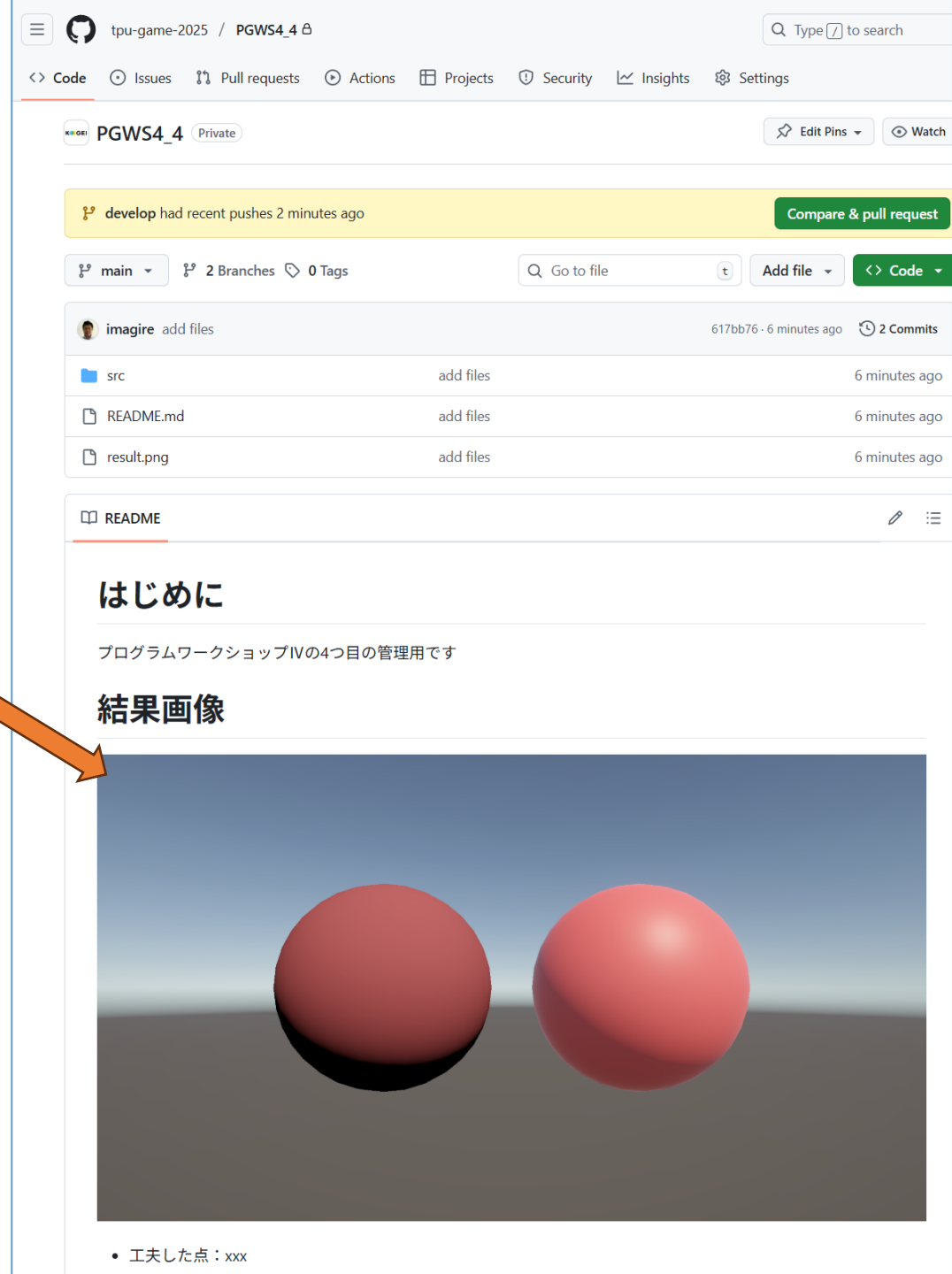
シェーダグラフ

2025年度 プログラムワークショップⅣ (4)

本日の課題

- https://github.com/tpu-game-2025/PGWS4_4
  をforkして、「結果画像」に結果を貼ってください
  - まずは右図と同じものを作って下さい
  - 今回の範囲で何か試してください



シェーダグラフ入門

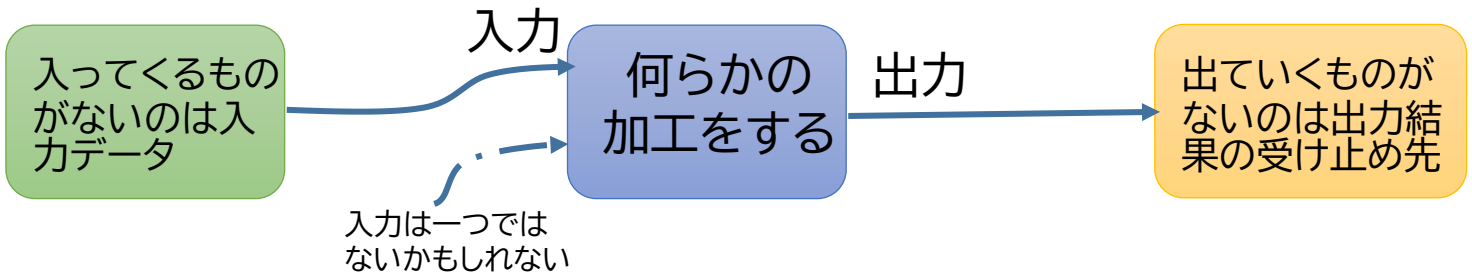
- シェーダグラフの追加
- 色を変える
- ノードの追加
- ノードの種類
- Lit Shader Graph

シェーダグラフ入門

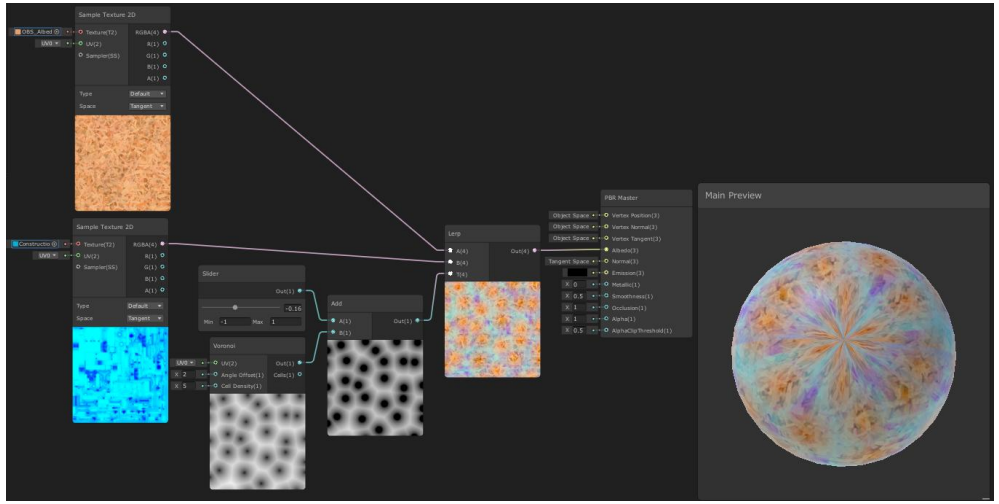
- シェーダグラフの追加
- 色を変える
- ノードの追加
- ノードの種類
- Lit Shader Graph

