

# トゥーンレンダリング

2025年度 プログラムワークショップIV (6)

# アジェンダ

- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トーンシェーディング
- NPR: 疑似ハーフトーン

# アジェンダ

- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トーンシェーディング
- NPR: 疑似ハーフトーン

# トゥーンレンダリング

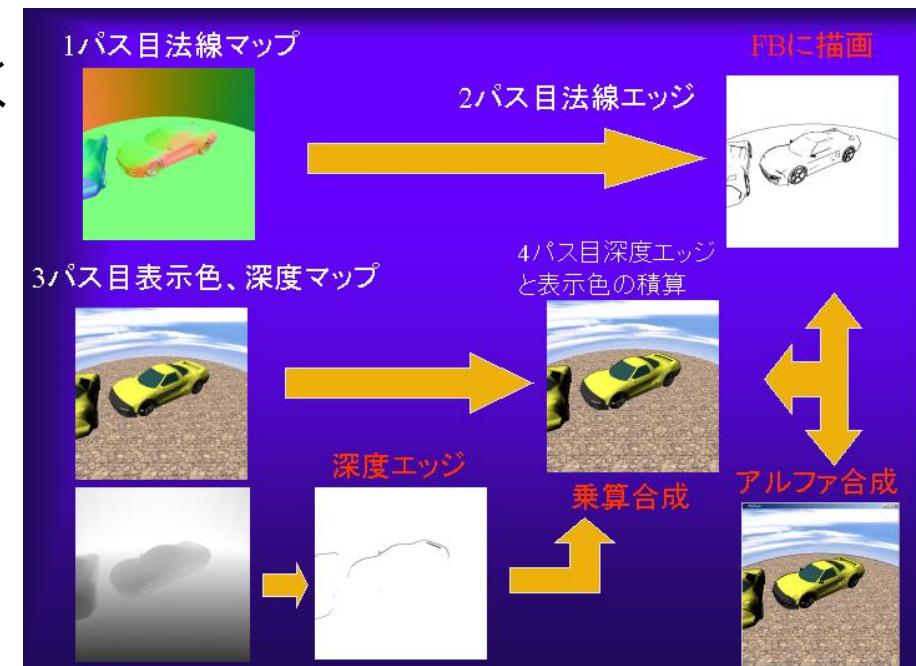
- アニメーションのように見せるレンダリング
  - 段階的陰影
  - 輪郭
  - ハッチング
    - 線画調
    - スクリーントーン
  - 誇張表現
    - 手前を大きく見せるなど
  - 残像？
- NPR(Non-Photorealistic Rendering: 非写実的レンダリング)の一つ
  - Nonが否定的なイメージを与えるので、Stylized Renderingともいう
  - ツゥーンでないNPR
    - セピア調・モノクロ
    - ランプシェーディング

# アジェンダ

- トゥーンレンダリングとは
- 輪郭
  - **概要**
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トーンシェーディング
- NPR: 疑似ハーフトーン

# 輪郭

- 物体の輪郭に線を引く
  - 背面法
    - 裏返して拡大したオブジェクトを単色で描画
  - ポストエフェクト法
    - レンダリングした画像について輪郭部分に線を引く
      - 深度変化が大きい
      - 色が大きく変わる
      - 法線変化が大きい
      - オブジェクトIDが異なる
      - どこかに何かのデータを仕込んで書きだす
        - アーティストの制御性を確保する



# とても良い資料

- [https://cedil.cesa.or.jp/cedil\\_sessions/view/2952](https://cedil.cesa.or.jp/cedil_sessions/view/2952)
- 同じ内容の動画
  - <https://www.youtube.com/watch?v=zCQVPeFcNUw>

## 3Dでキレイな線を引くために。ギルティギアシリーズのトゥンライン制御テクニック

タグ： [CEDEC 2024 VA](#)

日時： 2024年08月22日(木)14時40分～15時40分 形式： レギュラーセッション(60分)

受講スキル： トゥーン表現に興味のある3Dモデラー、シェーダーアーティスト、およびグラフィックプログラマー

受講者が得られるであろう知見：

背面法の基礎的な理屈から、応用テクニック各種をギルティギアシリーズでの実例を交えて紹介。定番手法である「背面法」を、さらに使いこなすための実践的な工夫の数々を知ることができます。

### セッションの内容

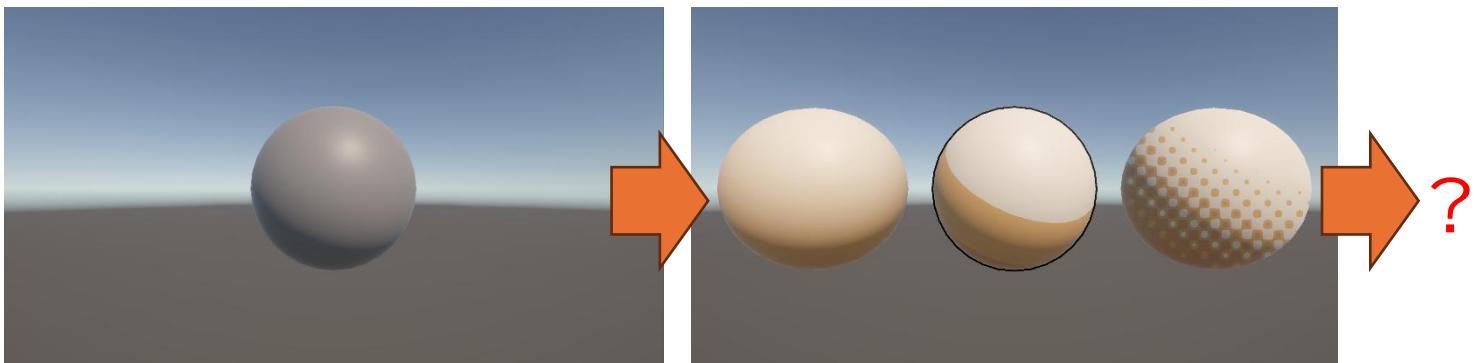
本セッションでは格闘ゲーム「ギルティギア」シリーズのビジュアルを支える「背面法アウトライン」を「意図したとおりに制御する」ための様々なテクニックと考え方を紹介します。

近年多数のタイトルが採用しているトゥーン表現。今セッションではそのトゥーン表現において非常に重要な役割を果たす「輪郭線」にフォーカスして解説していきます。

20年以上も使われ続けている定番手法「背面法」ですが、一見簡単そうに見えて、意図したとおりに制御するためには様々な工夫とテクニックが必要になります。今セッションではこの背面法による輪郭線の描画について、いまさら聞けない基礎的な理屈の部分から、実際のゲームで使用されている応用テクニックまで幅広く解説します。

# 本日の課題

- [https://github.com/tpu-game-2025/PGWS4\\_6\\_toon](https://github.com/tpu-game-2025/PGWS4_6_toon)  
をforkして、「結果画像」に結果を貼ってください
  - ・今回の範囲で何か試してください



tpu-game-2025 / PGWS4\_6\_toon

Code Issues Pull requests Actions Projects Security Insights Settings

Edit Pins Watch

PGWS4\_6\_toon Private

feature/halftoning had recent pushes 2 minutes ago Compare & pull request

develop had recent pushes 2 minutes ago Compare & pull request

main 6 Branches 5 Tags Go to file Add file Code

imagine initial state e03af68 · 5 hours ago 2 Commits

.vscode initial state 5 hours ago

src initial state 5 hours ago

README.md initial state 5 hours ago

result.png initial state 5 hours ago

README

## はじめに

プログラムワークショップIVの管理用です

## 結果画像

src - SampleScene - Windows, Mac, Linux - Unity 6.2 (6000.2.6f2) <DX12>

File Edit Assets GameObject Component Services Jobs Window Help

Unity 6 T1 Asset Store Unity Version Control

Hierarchy SampleScene Main Camera Directional Light Global Volume Sphere

Inspector

Main Camera Static

Transform Pcx 0 Y 0 Z -2 Rcx 0 Y 0 Z 0 Sx 1 Y 1 Z 1

Projection Perspective Field of View Ax Vertical Field of View 60 Clipping Planes Near 0.3 Far 1000 Physical Camera

Rendering Renderer Default Post Processing Anti-aliasing Stop NaNs Dithering Render Shadows Priority -1 Opaque Texture Use semi Depth Texture Use semi Depth Mask Use semi Occlusion Culling Everytri

Project Console

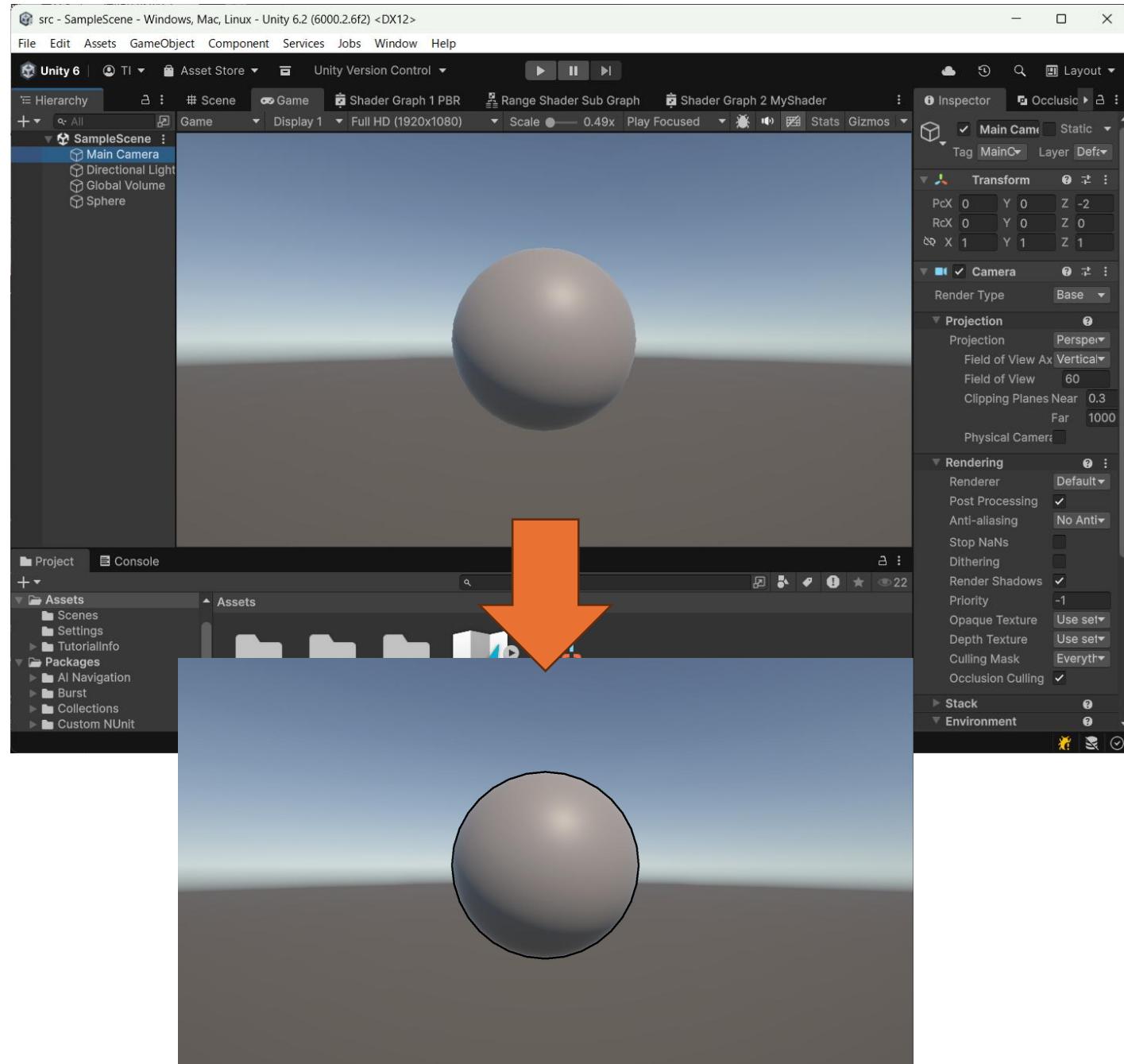
Assets Scenes Settings TutorialInfo Packages AI Navigation Burst Collections Custom NUI

