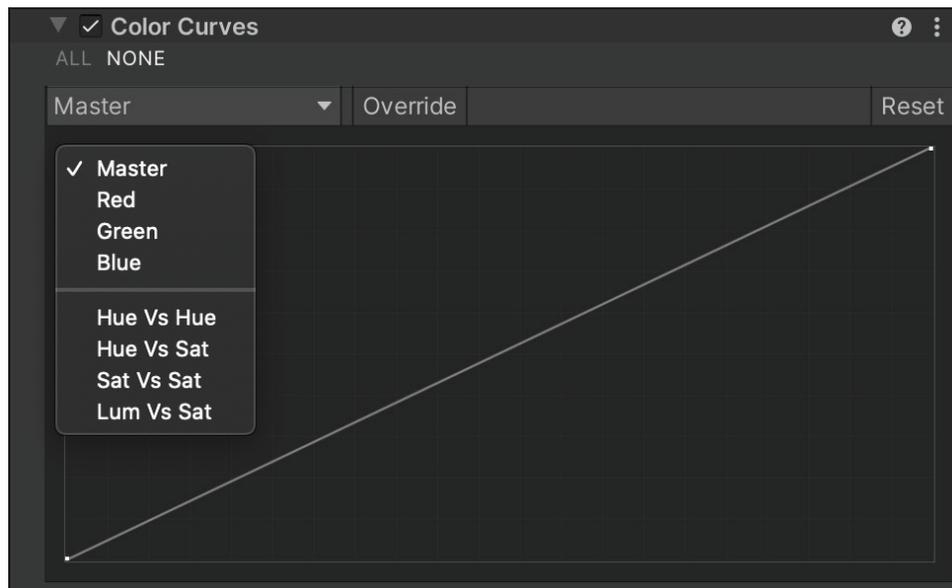
この HDRP サンプルプロジェクトでは、各部屋のローカルボリュームに「White Balance」のオーバーライドが含まれています。



White Balance

## Color Curves

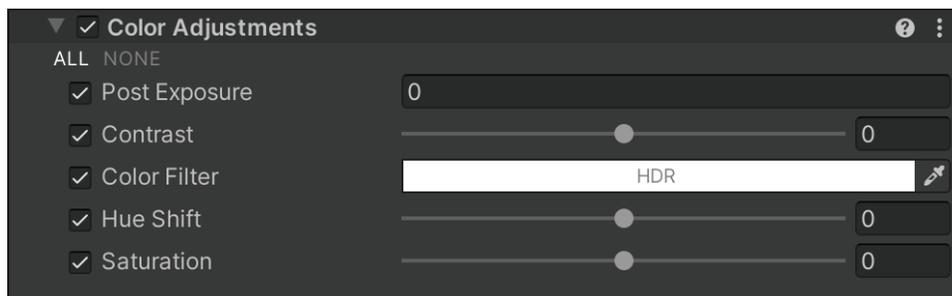
グレーディングカーブを使用して、色相、彩度、明るさの特定の範囲を調整できます。8つのグラフのいずれかを選択し、色とコントラストをリマップします。



Color Curves

## Color Adjustments

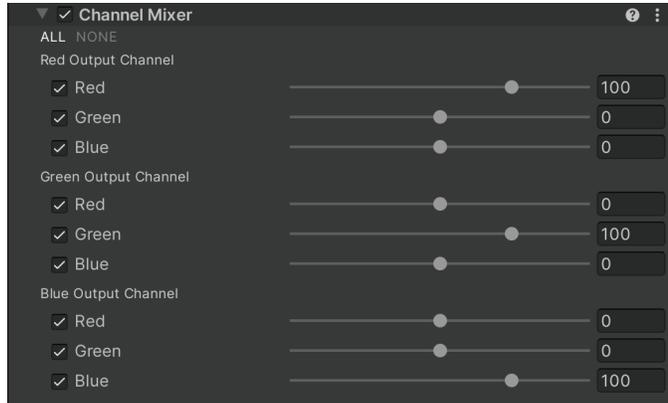
このエフェクトを使用して、最終的にレンダリングされる画像の全体的なトーン、明度、色相、コントラストを調整します。



Color Adjustments

## Channel Mixer

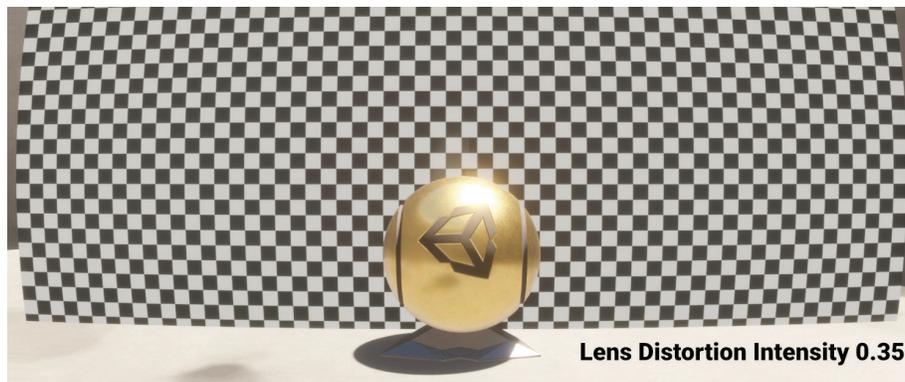
「Channel Mixer」では、カラーチャンネルの「組み合わせ」に、さらに別のカラーチャンネルを適用できます。RGB 出力を選択してから、いずれかの入力の影響度を調整します。たとえば、「Red Output Channel」で「Green」の影響度を上げると、画像内のすべての緑色の部分が赤みがかって表示されます。



Channel Mixer

## Lens Distortion

「Lens Distortion」では、現実世界のレンズの製造過程で発生する歪みによって生じる放射パターンのシミュレーションを行います。これを設定すると、特にズームレンズや広角レンズを使用したときに、直線がわずかに曲がって見えます。



「Lens Distortion」を使用すると、放射パターンの歪みを画像に適用します。

## Vignette

「Vignette」では現実世界の写真に生じる効果を再現し、画像の端部の明度や彩度を下げます。この現象は、広角レンズを使用した場合や、レンズフードや重ね付けしたフィルターリングによって光が遮られた場合に発生することがあります。見る人の視線を画面中央に集めるために、このエフェクトが使用されることもあります。