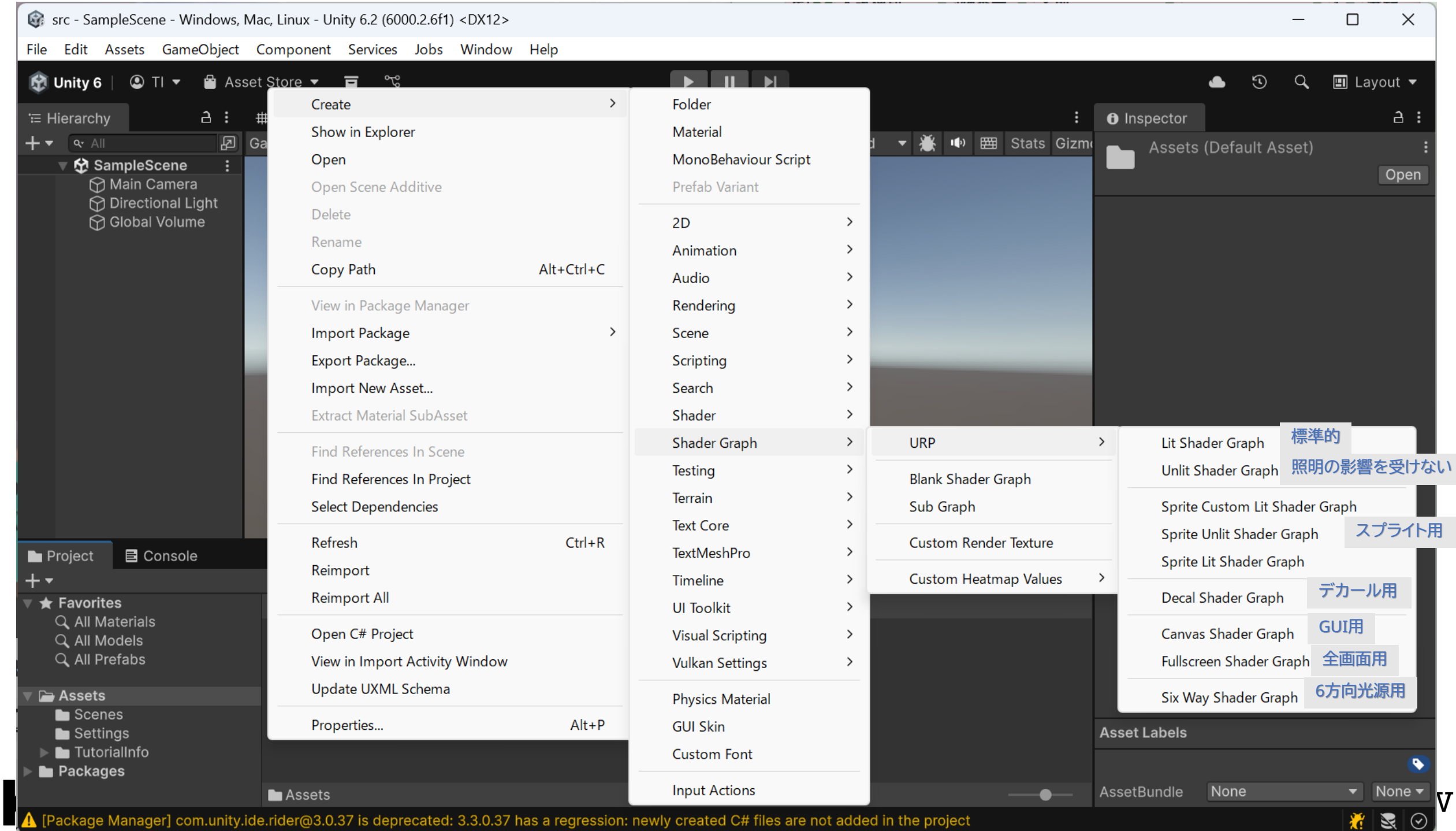
ShaderGraph

- ノードベースで シェーダーを作成出来る機能
  - ノードを線で繋いで結果を次の処理に回す

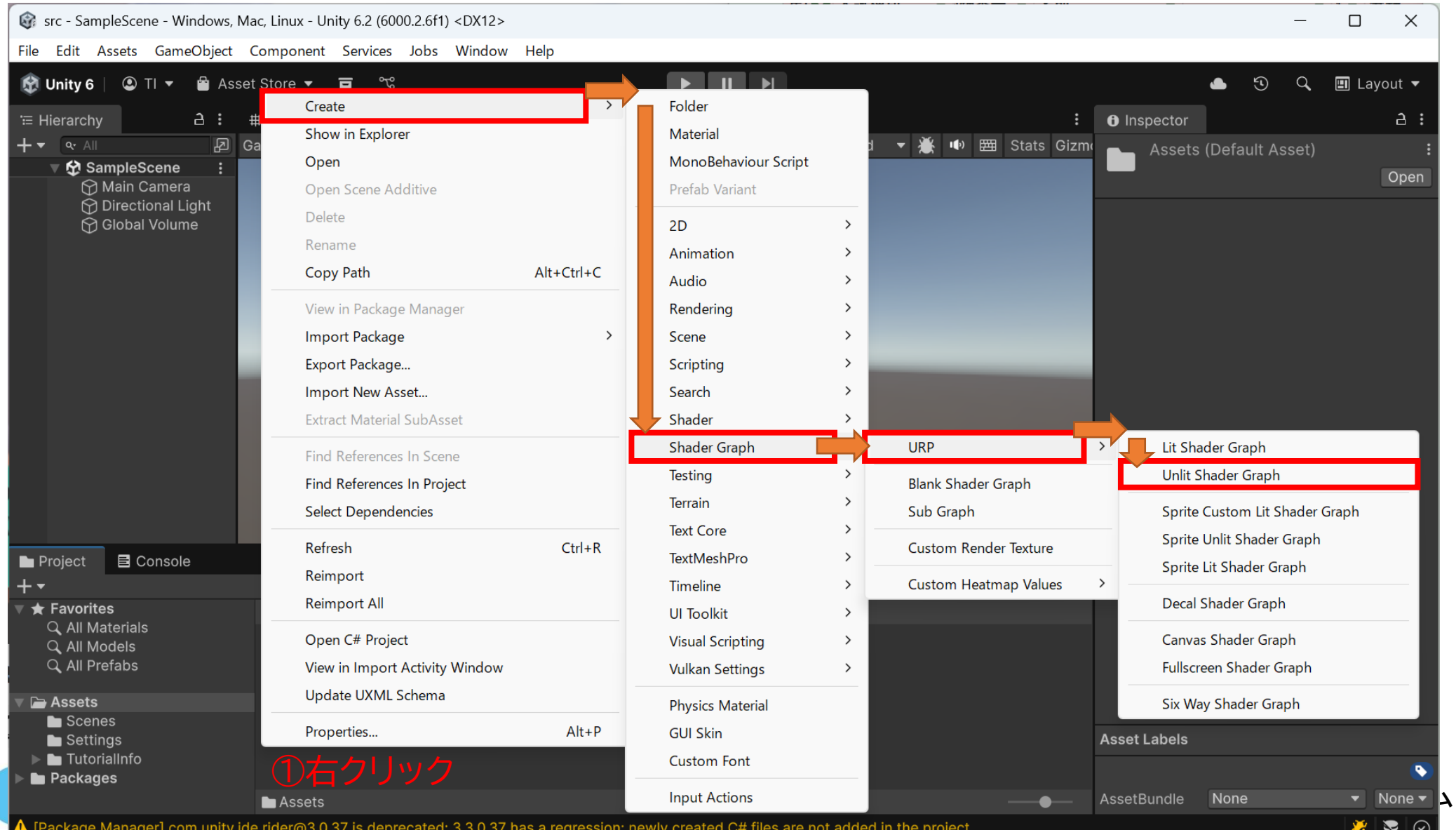


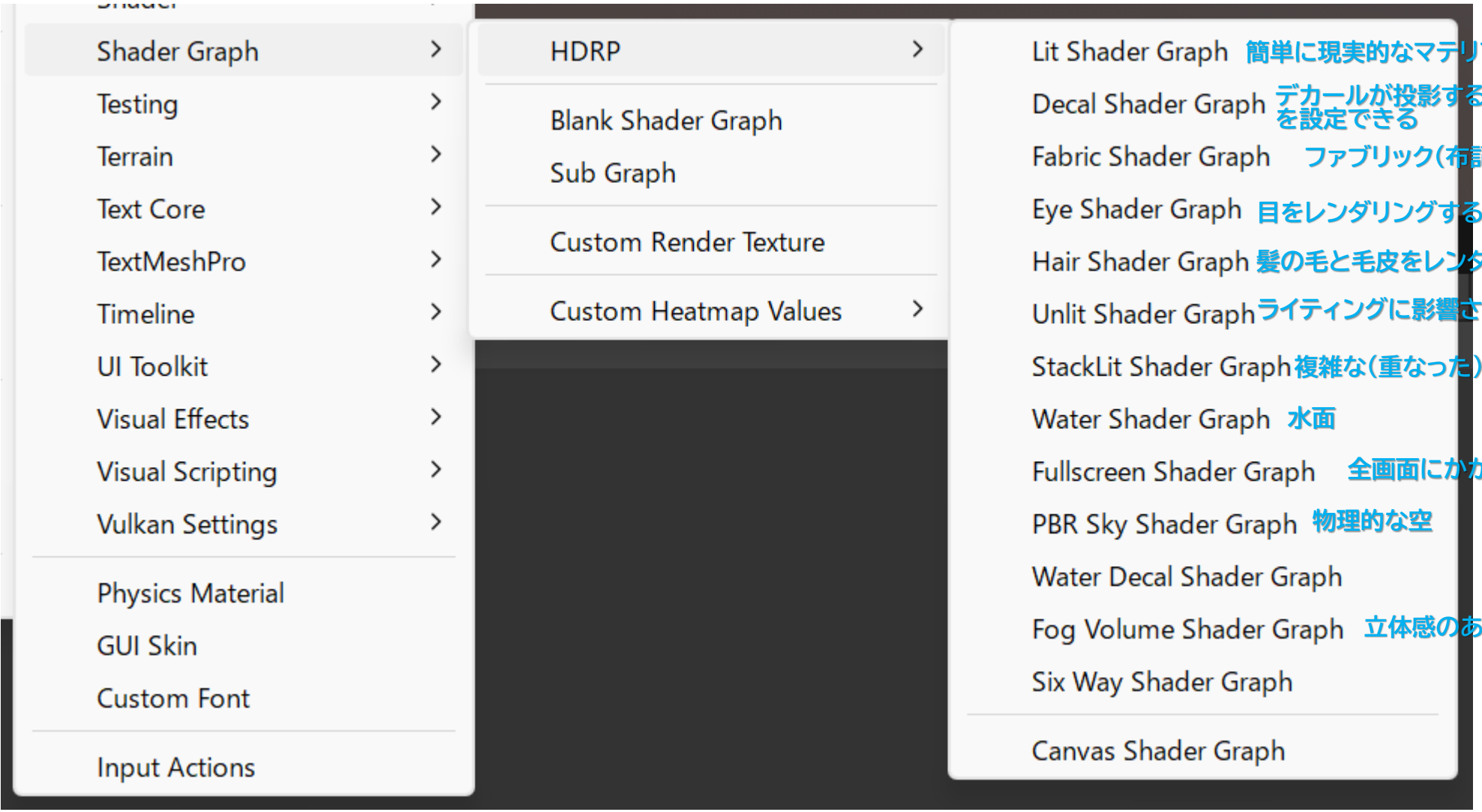
- プログラマ外の人でもシェーダを作れそう
  - 重いシェーダが作られる傾向にあるので、後から最適化できる範囲で使用する必要性あり
- 結果がすぐにプレビューできる！