Scenes Settings TutorialInfo InputSystem README

• 工夫した点：xxx

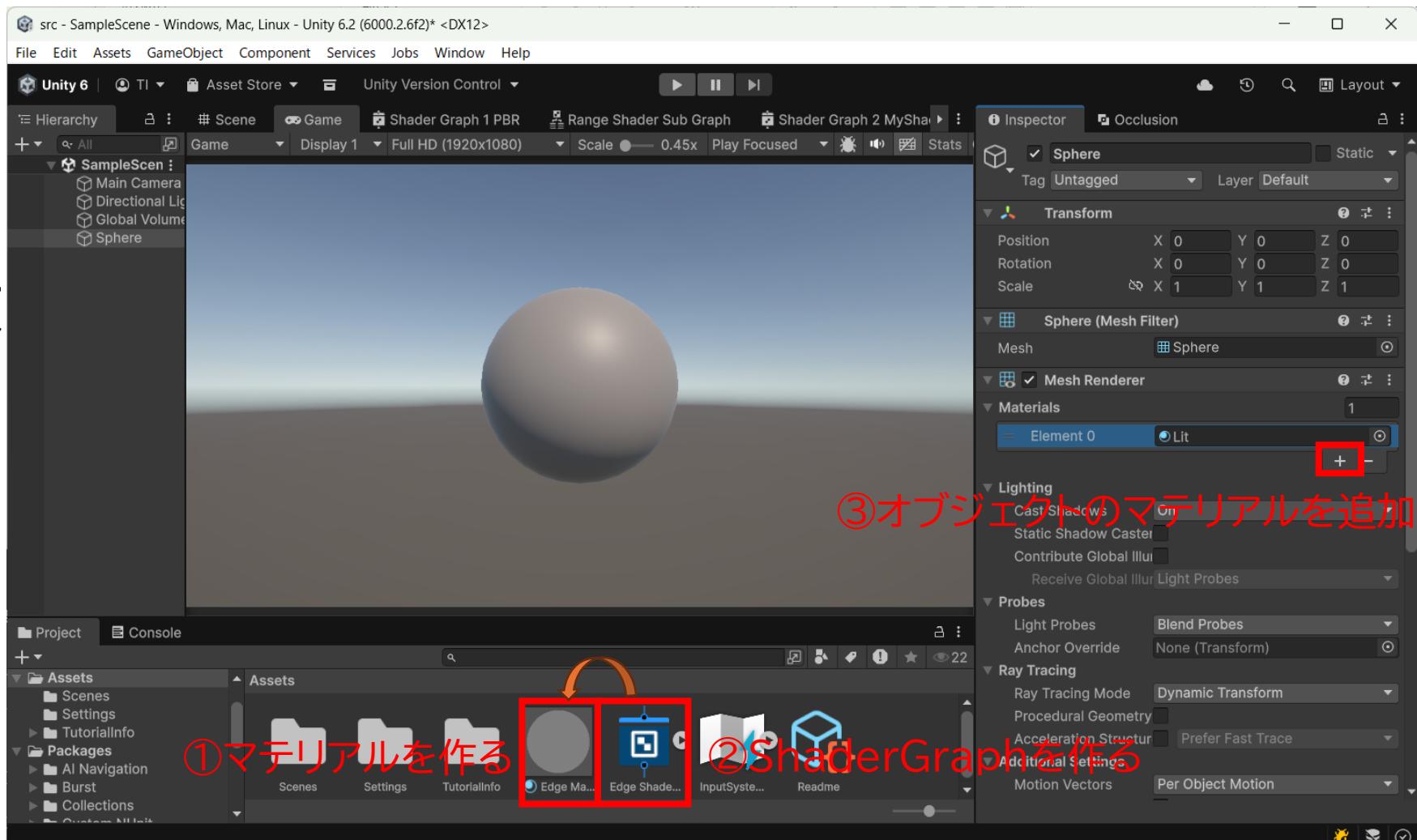
# アジェンダ

- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トゥーンシェーディング
- NPR: 疑似ハーフトーン

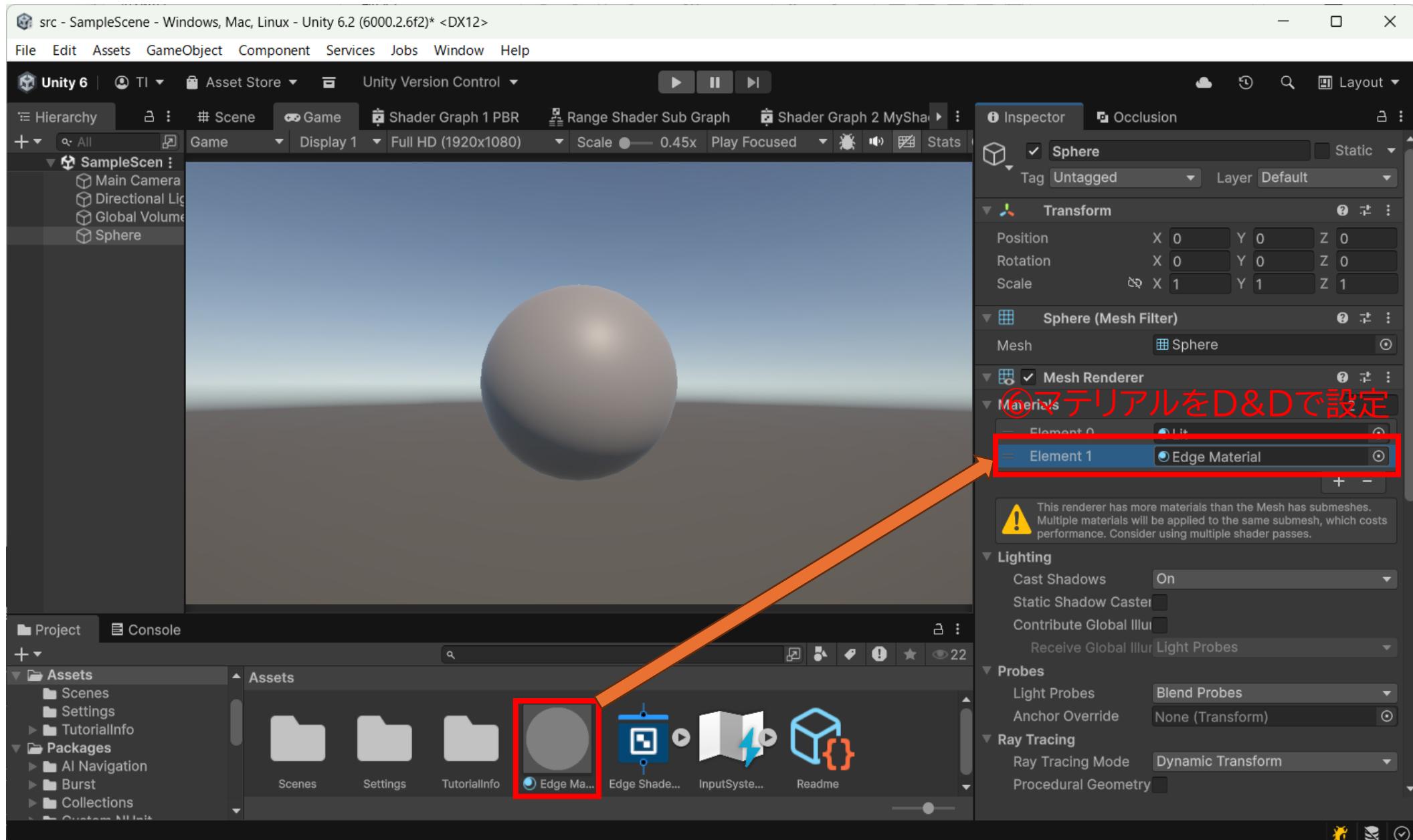


# マテリアルの追加

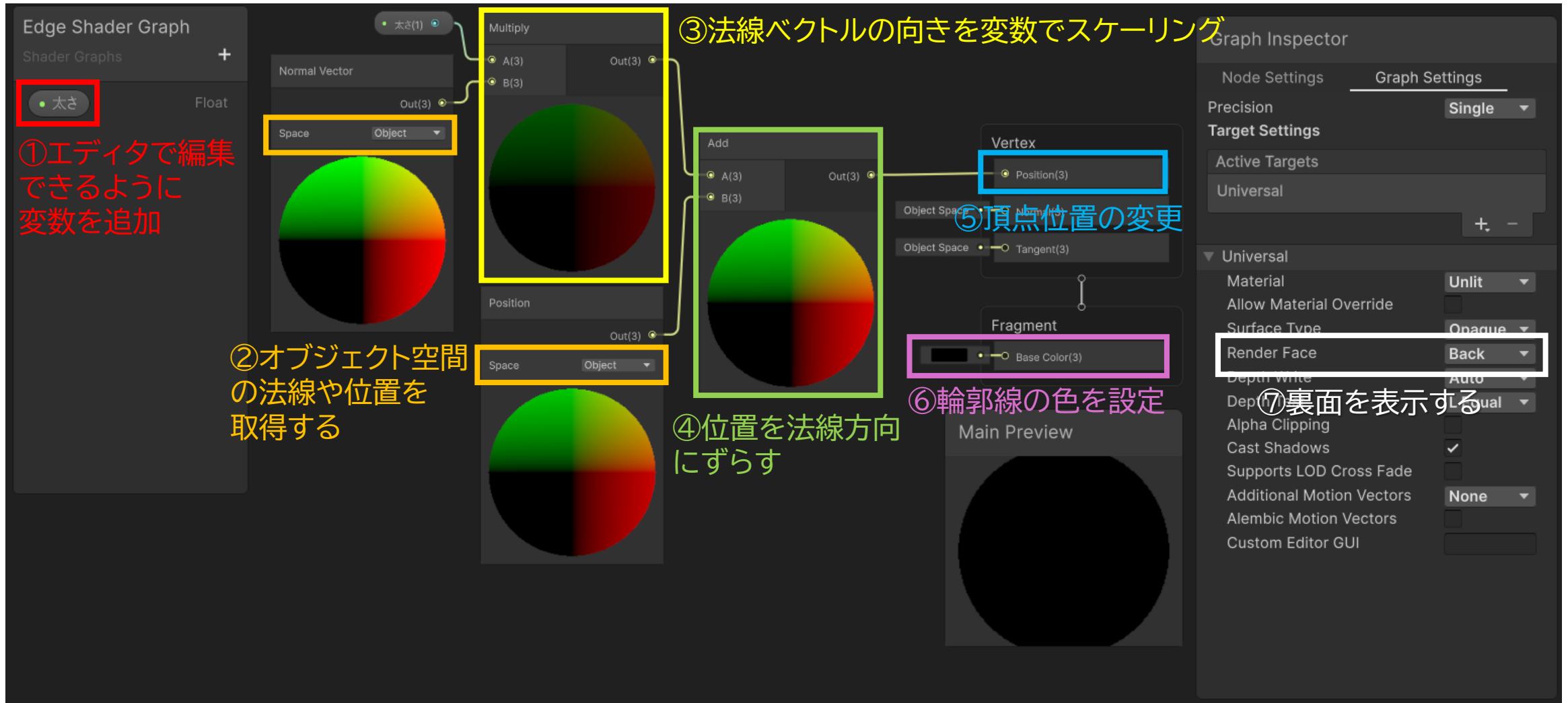
- ShaderGraphではマルチパスが使えないなので、マテリアルを複数つけてみる