「Vignette」を使用すると、フレームの端部が暗くなります。

## Motion Blur

現実世界では、カメラの露出時間よりも速く移動する物体を撮影すると、線状になったりぼやけたりした状態で写ります。「Motion Blur」のオーバーライドで、このような効果のシミュレーションを行うことができます。

パフォーマンスコストを最小限に抑えるためには、「Sample Count」を減らし、「Minimum Velocity」を上げ、「Maximum Velocity」を下げます。「Additional Properties」で「Camera Clamp Mode」のパラメーターを下げることもできます。



実際の写真からのレンズフレア。レンズフレアの物理的な特徴については、[Wikipediaの記事](#)に詳しく説明されています。

# レンズフレア

レンズフレアとは、カメラのレンズに明るい光が当たったときに発生するアーティファクトです。レンズフレアは、1つの明るいグレアとして現れる場合と、カメラの開口形状に沿った多数の色付き多角形のフレアとして現れる場合があります。現実の世界ではフレアは好ましくない効果をもたらしますが、物語や芸術の目的では有益な働きをする場合もあります。例えば、強いレンズフレアを使用することで、プレイヤーの注意を引いたり、設定やシーンの雰囲気を変えたりすることができます。レンズフレアは、レンダリングプロセスの後段で描画されるため、基本的にはポストプロセスエフェクトです。

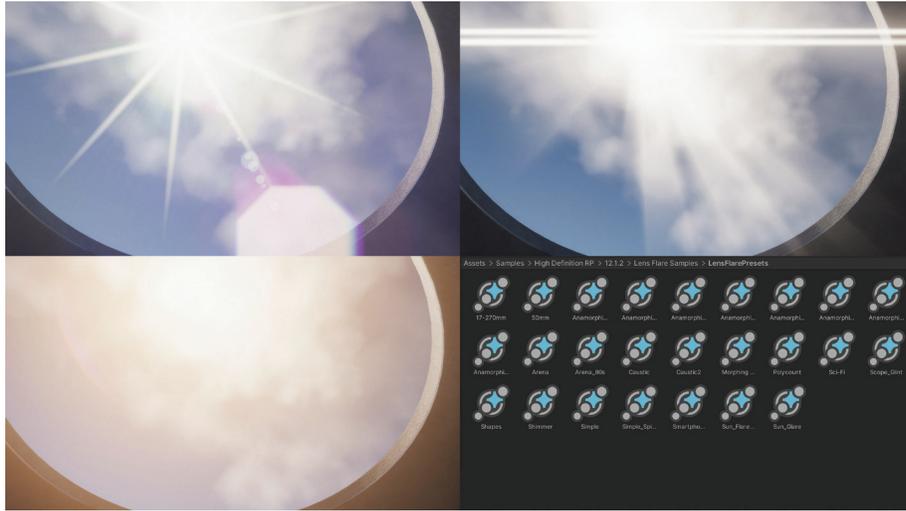
フレアの外観はフレアアセットで指定されますが、そのエフェクトをレンダリングするためには、シーンビューで **Lens Flare (SRP)** コンポーネントをオブジェクトに追加する必要があります。例えば、フレアを発生させる物体や光源などです。エフェクトの一般的なパラメーターである強度、スケール、オクルージョンは、このコンポーネントによって制御されます。また、フレアがカメラビューの外にあるときに画面外で実行することもできますが、その場合でも、フレアが静的な光源から来るときなど、一部のフレアエフェクトはシーンに投影されます。

レンズフレアを使いこなすには、Package Manager からサンプルをインストールするのが一番です。そうすることで、事前に定義された一連のフレアアセットが追加され、レンズフレアエフェクトを含むように HDRP サンプルシーンが変更されます。また、レンズフレアを閲覧できるテストシーンも収録されているので、独自に作成する際の参考になります。

▼ Features	
● Engineering 7 packages	
▼ Packages - Unity	
▶ Cinemachine 2.8.2	
▶ High Definition RP 12.1.2	<b>High Definition RP</b> <span>Release</span>
▶ Input System 1.2.0	Unity Technologies
▶ JetBrains Rider Editor 3.0.7	Version 12.1.2 - December 11, 2021
▶ Test Framework 1.1.29	<a href="#">View documentation</a> · <a href="#">View changelog</a> · <a href="#">View licenses</a>
▶ TextMeshPro 3.0.6	The High Definition Render Pipeline (HDRP) is a high-fidelity Scriptable Render Pipeline built by Unity to target modern (Compute Shader compatible) platforms. HDRP utilizes Physically-Based Lighting techniques, linear lighting, HDR lighting, and a configurable hybrid Tile/Cluster deferred/Forward lighting architecture and gives you the tools you need to create games, technical demos, animations, and more to a high graphical standard.
▶ Timeline 1.6.3	▼ Samples
▶ Tutorial Framework 2.1.1	Additional Post-processing Data 35.34 MB <span>Import</span>
▶ Unity UI 1.0.0	Procedural Sky 22.39 KB <span>Import</span>
▶ Version Control 1.15.4	Particle System Shader Samples 2.26 MB <span>Import</span>
▶ Visual Scripting 1.7.6	Material Samples 69.32 MB <span>Import</span>
▶ Visual Studio Code Editor 1.2.4	✓ Lens Flare Samples 25.99 MB <span>Reimport</span>
▶ Visual Studio Editor 2.0.12	Local Volumetric 3D Texture Samples 418.38 KB <span>Import</span>

レンズフレアのサンプルは、Package Manager の「High Definition RP」にあります。

プロジェクトの「**Directional Light Sun**」を選択すると、ライトにアタッチされた Lens Flare (SRP) コンポーネントと、Lens Flare データアセットを確認できます。アセットを変更すると、さまざまなフレアエフェクトを観察することができます。



