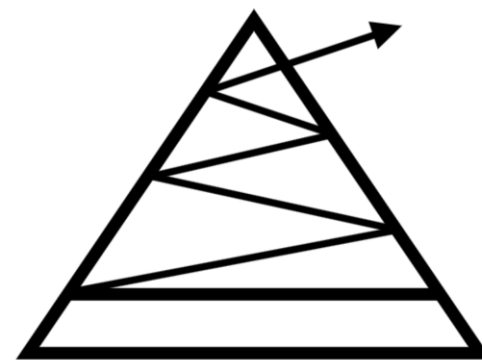


Takenoko Shader マニュアル

β1.0.0



目次

1. Takenoko Shaderについて
2. Render Mode
3. Main Parameter
4. Emission
5. Thin-Film
6. Lightmap
7. Debug
8. ライセンス、連絡先



Takenoko Shader
PBR Standard Shader

Takenoko Shaderについて

Takenoko Shaderについて

- 特殊なマテリアルの再現を目的とする PBR Standard Shaderです
- Unity Standard Shaderの強化に加えて、薄膜干渉を物理的に再現できます
- 現在のバージョンは61.0.0です

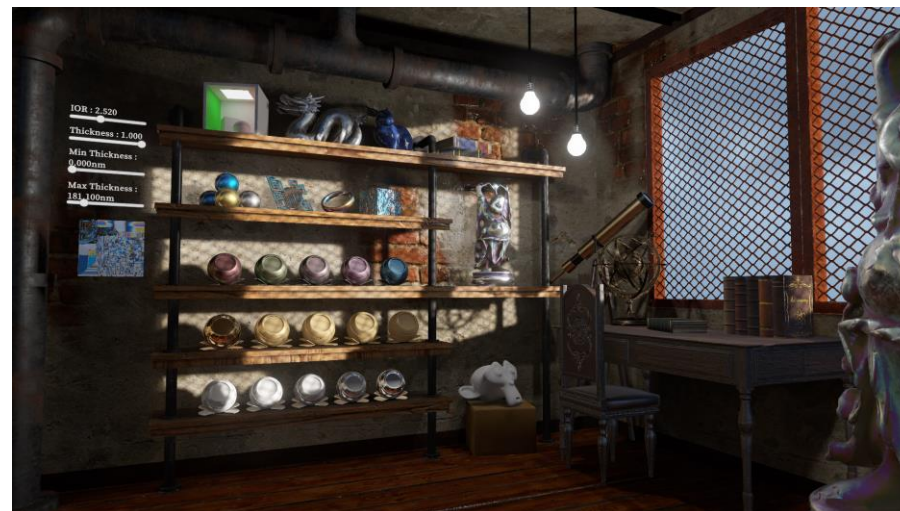


Takenoko Shaderについて

- ・ サンプルワールドを公開しています

<https://vrchat.com/home/launch?worldId=wrld043282b4-71af-4773-9d9b-c24720142eb8>

- ・ Thin-Filmについて体験することが出来ます

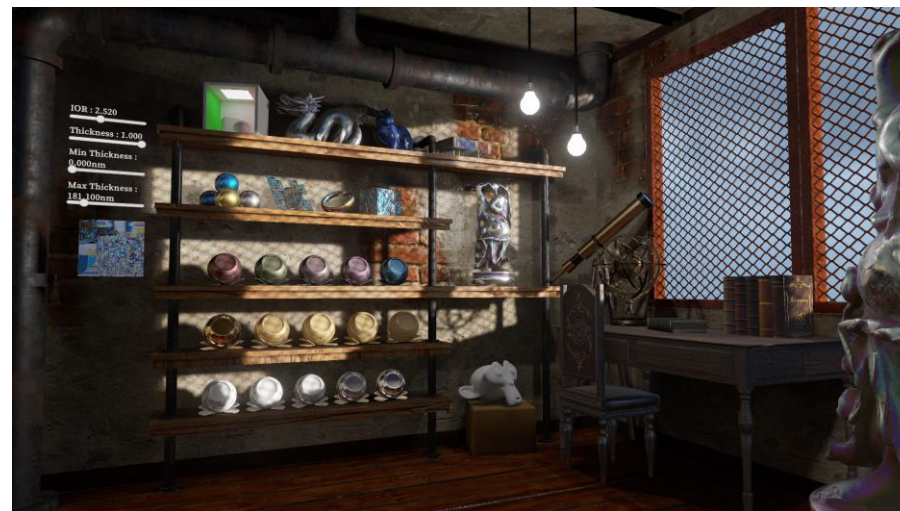


Takenoko Shaderについて

- ・ サンプルワールドを公開しています

<https://vrchat.com/home/launch?worldId=wrld043282b4-71af-4773-9d9b-c24720142eb8>

- ・ Thin-Filmについて体験することが出来ます



Features

- Metallic-Roughness ワークフロー
- 薄膜干渉(Thin-Film)
- SH, Mono SH Lightmap対応(Bakery必須)
- 複数Lightmap切り替え

インポート方法

- Booth または GitHubのReleaseより
Unity Packageをインストール

Booth

<https://kinankocraft.booth.pm/items/5267948>

GitHub

<https://github.com/kinakomoti-321/Takenoko-Shader>

- Boothでは最新版(現在B1.0.0)が配布されています



インポート方法

- Booth または GitHubのReleaseより
Unity Packageをインストール

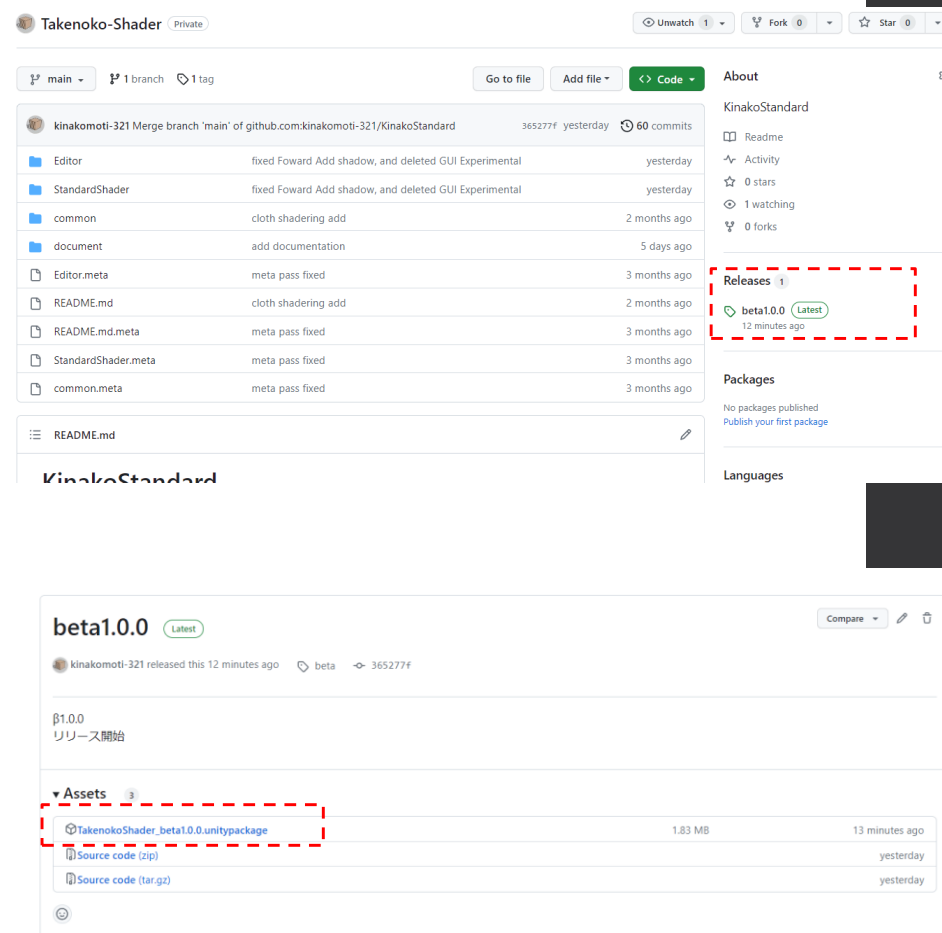
Booth

<https://kinankocraft.booth.pm/items/5267948>

GitHub

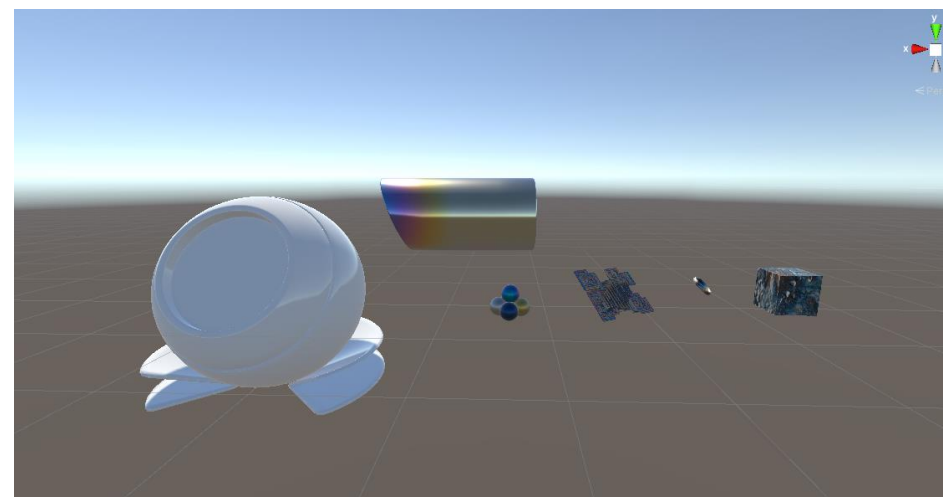
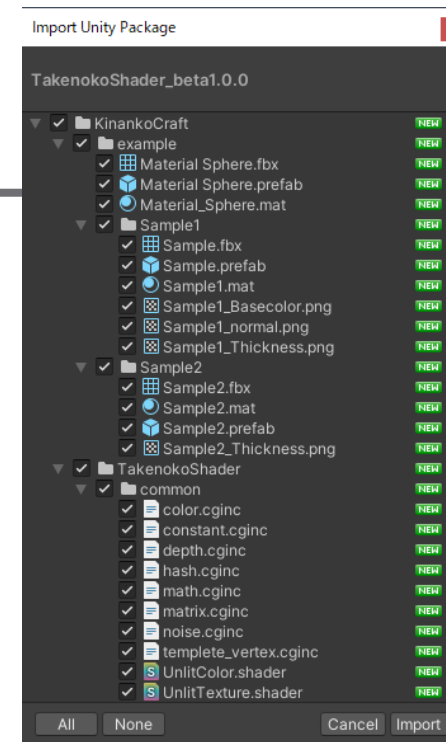
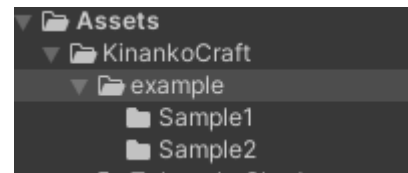
<https://github.com/kinakomoti-321/Takenoko-Shader>