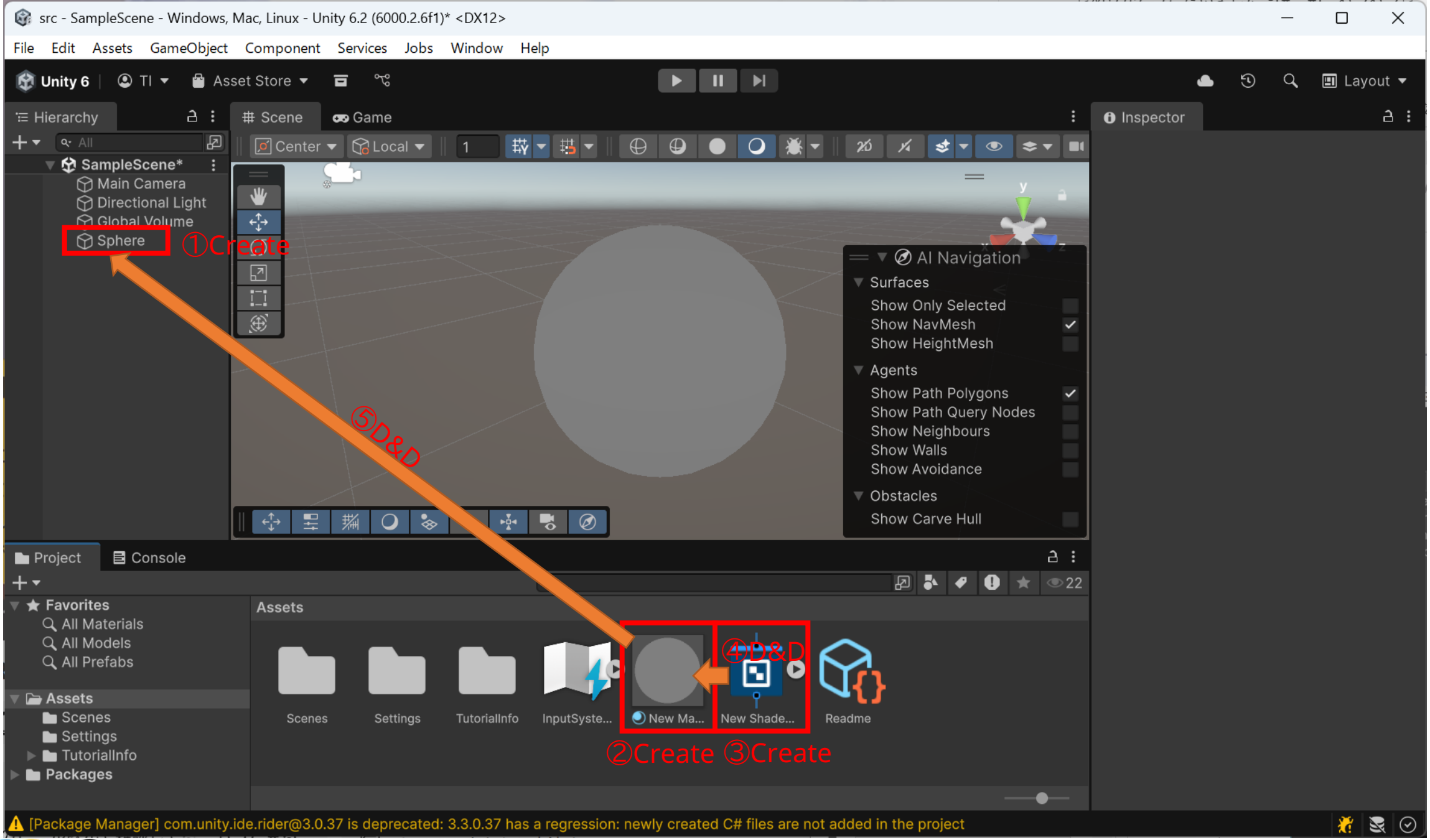


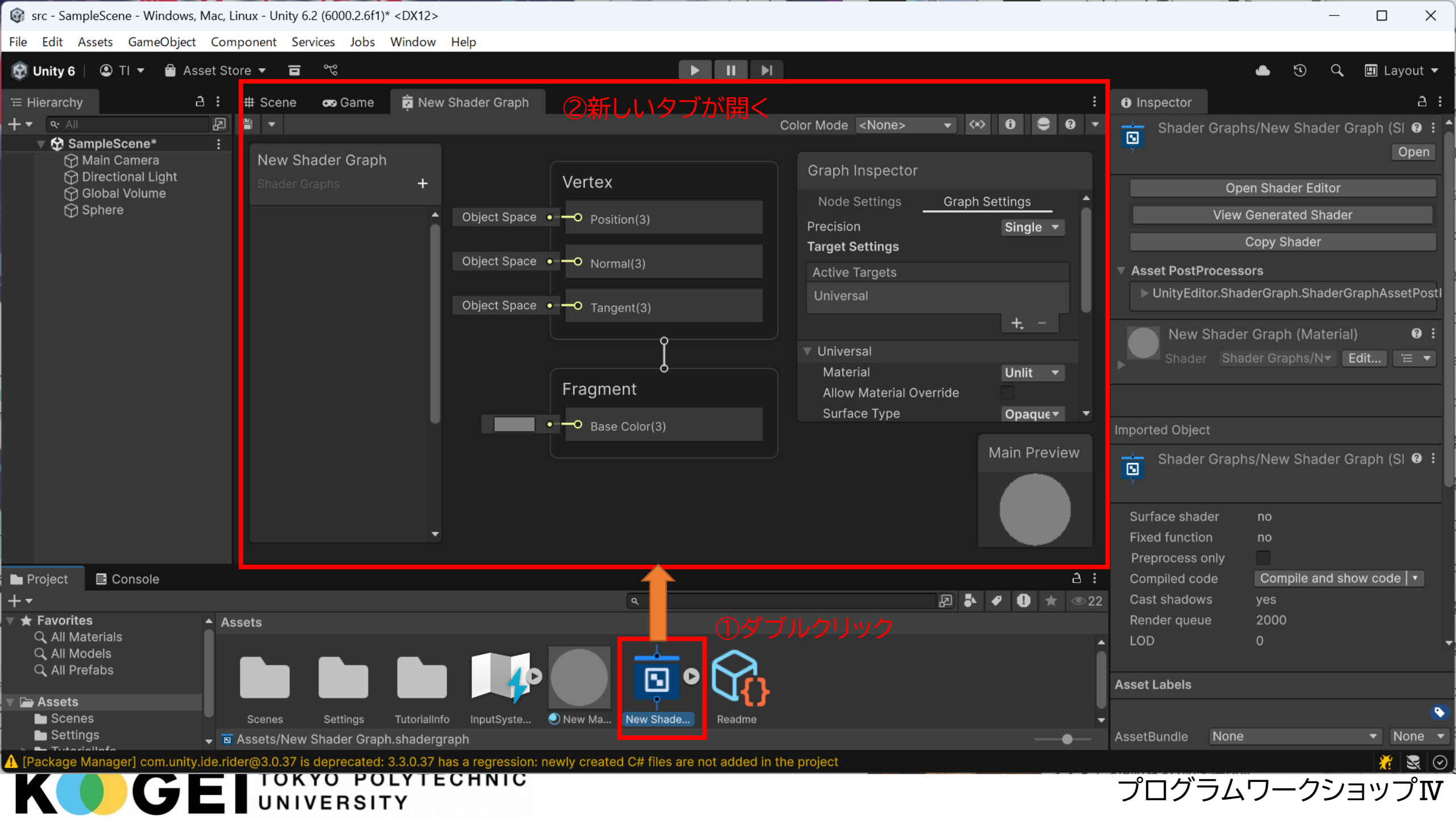

src - SampleScene - Windows, Mac, Linux - Unity 6.2 (6000.2.6f1) <DX12>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
Asset Store
Layout
Hierarchy
SampleScene
Main Camera
Directional Light
Global Volume
Inspector
Assets (Default Asset)
Open
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Settings
TutorialInfo
Packages
Asset Labels
AssetBundle
None
None
Create
Show in Explorer
Open
Open Scene Additive
Delete
Rename
Copy Path Alt+Ctrl+C
View in Package Manager
Import Package
Export Package...
Import New Asset...
Extract Material SubAsset
Find References In Scene
Find References In Project
Select Dependencies
Refresh Ctrl+R
Reimport
Reimport All
Open C# Project
View in Import Activity Window
Update UXML Schema
Properties... Alt+P
Folder
Material
MonoBehaviour Script
Prefab Variant
2D
Animation
Audio
Rendering
Scene
Scripting
Search
Shader
Shader Graph
Testing
Terrain
Text Core
TextMeshPro
Timeline
UI Toolkit
Visual Scripting
Vulkan Settings
Physics Material
GUI Skin
Custom Font
Input Actions
URP
Blank Shader Graph
Sub Graph
Custom Render Texture
Custom Heatmap Values
Lit Shader Graph
標準的
Unlit Shader Graph
照明の影響を受けない
Sprite Custom Lit Shader Graph
Sprite Unlit Shader Graph
スプライト用
Sprite Lit Shader Graph
Decal Shader Graph
デカール用
Canvas Shader Graph
GUI用
Fullscreen Shader Graph
全画面用
Six Way Shader Graph
6方向光源用
[Package Manager] com.unity.ide.rider@3.0.37 is deprecated: 3.3.0.37 has a regression: newly created C# files are not added in the project

Unlit Shader Graph




Shader Graph
Testing
Terrain
Text Core
TextMeshPro
Timeline
UI Toolkit
Visual Effects
Visual Scripting
Vulkan Settings
Physics Material
GUI Skin
Custom Font
Input Actions
HDRP
Blank Shader Graph
Sub Graph
Custom Render Texture
Custom Heatmap Values
Lit Shader Graph 簡単に現実的なマテリアルを作成することができる
Decal Shader Graph デカールが投影するマテリアルのどの属性に影響を与えるかを設定できる
Fabric Shader Graph ファブリック(布記事)を作成できる
Eye Shader Graph 目をレンダリングするときにまず利用するもの
Hair Shader Graph 髪の毛と毛皮をレンダリングするときにまず利用するもの
Unlit Shader Graph ライティングに影響されない
StackLit Shader Graph 複雑な(重なった)マテリアルをレンダリング
Water Shader Graph 水面
Fullscreen Shader Graph 全画面にかかる演出
PBR Sky Shader Graph 物理的な空
Water Decal Shader Graph
Fog Volume Shader Graph 立体感のある霧
Six Way Shader Graph
Canvas Shader Graph

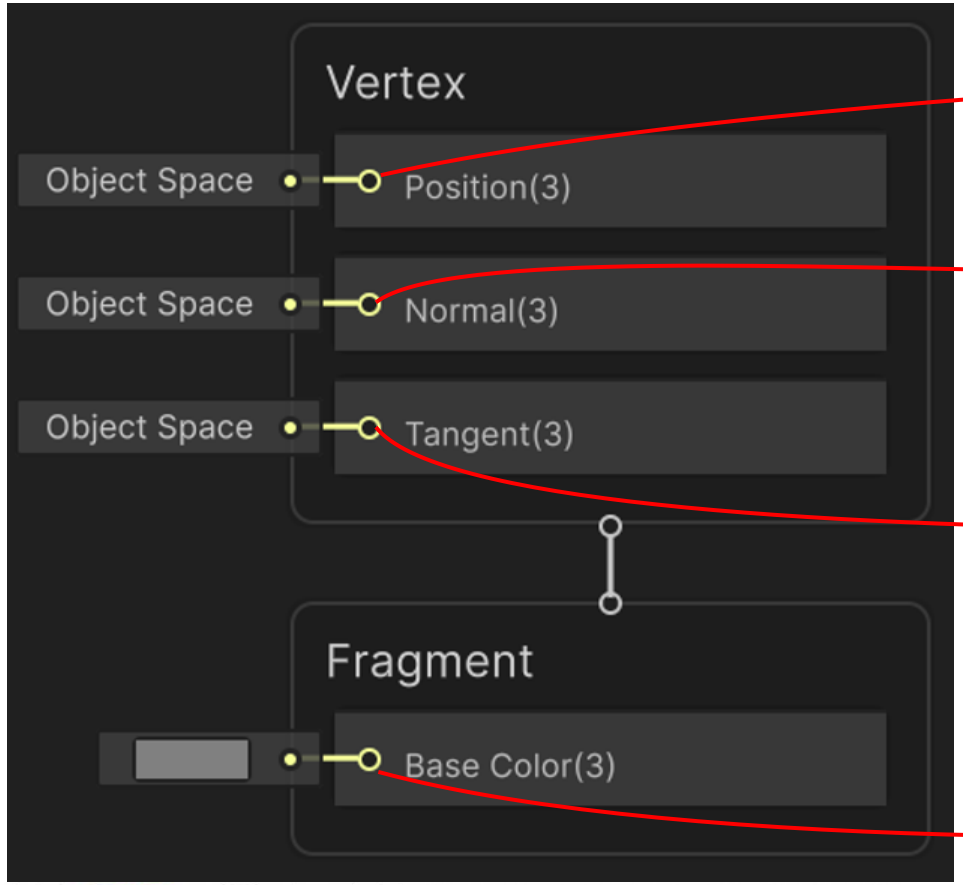

src - SampleScene - Windows, Mac, Linux - Unity 6.2 (6000.2.6f1)* <DX12>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
Asset Store
Layout
Hierarchy
Scene
Game
Inspector
SampleScene*
Main Camera
Directional Light
Global Volume
Sphere
①Create
⑤D&D
AI Navigation
Surfaces
Show Only Selected
Show NavMesh
Show HeightMesh
Agents
Show Path Polygons
Show Path Query Nodes
Show Neighbours
Show Walls
Show Avoidance
Obstacles
Show Carve Hull
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Settings
TutorialInfo
Packages
Assets
Scenes
Settings
TutorialInfo
InputSyste...
New Ma...
New Shade...
Readme
④D&D
②Create ③Create
[Package Manager] com.unity.ide.rider@3.0.37 is deprecated: 3.3.0.37 has a regression: newly created C# files are not added in the project


src - SampleScene - Windows, Mac, Linux - Unity 6.2 (6000.2.6f1)* <DX12>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
Asset Store
Layout
Hierarchy
SampleScene*
Main Camera
Directional Light
Global Volume
Sphere
Scene
Game
New Shader Graph
②新しいタブが開く
Color Mode <None>
New Shader Graph
Shader Graphs
Vertex
Object Space
Position(3)
Object Space
Normal(3)
Object Space
Tangent(3)
Fragment
Base Color(3)
Graph Inspector
Node Settings
Graph Settings
Precision
Single
Target Settings
Active Targets
Universal
Universal
Material
Unlit
Allow Material Override
Surface Type
Opaque
Main Preview
Inspector
Shader Graphs/New Shader Graph (SI
Open
Open Shader Editor
View Generated Shader
Copy Shader
Asset PostProcessors
UnityEditor.ShaderGraph.ShaderGraphAssetPostI
New Shader Graph (Material)
Shader
Shader Graphs/N
Edit...
Imported Object
Shader Graphs/New Shader Graph (SI
Surface shader no
Fixed function no
Preprocess only
Compiled code
Compile and show code
Cast shadows yes
Render queue 2000
LOD 0
Asset Labels
AssetBundle None None
Project
Console
①ダブルクリック
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Settings
Assets
Scenes
Settings
TutorialInfo
InputSyste...
New Ma...
New Shade...
Readme
Assets/New Shader Graph.shadergraph
[Package Manager] com.unity.ide.rider@3.0.37 is deprecated: 3.3.0.37 has a regression: newly created C# files are not added in the project