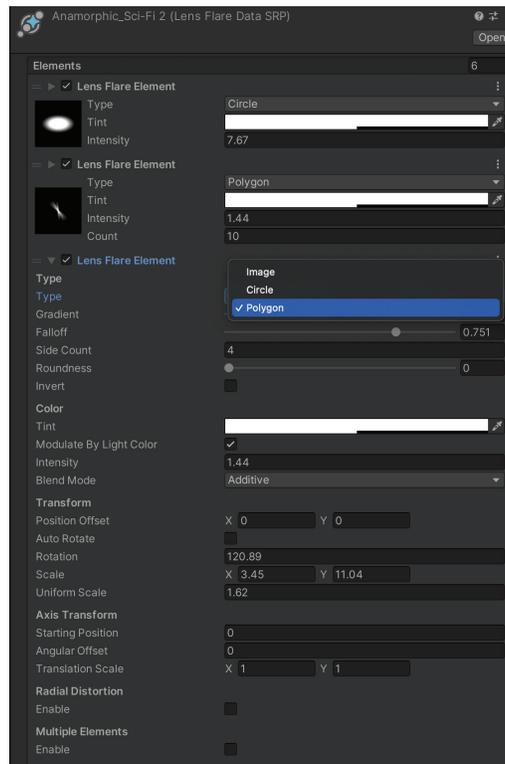
レンズフレアのサンプルには、リアルなレンズエフェクトや架空のレンズエフェクトなどのさまざまなプリセットが用意されています。

レンズフレアはレンズフレア要素で作られており、各要素はフレアが生み出すさまざまなアーティファクトを表しています。要素の形状には、ポリゴン、円、カスタム画像などがあります。

要素のパラメーターで、色、トランスフォームの位置、デフォーメーションのほか、複数の要素を含んだレンズフレアエフェクトの一部である場合のフレア要素の配置、色、スケールを微調整することが可能です。フレア要素を光源にアタッチすれば色合いが使用できるので、同じフレアアセットをさまざまな光源で再利用することができます。

レンズフレアのしくみについて詳しくは、[こちらのプレゼンテーション](#)をご覧ください。

フレアアセットとその要素および使用可能なパラメーター。カスタムフレアエフェクトの独自のライブラリを作成して、ゲームに適した雰囲気を作り出すこともできます。



# 動的解像度

CPU がボトルネックにならない場合、性能に大きく影響する可能性があるのは解像度です。動的解像度は、レンダリング解像度を下げ、出力画面の解像度に合わせて結果をスケールするものです。このアップスケールを行うためのフィルターが 2021 LTS で大幅に改良され、70% 以下の解像度でレンダリングしても、ほとんどビジュアルを損なわなくなりました。

Unity 2021 LTS の HDRP には、最新のスーパーサンプリング技術のいくつかを使用した複数の選択肢が用意されています。

## NVIDIA DLSS (NVIDIA RTX GPU および Windows 向け)

Unity 2021 LTS の HDRP では、NVIDIA DLSS (ディープラーニングスーパーサンプリング) がネイティブにサポートされます。NVIDIA DLSS は高度な AI レンダリングを使用しており、ほんのわずかなピクセルを従来のようにレンダリングするだけで、ネイティブ解像度に匹敵する画質を実現できます。

リアルタイムレイトレーシングと NVIDIA DLSS により、より高いフレームレートと解像度で動作する美しい世界を NVIDIA RTX GPU 上で作成することができます。また、DLSS は従来のラスタライズされたグラフィックスに対しても、大幅な性能向上を実現しています。詳細については、[NVIDIA の Unity 開発者ページ](#)、[ブログ記事](#)、[ドキュメント](#)を参照してください。



24 Entertainment の『Naraka Bladepoint』などのゲームを 4K で動作させる DLSS 技術の内部的なしくみについて詳しくは、[こちらのブログ](#)をご覧ください。

## AMD FSR (クロスプラットフォーム)

Unity 2021 LTS では、AMD FidelityFX™ Super Resolution が利用できるようになりました。HDRP および URP 用の FSR がビルトインでサポートされます。HDRP アセットとカメラで動的解像度を有効にし、アップスケールフィルターオプションから「**FidelityFX Super Resolution 1.0**」を選択すると FSR を使用できます。

AMD FidelityFX Super Resolution (FSR) は、低解像度の入力から高解像度のフレームを生成するためのオープンソースの高品質ソリューションです。特に高品質のエッジを作成することに重点を置いた最先端のアルゴリズム群が使用されており、ネイティブ解像度で直接レンダリングする場合と比較して大幅な性能向上が得られます。FSR は、ハードウェアレイトレーシングのようなコストのかかるレンダリング処理に対して「実用的なパフォーマンス」を実現します。詳細については、AMD の Unity [ウェブページ](#) および [フォーラム](#) を参照してください。



[Steam](#) で配信中の Spaceship デモで FSR のテクノロジーをお試しください。

### TAA Upscale (クロスプラットフォーム)

Temporal Anti-Aliasing (TAA) Upscale は、時間積分を用いてシャープな画像を生成する手法です。Unity では、通常のアンチエイリアシングと並行してこの手法が使用されます。HDRP は、このアップスケールフィルターをポストプロセスの前に、TAA ステップと同時に実行します。つまり、使用できるのは、TAA アンチエイリアシング手法のみということです。このフィルターは、他のアンチエイリアシング手法とは互換性がありません。Temporal Anti-Aliasing (TAA) Upscale はフレームごとにアンチエイリアシングを実行します。つまり、動的解像度を有効にすると、スクリーンパーセンテージの解像度の比率が 100% であってもアンチエイリアシングが実行されます。詳細については、[TAA Upscale についての注意事項](#)に関するドキュメントセクションを参照してください。



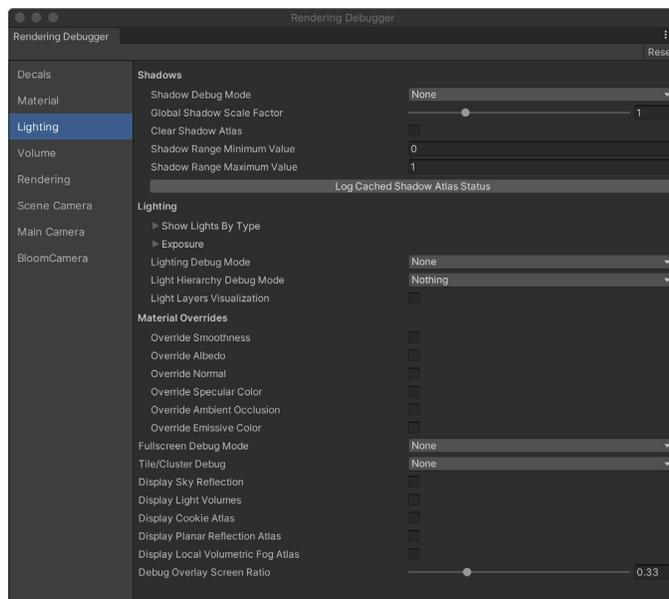
Temporal Anti-Aliasing Upscale 法を用いた左の画像 (A) は、シャープで輪郭がはっきりしています。右の画像 (B) は、Catmull-Rom アップスケール法を用いたもので、あまり鮮明に見えません。