- GitHubのリリースでは過去バージョンも
配布しています



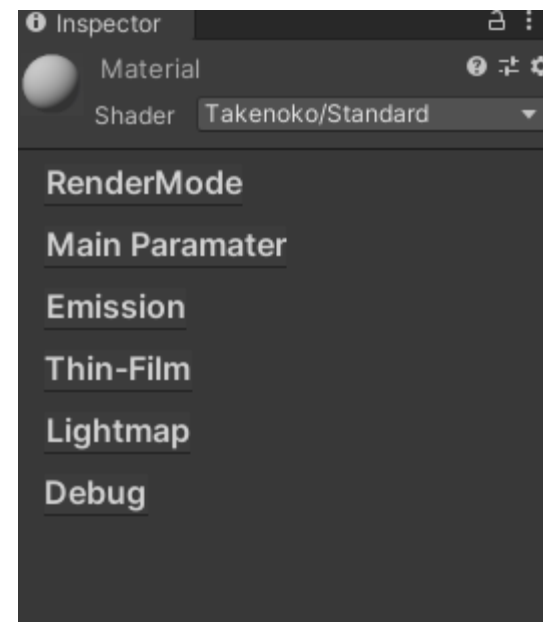
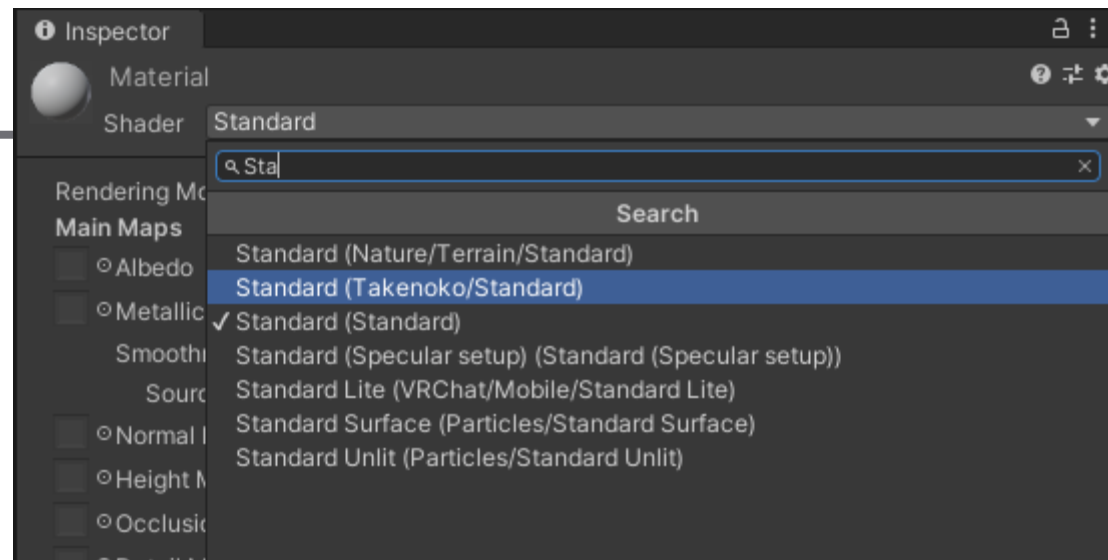
インポート方法

- インストールしたUnity PackageをUnityにインポート
- exampleにいくつかサンプルのPrefabがあります
- 各Prefabを出して見てマテリアルエラーが出なければ正常にインポート出来ています



インポート方法

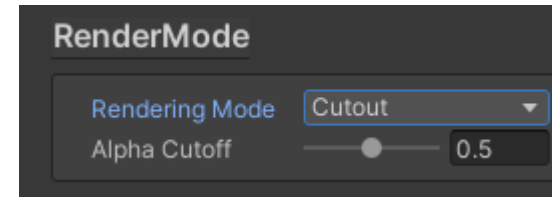
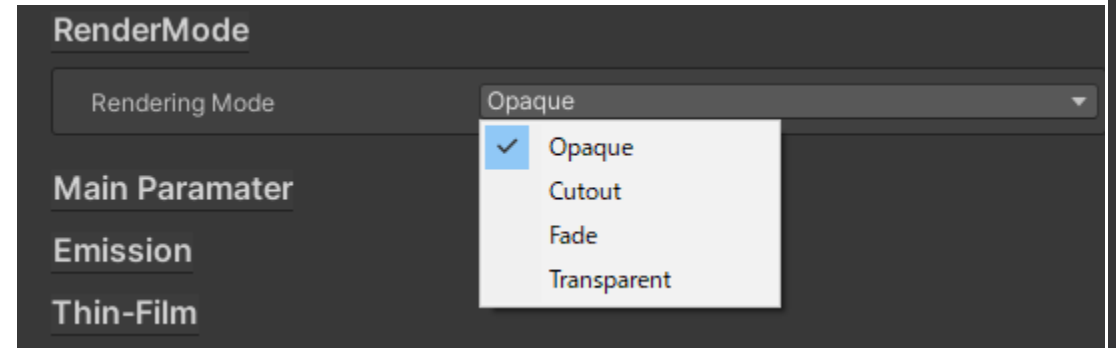
- Takenoko Shaderを使用した場合、
- マテリアルの検索欄から Takenoko/Standardを選択することで制作することができます。



Render Mode

Render Mode

- Opaque, Cutout, Fade, Transparent
を選択することが出来ます
- Cutoutでは切り取るアルファの閾値
を操作することが出来ます
- アルファ値はBaseColorのアルファ値
から取得しています

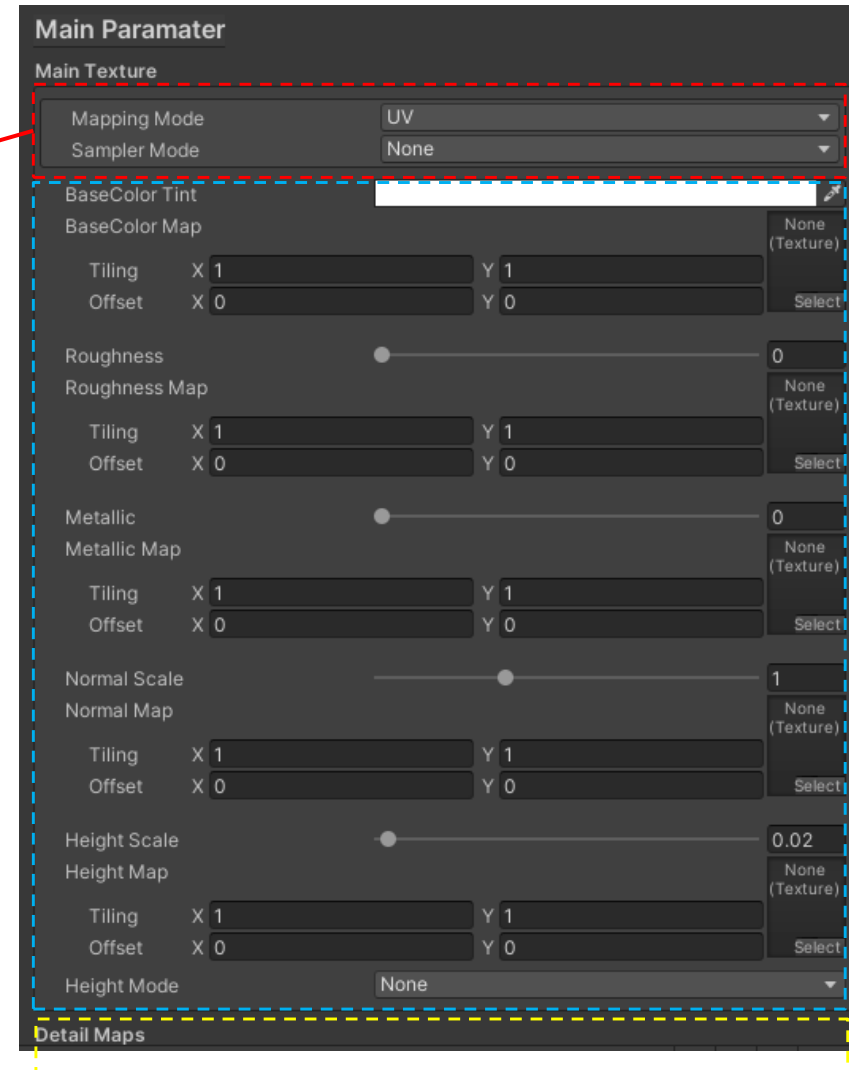


Main Parameter

Main Parameter

- 基本的なマテリアルのパラメータの設定する部分です

Textureの貼り方についての設定項目です
シームレスにしたり、UVがなくても貼り付けられる機能などがあります

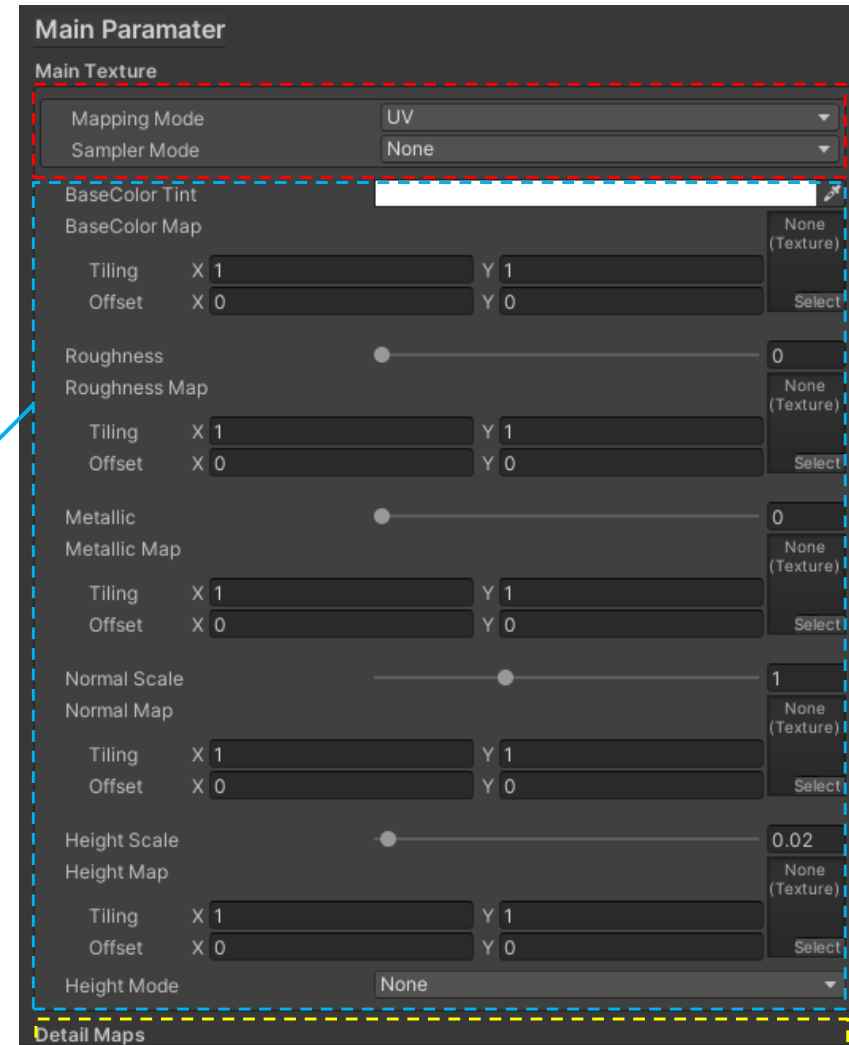


Main Parameter

- 基本的なマテリアルのパラメータの設定する部分です

マテリアルの基本パラメータの指定部分です
各項目はテクスチャを適用できます

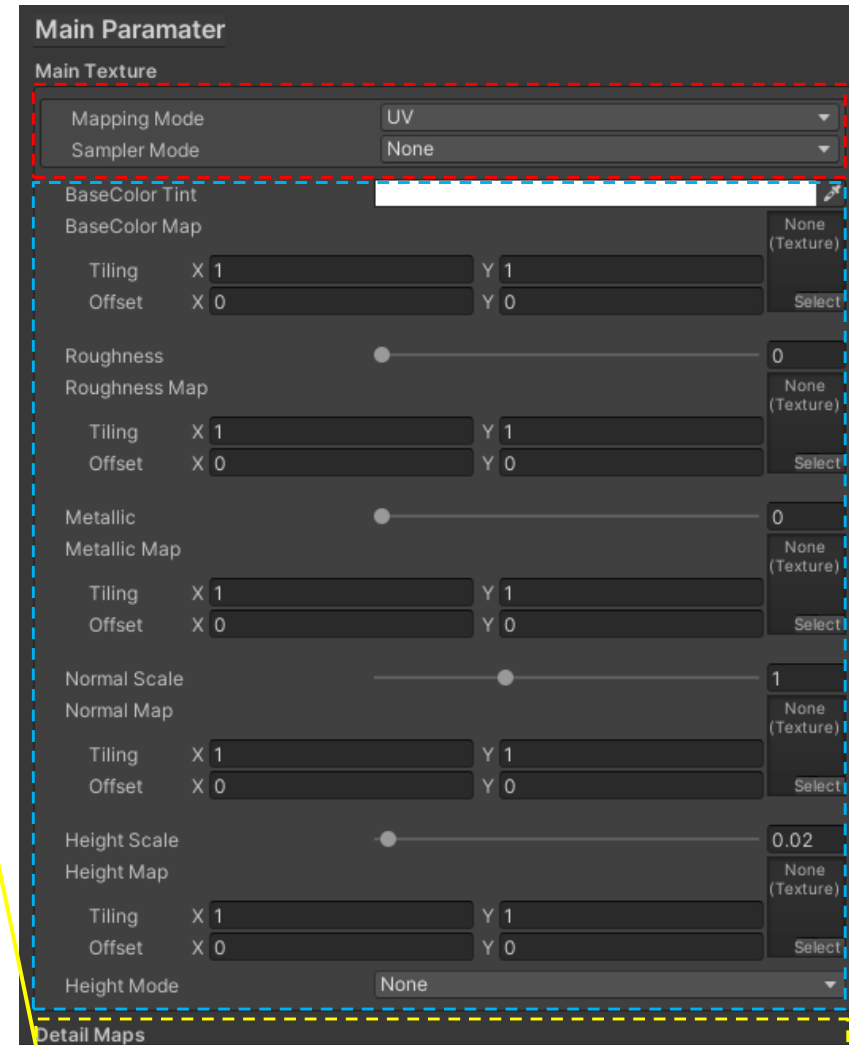
- BaseColor
- Roughness
- Metallic
- Normal
- Height