続き

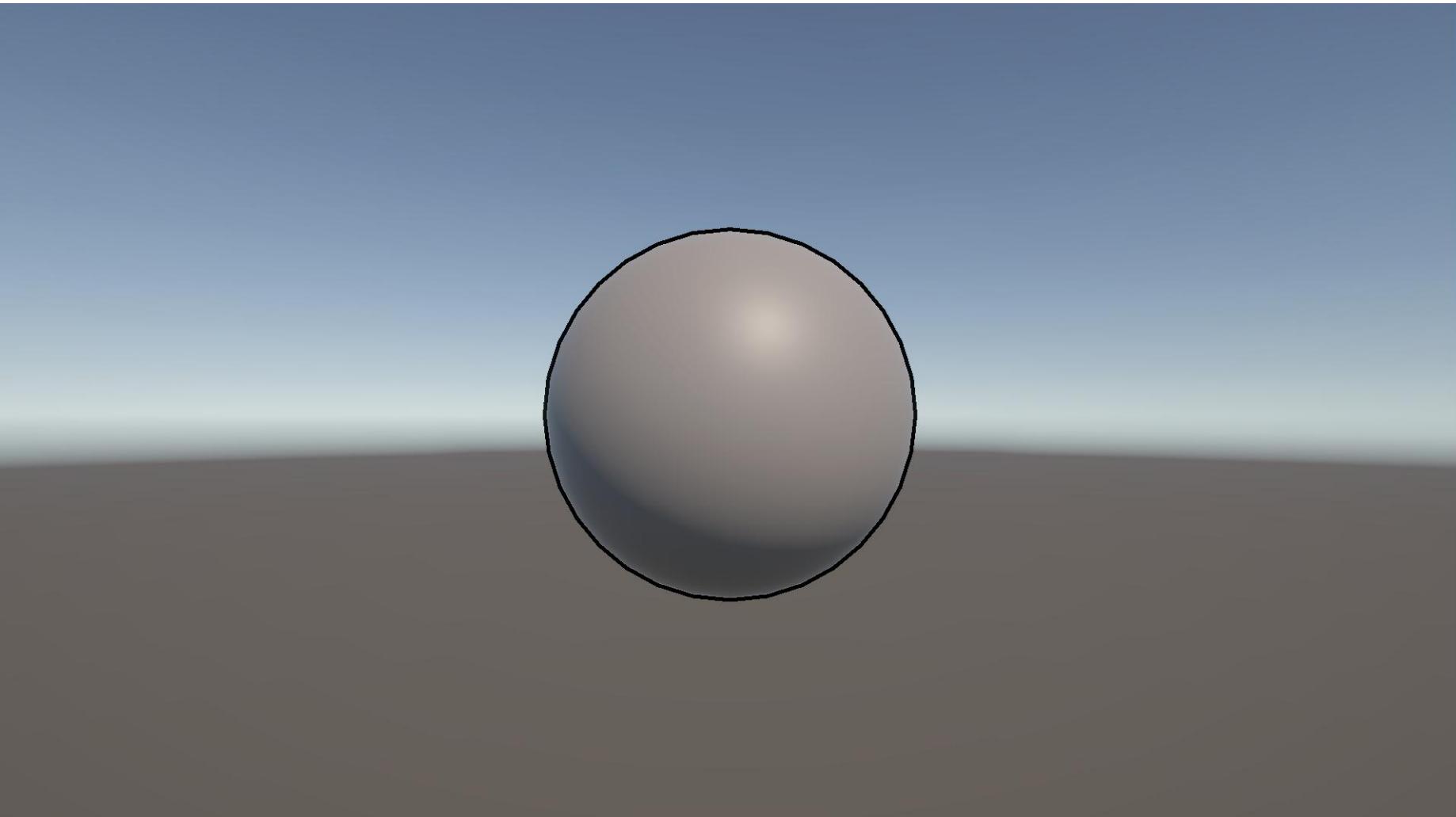


# シェーダグラフ



# やってみよう

- ・幅:0.02の例

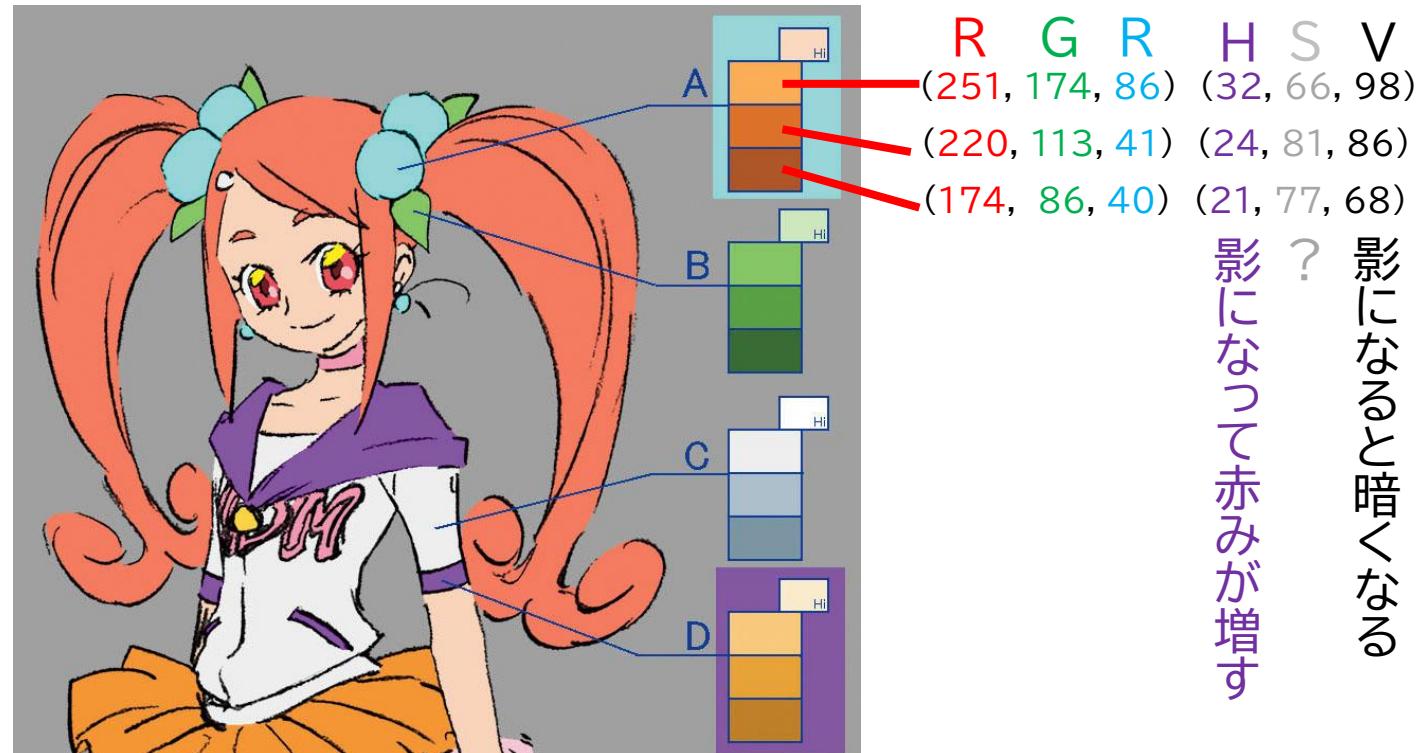


# アジェンダ

- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トーンシェーディング
- NPR: 疑似ハーフトーン

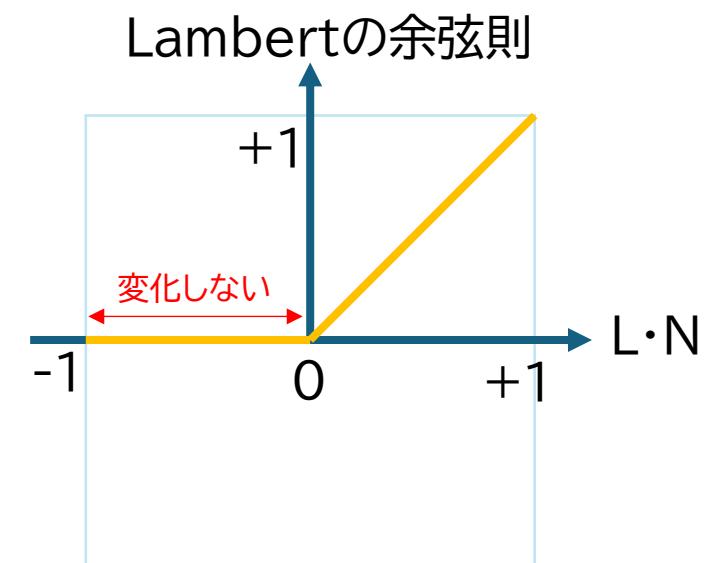
# 段階的陰影

- ・アニメーションでは色彩設計の専門職の人が色を決める
  - ・影になっている部分は単に暗いわけではない
    - ・色相も変わっている
  - ・プログラマーとしては、アーティストが制御できることが大切
    - ・制御しやすい方が良い



# Ramp Shading

- Lambertの余弦則は負の値を0にクランプ
  - ・負の領域では法線の情報が活用できていないのでもったいない
  - ・負の領域にも色をつけるのはどうか？
- 余弦則の値の色をグラデーションで指定することでより豊かな表現を実現する

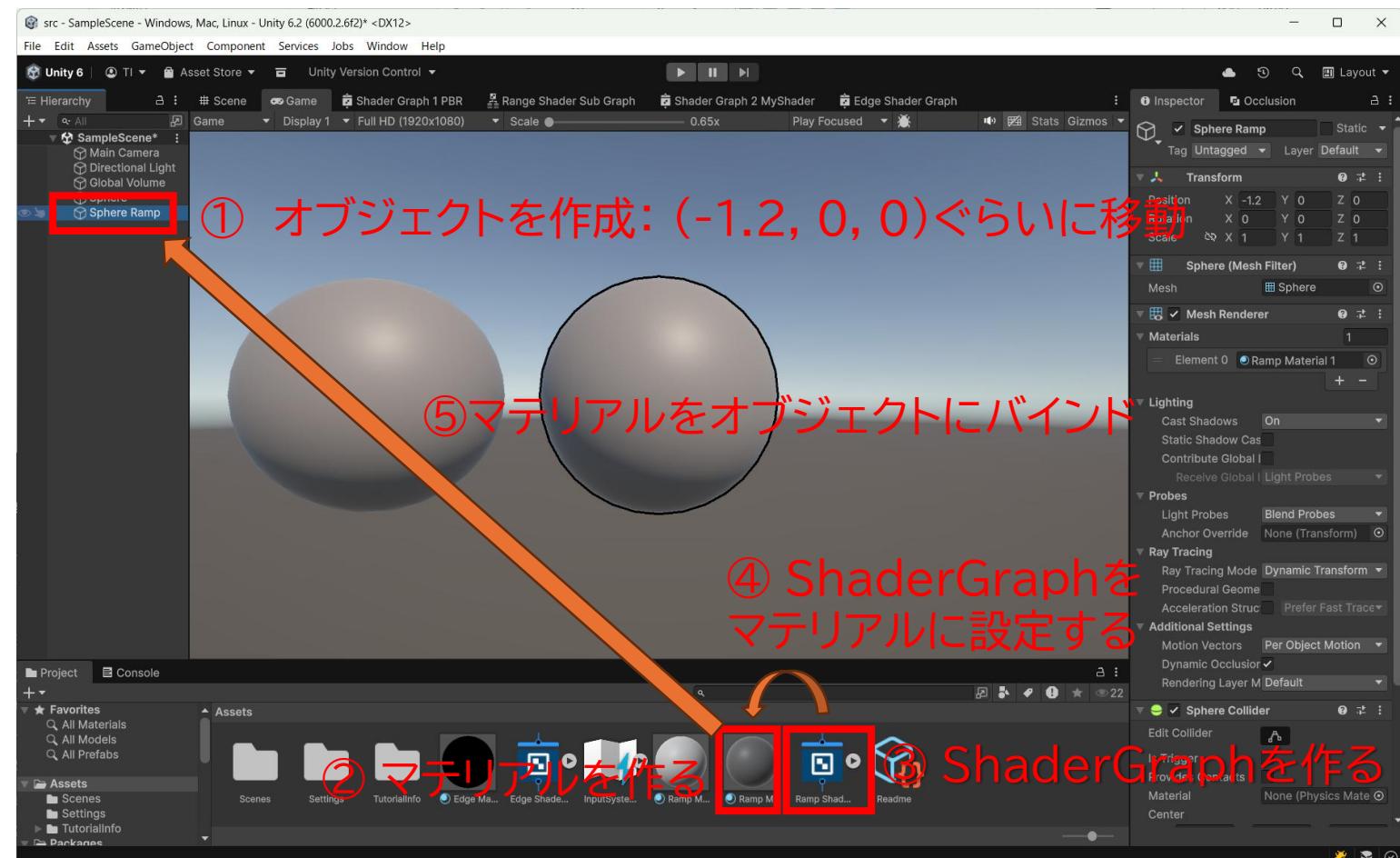


# アジェンダ

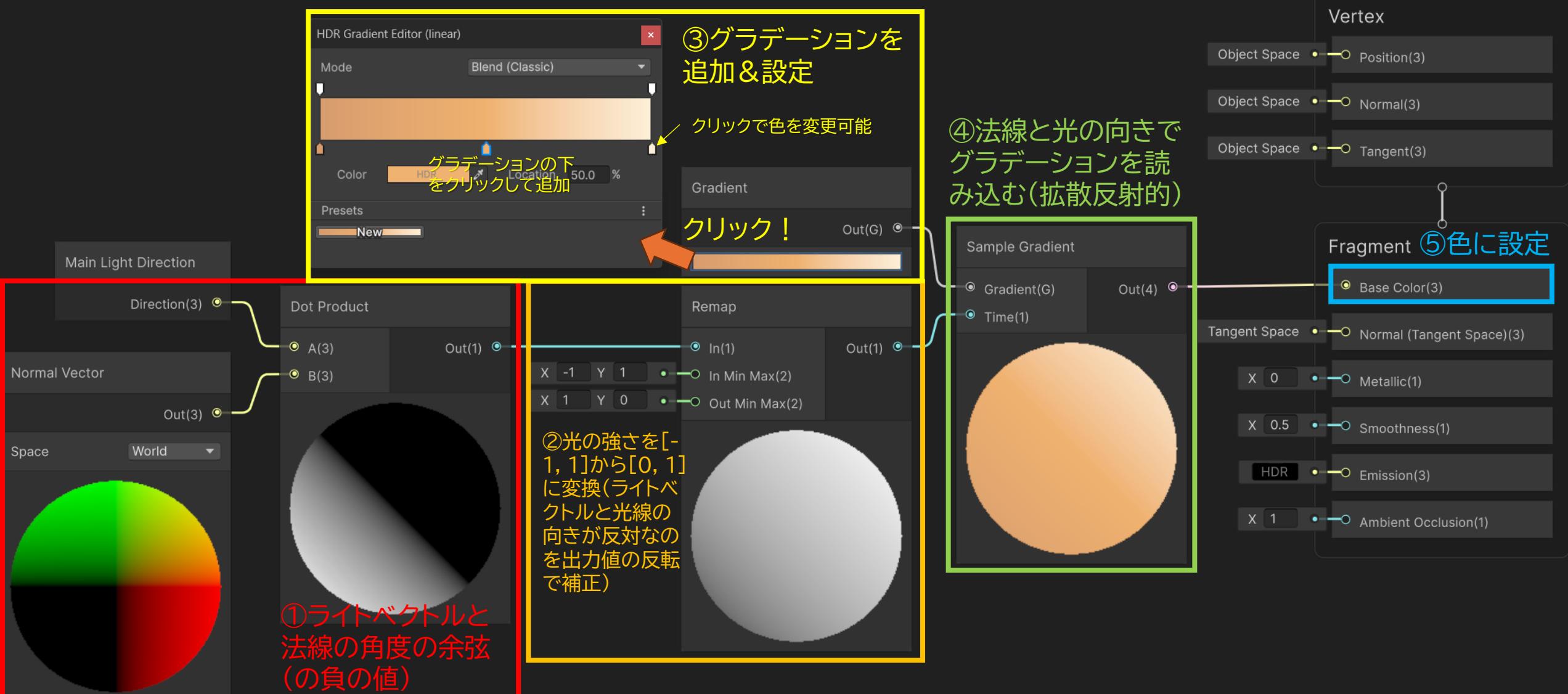
- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トーンシェーディング
- NPR: 疑似ハーフトーン

# 準備

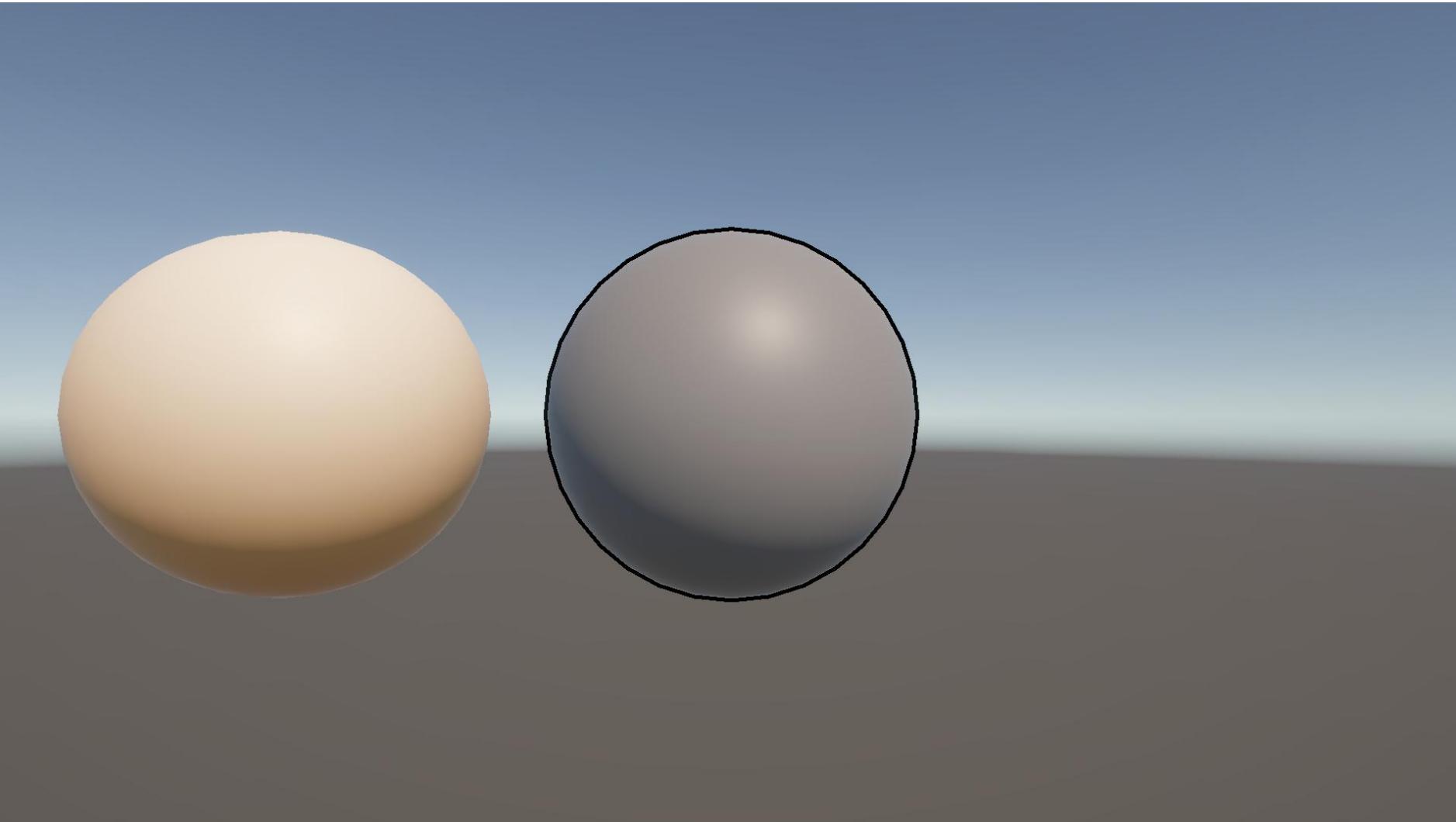
1. オブジェクトを追加
2. マテリアルを追加
3. Lit Shader Graph  
を追加
4. Shader Graphを  
マテリアルに設定
5. マテリアルをオブ  
ジェクトにバインド