HDRP Asset で強制的にスケールするか、ロジックをコーディングして動的にスケールを調整することができます。

プロジェクトで動的解像度を設定し、ニーズに合った最適なアルゴリズムを選択する方法についてのガイダンスを得るには、ドキュメントの[こちらのページを参照](#)してください。

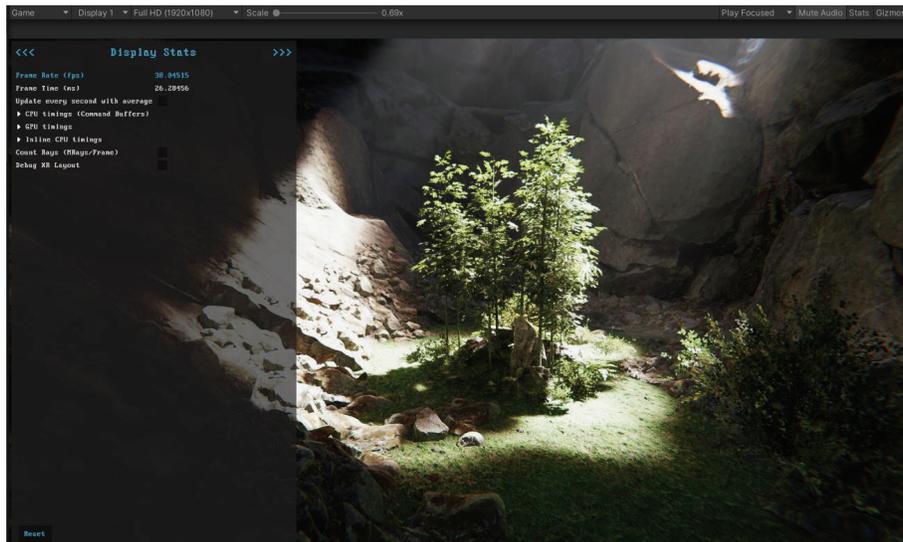
# Rendering Debugger

「Rendering Debugger」ウィンドウ（「**Window > Analysis > Rendering Debugger**」）には、スクリプタブルレンダーパイプライン用のデバッグツールと可視化ツールがあります。左側はカテゴリ別に分類されています。各パネルでは、ライティング、マテリアル、ボリューム、カメラなどの問題を分割できます。



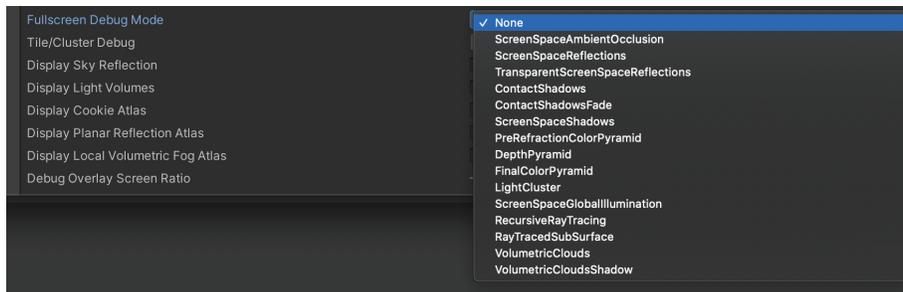
Rendering Debugger

Rendering Debugger は、ゲームビューの再生モードで実行時に利用できるほか、開発ビルドで作成された Player で利用することもできます。**Ctrl + Backspace** キーを使用するか、ゲームコントローラーの 2 本のスティックを押すと、そのメニューを開くことができます。



ゲームビューまたは Player における「Rendering Debugger」ウィンドウ

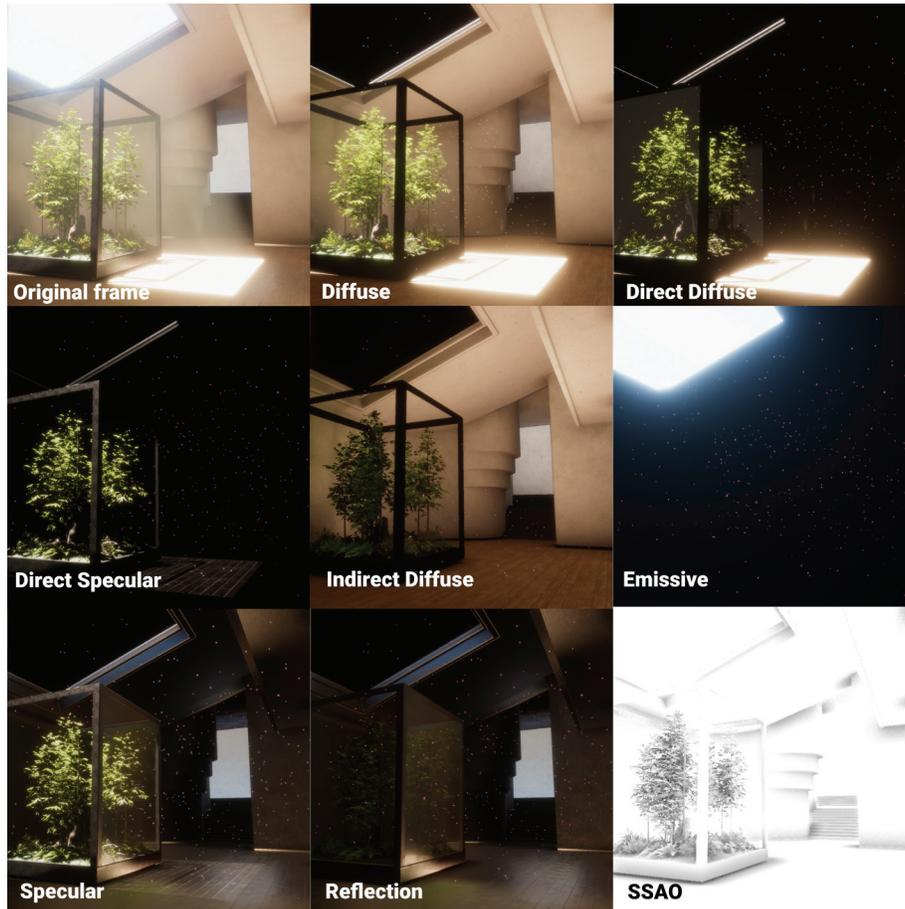
このデバッガーは、特定のレンダリングパスのトラブルシューティングに役立ちます。「Lighting」パネルで「**Fullscreen Debug Mode**」と入力し、デバッグ機能を選択できます。