Main Parameter

- 基本的なマテリアルのパラメータの設定する部分です

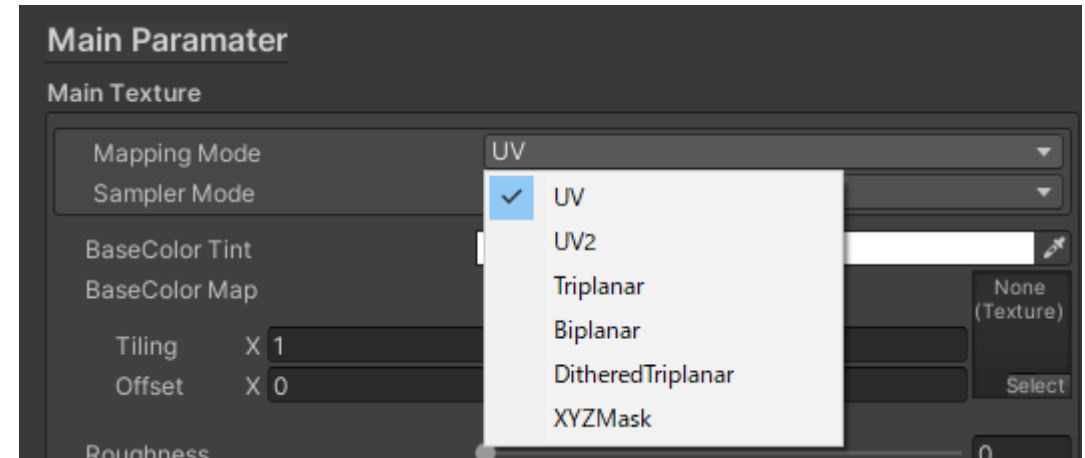
Detail Map(Secondary Map)を指定する部分です
細かい模様を入れたい時などに使用します



Main Parameter / Mapping Mode

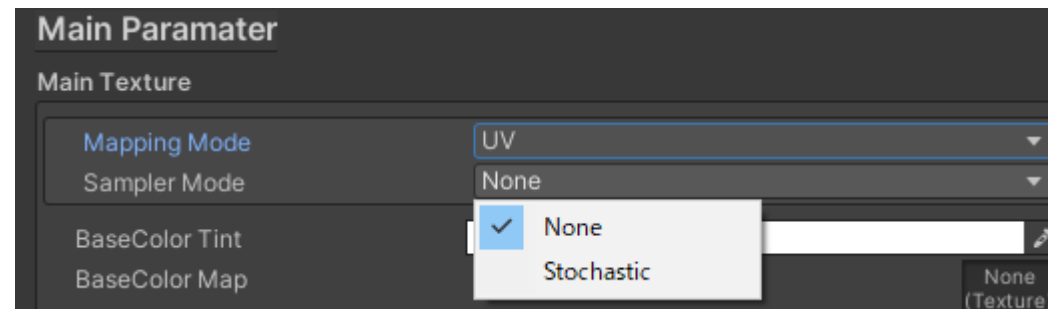
- テクスチャのマッピング方法を選択できます

1. UV : UVからマッピングします
2. UV2 : UV2からマッピングします
3. Triplanar : 法線と位置からマッピングする手法です(主にUVがないモデル、巨大なモデルに用います)
4. Biplanar : Triplanarより軽量な方法です
5. Dithered Triplanar : さらに軽量ですがつなぎ目がジャギジャギしてるやつです
6. XYZ Mask : XYZ方向からテクスチャを投影する方法です



Main Parameter / Sampler Mode

- テクスチャのサンプリング方法を選択できます



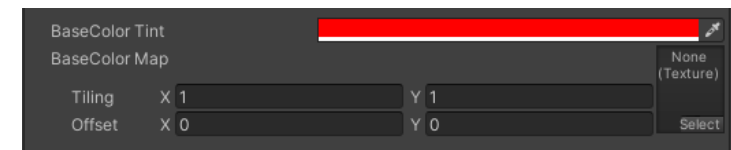
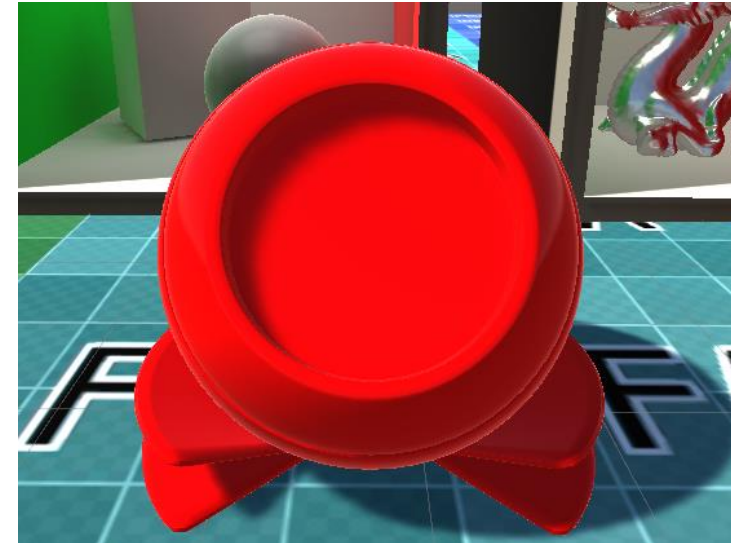
1. None : デフォルトのサンプリング方法
2. Stochastic : 自動的にシームレスなテクスチャリングをしてくれる方法です (巨大なモデルなどに使用します)

Main Parameter / BaseColor

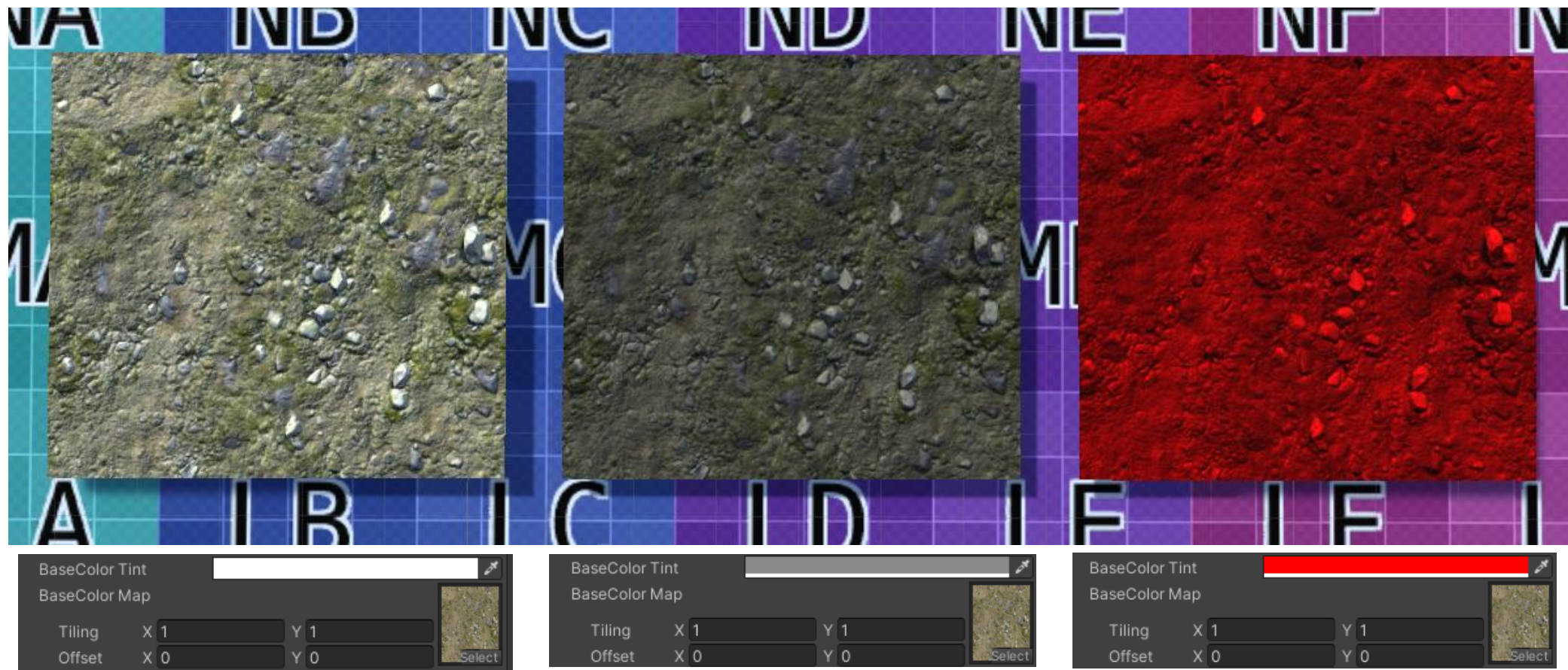
- BaseColor TintはMapがない場合はそのままカラーとして働きます

- Mapがある場合はBaseColor Tintの値と乗算した値を使用します。

$$\text{BaseColor} = \text{BaseColor Map} * \text{BaseColor Tint}$$

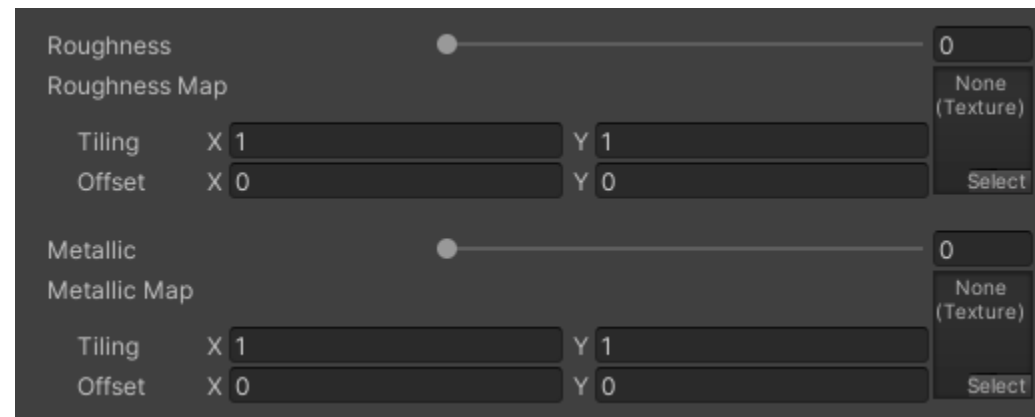


Main Parameter / BaseColor



Main Parameter / Roughness, Metallic

- パラメーターとMapの関係はBaseColorと同様です
- 各MapはRGBの内、Rの値を使用します



Main Parameter / Normal

- Normal Scaleはノーマルマップの強さを調節する項目です[0~3]
- 0ならノーマルマップがないのと同じ
- 1はデフォルト、それ以上になると傾きが強くなっていきます