# Shader Graph

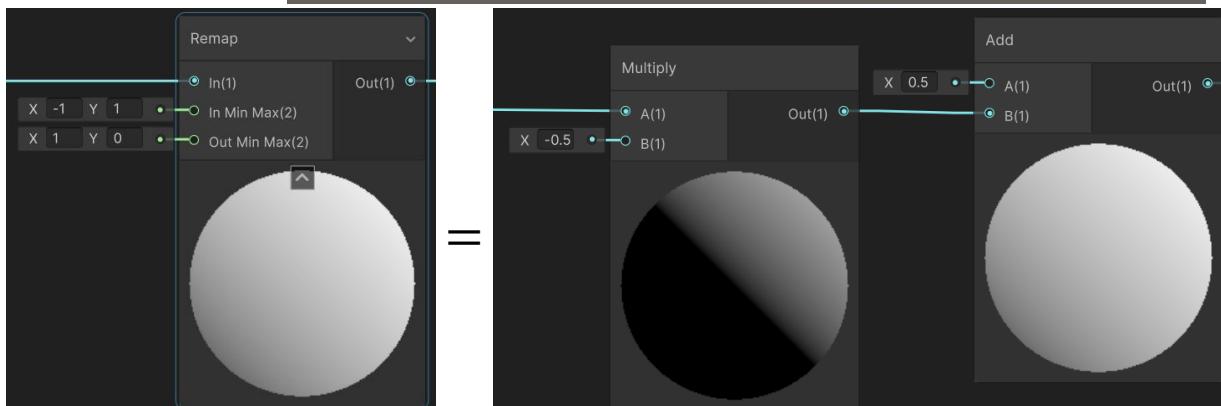
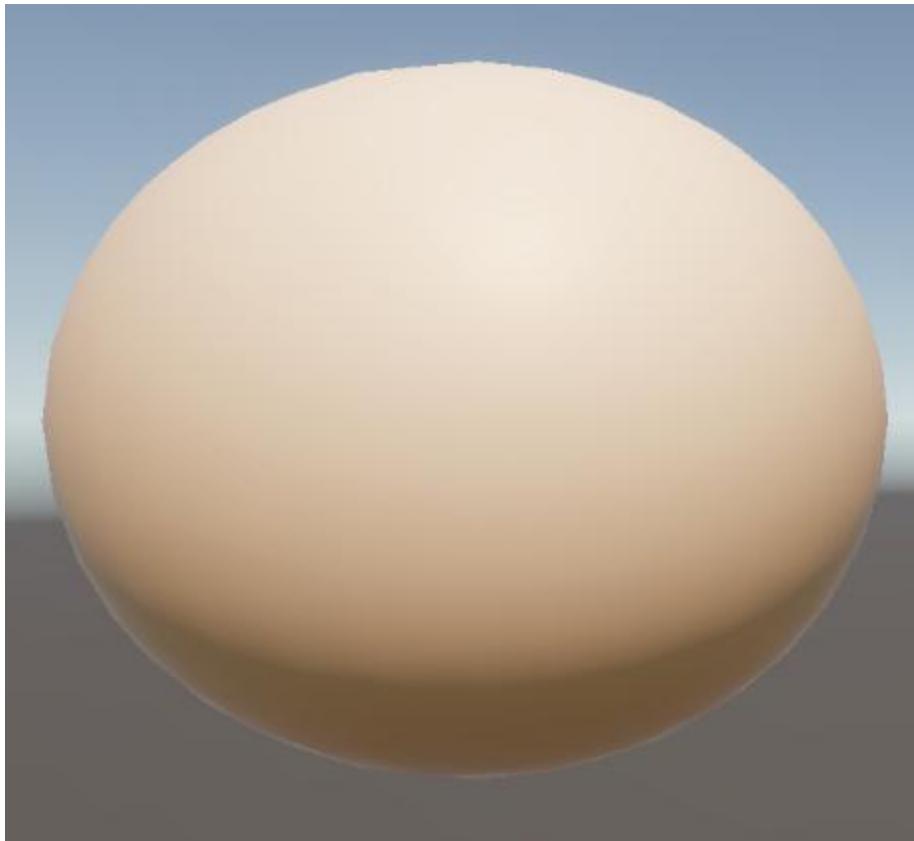


やってみよう

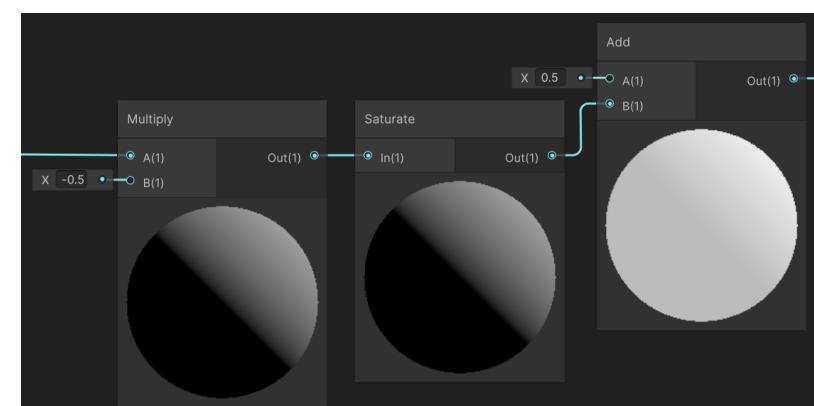
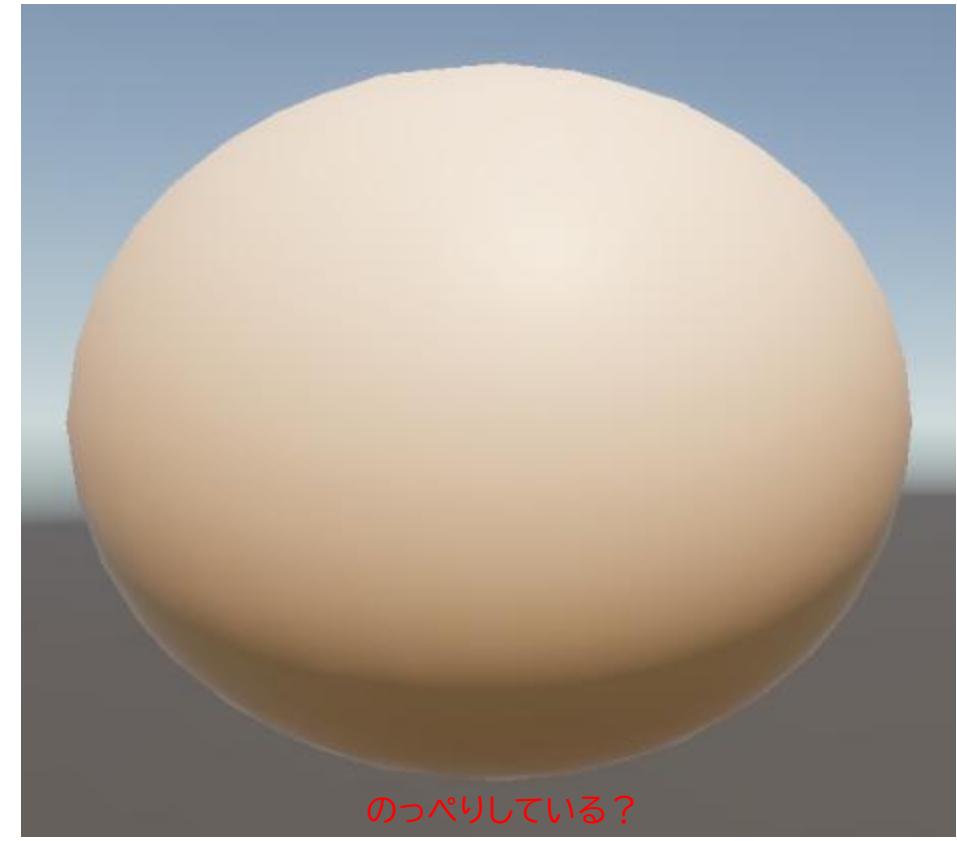


# 比較

Ramp Shading



Lambertの余弦則での変換(負の値をクランプ)



# アジェンダ

- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トーンシェーディング
- NPR: 疑似ハーフトーン

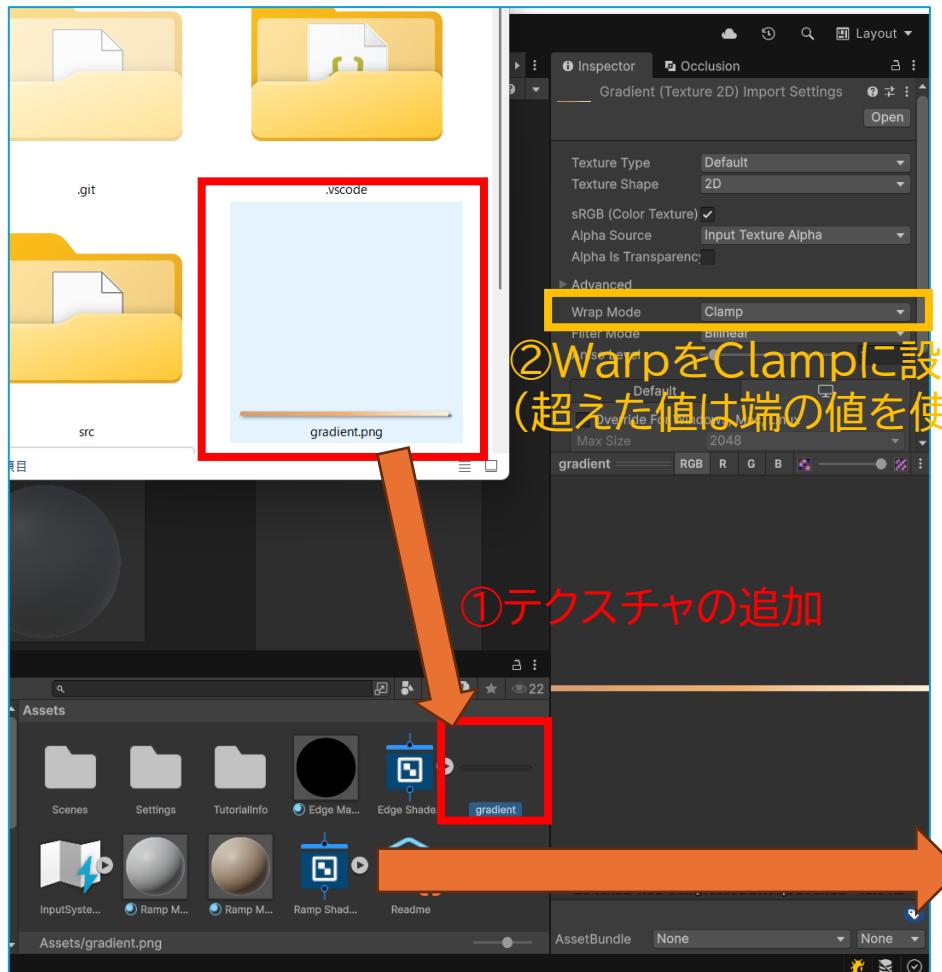
# テクスチャによる表現

- ・グラデーションをデザイナーが画像ファイルとして作成し、テクスチャとして読み込む
  - ・長所
    - ・デザイナーが独立して制御できる
      - PhotoShopなど使い慣れたツールを使え
      - Unityでのコンフリクトが起きない
    - ・速度が速い？
      - グラデーションが複雑になれば、テクスチャ読み込みの方が早い(はず)
  - ・短所
    - ・グラデーションの変化が解像度による(一般的に決める方法はない)
      - 疑似輪郭が出やすい
    - ・計算の方が補間の精度が高くなりやすい(高いテクスチャ解像度が必要)
    - ・メモリ使用量が多い(定数バッファと比べて)