「Fullscreen Debug Mode」オプション

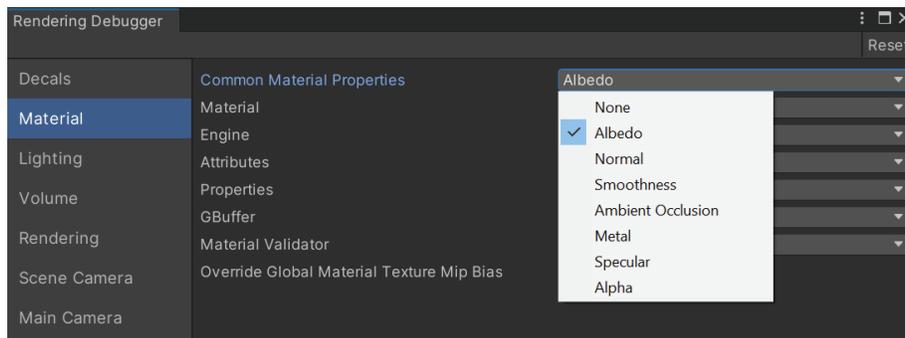
デバッグモードでは、さまざまな情報を手掛かりにして、ライティングやシェーディングの問題の原因を具体的に突き止めることができます。左側のパネルには、カメラ、マテリアル、ボリウムなどの重要な統計情報が表示されており、レンダリングの最適化に役立てることができます。

フルスクリーンのデバッグモードが有効な場合、シーンビューとゲームビューが切り替わり、特定の機能が一時的に可視化されます。この機能を利用して、効率的に診断を進めることができます。

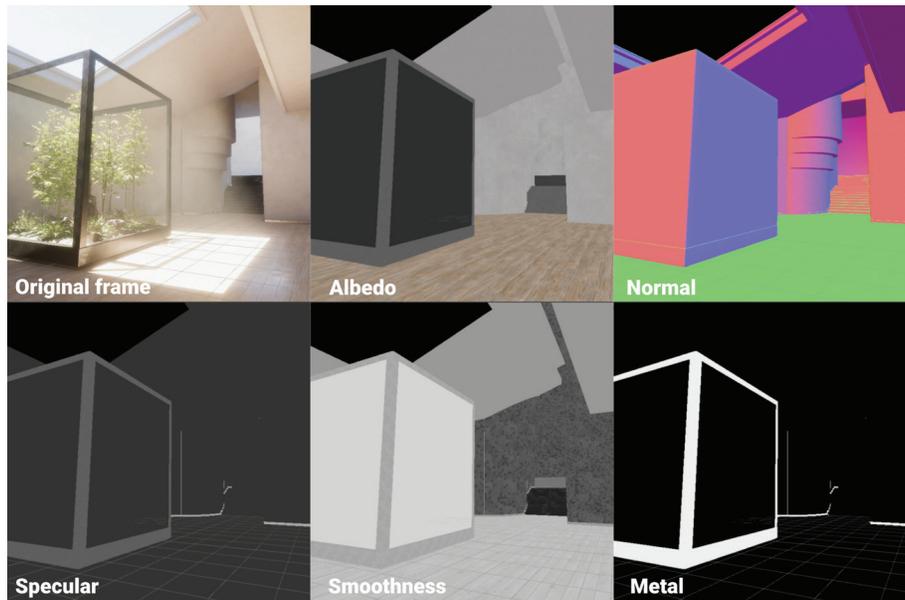


「Lighting Debug Mode」と「Fullscreen Debug Mode」は、シーン内の光源を把握するのに有効です。

一般的なマテリアルプロパティのデバッグを行うこともできます。「Material」スクリーンで、「Common Material Properties」の中から、「Albedo」、「Normal」、「Smoothness」、「Specular」などを選択します。

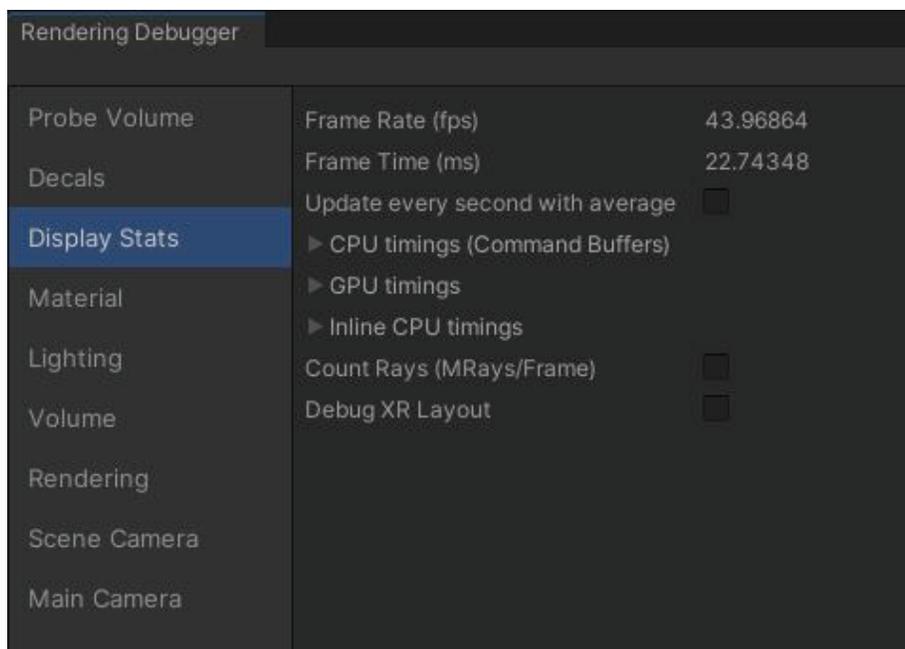


Common Material Properties (マテリアルの基本プロパティ)