Normal Scale = 0

Normal Scale = 1

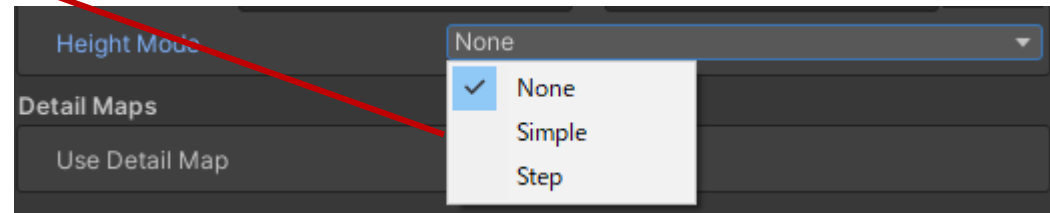
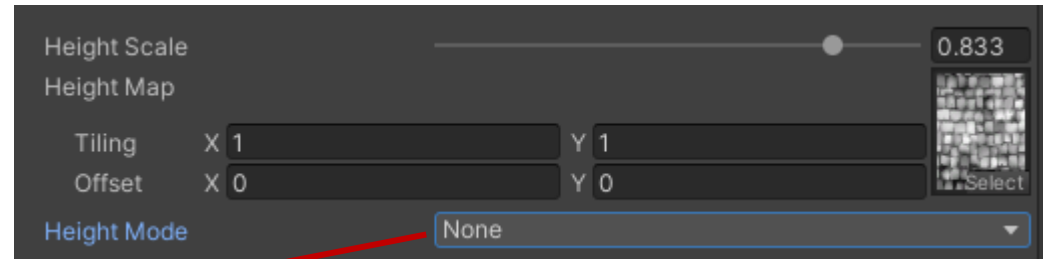
Normal Scale = 3



Main Parameter / Height

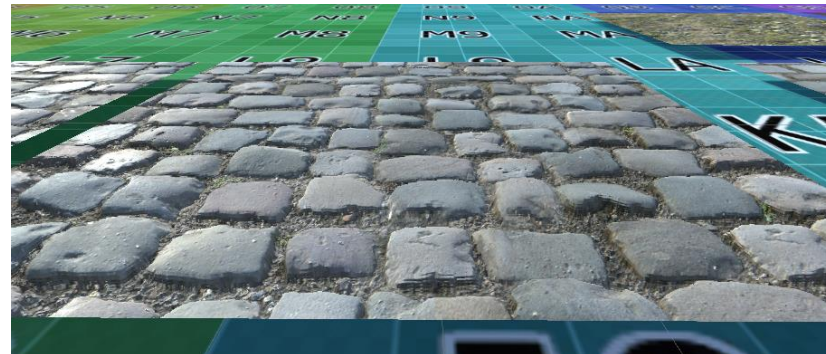
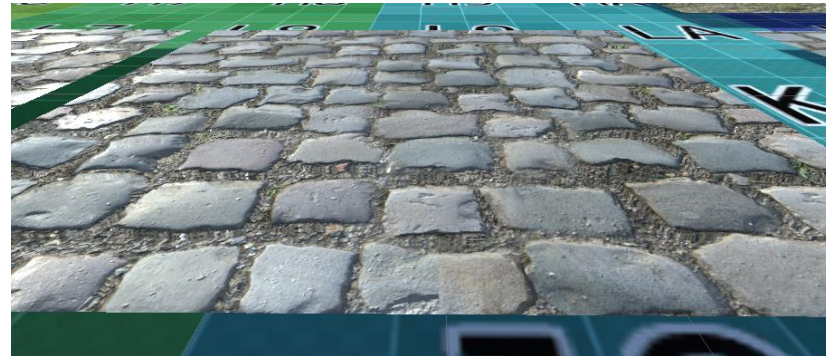
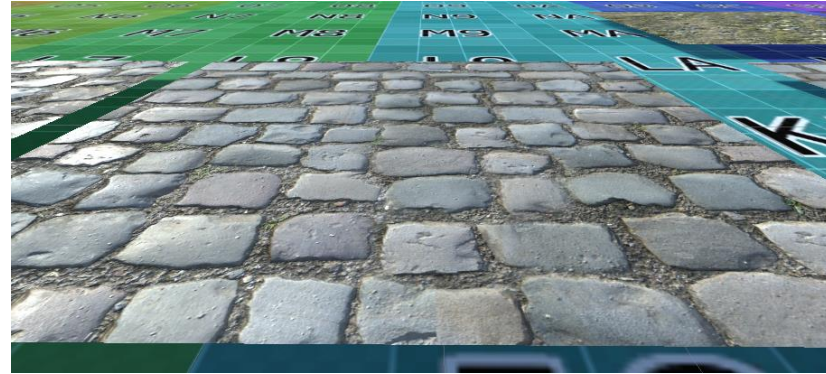
- HeightMapを凹凸の高さとし、疑似的に凹凸を表現する機能です

Height機能を使う場合HeightModeを「Simple」,「Step」のどちらかを選択してください



Main Parameter / Height

- None (デフォルト)
Height機能をオフにするモードです
- Simple
軽量ですが品質は悪いHeight機能です
- Simple
重いですが品質が良いHeight機能です

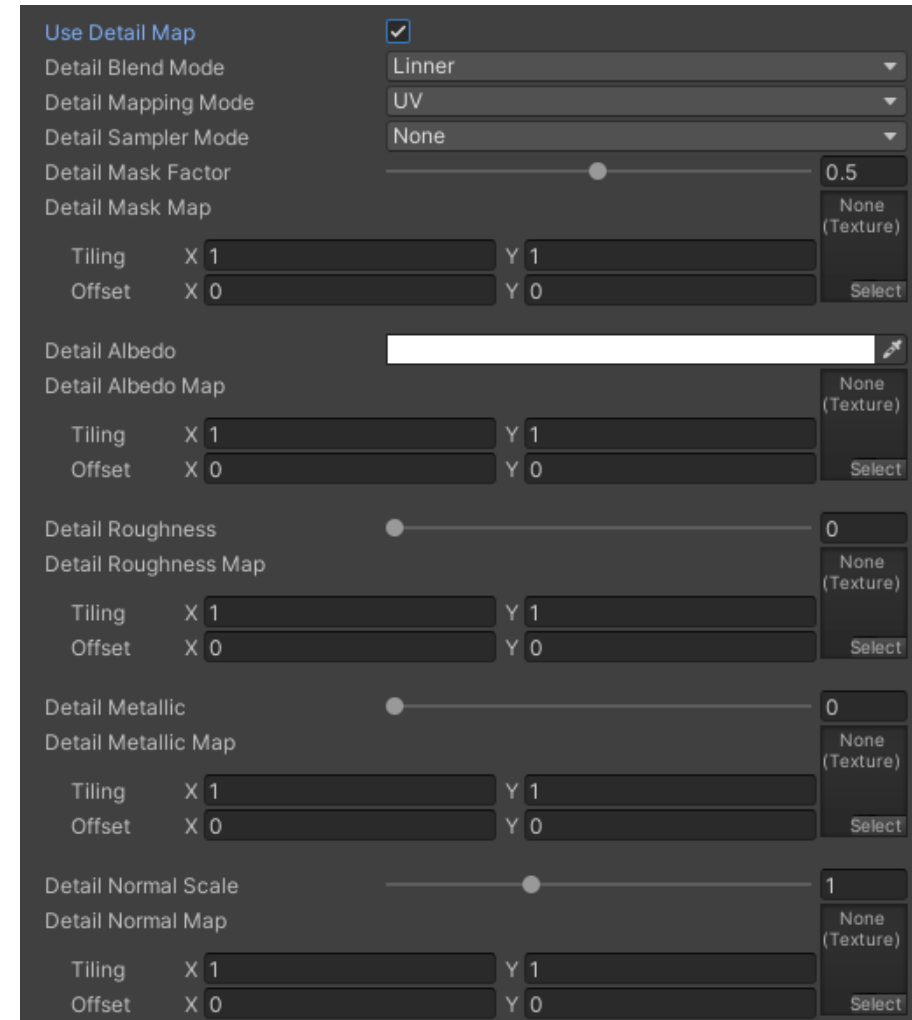


Main Parameter / Detail Map

- Main Textureに更に重ねる2枚目のTextureです

<https://docs.unity3d.com/ja/2019.4/Manual/StandardShaderMaterialParameterDetail.html>