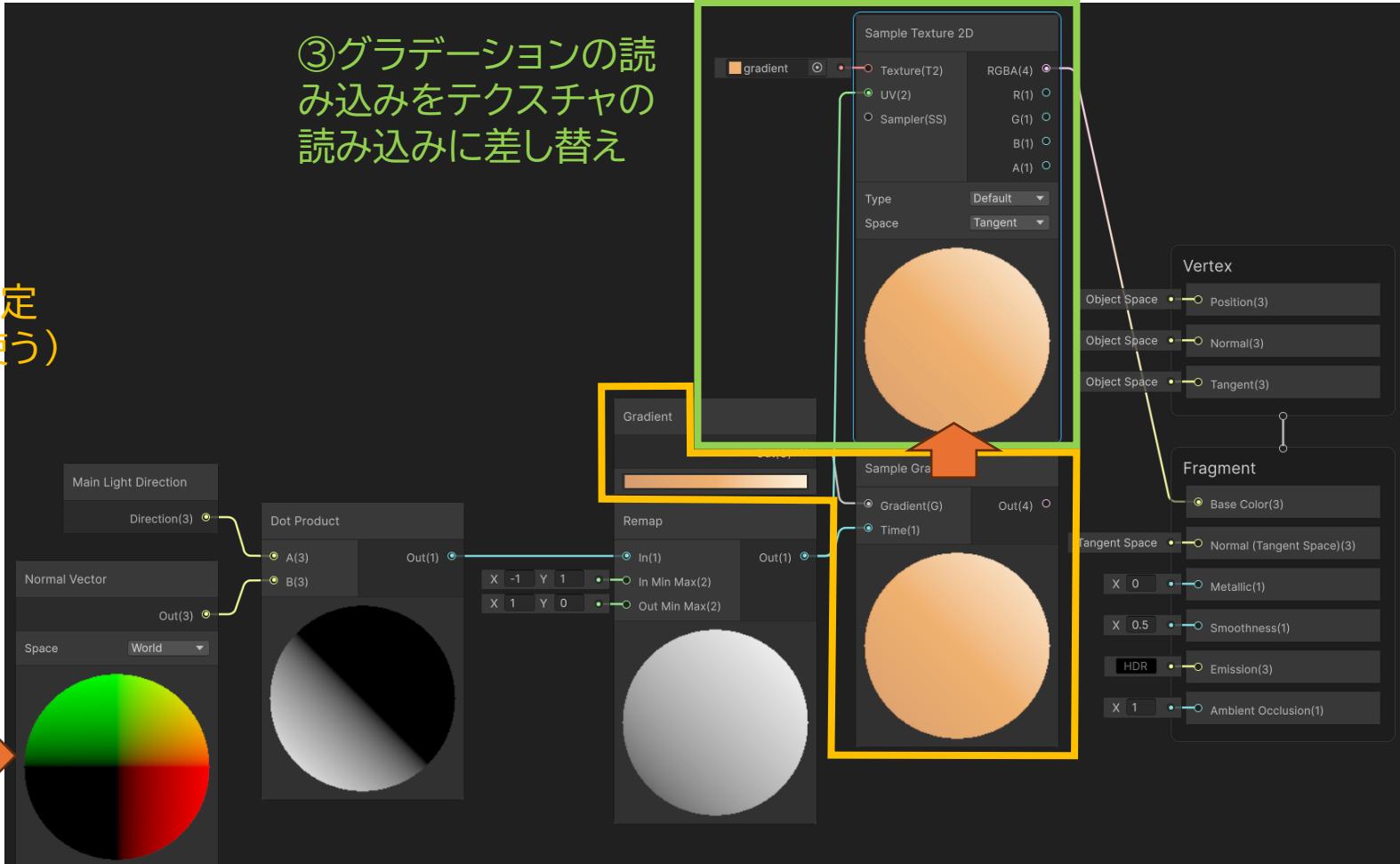


gradient.png

# 実装

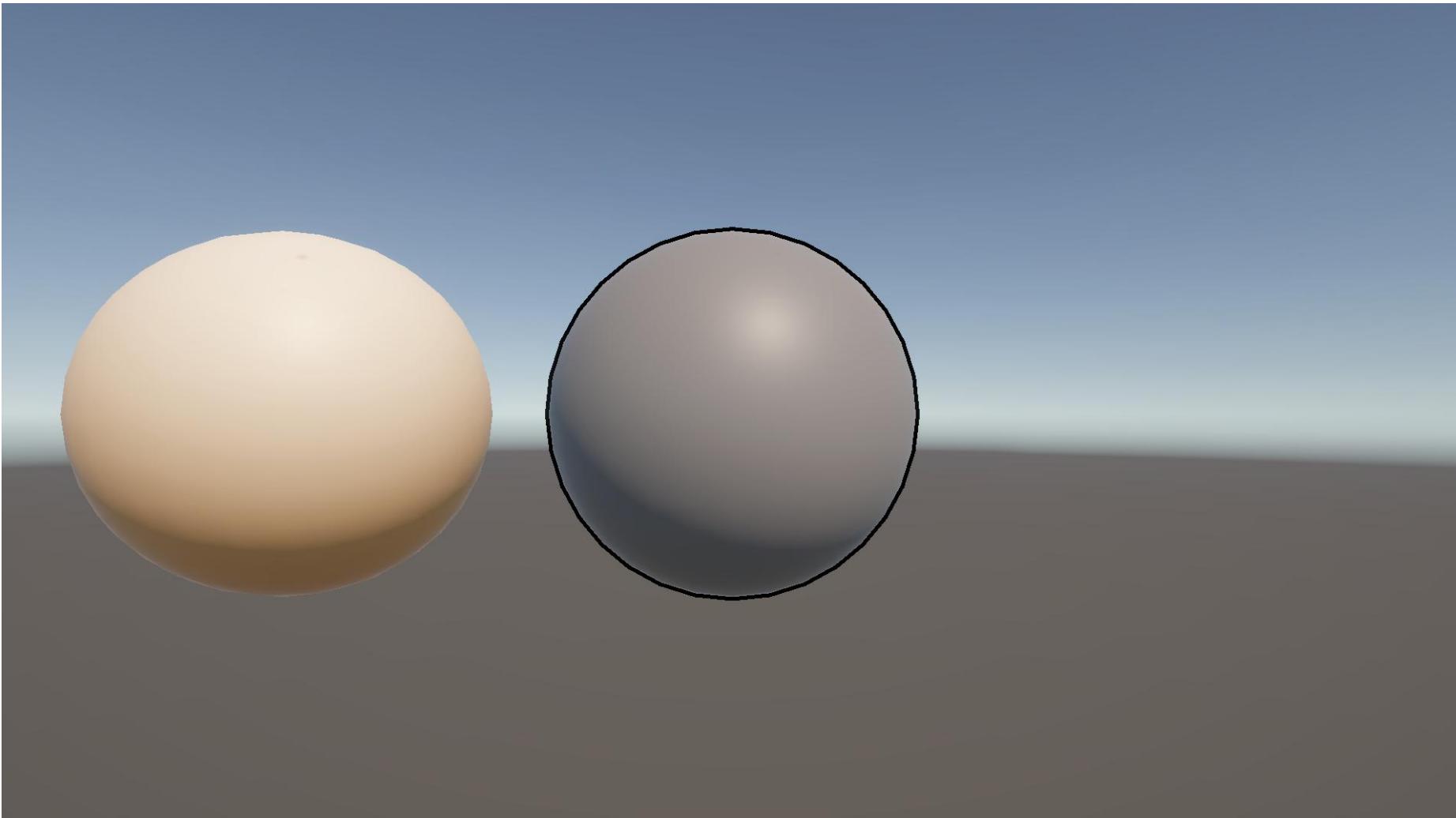


② Warp を Clamp に設定  
(超えた値は端の値を使う)



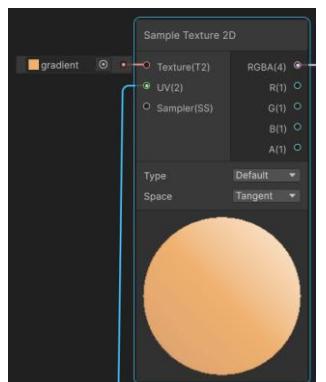
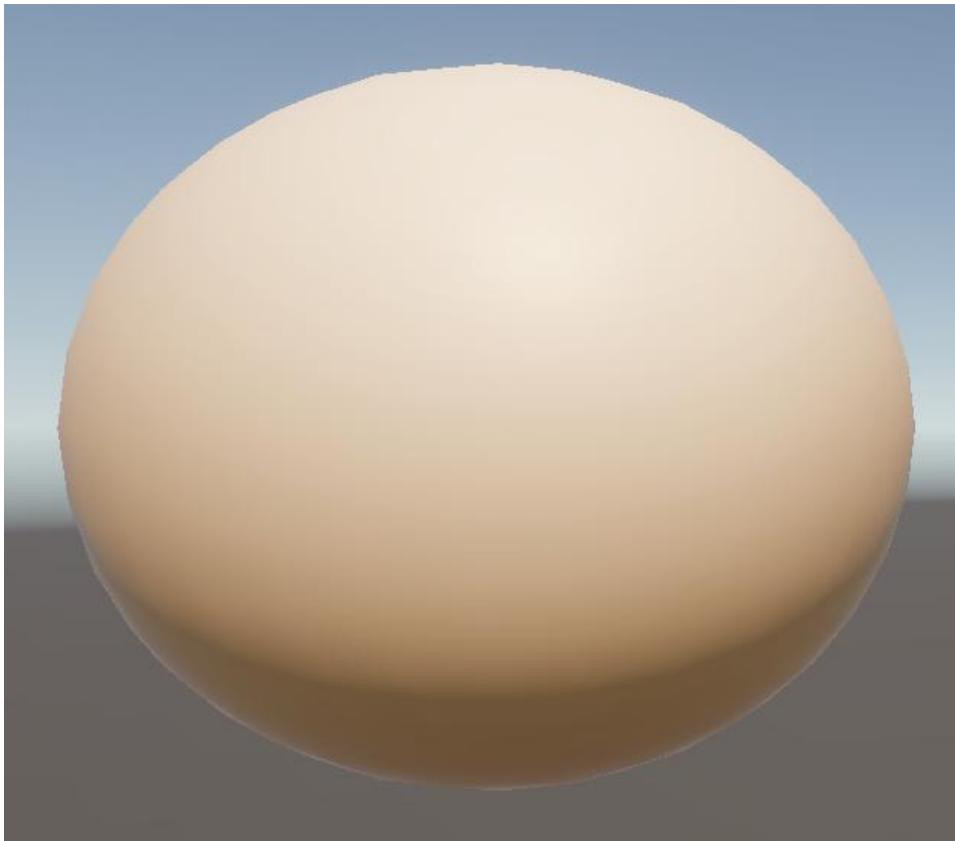
③グラデーションの読み込みをテクスチャの読み込みに差し替え

やってみよう



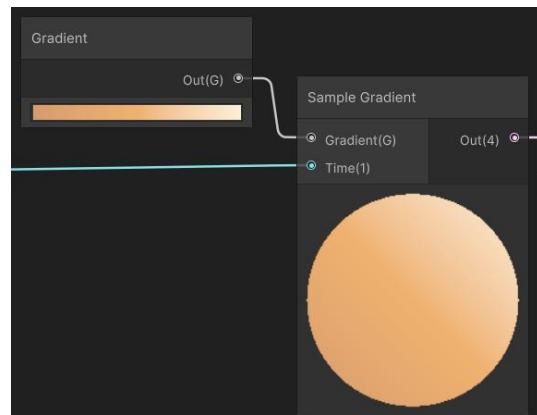
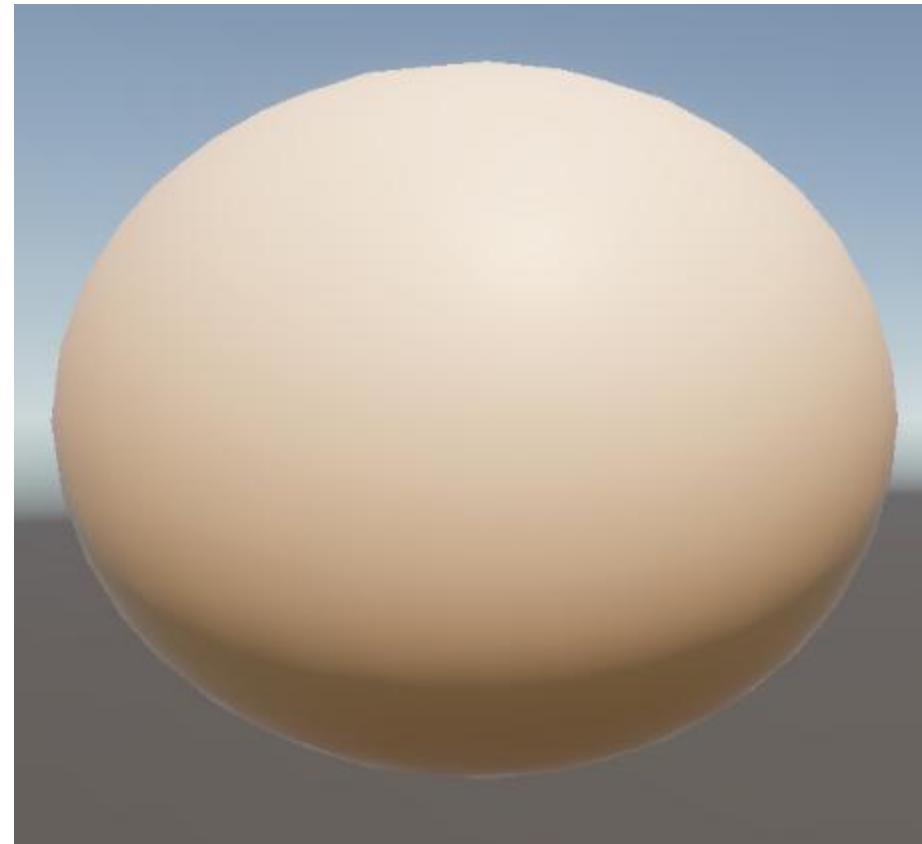
比較

Ramp Shading (texture)



1556px  
(大き目のサイズで値自体に意味はない)

Ramp Shading (Gradient node)