Shader Graph ウィンドウ

やっていること

- コードの一部を自由に指定できるようにする

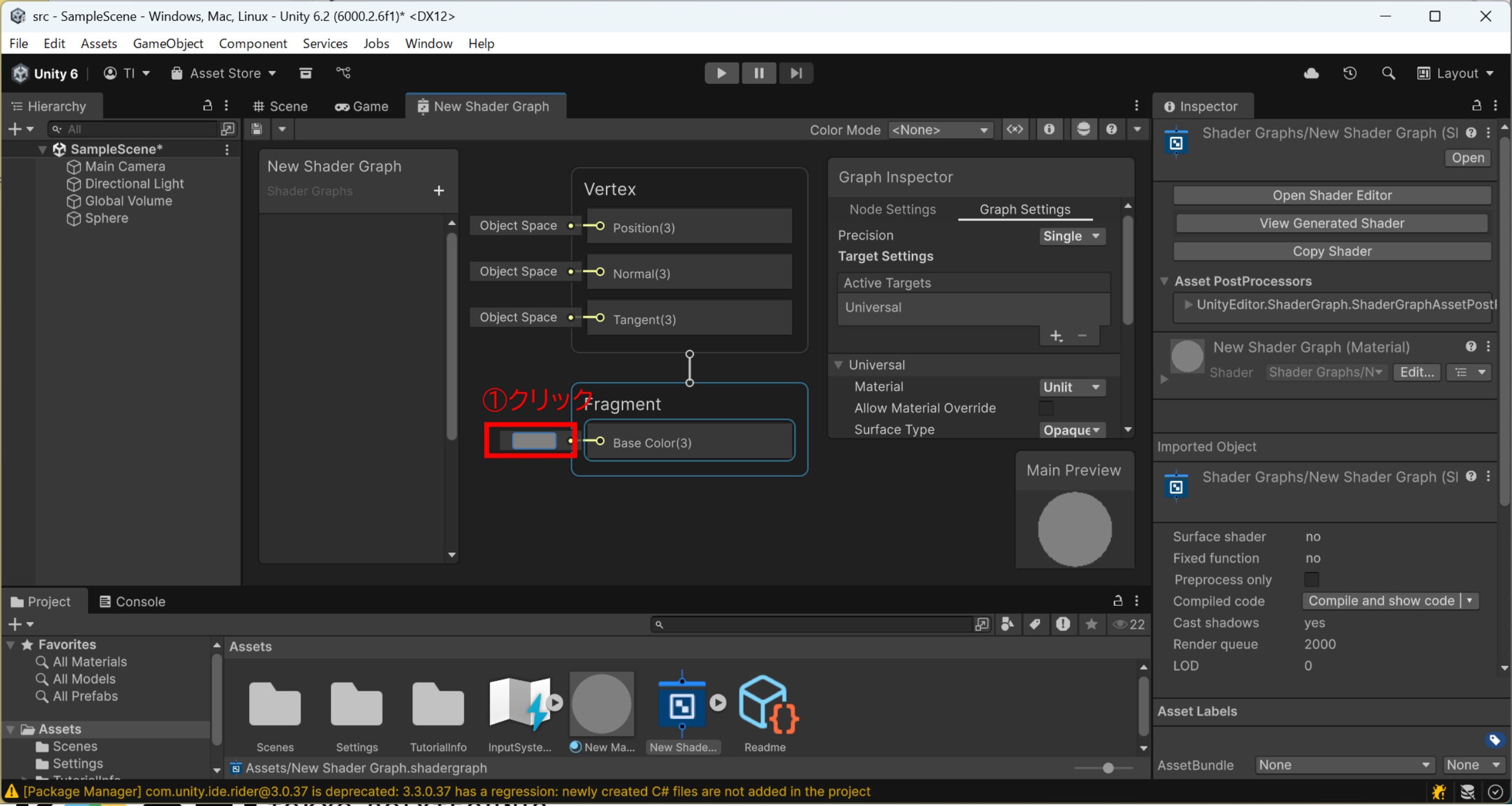


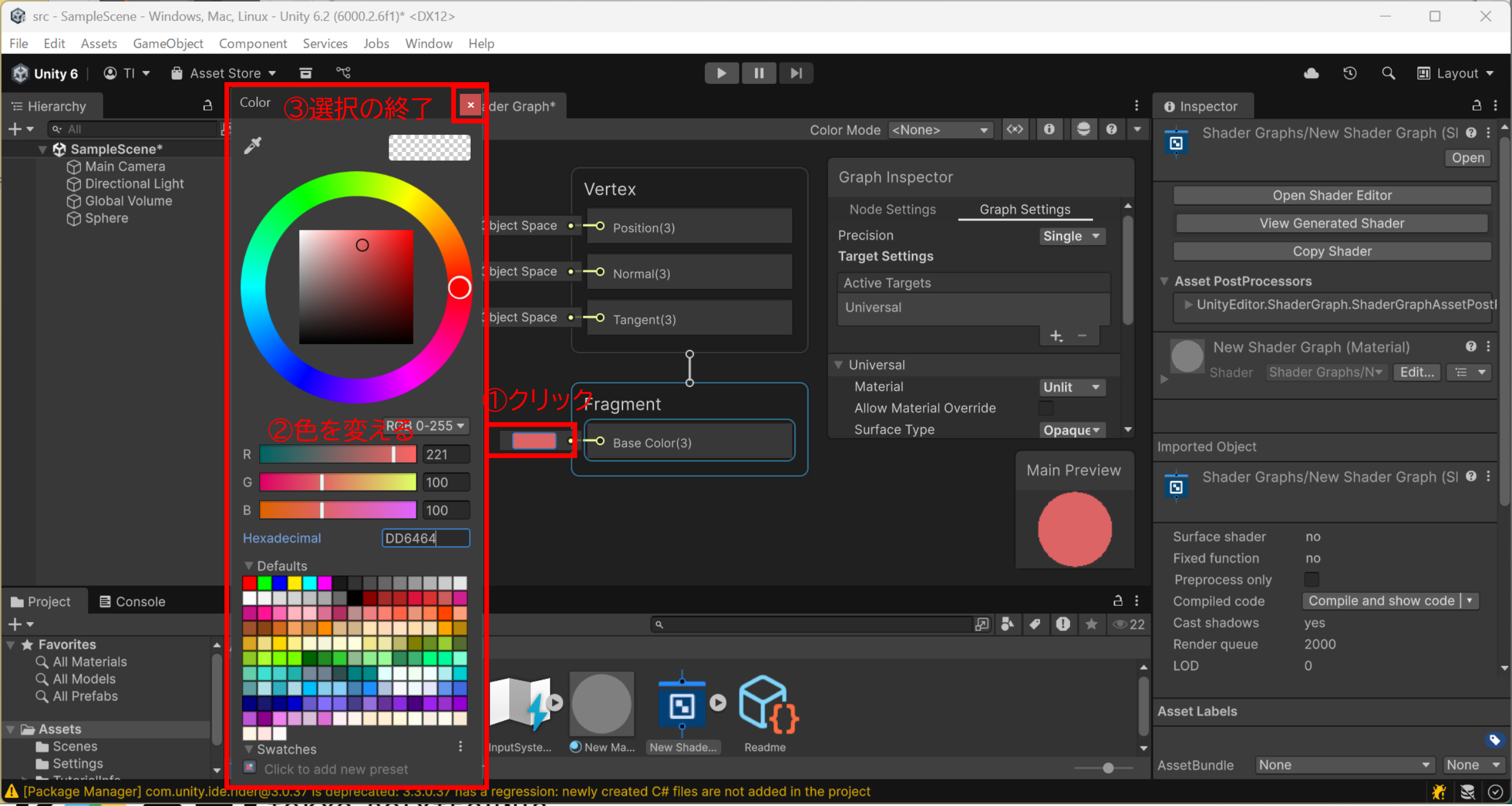
大枠は決まっている(種類ごと)

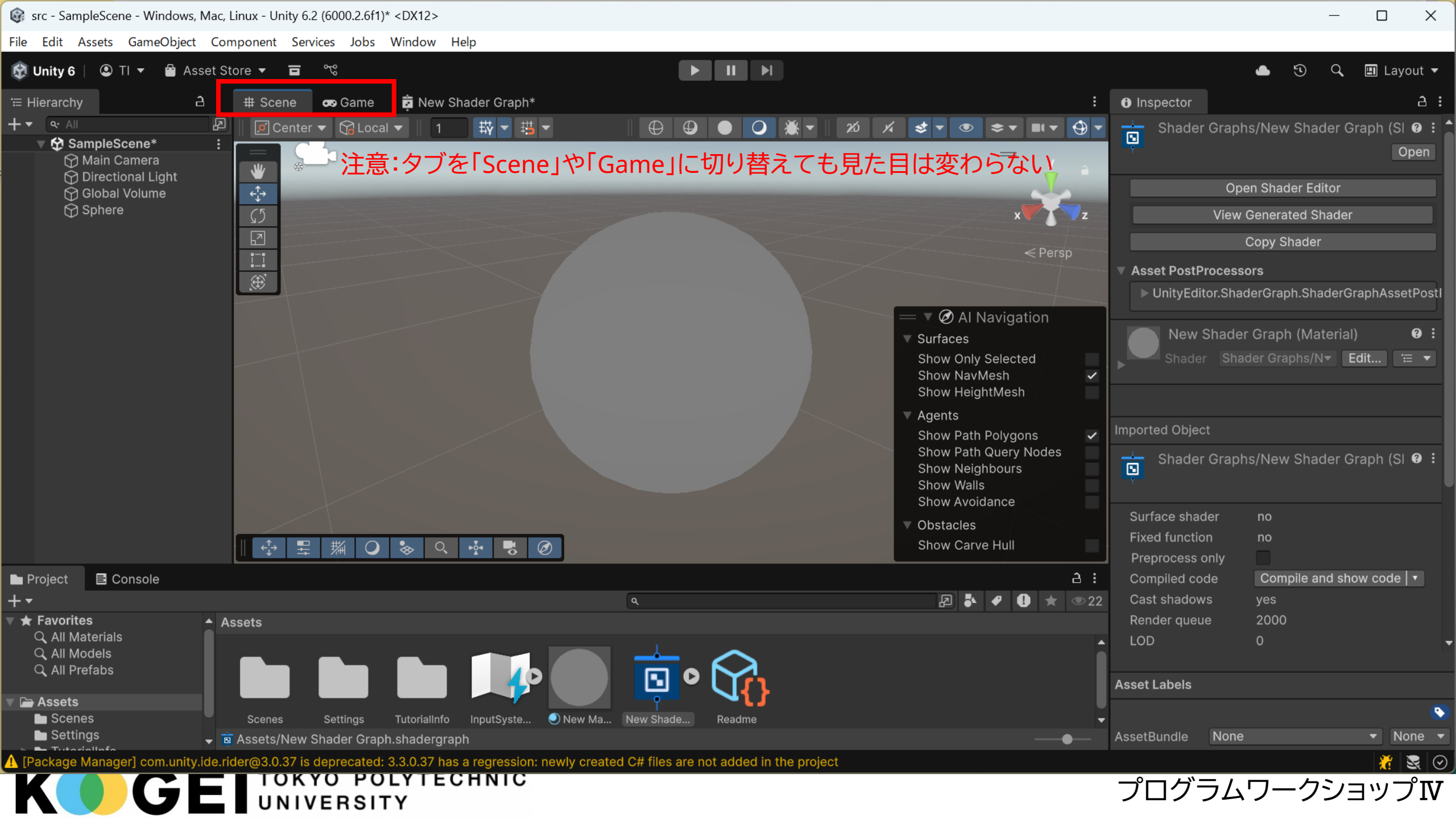
```
42          Varyings vert(Attributes IN)
43          {
44              float3 Position = ...;
45              float3 Normal = ...;
46              float3 tangent = ...;
47              Varyings OUT;
48              OUT.positionHCS = TransformObjectToHClip(Position);
49              return OUT;
50          }
51
52          half4 frag(Varyings IN) : SV_Target
53          {
54              half3 Base_Color = ...;
55              return half4(Base_Color, 1);
56          }
```

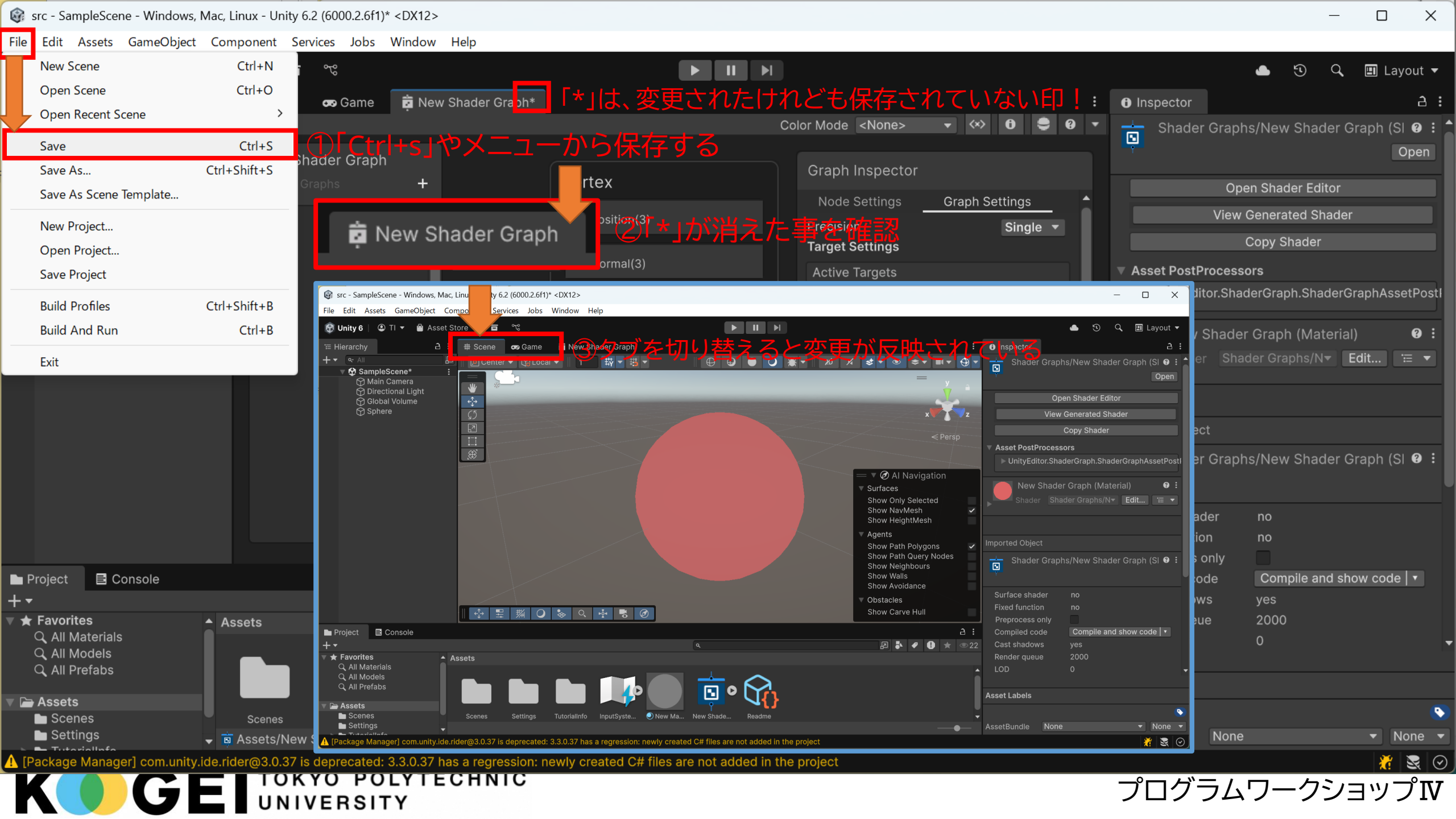
シェーダグラフ入門

- シェーダグラフの追加
- 色を変える
- ノードの追加
- ノードの種類
- Lit Shader Graph


src - SampleScene - Windows, Mac, Linux - Unity 6.2 (6000.2.6f1)* <DX12>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
Asset Store
Layout
Hierarchy
Scene
Game
New Shader Graph
Inspector
SampleScene*
Main Camera
Directional Light
Global Volume
Sphere
Color Mode <None>
New Shader Graph
Shader Graphs
Vertex
Object Space
Position(3)
Object Space
Normal(3)
Object Space
Tangent(3)
①クリック
Fragment
Base Color(3)
Graph Inspector
Node Settings
Graph Settings
Precision
Single
Target Settings
Active Targets
Universal
Universal
Material
Unlit
Allow Material Override
Surface Type
Opaque
Main Preview
Shader Graphs/New Shader Graph (SI
Open
Open Shader Editor
View Generated Shader
Copy Shader
Asset PostProcessors
UnityEditor.ShaderGraph.ShaderGraphAssetPostI
New Shader Graph (Material)
Shader
Shader Graphs/N
Edit...
Imported Object
Shader Graphs/New Shader Graph (SI
Surface shader no
Fixed function no
Preprocess only
Compiled code
Compile and show code
Cast shadows yes
Render queue 2000
LOD 0
Asset Labels
AssetBundle None None
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Settings
Assets
Scenes
Settings
TutorialInfo
InputSyste...
New Ma...
New Shade...
Readme
Assets/New Shader Graph.shadergraph
[Package Manager] com.unity.ide.rider@3.0.37 is deprecated: 3.3.0.37 has a regression: newly created C# files are not added in the project