- 細かい模様を別途に入れるなどの用途に使います
- 基本的にMainTextureと同様に設定できます



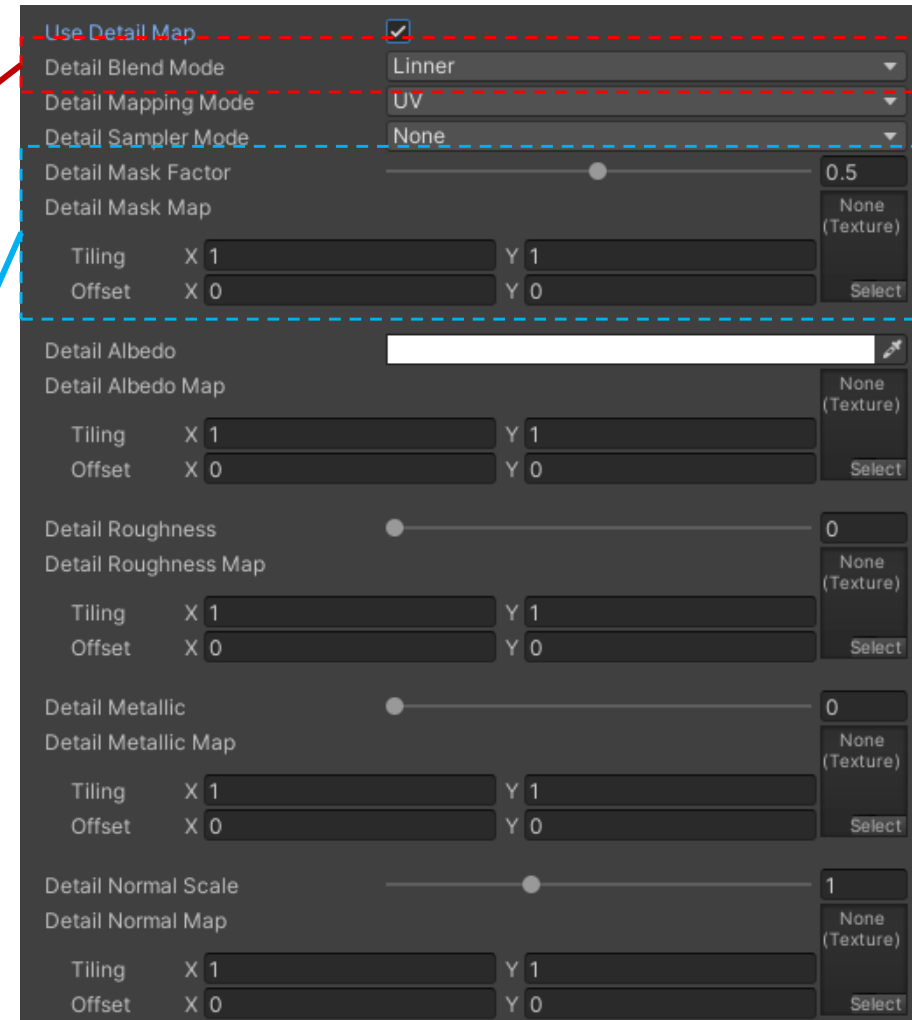
Main Parameter / Detail Map

Detail Blend Mode

- Detail Mapのブレンド方法をここで設定可能です
- ※Normal Mapのブレンド方法は変わりません

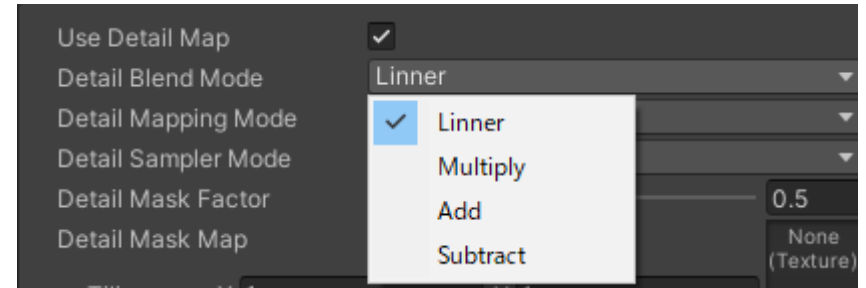
Detail Mask

- DetailMapの強さはMaskで制御することが出来ます
- Maskの値0なら完全にDetailMapの影響はなくなります



Main Parameter / Detail Map

- ブレンド方法は4種類あります
- Linner
- Maskの値が1の時、Detail Mapで完全に上書きします



Mask = 0.0

Mask = 0.5

Mask = 1.0



Main Map

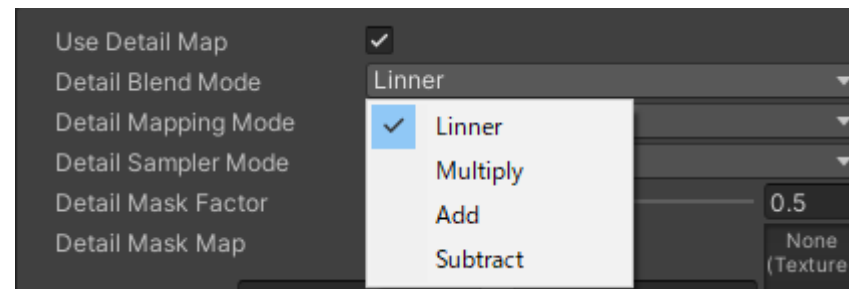


Detail Map



Main Parameter / Detail Map

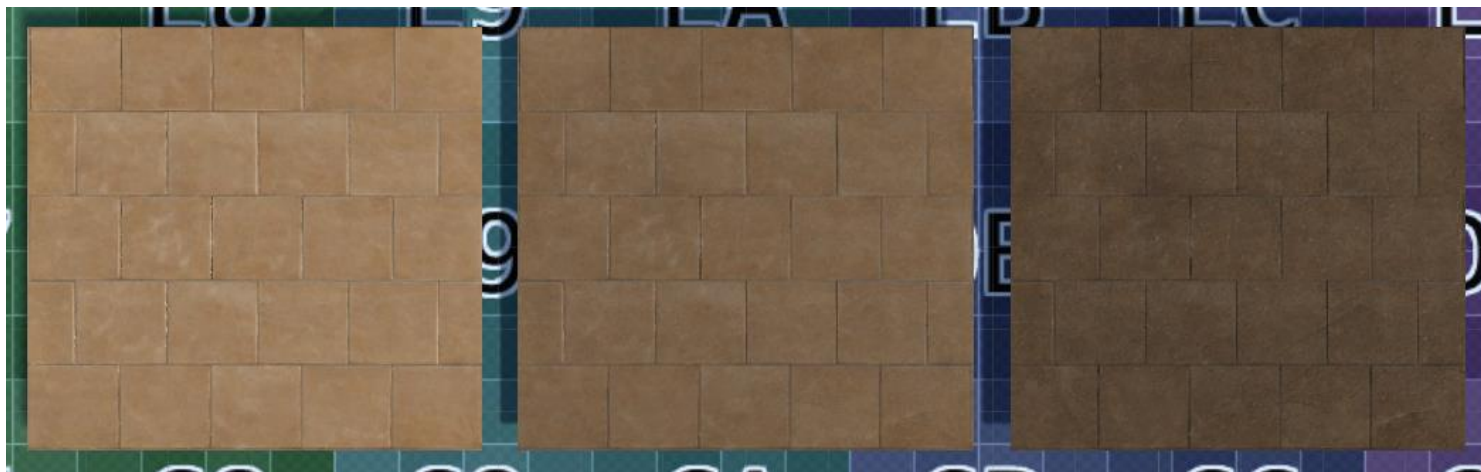
- Mutiply
- Maskの値が1の時、Main MapにDetail Mapを乗算した値を用います



Mask = 0.0

Mask = 0.5

Mask = 1.0



Main Map

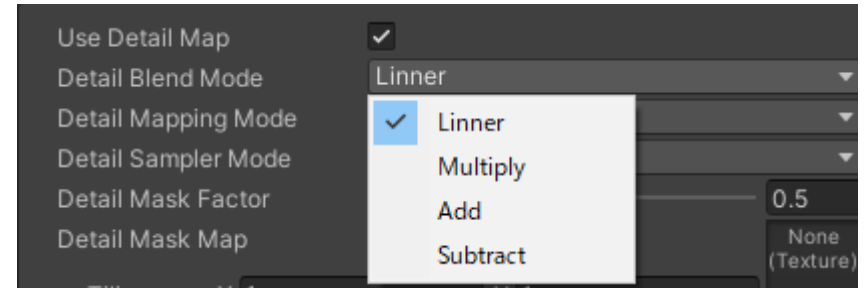


Detail Map



Main Parameter / Detail Map

- Add
- Maskの値が1の時、Main MapにDetail Mapを加算した値を用います



Mask = 0.0

Mask = 0.5

Mask = 1.0



Main Map

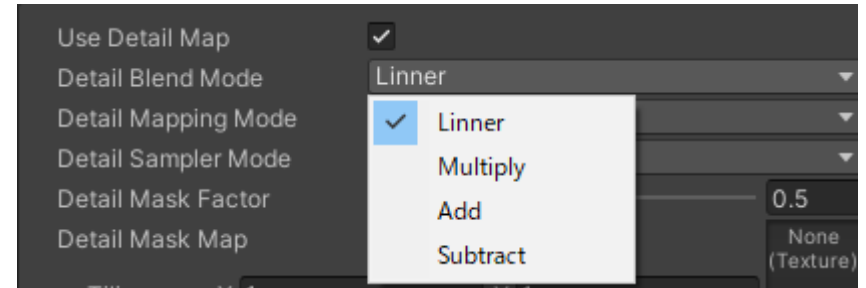


Detail Map



Main Parameter / Detail Map

- Subtract
- Maskの値が1の時、Main MapにDetail Mapを減算した値を用います



Mask = 0.0

Mask = 0.5

Mask = 1.0



Main Map



Detail Map



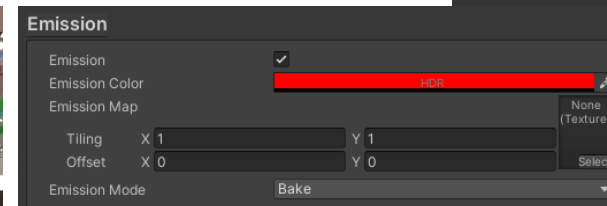
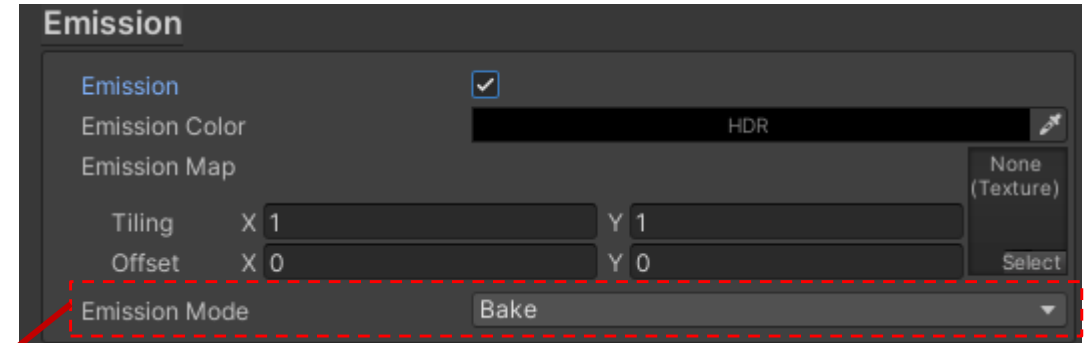
Emission

Emission

- 発光の材料を設定する部分です
- 基本的にUnity Standard Shaderと同様のパラメーターです

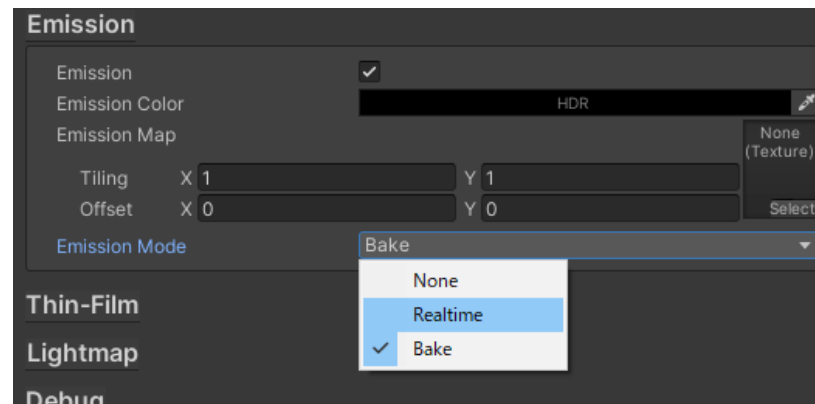
Emission Mode

Bake時の光源の扱いを設定する部分です

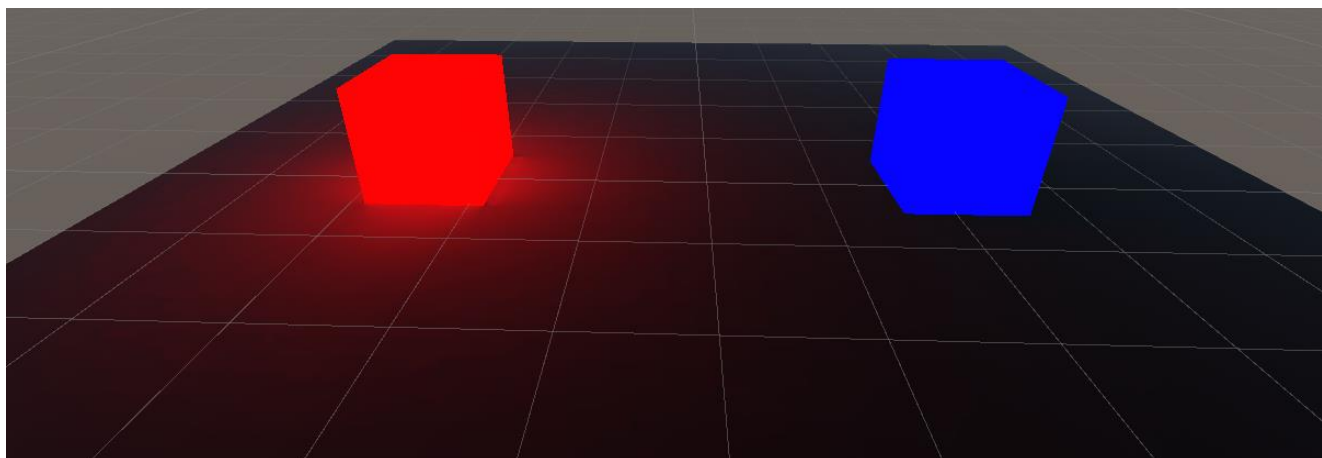


Emission / Emission Mode

- デフォルトではライトマップのベイク時に光源として扱われます(Bake)
- Staticでもベイク光源として扱いたくない場合はRealtimeに設定してください



Emission Mode = Bake Emission Mode = Realtime



※全てStaticに設定しています

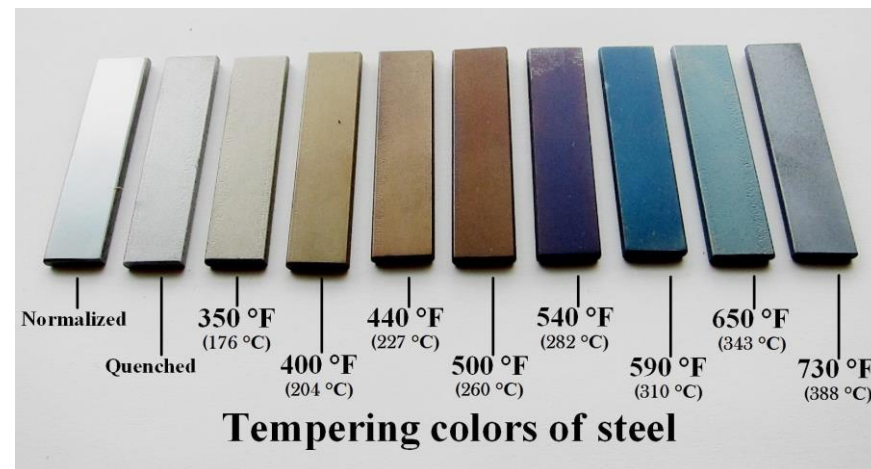
Thin-Film (薄膜干涉)