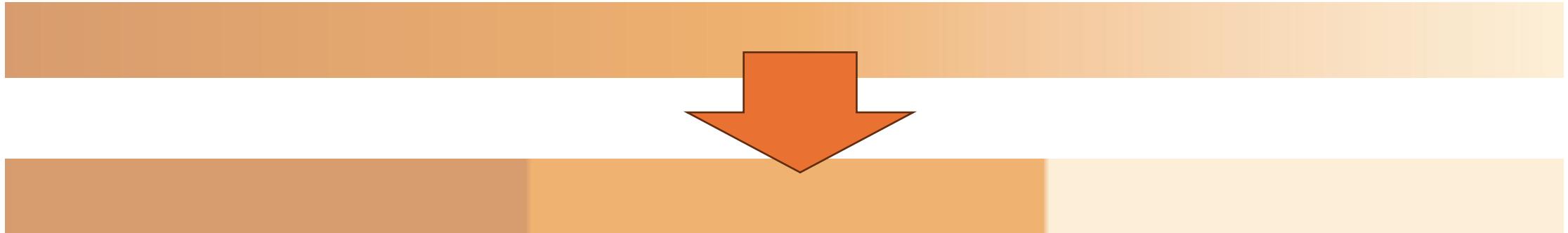
プログラムワークショップIV

# アジェンダ

- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トゥーンシェーディング
- NPR: 疑似ハーフトーン

# トーンシェーディング

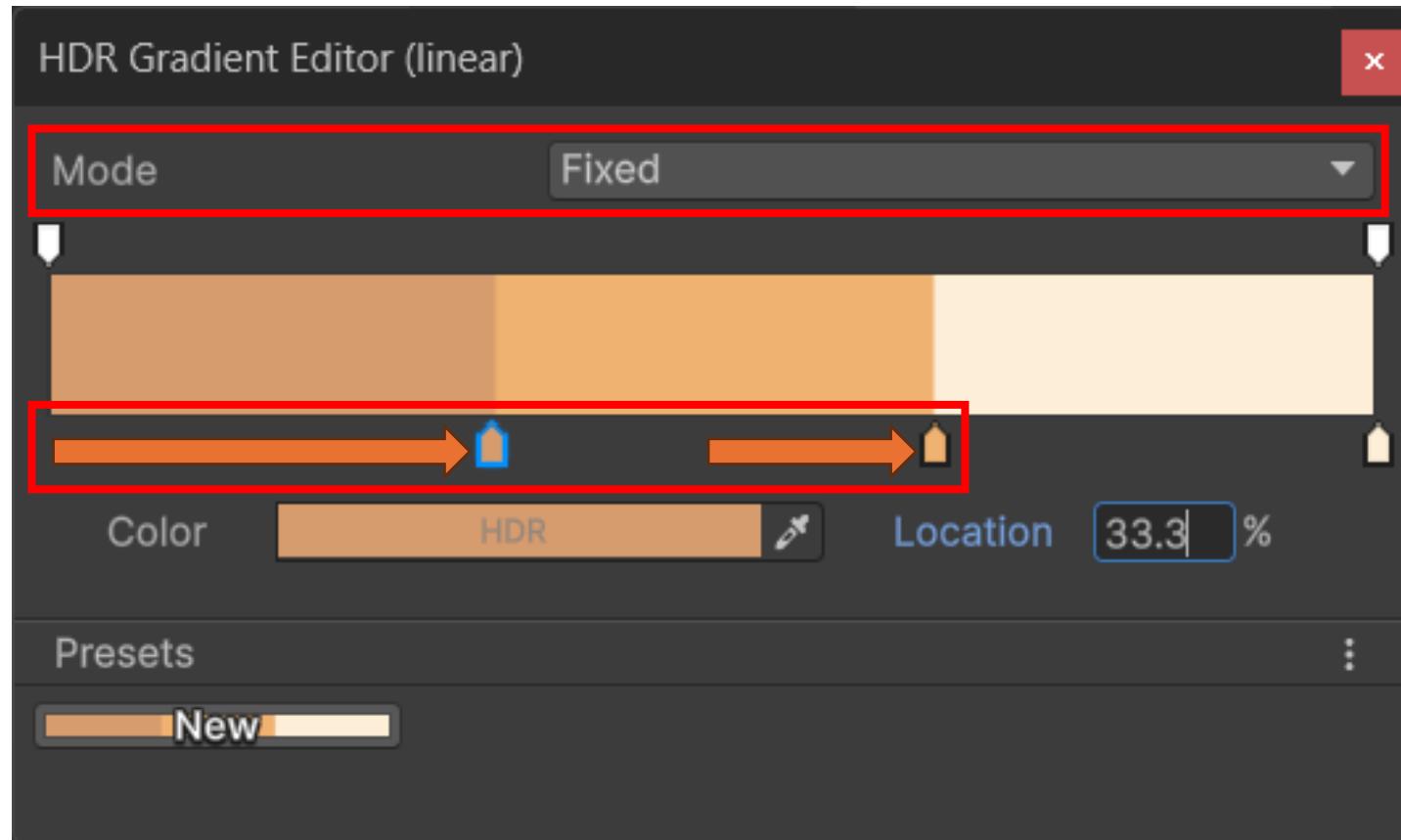
- ・色の変化をなだらかではなくて、段階的にする



# グラデーション設定

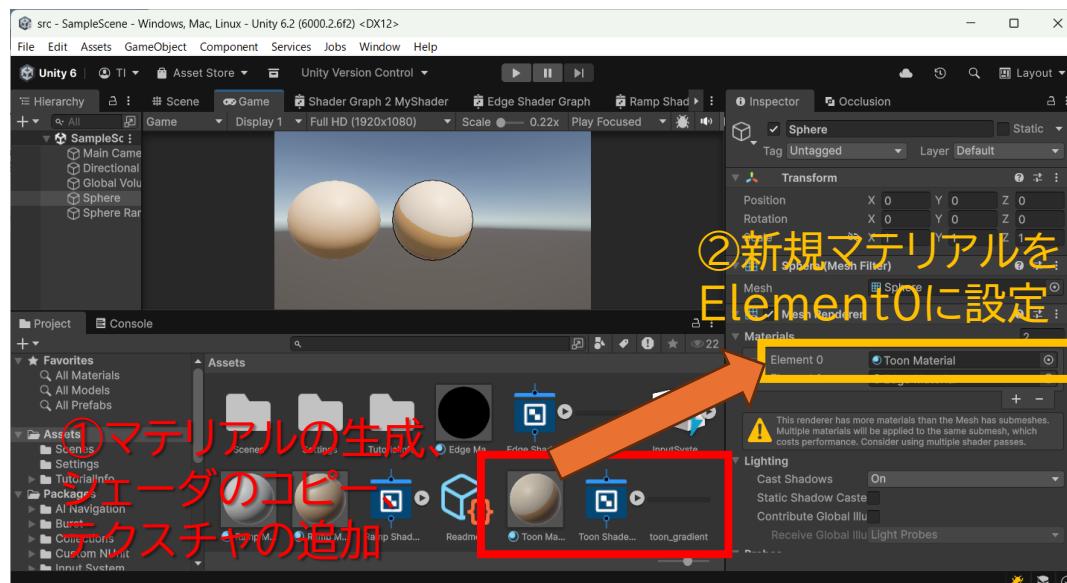
- ModeをFixedにすると段階的な変化に変わる

色の指定位置を変更

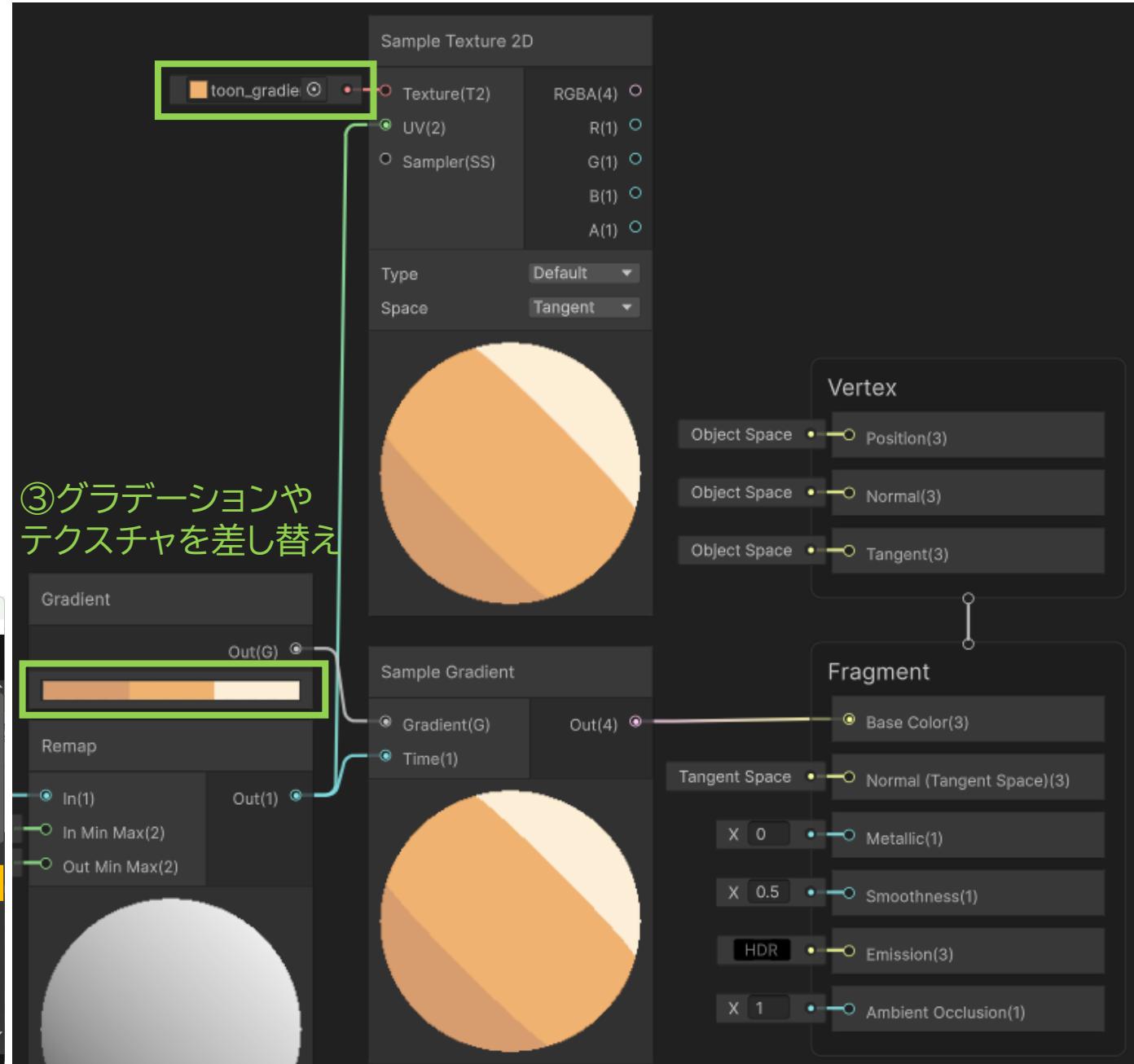


# 実装

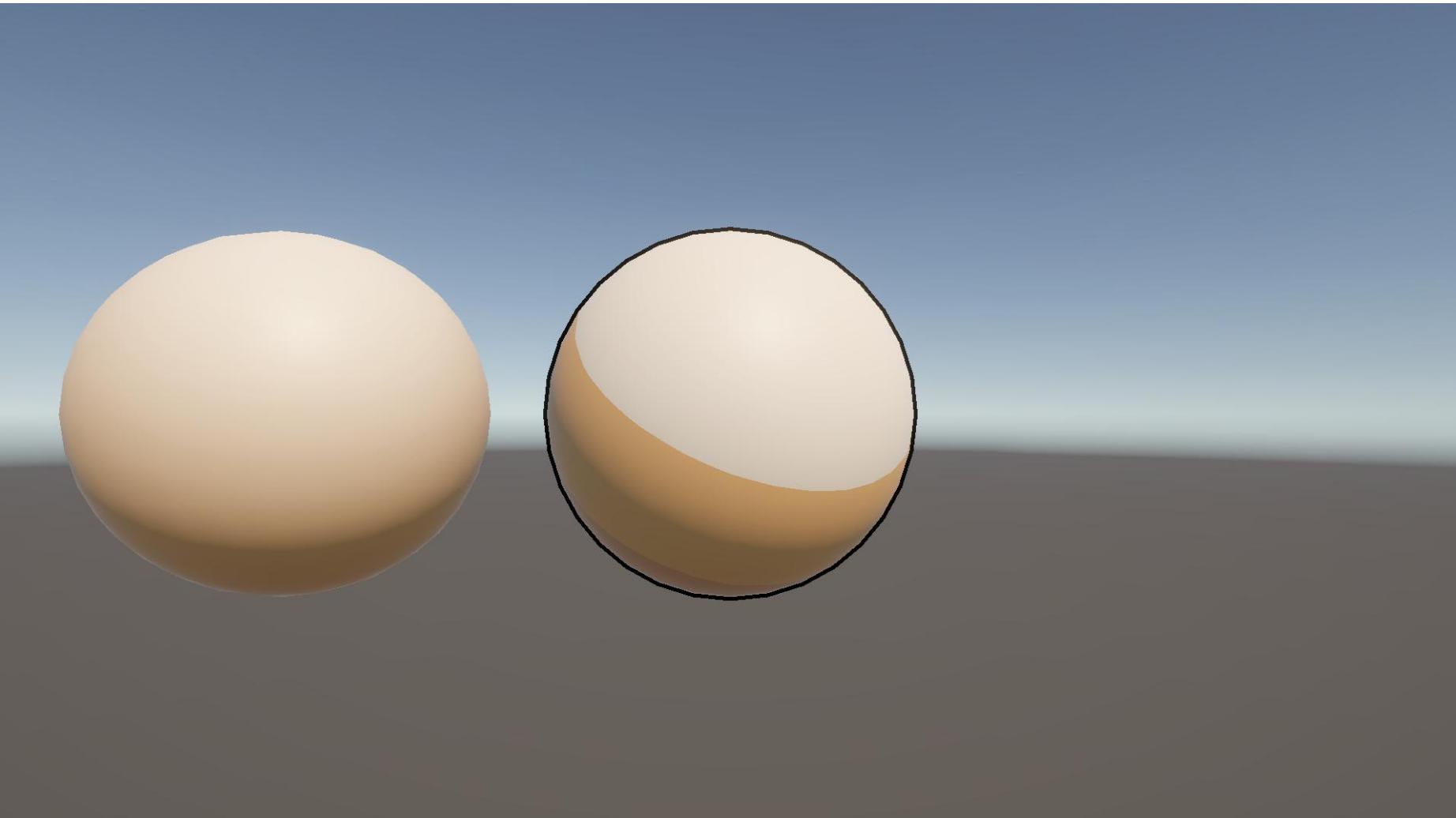
1. 新しくマテリアル、シェーダー、テクスチャ(段階的)を用意
  - ・シェーダーはコピー
2. マテリアルをSphereに設定
3. グラデーションやテクスチャを差し替え