レンダラーパイプラインデバッガーを使用すると、マテリアルのトラブルシューティングを行えます。

パフォーマンスをデバッグするには、再生モード時に「Rendering Debugger」ウィンドウの統計パネルにアクセスします。



再生モードの「Rendering Debugger」のパフォーマンス統計

詳細については、[レンダラーパイプラインデバッガーに関するドキュメント](#)をご覧ください。

# レイトレーシング (プレビュー)

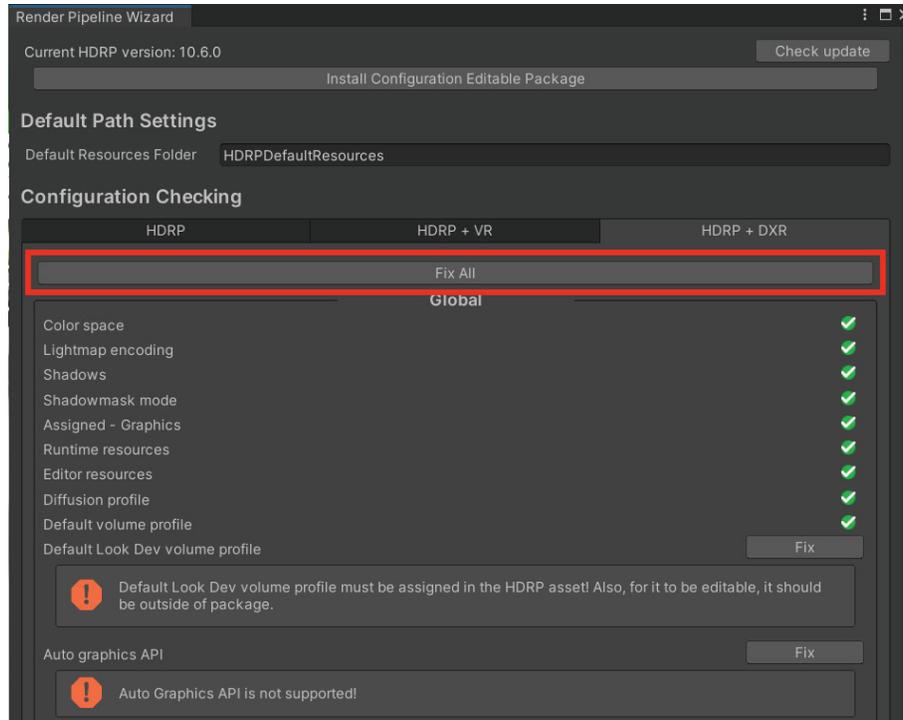
レイトレーシングは、従来のラスライゼーションよりも訴求力のあるレンダリングを生成できる手法です。従来は計算コストが大きい処理でしたが、近年のハードウェアアクセラレーションによるレイ交差（トレーシング）の発達により、レイトレーシングのリアルタイム利用が可能になりました。

HDRP におけるレイトレーシングは、フォールバックとして依然としてラスライズレンダリングに依存するハイブリッドシステムであり、一部の GPU ハードウェアと DirectX 12 API のサブセットによるレイトレーシングのプレビューサポートを含んでいます。システム要件の一覧については、「[Getting started with ray tracing](#)」（英語）を参照してください。

## 設定

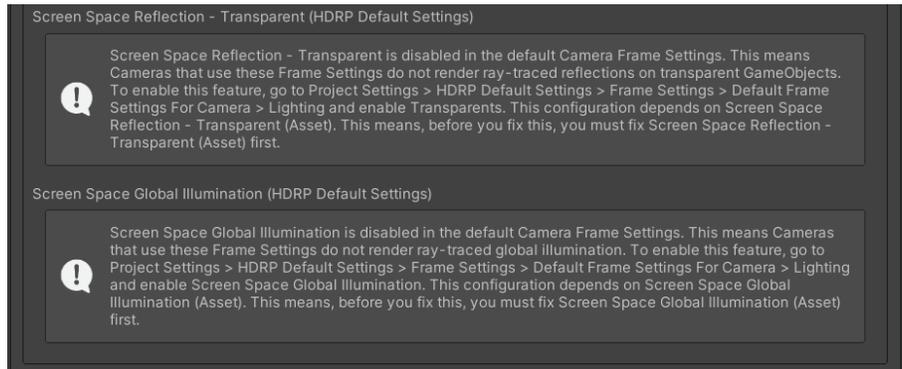
レイトレーシング（プレビュー版）を有効にするためには、HDRP プロジェクトのデフォルトのグラフィックス API を DirectX 12 に変更する必要があります。

「Render Pipeline Wizard」（「**Window > Render Pipeline > HD Render Pipeline Wizard**」）を開きます。「**HDRP + DXR**」タブで「**Fix All**」をクリックすると、エディターを再起動するように求められます。



「Render Pipeline Wizard」でレイトレーシングを有効にします。

<sup>2</sup>Unity 2021 では、「HDRP Wizard」は「Window > Rendering > HDRP Wizard」にあります



指示に従って無効になっている機能を修正します。

レイトレーシングを**手動**で設定することもできます。

プロジェクトでレイトレーシングを有効にしたら、「**HDRP Global**」または「**Camera Frame Settings**」でもレイトレーシングが有効になっていることを確認します。「**Build Settings**」で、互換性のある 64 ビットアーキテクチャーを使用していることを確認します。また、「**Edit > Rendering > Check Scene Content for HDRP Ray Tracing**」から、シーンのオブジェクトを検証します。