薄膜干渉とは

- ・ 薄膜干渉とは物体表面に薄い膜が張られることで生じる構造色的一种です

Example

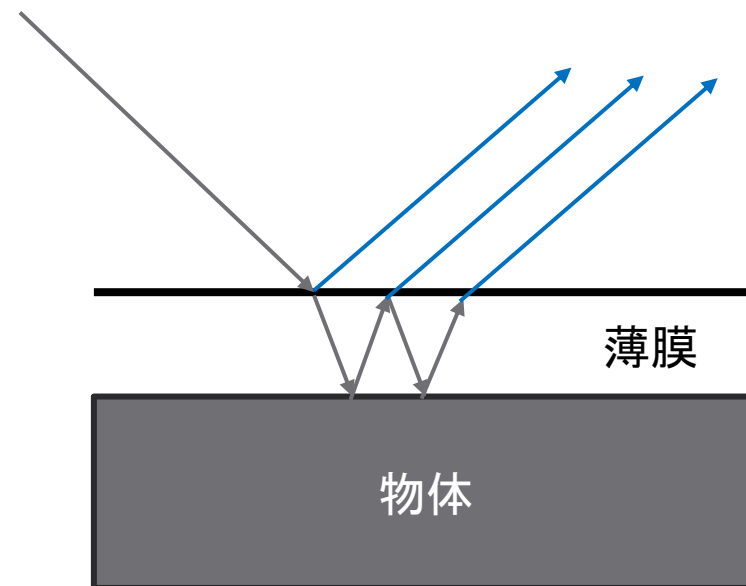
- ・ 金属の焼け(チタングルー, ビスマス結晶など)
- ・ シャボン玉の虹色
- ・ 玉虫の虹色



・ Wikipedia 薄膜干渉より引用

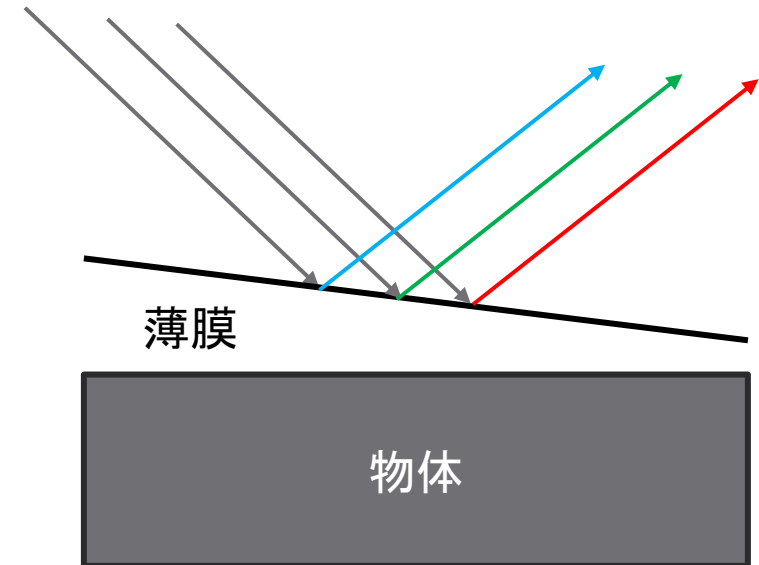
薄膜干渉の仕組み

- 物体に薄膜が存在する場合、薄膜内で反射した光同士が干渉し合います
- この干渉によって特定の色(波長)の光が弱まったり強くなったりして、色の偏りが生まれます
- この色の偏りは薄膜の「厚み」「屈折率」によって変化します



薄膜干渉の仕組み

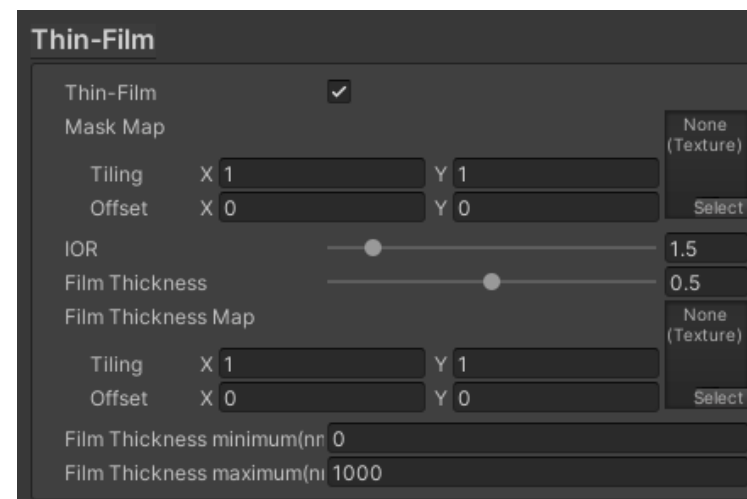
- 様々な色が現れるのは厚みが場所ごとに異なるためです
- 厚みに変化を持たせることで現実味のある薄膜干渉を作ることができます



薄膜干渉に必要なパラメーター

- Takenoko Shaderでは以下のパラメーターで薄膜干渉の色を計算できるようにしています

1. 薄膜の屈折率 IOR
2. 薄膜の厚み Thickness

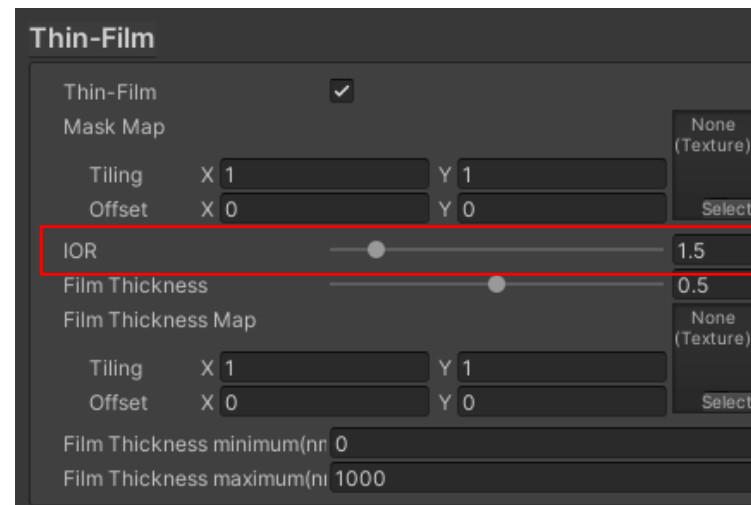


Thin-Film / IOR

- 薄膜の屈折率を設定する項目です
- 一般に見る薄膜の材質と屈折率は以下の通りです

材質	屈折率(IOR)
油	1.2 ~ 1.5
石鹼水	1.33
金属の酸化膜	2.0 ~ 3.0

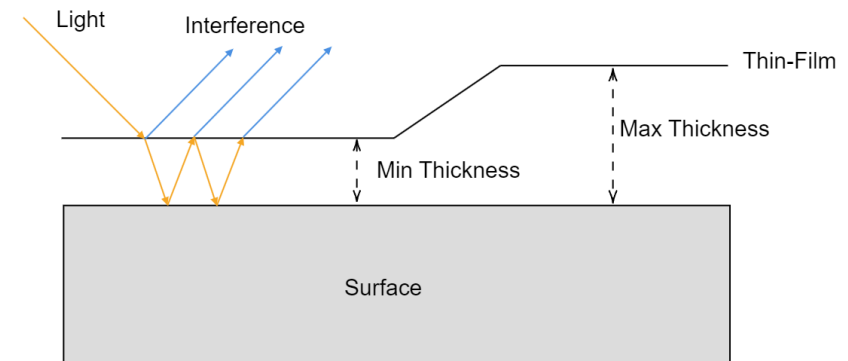
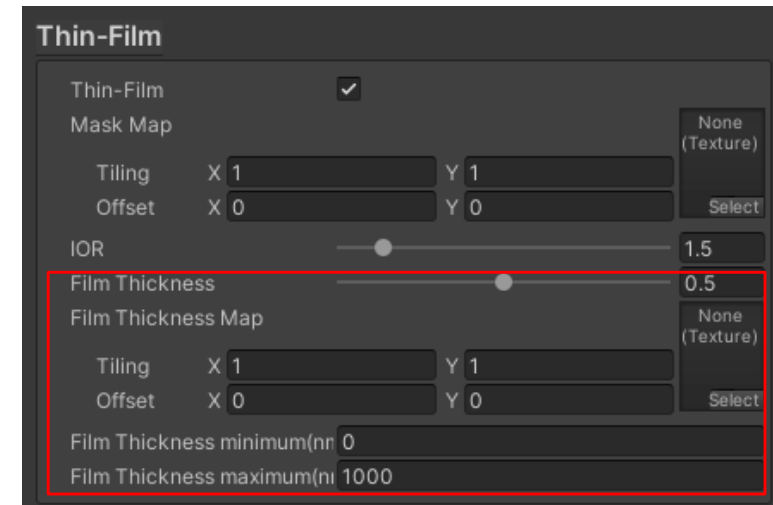
- これを目安に設定するとそれっぽくなります



Thin-Film / Thickness

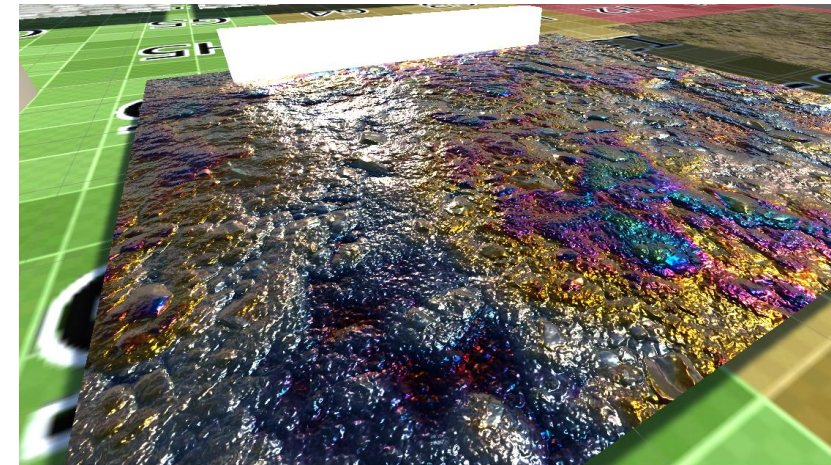
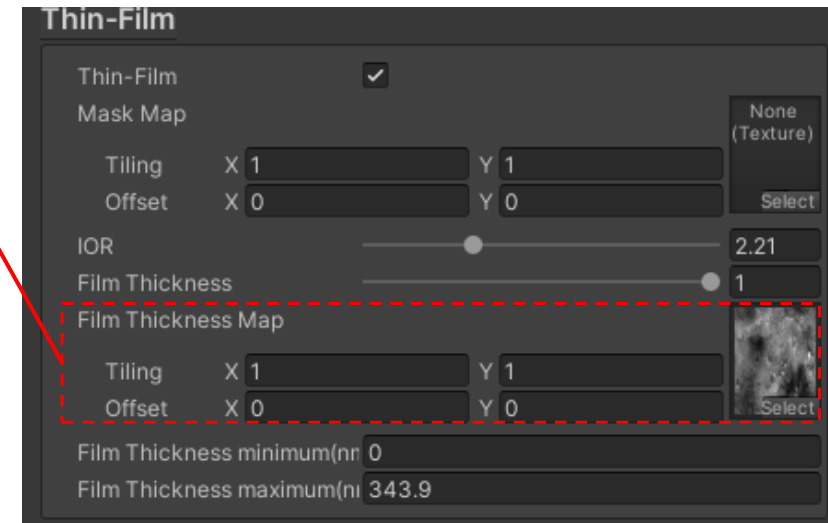
- Thicknessに関わる項目です
- 具体的な厚みの長さは下の二つの項目で設定することが出来ます。
- Film Thickness minimum : 厚みの下限値(nm)
- Film Thickness maximum : 厚みの上限値(nm)