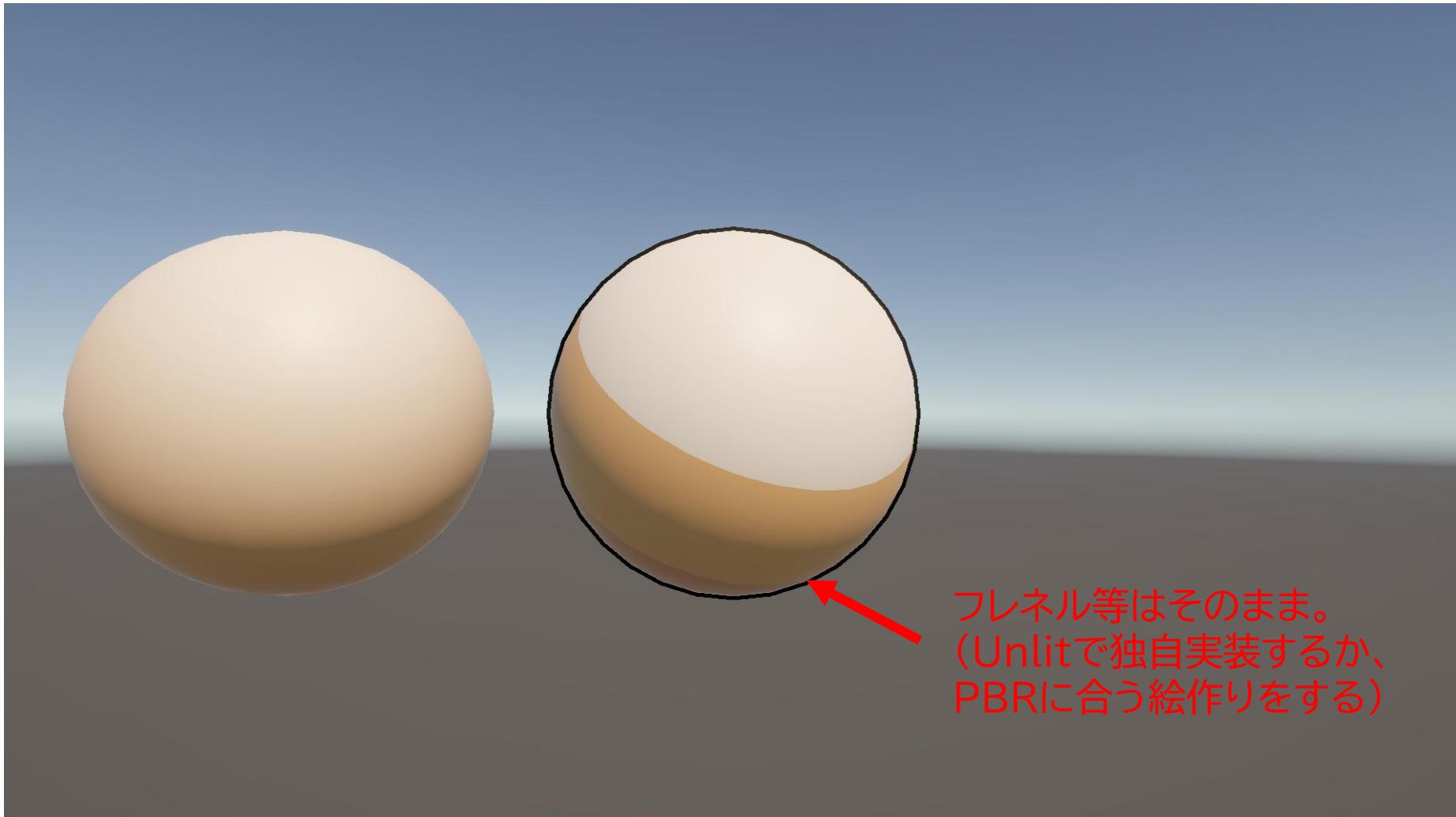
②新規マテリアルを  
Element0に設定



やってみよう



このツウーンはPBR的(ベースカラーのみ)

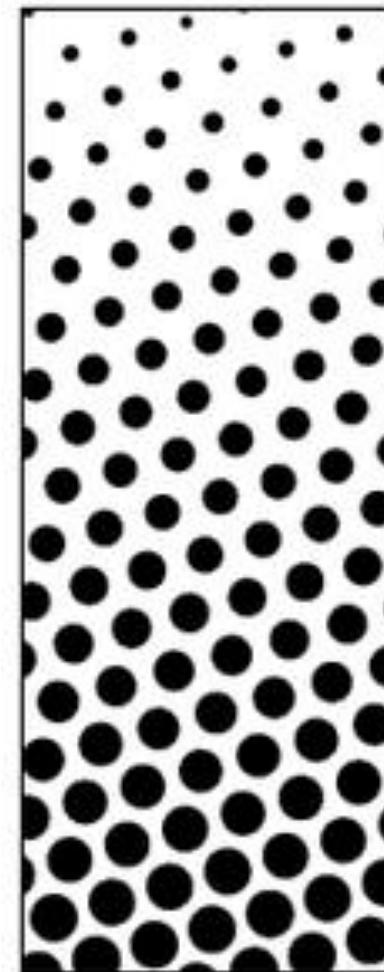
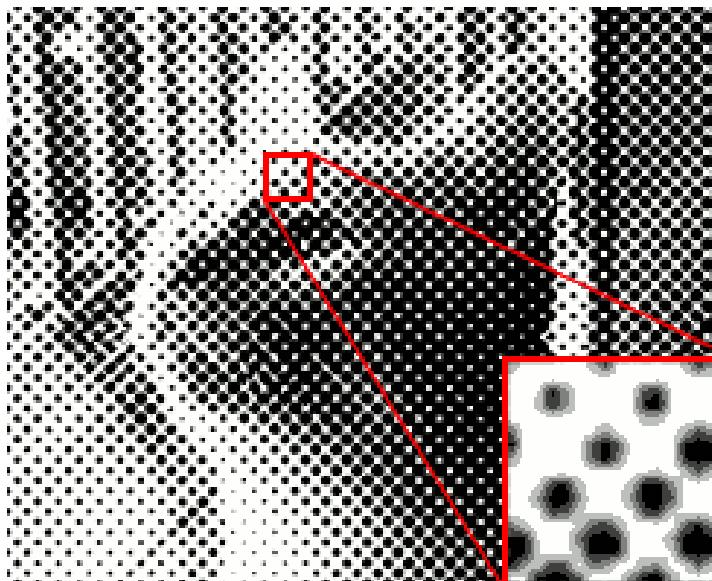


# アジェンダ

- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トーンシェーディング
- **NPR: 疑似ハーフトーン**

# NPR:さらなる発想ができるようになろう

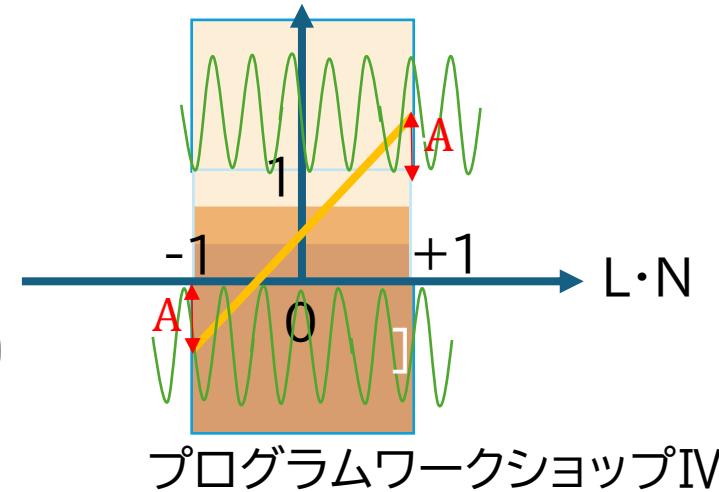
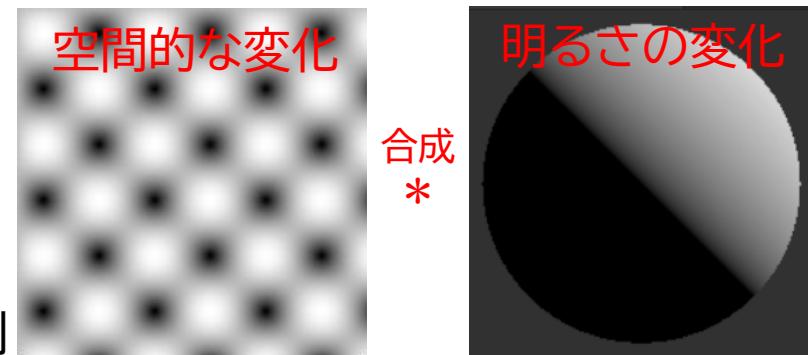
- ・ハーフトーン(網点)
  - ・白黒を黒いインクの大きさで表現
    - ・白:黒いインクを置かない
    - ・黒:黒いインクで埋められる



# 疑似ハーフトーン

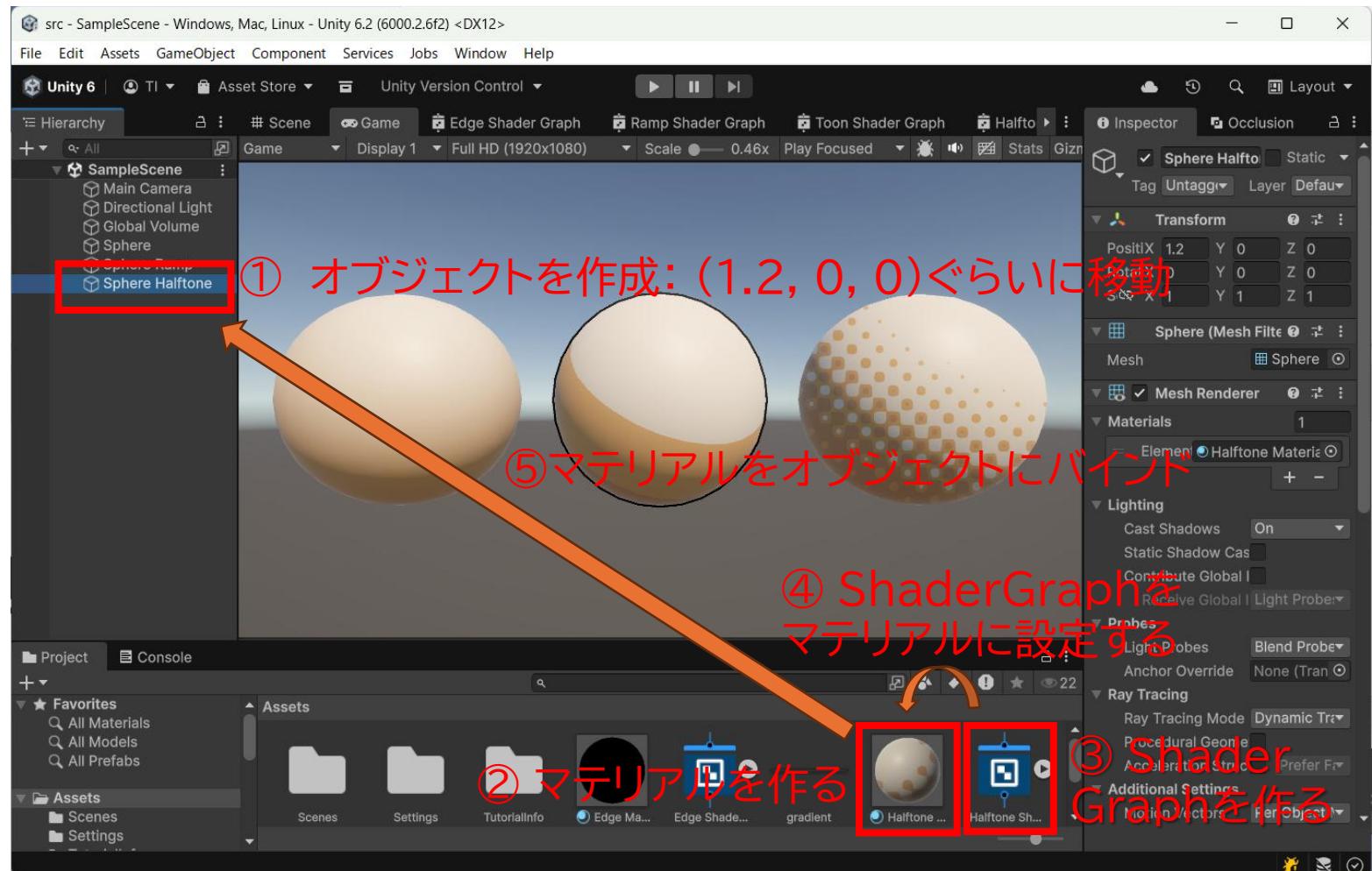
- 実際に黒い点を置くこともできるが、ピクセルシェーダでの処理は大変なので、疑似的な方法を考える
  - ピクセル単位でそれっぽく判断
    - 空間的な縞模様を考える
      - 最も明るい場所と暗い場所では影響を受けない
      - 間の明るさでは縞模様の影響を受けるLambertの余弦則
  - 空間的に縞模様の変化を考える
    - rampに入れる値について
      - $L \cdot N = +1$ : 縞模様で減ったとしても1
      - $L \cdot N = -1$ : 縞模様で増やしたとしても0(以下)