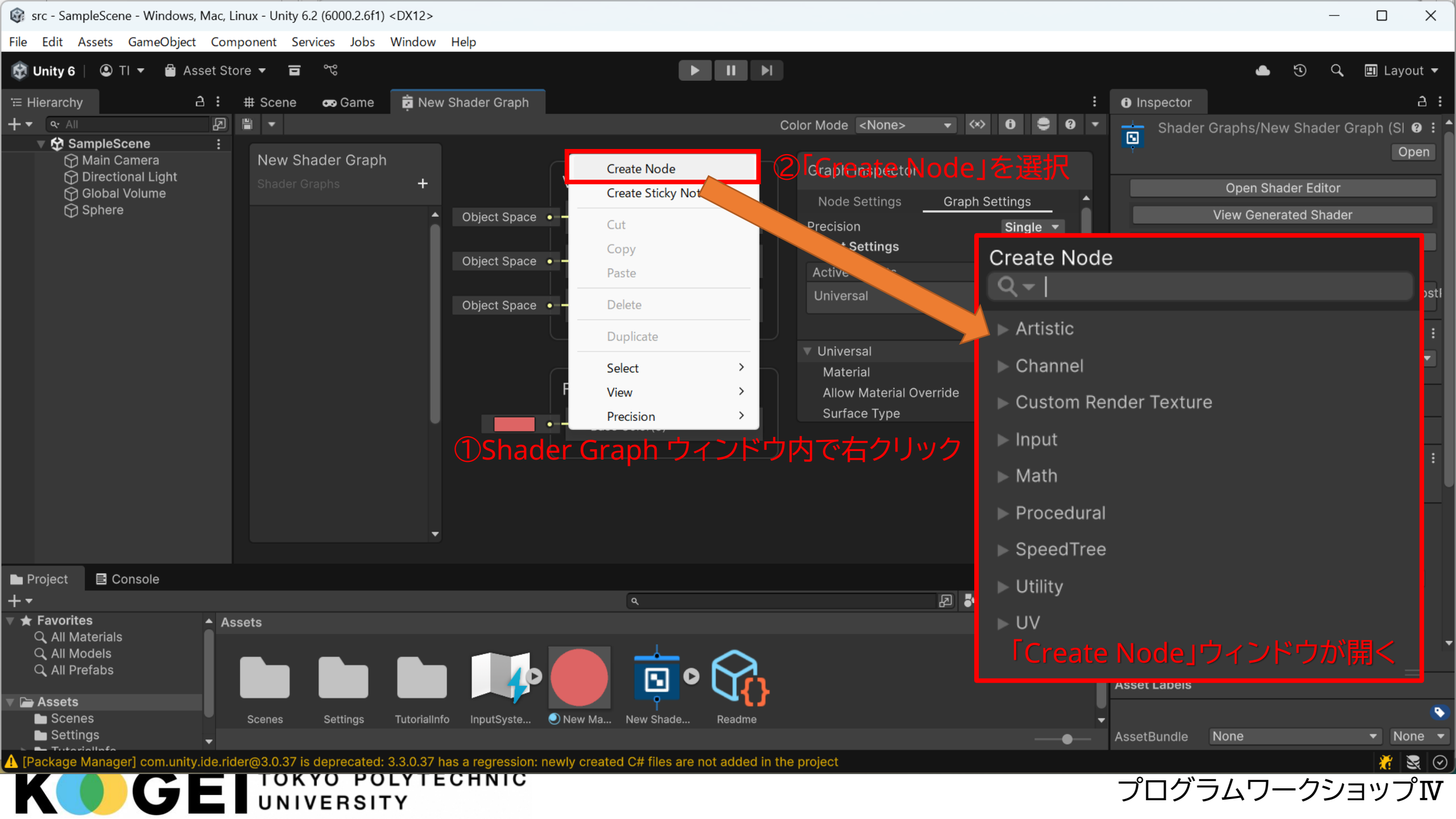

③選択の終了
②色を変える
①クリック
Color
R 221
G 100
B 100
Hexadecimal DD6464
Vertex
Position(3)
Normal(3)
Tangent(3)
Fragment
Base Color(3)
Graph Inspector
Node Settings
Graph Settings
Precision Single
Target Settings
Active Targets
Universal
Material Unlit
Allow Material Override
Surface type Opaque
Main Preview
Hierarchy
SampleScene*
Main Camera
Directional Light
Global Volume
Sphere
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Settings
Inspector
Shader Graphs/New Shader Graph (SI
Open
Open Shader Editor
View Generated Shader
Copy Shader
Asset PostProcessors
New Shader Graph (Material)
Imported Object
Surface shader no
Fixed function no
Preprocess only
Compiled code Compile and show code
Cast shadows yes
Render queue 2000
LOD 0
Asset Labels
AssetBundle None
[Package Manager] com.unity.ide.rider@3.0.37 is deprecated. 3.3.0.37 has a regression: newly created C# files are not added in the project


src - SampleScene - Windows, Mac, Linux - Unity 6.2 (6000.2.6f1)* <DX12>
File Edit Assets GameObject Component Services Jobs Window Help
Unity 6
Asset Store
Layout
Hierarchy
Scene
Game
New Shader Graph*
Inspector
SampleScene*
Main Camera
Directional Light
Global Volume
Sphere
Center
Local
注意:タブを「Scene」や「Game」に切り替えても見た目は変わらない
Persp
AI Navigation
Surfaces
Show Only Selected
Show NavMesh
Show HeightMesh
Agents
Show Path Polygons
Show Path Query Nodes
Show Neighbours
Show Walls
Show Avoidance
Obstacles
Show Carve Hull
Shader Graphs/New Shader Graph (SI
Open
Open Shader Editor
View Generated Shader
Copy Shader
Asset PostProcessors
UnityEditor.ShaderGraph.ShaderGraphAssetPostI
New Shader Graph (Material)
Shader
Shader Graphs/N
Edit...
Imported Object
Shader Graphs/New Shader Graph (SI
Surface shader no
Fixed function no
Preprocess only
Compiled code Compile and show code
Cast shadows yes
Render queue 2000
LOD 0
Asset Labels
AssetBundle None None
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Settings
Assets
Scenes
Settings
TutorialInfo
InputSyste...
New Ma...
New Shade...
Readme
Assets/New Shader Graph.shadergraph
[Package Manager] com.unity.ide.rider@3.0.37 is deprecated: 3.3.0.37 has a regression: newly created C# files are not added in the project


File Edit Assets GameObject Component Services Jobs Window Help
New Scene Ctrl+N
Open Scene Ctrl+O
Open Recent Scene
Save Ctrl+S
Save As... Ctrl+Shift+S
Save As Scene Template...
New Project...
Open Project...
Save Project
Build Profiles Ctrl+Shift+B
Build And Run Ctrl+B
Exit
「*」は、変更されたけれども保存されていない印！
①「Ctrl+s」やメニューから保存する
②「*」が消えた事を確認
③タブを切り替えると変更が反映されている
New Shader Graph
Graph Inspector
Node Settings
Graph Settings
Target Settings
Active Targets
Inspector
Open Shader Editor
View Generated Shader
Copy Shader
Asset PostProcessors
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Settings
[Package Manager] com.unity.ide.rider@3.0.37 is deprecated: 3.3.0.37 has a regression: newly created C# files are not added in the project

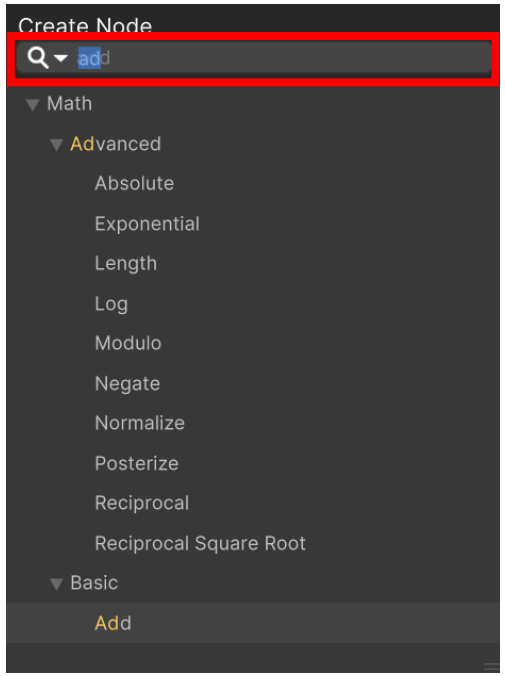
シェーダグラフ入門

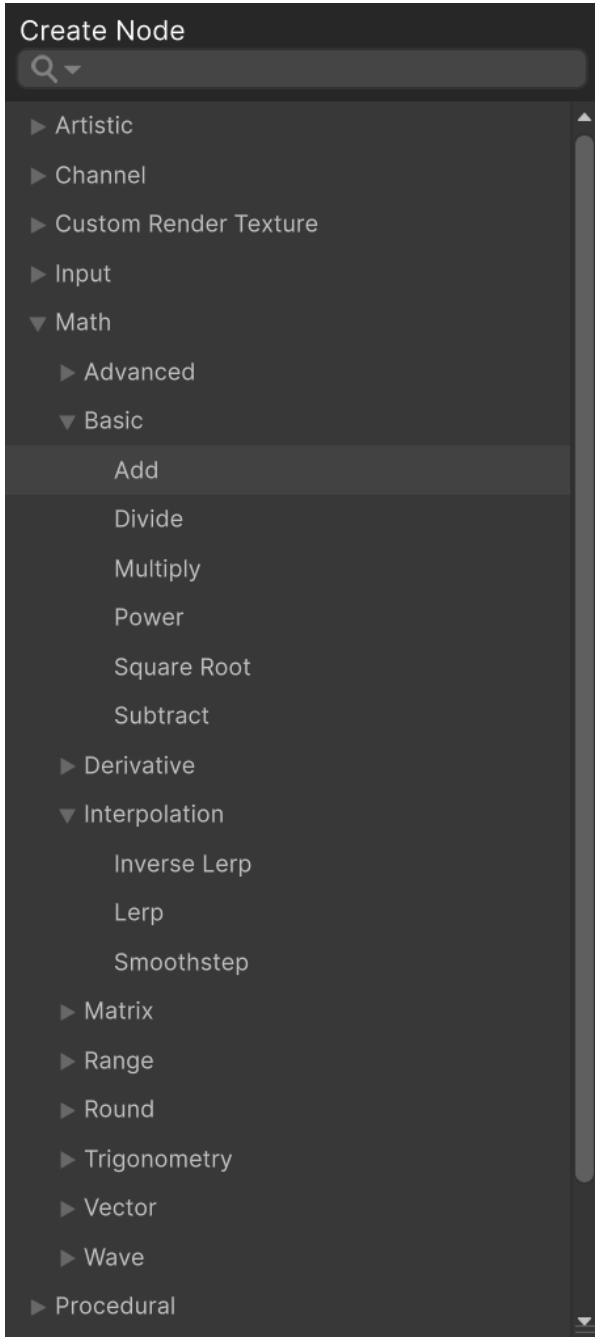
- シェーダグラフの追加
- 色を変える
- ノードの追加
- ノードの種類
- Lit Shader Graph


File Edit Assets GameObject Component Services Jobs Window Help
Hierarchy
Scene
Game
New Shader Graph
Inspector
SampleScene
Main Camera
Directional Light
Global Volume
Sphere
New Shader Graph
Shader Graphs
Color Mode
Create Node
Create Sticky Note
Cut
Copy
Paste
Delete
Duplicate
Select
View
Precision
Object Space
Object Space
Object Space
Node Settings
Graph Settings
Precision
Single
Universal
Material
Allow Material Override
Surface Type
Shader Graphs/New Shader Graph (SI
Open
Open Shader Editor
View Generated Shader
②「Create Node」を選択
①Shader Graph ウィンドウ内で右クリック
Create Node
Artistic
Channel
Custom Render Texture
Input
Math
Procedural
SpeedTree
Utility
UV
「Create Node」ウィンドウが開く
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Settings
Assets
Scenes
Settings
TutorialInfo
InputSyste...
New Ma...
New Shade...
Readme
Asset Labels
AssetBundle
None
None
[Package Manager] com.unity.ide.rider@3.0.37 is deprecated: 3.3.0.37 has a regression: newly created C# files are not added in the project