## オーバーライド

レイトレーシングの導入により、新しい Volume オーバーライドがいくつか追加され、HDRP の既存のオーバーライドが多数強化されています。

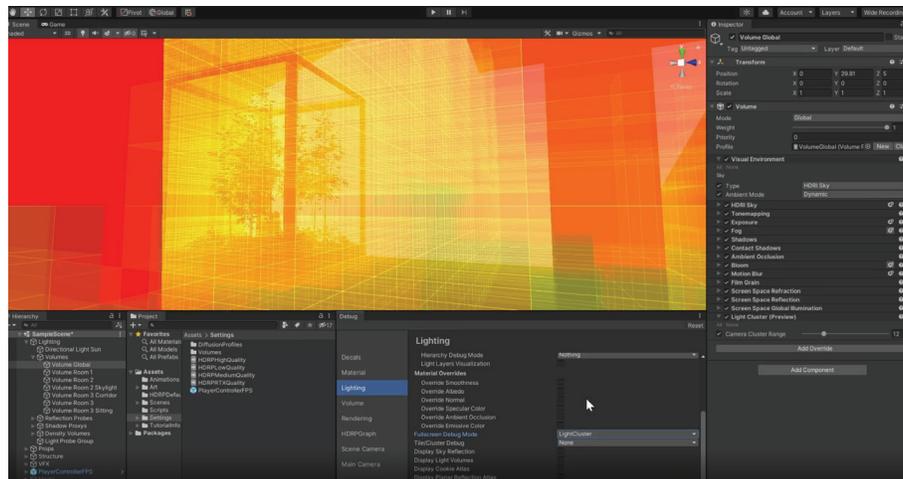
- **レイトレーシングされたアンビエントオクルージョン**：スクリーンスペースアンビエントオクルージョンに代わって、レイトレーシングされたアンビエントオクルージョンが導入されます。スクリーンスペースアンビエントオクルージョンと異なり、レイトレーシングアンビエントオクルージョンでは、画面外のジオメトリを使用してオクルージョンを生成できます。これにより、フレームの端に向かうにつれてエフェクトが消えたり、不正確になったりする問題が解消されます。



スクリーンスペースアンビエントオクルージョンとレイトレーシングされたアンビエントオクルージョン

- **ライトクラスター**：HDRP によって、カメラを中心とした軸平行のグリッドにシーンが分割されます。HDRP は、この構造体を使用して、光線がサーフェスに当たったときのライティングに寄与する可能性がある一連の局所的ライトを判別します。その後、特定のエフェクトの反射光を計算します（レイトレーシングされたリフレクション、レイトレーシングされたグローバルイルミネーションなど）。考慮すべきゲームオブジェクトとライトを包含するようにこの構造体の範囲を変更するには、「**Camera Cluster Range Volume Override**」を使用します。

「HDRP Debug」モードは、「Windows > Analysis > Rendering Debugger > Lighting > Full Screen Debug mode」から利用できます。これによって、赤でハイライトされたライトクラスターセルを可視化し、ライト数が HDRP アセットの「Maximum Lights per Cell」に達した場所を示すことができます。この設定を調整することで、不要なライトの漏れやアーティファクトを減らすことができます。



デバッグモードでのレイトレーシングライトクラスター

- **レイトレーシングされたグローバルイルミネーション**：スクリーンスペースイルミネーションとライトプローブに代わる機能で、間接的な反射光のシミュレーションを行います。レイトレーシングされたグローバルイルミネーションの計算はリアルタイムで実行されます。オフラインでのライトマップのベイキングプロセスに長い時間をかけなくても、同等の結果を実現できます。

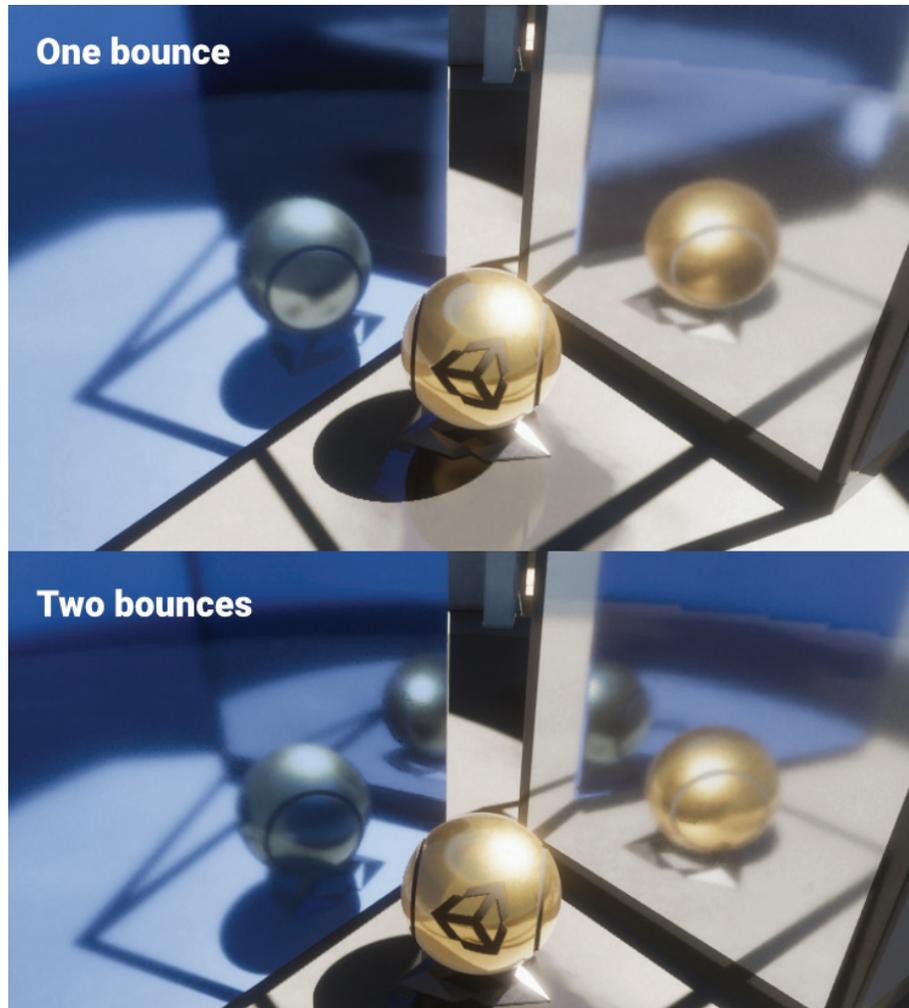
複数の反射やサンプルからメリットを得られる複雑な屋内環境には、「Quality」設定を使用します。「Performance」モード（1つのサンプルと1つの反射に限定）は、ライティングのほとんどがメインのディレクショナルライトによって行われる屋外の状況に適しています。



レイトレーシングされたグローバルイルミネーションによって、反射光がリアルタイムで表示されます。

- **レイトレーシングされたリフレクション**：レイトレーシングされたリフレクションでは、リフレクションプロップやスクリーンスペースリフレクションよりも質の高いリフレクションを再現できます。画面外のメッシュは、生成されたリフレクションで適切に表示されます。