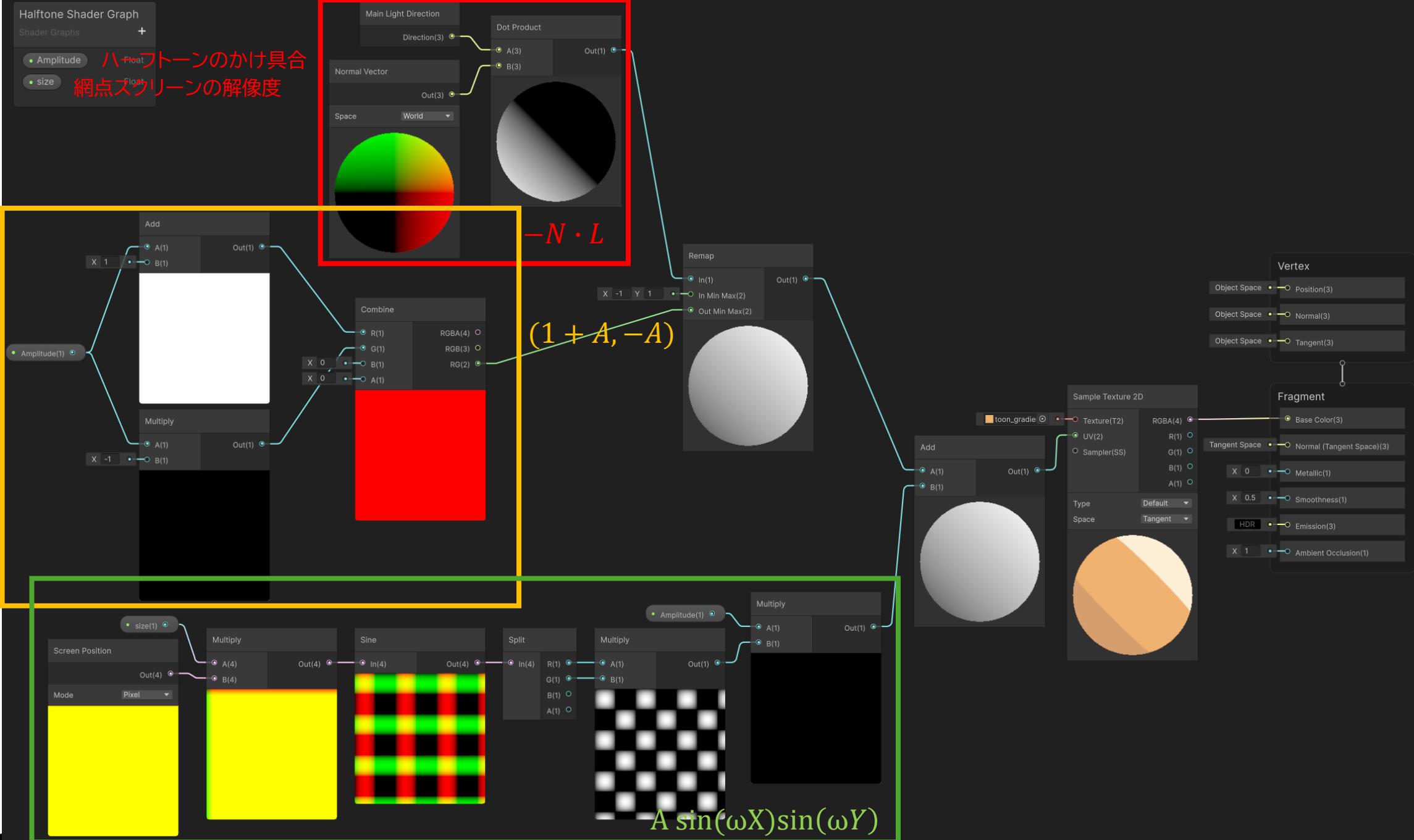
そのような例  $I = Ramp\left(\left(\frac{1}{2} + A\right)(N \cdot L) + \frac{1}{2} + A \sin(\omega X)\sin(\omega Y)\right)$



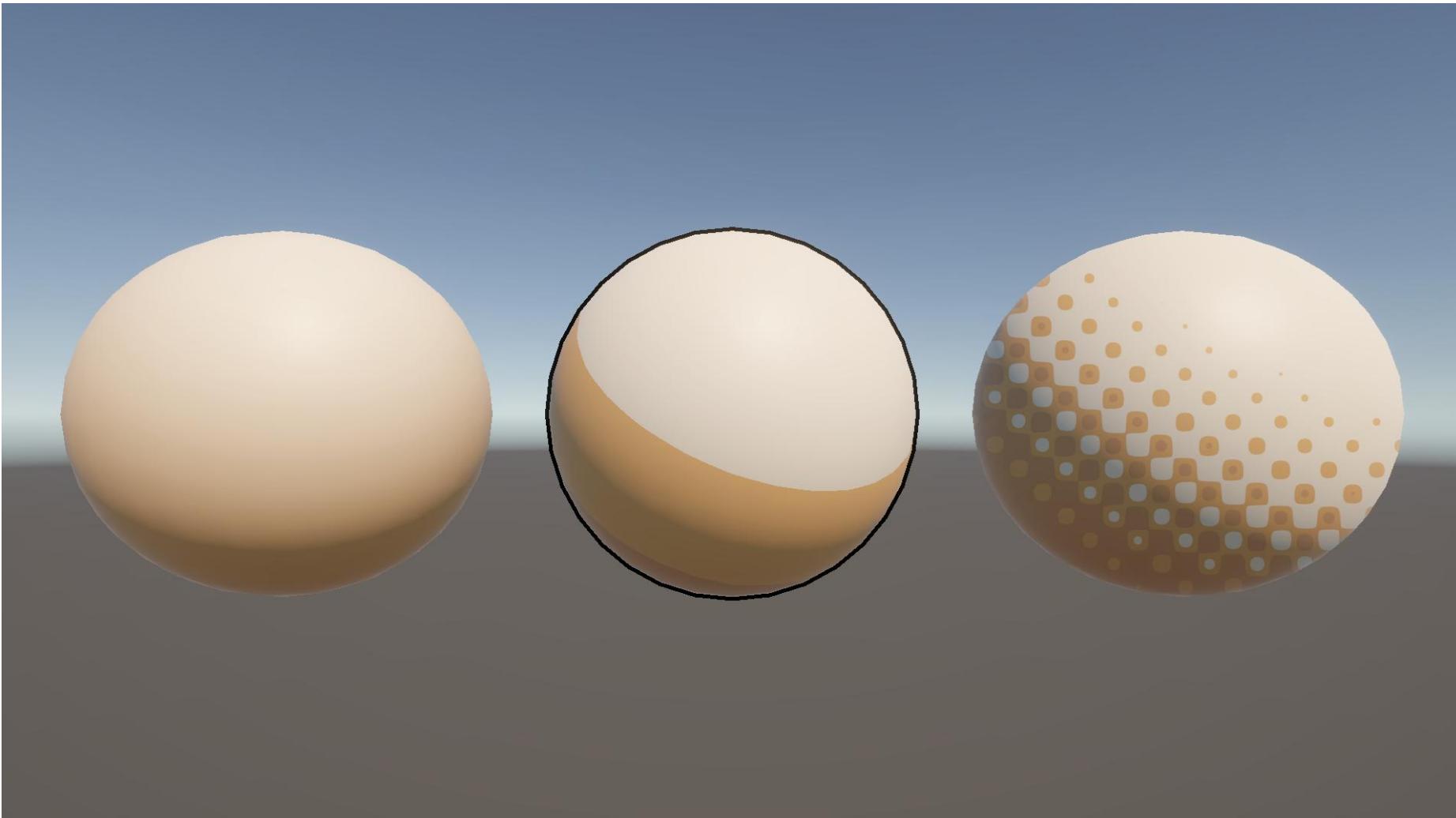
# 準備

1. オブジェクトを追加
2. マテリアルを追加
3. Lit Shader Graph  
をトゥーンから複製
4. Shader Graphを  
マテリアルに設定
5. マテリアルをオブ  
ジェクトに BIND

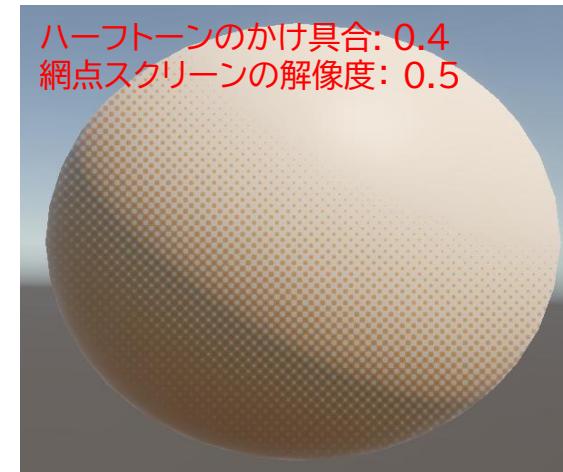
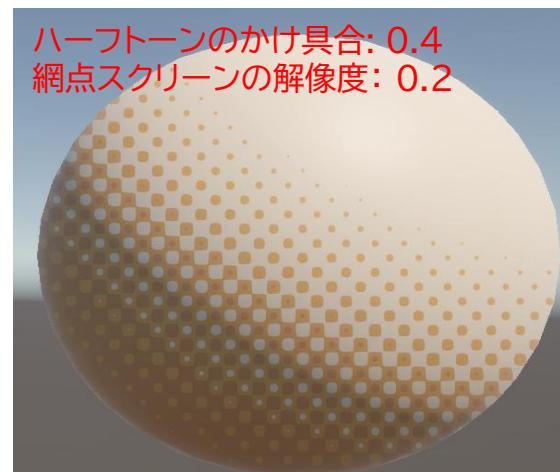
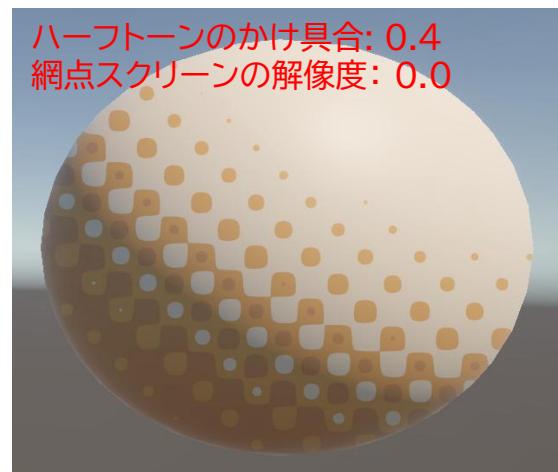
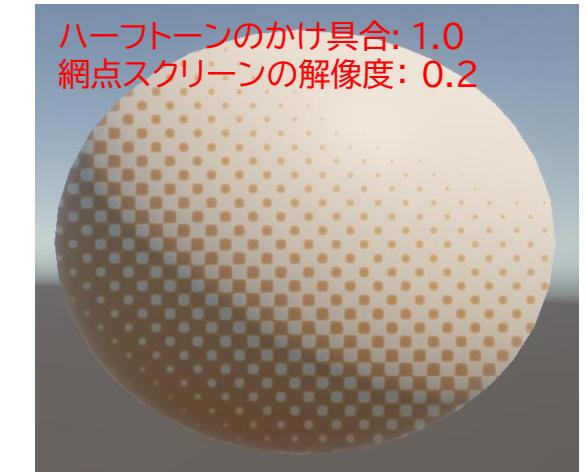
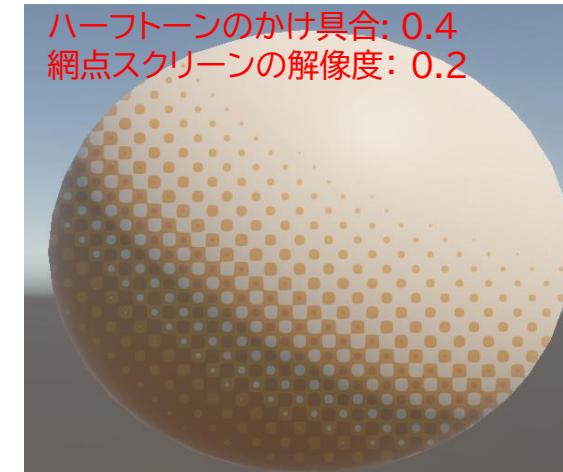
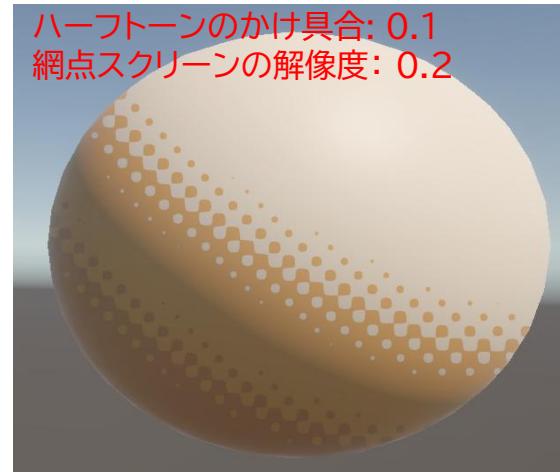
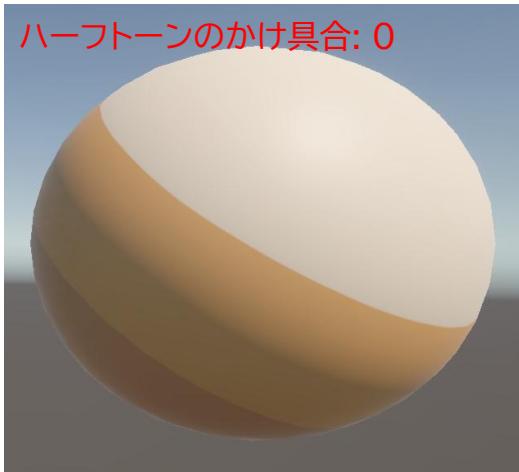




やってみよう



# パラメータ変化



# まとめ

- トゥーンレンダリングとは
- 輪郭
  - 概要
  - 背面法による輪郭生成
    - ソフトエッジのポリゴン形状の輪郭
- 段階的陰影
  - ランプシェーディング
    - グラデーション
    - テクスチャ
  - トゥーンシェーディング
    - 段階的な色分け
- NPR: 疑似ハーフトーン
  - ドットを使ったトゥーンの表現