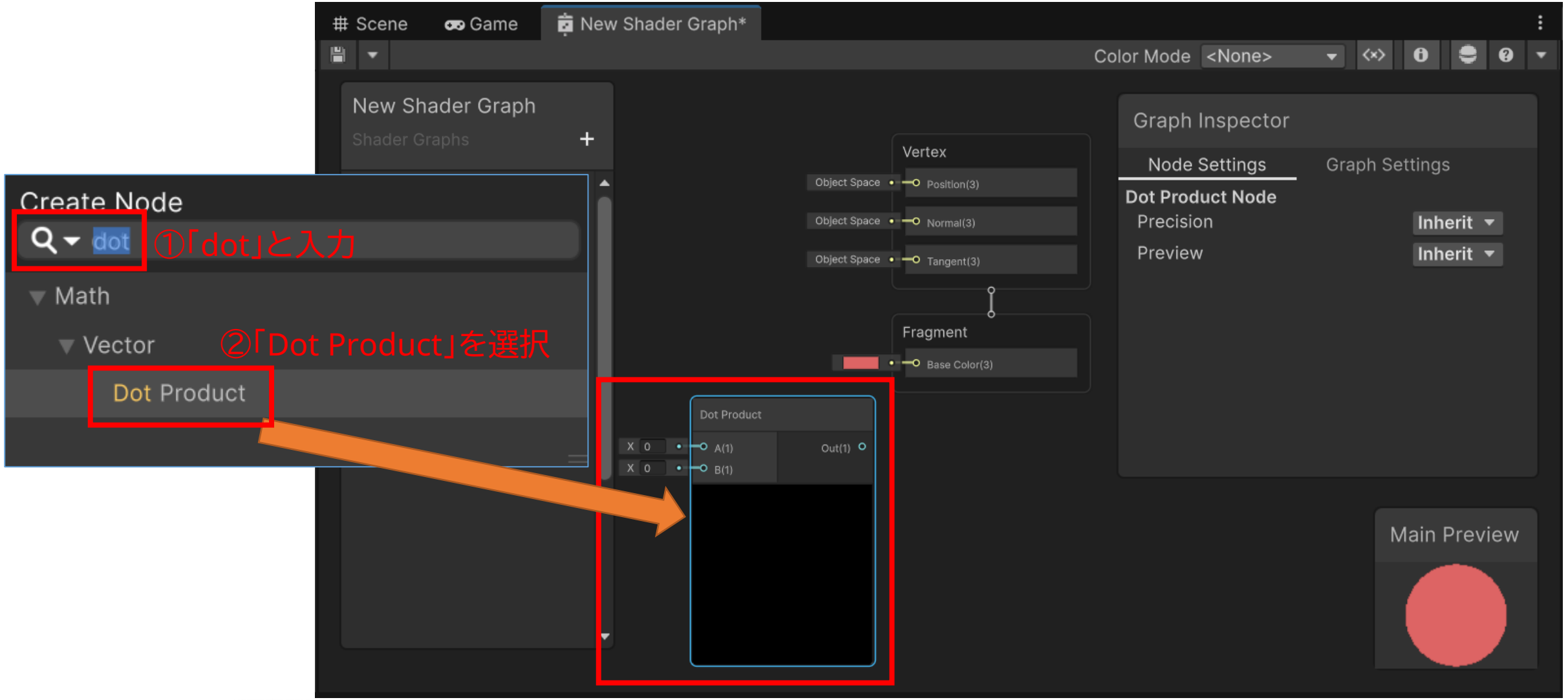
Create Node ウィンドウ

- 沢山のノードがある
- ▶ を押すとサブメニューが開く
  - ▼は、現在サブメニューが開いている
- 検索バーで調べることも可能
  - その文字列が使われている単語だけが表示される

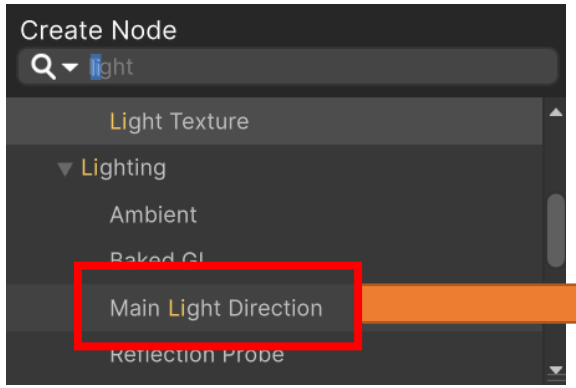




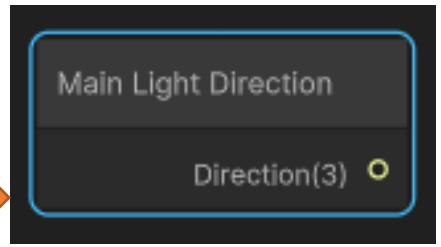
ドット積の追加

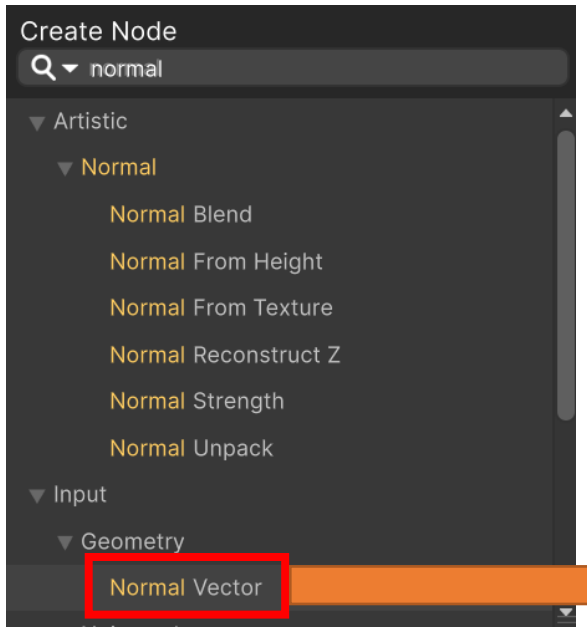

Scene
Game
New Shader Graph*
Color Mode
<None>
New Shader Graph
Shader Graphs
Create Node
dot
①「dot」と入力
Math
Vector
②「Dot Product」を選択
Dot Product
Vertex
Object Space
Position(3)
Object Space
Normal(3)
Object Space
Tangent(3)
Fragment
Base Color(3)
Dot Product
X 0
X 0
A(1)
B(1)
Out(1)
Graph Inspector
Node Settings
Graph Settings
Dot Product Node
Precision
Inherit
Preview
Inherit
Main Preview