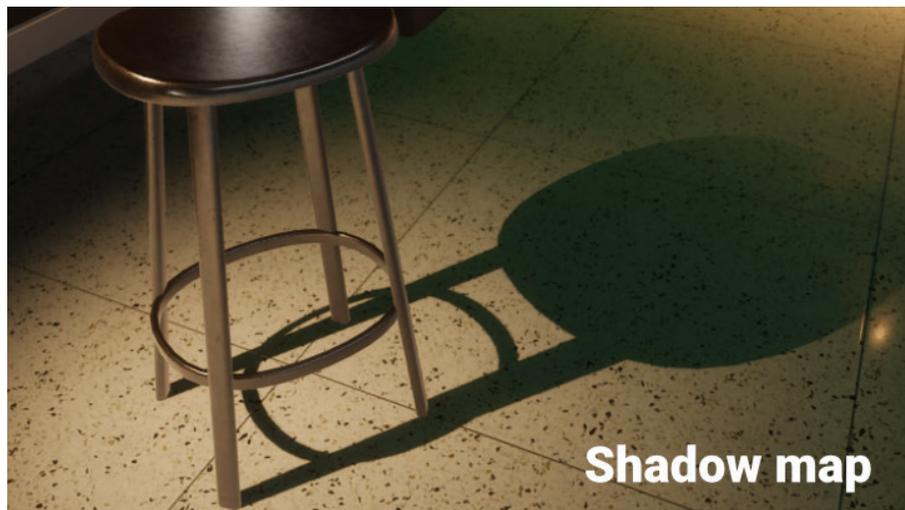
「**Minimum Smoothness**」と「**Smoothness Fade Start**」の値を調整すると、滑らかなサーフェスがレイトレーシングリフレクションを受け始めるしきい値を変更できます。必要に応じて「**Bounces**」を増やすこともできますが（合わせ鏡のような場合）、パフォーマンスコストに注意してください。



レイトレーシングされたリフレクションが、滑らかな鏡のような面にレンダリングされています。

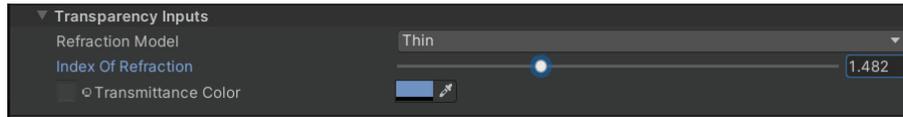
**レイトレーシングされたシャドウ**：ディレクショナル、ポイント、スポット、矩形エリアの各ライトのレイトレーシングシャドウを、不透明なゲームオブジェクトのシャドウマップの代わりに使用することができます。ディレクショナルライトを使用して、透明または半透明のゲームオブジェクトからレイトレーシングされたシャドウを投影することもできます。

レイトレーシングでは、キャストとの距離が離れるほど柔らかくなる、現実のような自然な影を表現することができます。

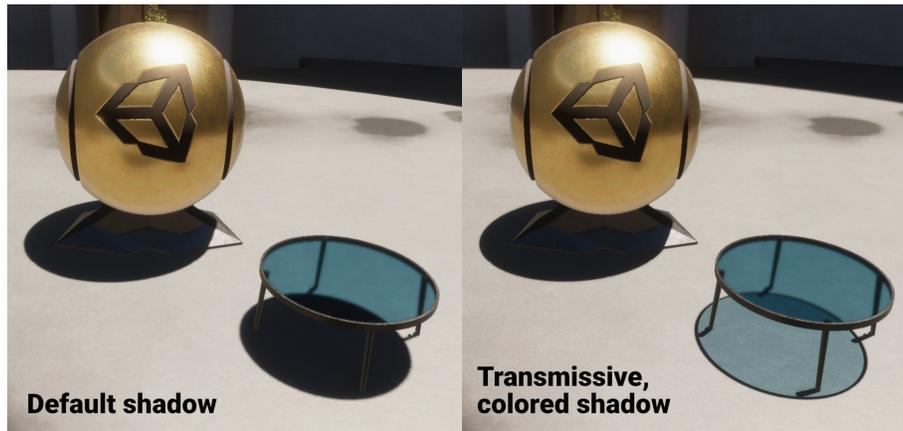


レイトレーシングされたシャドウはキャストから離れるにつれてぼやけるため、シャドウマッピングとは異なる効果が得られます。ディレクショナルライト、ポイントライト、スポットライトで、半透明のシャドウを生成することもできます。

HDRP のディレクショナルライトで、半透明の色付きシャドウを生成することもできます。この例では、ガラスのサーフェスによって、リアルな色合いのシャドウが床にキャストされています。



透明のシャドウキャストには「Transmittance Color」を使用します。



HD レンダーパイプラインのレイトレーシング機能（プレビュー）のウォークスルーについては、[HDRP でのレイトレーシングのアクティベートに関する動画](#)をご覧ください。詳しくは、HDRP のマイクロサイトで[レイトレーシングに関するドキュメント](#)をご覧ください。

### その他のリソース

- HDRP の機能に関する概要動画については、このガイドと合わせて、[HDRP で高品質なグラフィックスを作成する方法](#)をご覧ください。ことをおすすめします。
- ビルトインレンダーパイプラインから移行する場合は、[こちらのチャート](#)をご覧ください。両者の機能の違いが詳細にまとめられています。
- [HDRP に関するドキュメント](#)には、このパイプラインのあらゆる機能の詳細が含まれています。
- パフォーマンスを向上させるための HDRP の設定については、[こちらのブログ](#)で詳しく説明しています。

# さらに先へ：

このガイドをきっかけに、皆様が次のプロジェクトで HDRP に取り組んでいただければ幸いです。

さらに詳しく知りたい方は、「[3D Sample Project](#)」を Unity Hub から入手できるのでご利用ください。このガイドに関して質問やご意見がありましたら、この[フォーラムのスレッド](#)でお知らせください。

以下のその他のリソースもぜひご確認ください。また、[Unity ブログ](#)や [HDRP コミュニティフォーラム](#)では、いつでもヒントを得ることができます。

Unity は、リアルタイムコンテンツを作成するための最高のツールを用意して、アーティストや開発者の皆様を支援したいと考えています。ライティングは芸術であり、科学です。芸術と科学を融合させることで、魔法のような効果が生まれます。



[unity.com](https://unity.com)