Thicknessが0ならMin, 1ならMaxの厚みを使用します



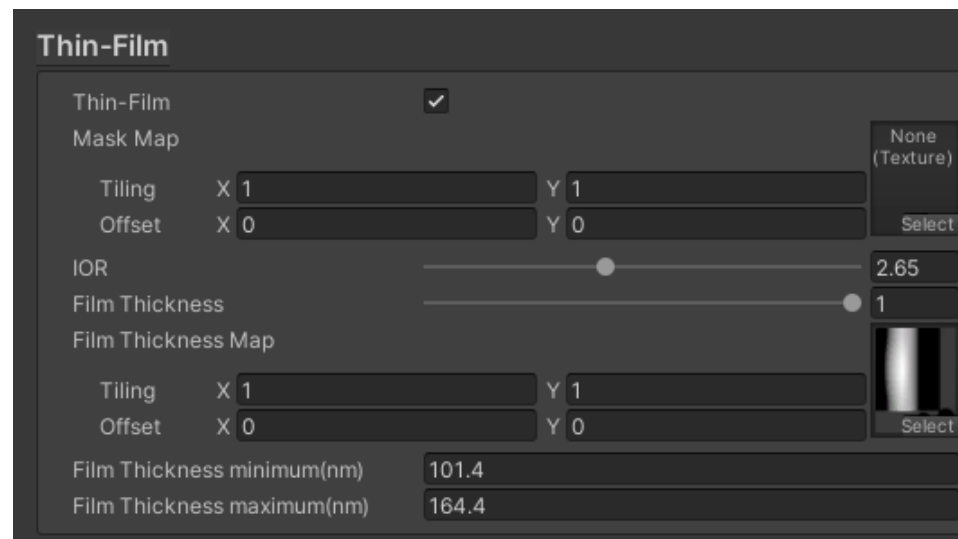
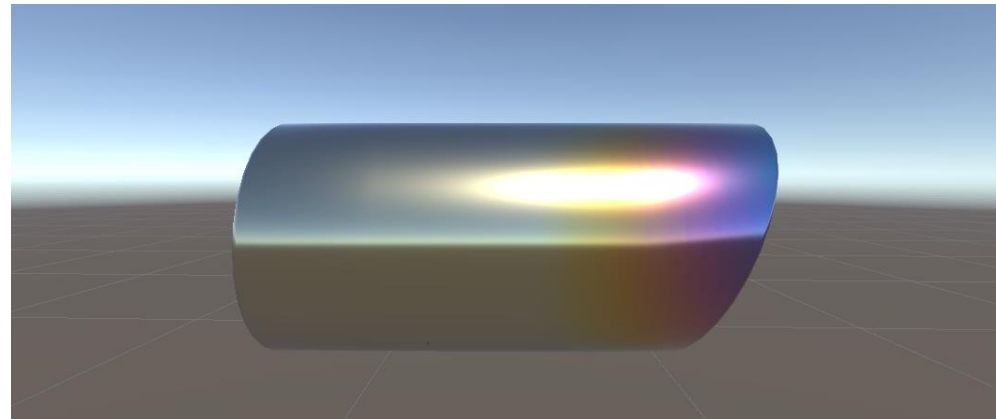
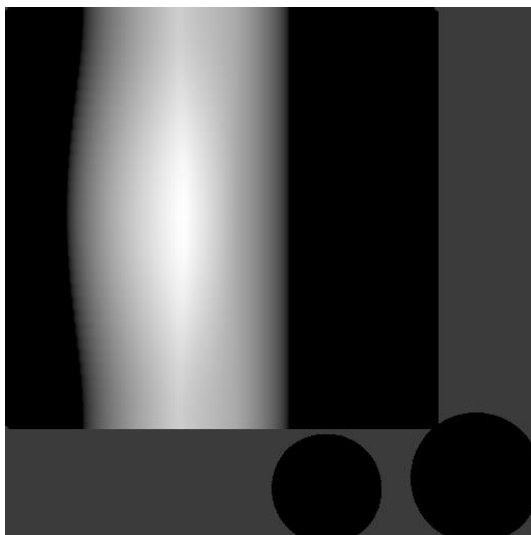
Thin-Film / Thickness

- グレースケールのテクスチャを厚みのテクスチャとして使用することが出来ます
- 現実的には薄膜の厚さは変化していることが普通なので適用するといいい見た目になります
- Roughness Map、AO Mapなどを流用してもいい感じに見えたりします



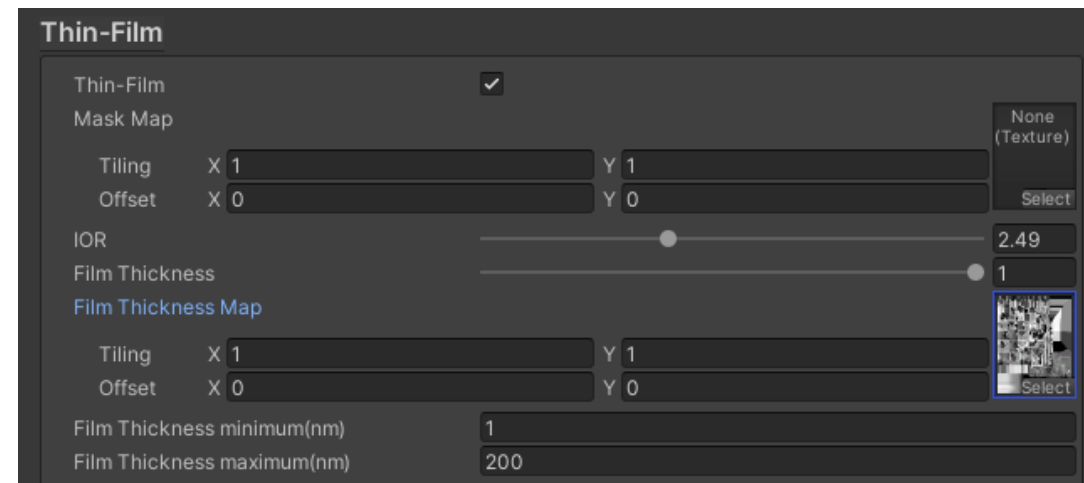
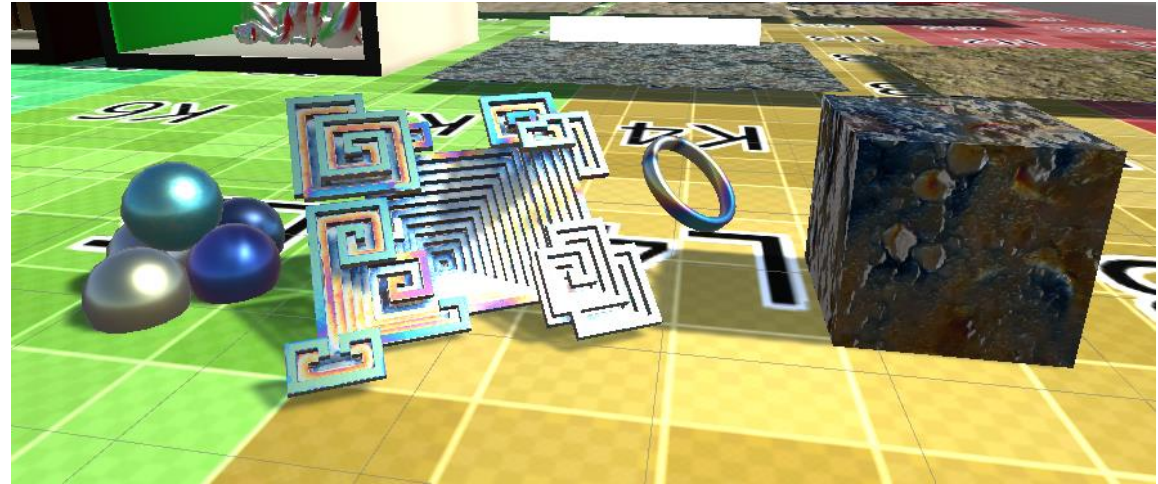
Example

- IOR : 2.65
- Thickness Minimum : 101.4
- Thickness Maximum : 164.4
- グラデーションを付けたThickness Mapを用意



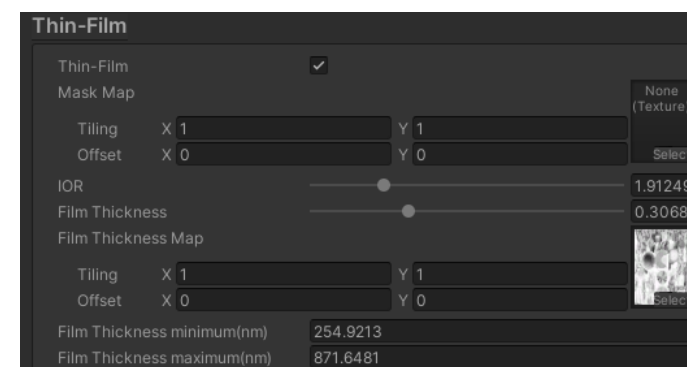
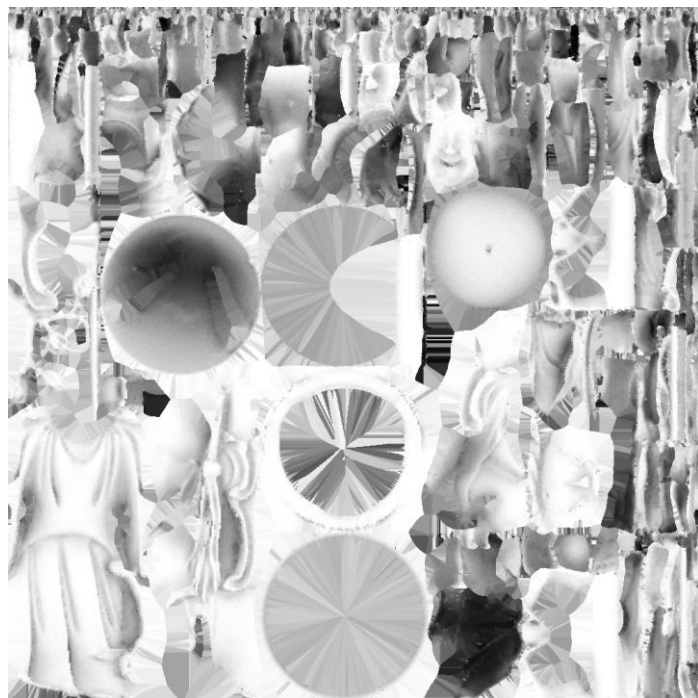
Example

- IOR : 2.49
- Thickness Minimum : 1
- Thickness Maximum :
- AO MapをThickness Mapとして使用



Example

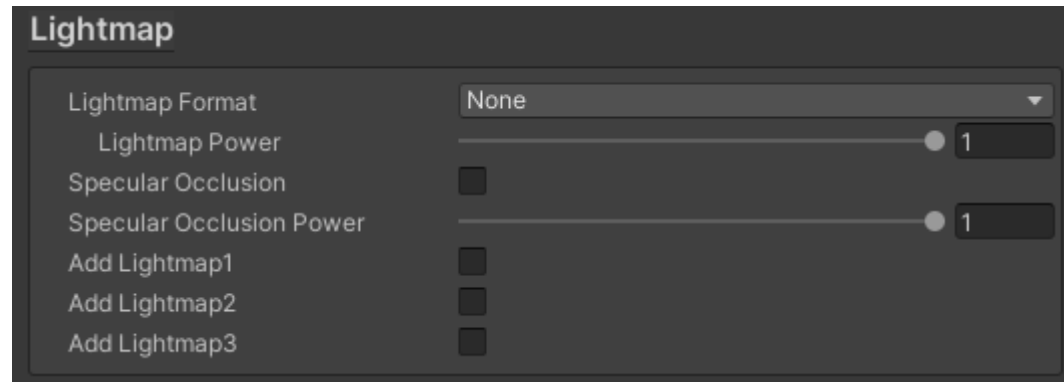
- AO MapをThickness Mapとして使用



Lightmap

Lightmap

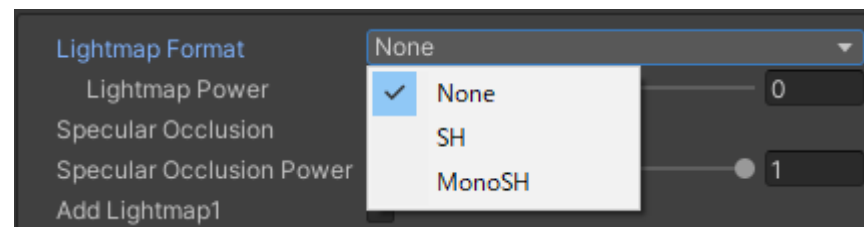
- Lightmapに関わる項目です
- Lightmapのフォーマット
- Lightmapの明るさ調整
- 疑似Specular反射
- テカリの抑制
- Lightmapの追加