その他のノードを追加

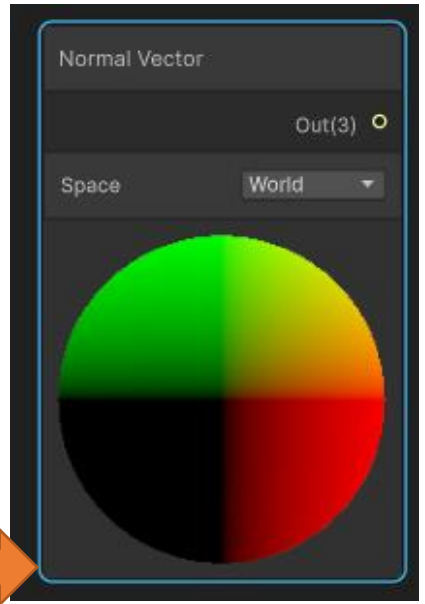


光源の向き
（ライトベクトルの逆）

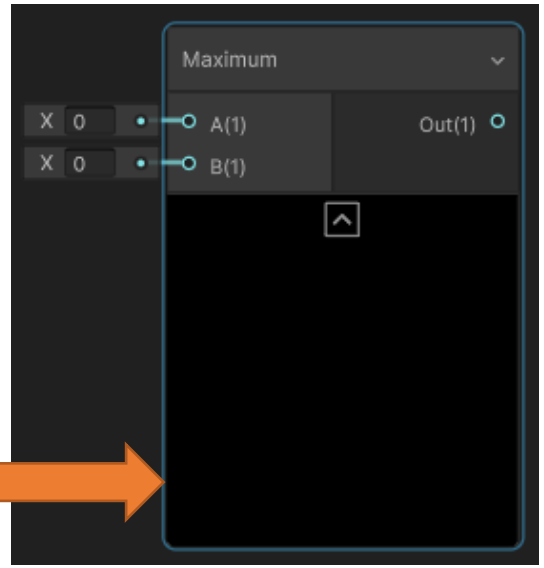




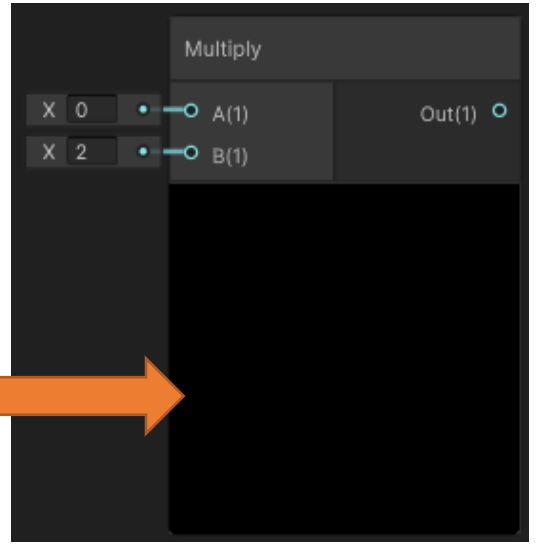
法線ベクトル



最大値

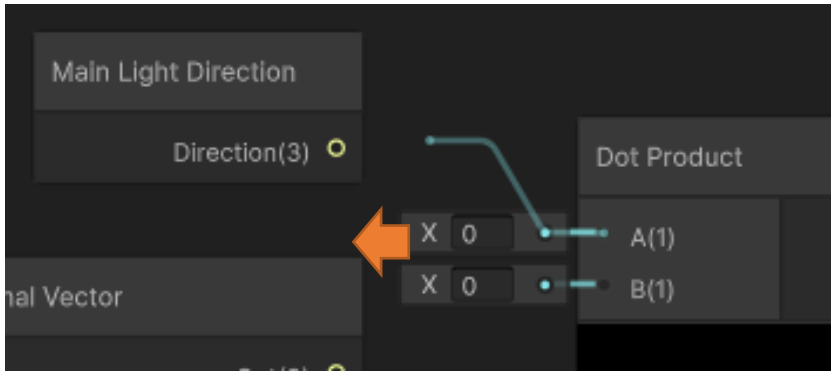


乗算

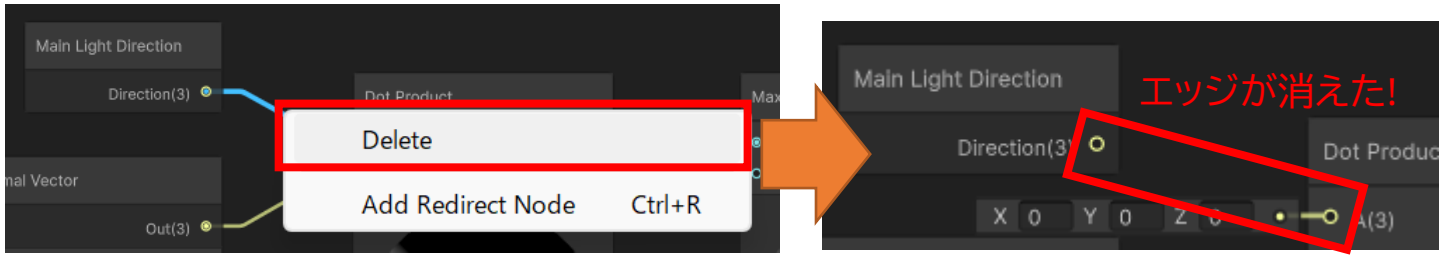


ノード

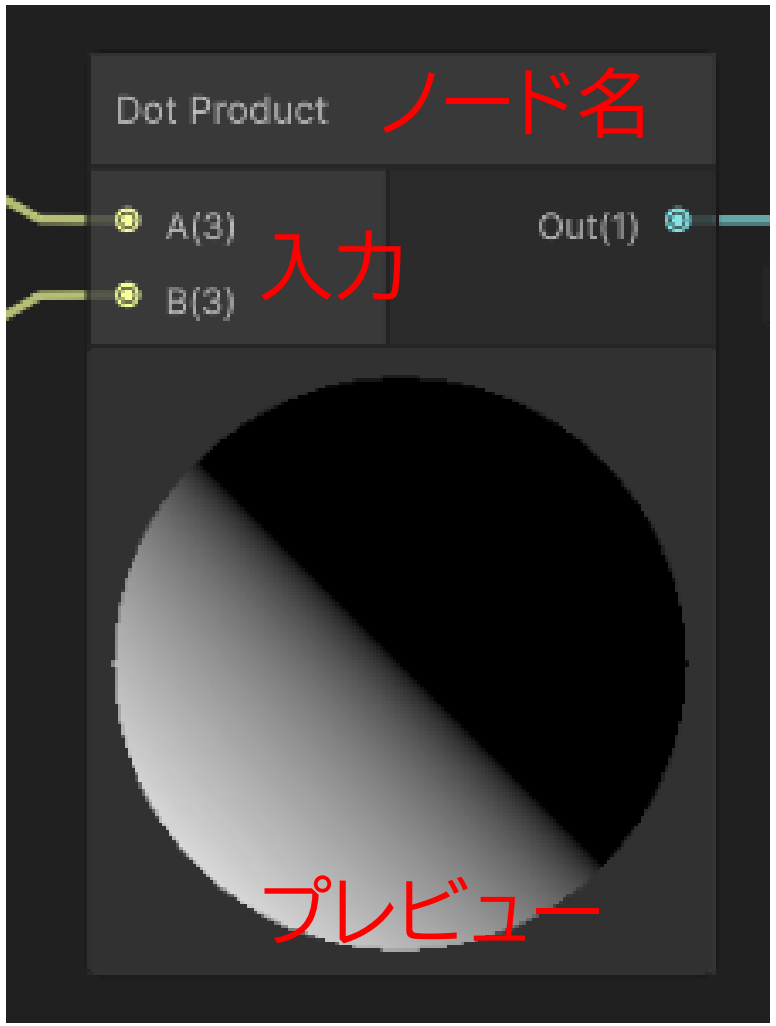
入力(出力)の「・」からドラッグすることで
エッジを引くことができる



エッジ上で右クリックして、「delete」
を選択することでエッジを削除できる

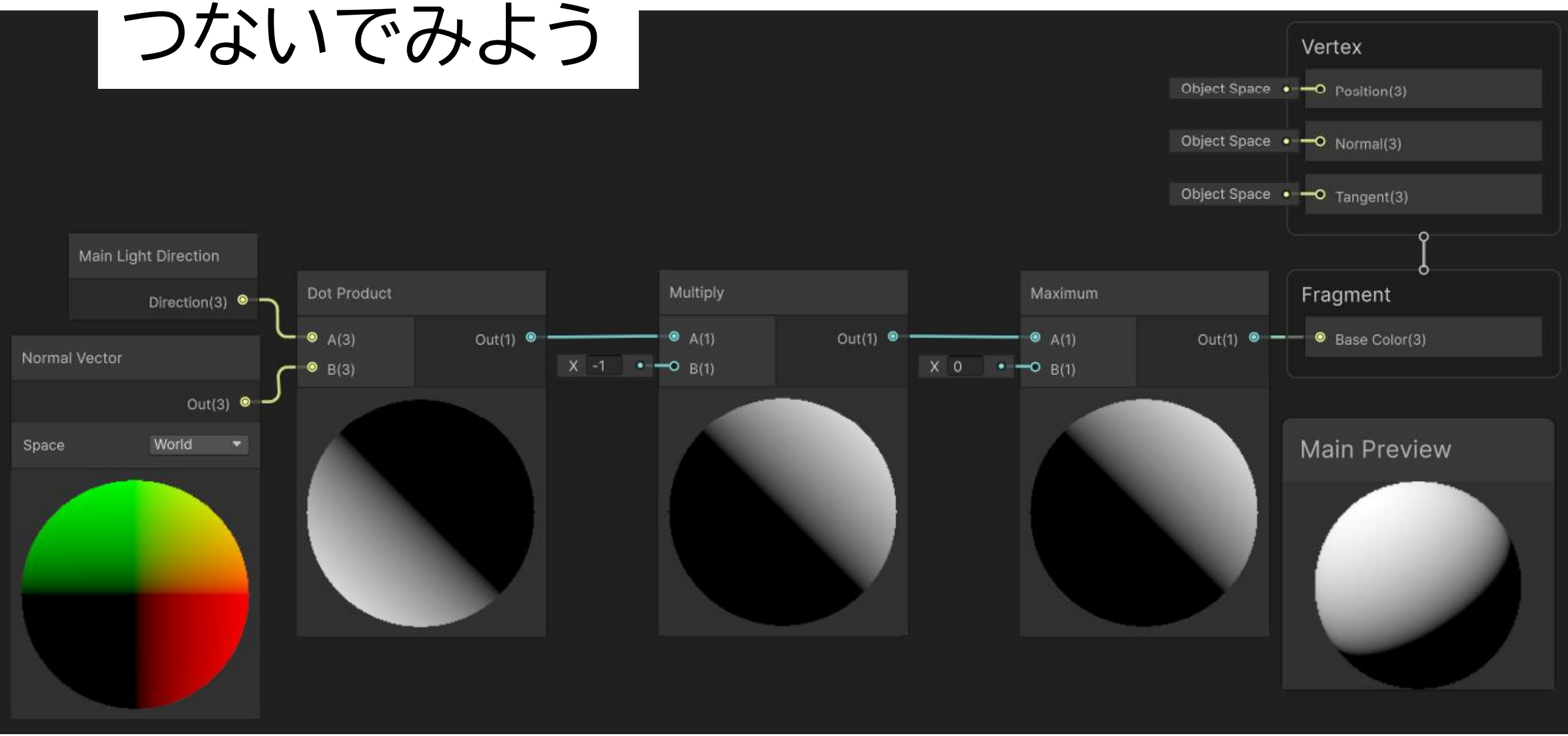


ノードをつまんでドラッグすると動かせる



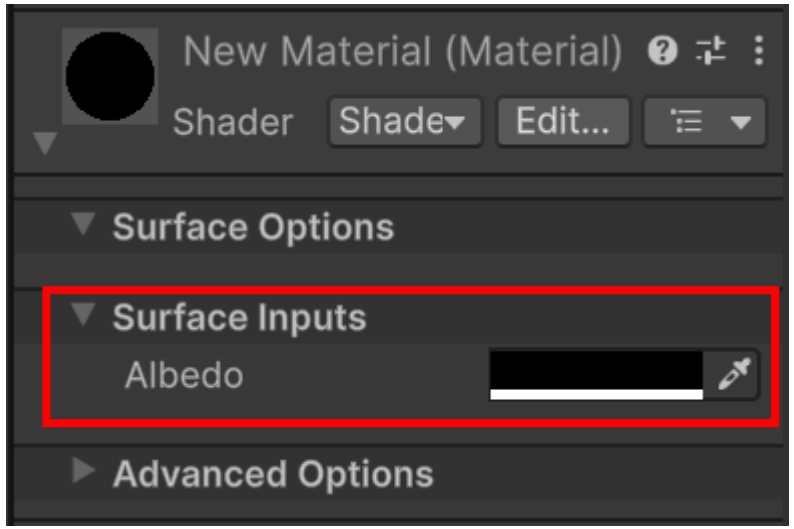
max(0, dot(Normal, - Light.Direction)
=max(0, - dot(Normal, Light.Direction)