Lightmap / Format

- 当シェーダーでは以下のLightmapのフォーマットに対応しています
- 利用するLightmapの形式を手動で選択する必要があります

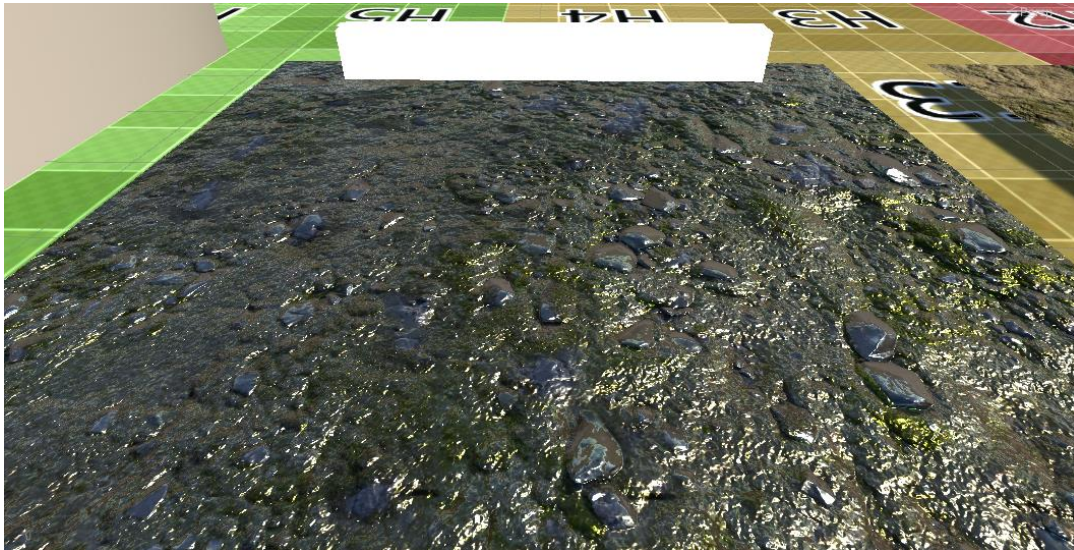
1. 標準のLightmap
2. SH Lightmap (Bakery必須)
3. Mono SH Lightmap (Bakery必須)



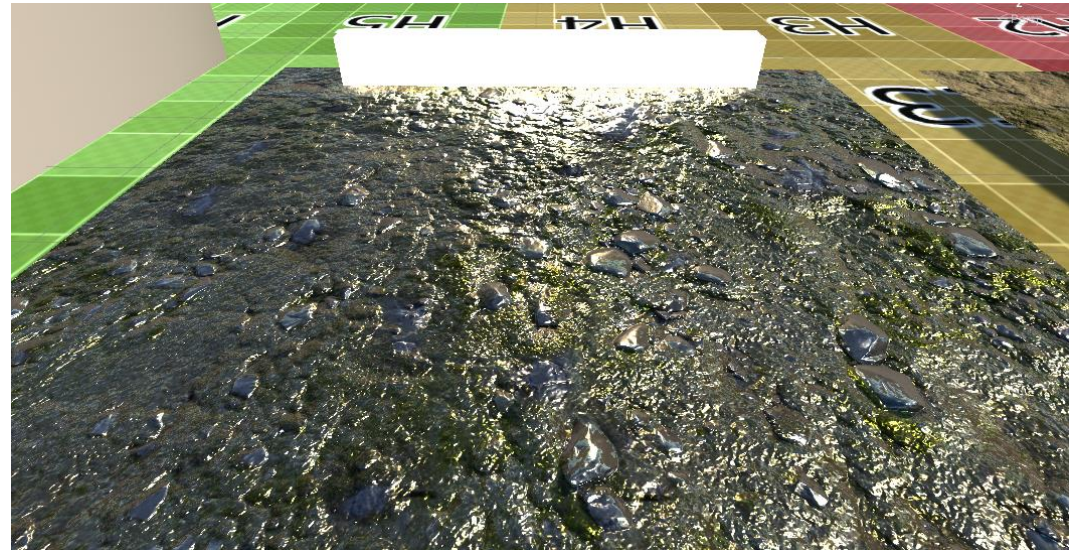
Lightmap / Specular Approximate

- SH, Mono SHのみで可能な機能です
- ハイライト(Specular光)をライトマップから疑似的に表現する方法です
- メタリックが強めな材料に効果的です

Specular Approximate OFF

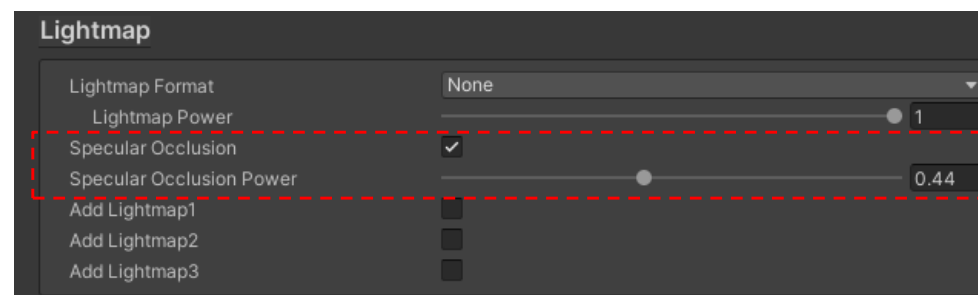


Specular Approximate ON



Lightmap / Specular Occlusion

- ライトマップにおいて影になる部分でハイライトを抑える機能です
- 隙間などが白くなる現象を防ぎます
- Powerで強さを調節できます



Specular Occlusion OFF

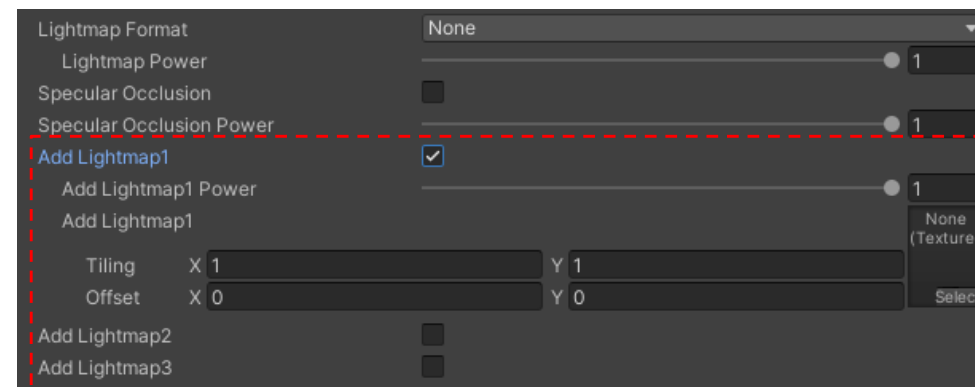


Specular Occlusion ON



Lightmap / Additional Lightmap

- 手動で計3枚までのライトマップを追加することが出来ます
- Add Lightmap にチェックを入れるとライトマップを指定する項目を表示できます
- 各ライトマップはLightmap Powerで明るさを調節できます

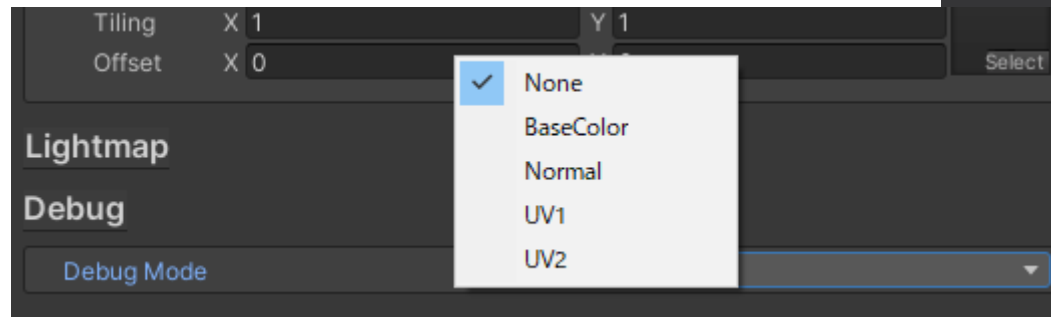


- ライトON,OFF演出などにご使用頂けます
- SH, Mono SH Lightmap非対応

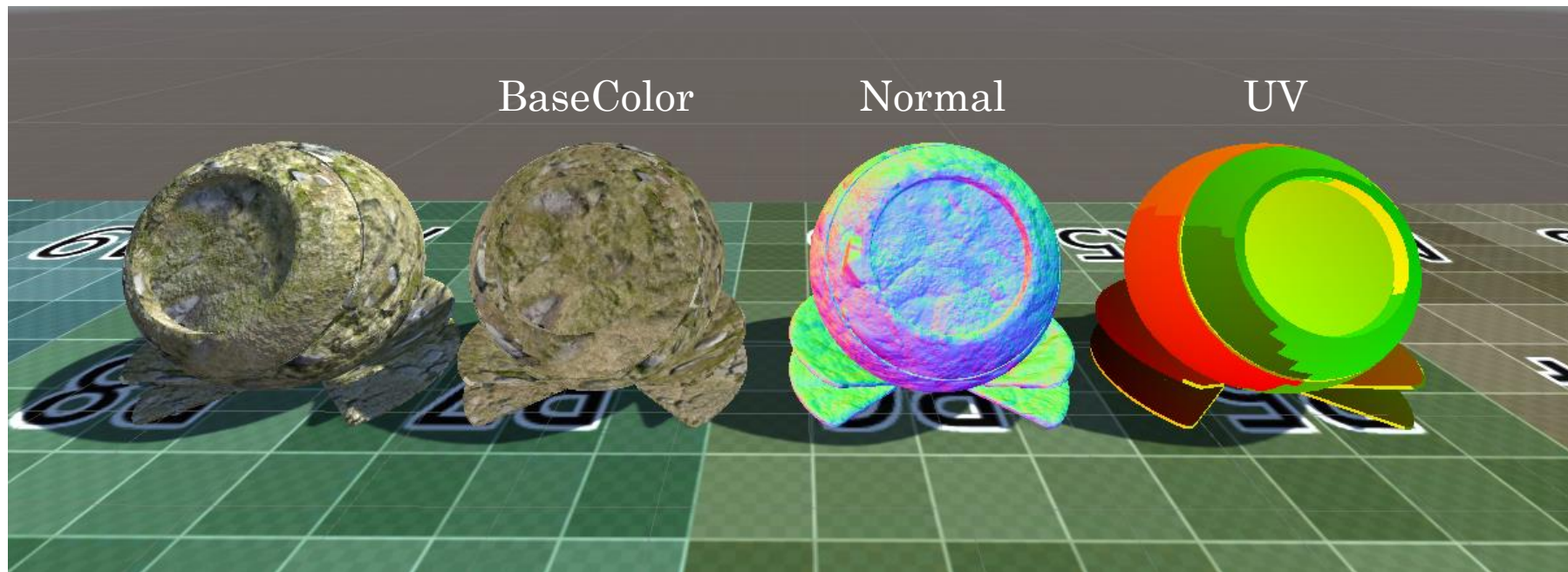
Debug

Debug

- ベースカラーやノーマルなど基本的な情報を表示することが出来ます
- モデルが正常なのかを確認する際にご使用ください
例.
 - Normalが正常なのか
 - UVがあるのかないのか



Debug



ライセンス、連絡先など

ライセンス

- ・ 当シェーダーはApache License 2.0の下配布しているため、以下の行為を許可します
 - ・ 利用料無料
 - ・ 商用利用（無償かつ無制限に使用可）
 - ・ 改変
 - ・ 複製
 - ・ 公開
 - ・ 再配布
 - ・ 改変したものの配布
 - ・ サブライセンス
 - ・ アプリケーションへの埋め込み、同梱
 - ・ 特許ライセンスの付与
- ・ クレジットへの表示は任意です

連絡先

- 質問やバグ報告、ご要望などがございましたらBooth または Twitterにご連絡ください
- Twitter @kinankomoti321
- Booth <https://kinankocraft.booth.pm/>
- Github <https://github.com/kinakomoti-321>