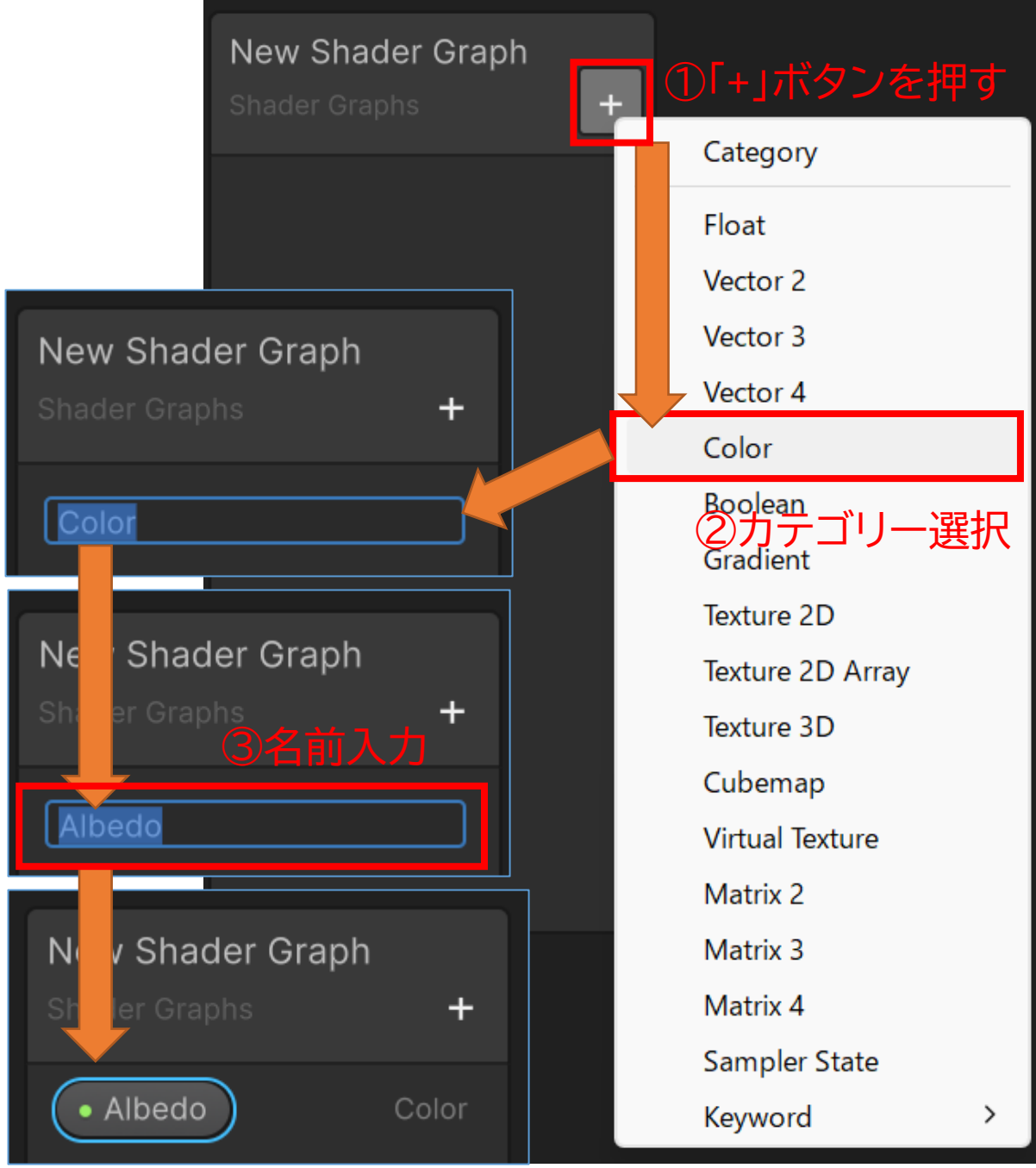
つないでみよう



ブラックボード（黒板）

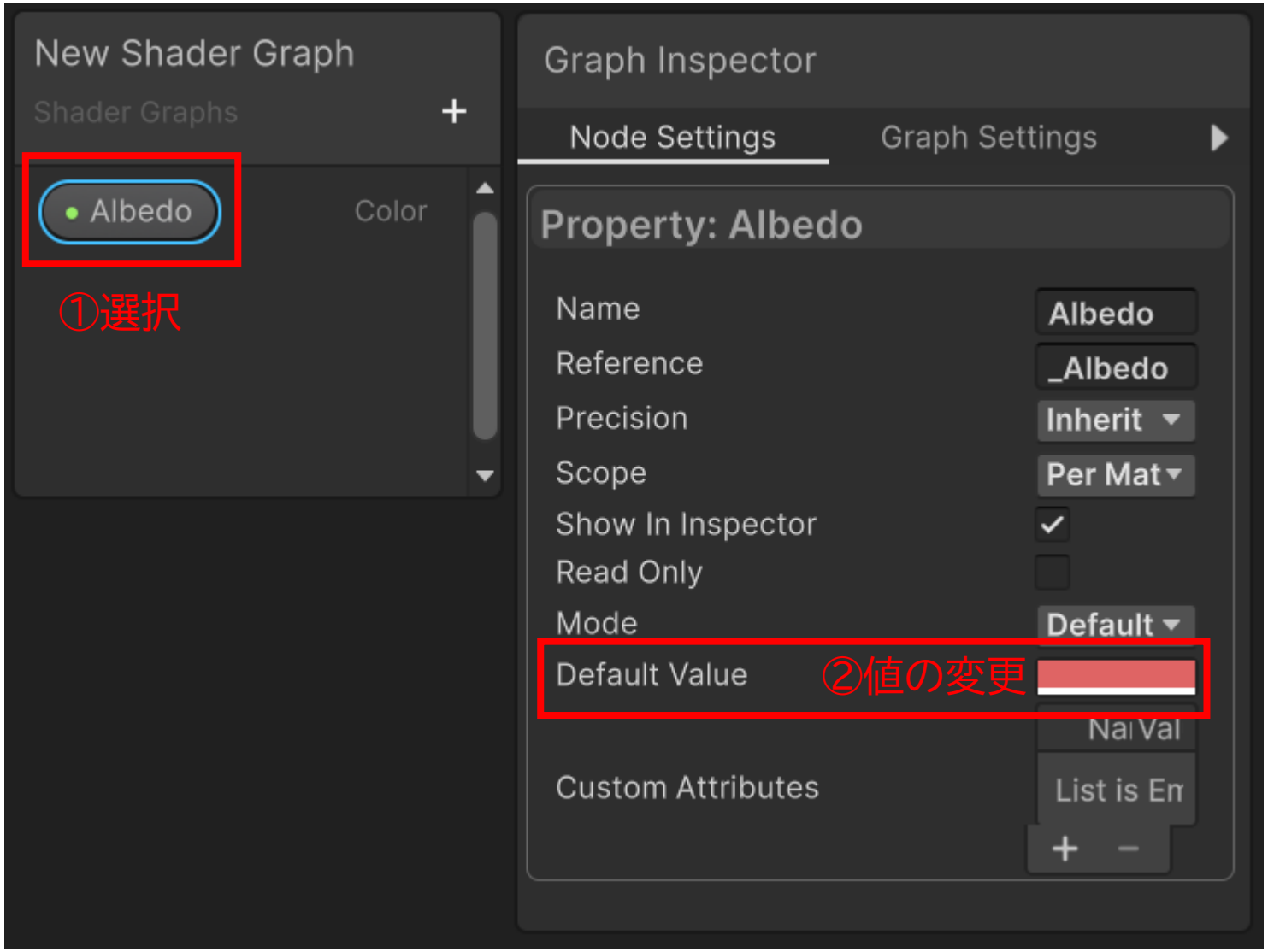
- インスペクターに表示される変数
- 追加するとインスペクターにも登録される





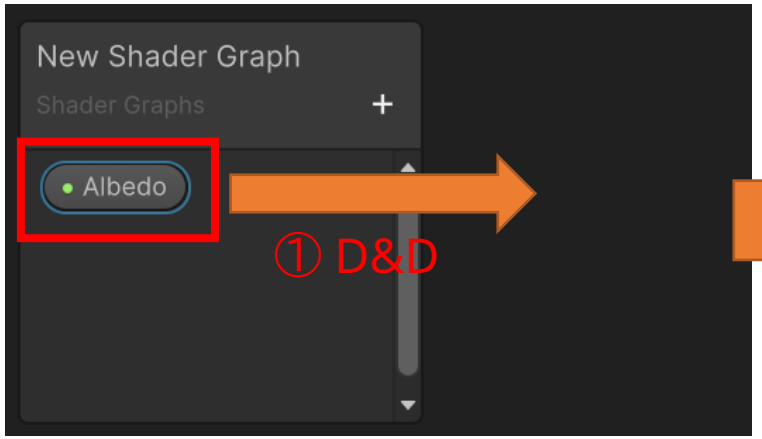
ノードの設定

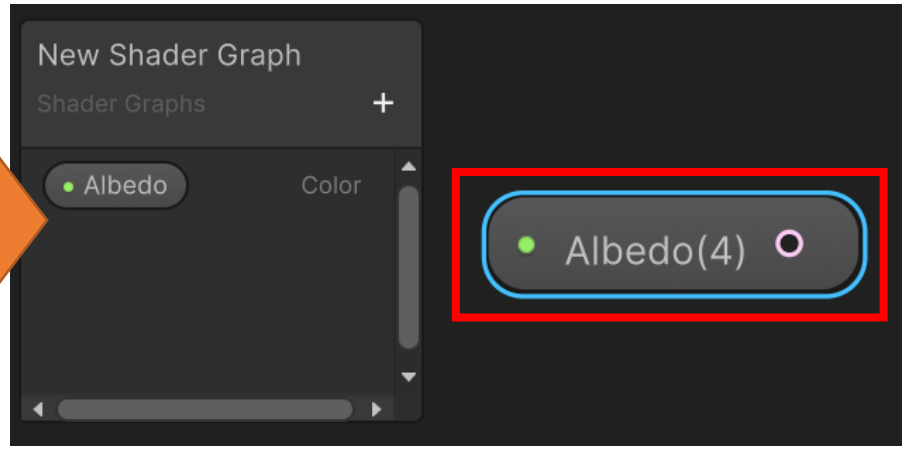
- 「Graph Inspector」の「Node Setting」
- デフォルト等を設定可能



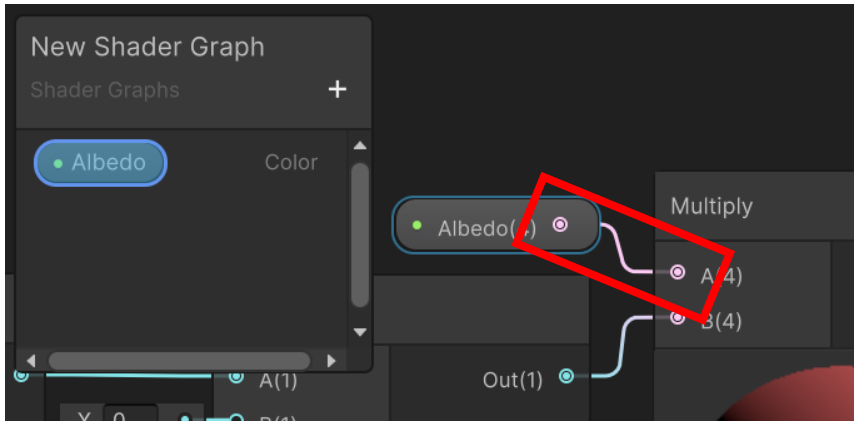
変数の使い方

- ブラックボードからShader Graphウィンドウにドラッグ＆ドロップ

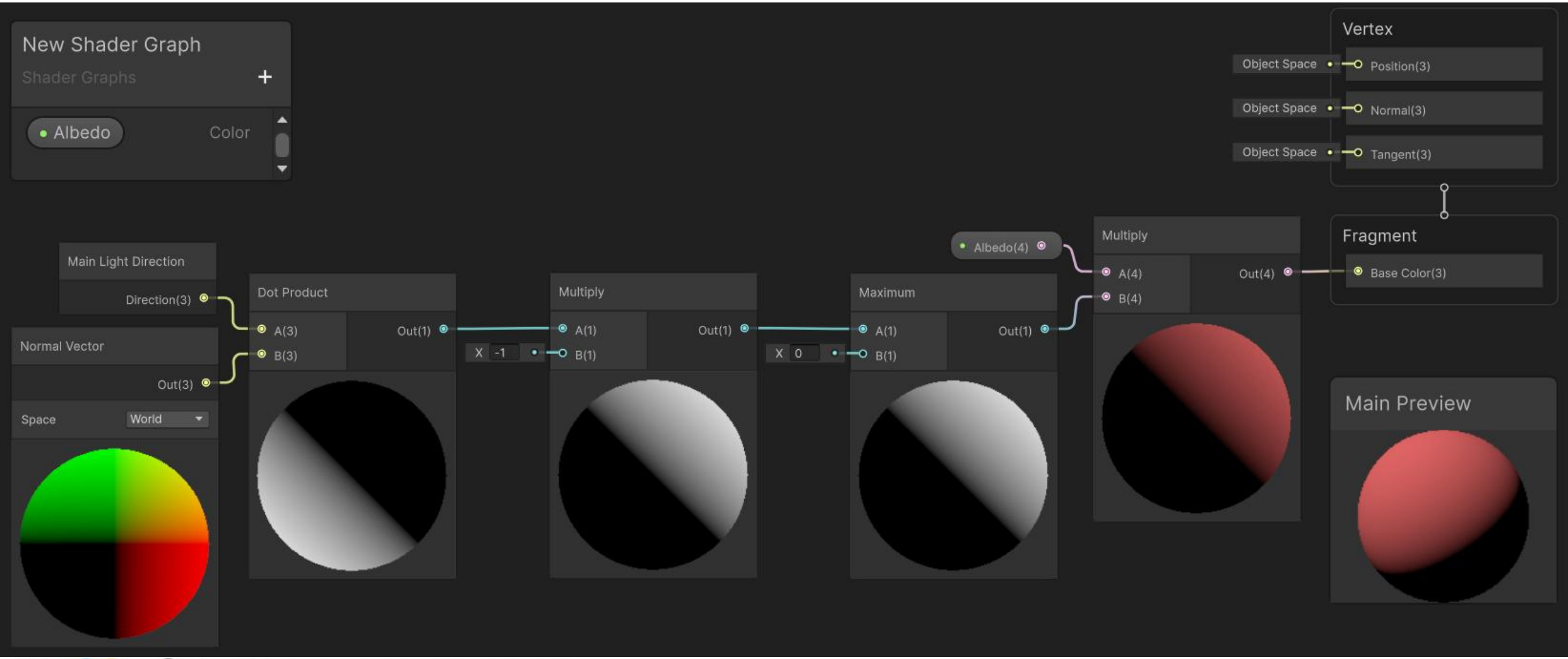




- 他のノードにつなげる
  - 選択中はブラックボードの変数も目立つ演出が行われる

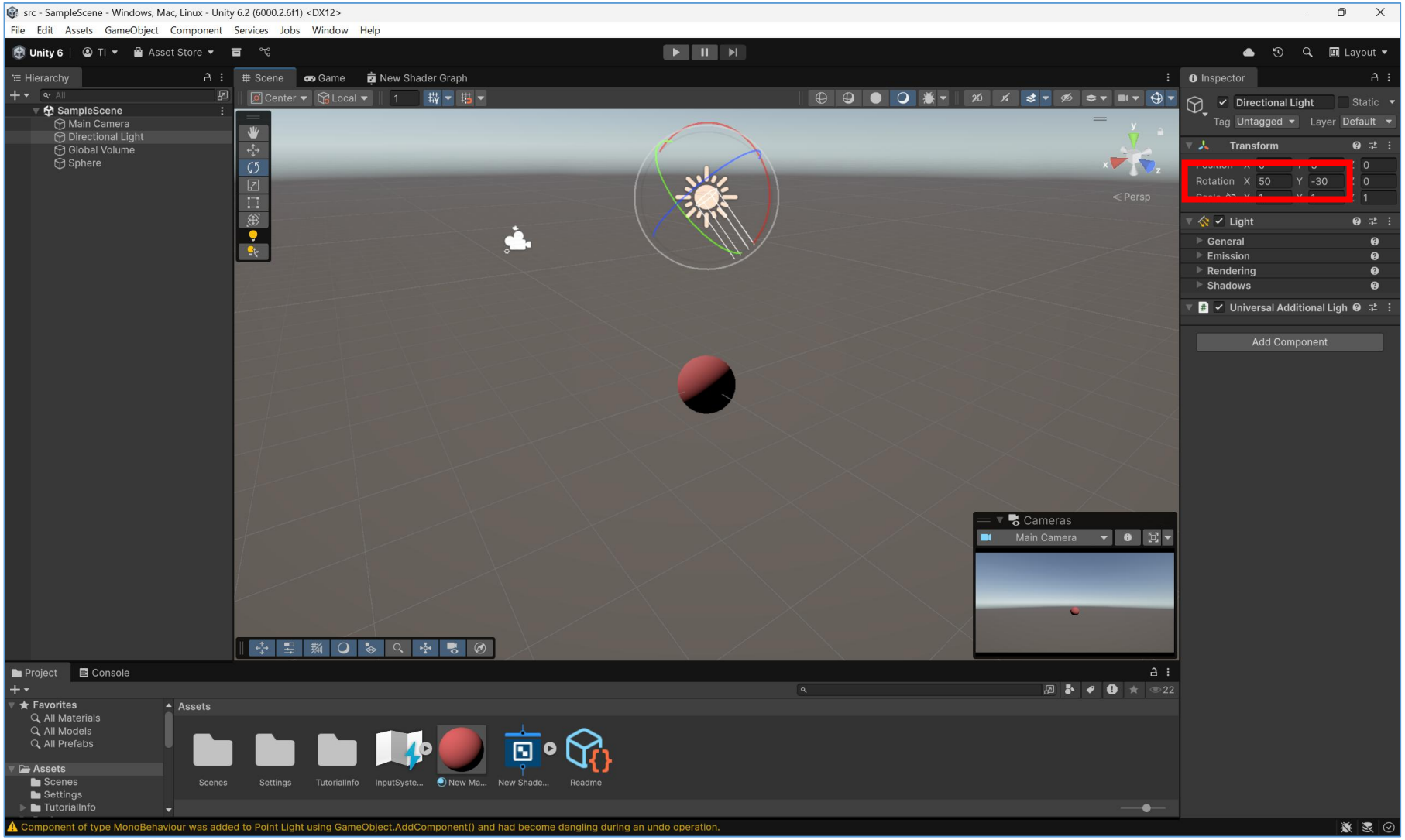


ランバート拡散反射


New Shader Graph
Shader Graphs
Albedo
Color
Vertex
Object Space
Position(3)
Object Space
Normal(3)
Object Space
Tangent(3)
Fragment
Base Color(3)
Main Light Direction
Direction(3)
Normal Vector
Out(3)
Space
World
Dot Product
A(3)
B(3)
Out(1)
Multiply
X -1
A(1)
B(1)
Out(1)
Maximum
X 0
A(1)
B(1)
Out(1)
Albedo(4)
Multiply
A(4)
B(4)
Out(4)
Main Preview

やってみよう

- 完成したら
  光源の向きを
  変更してみよう



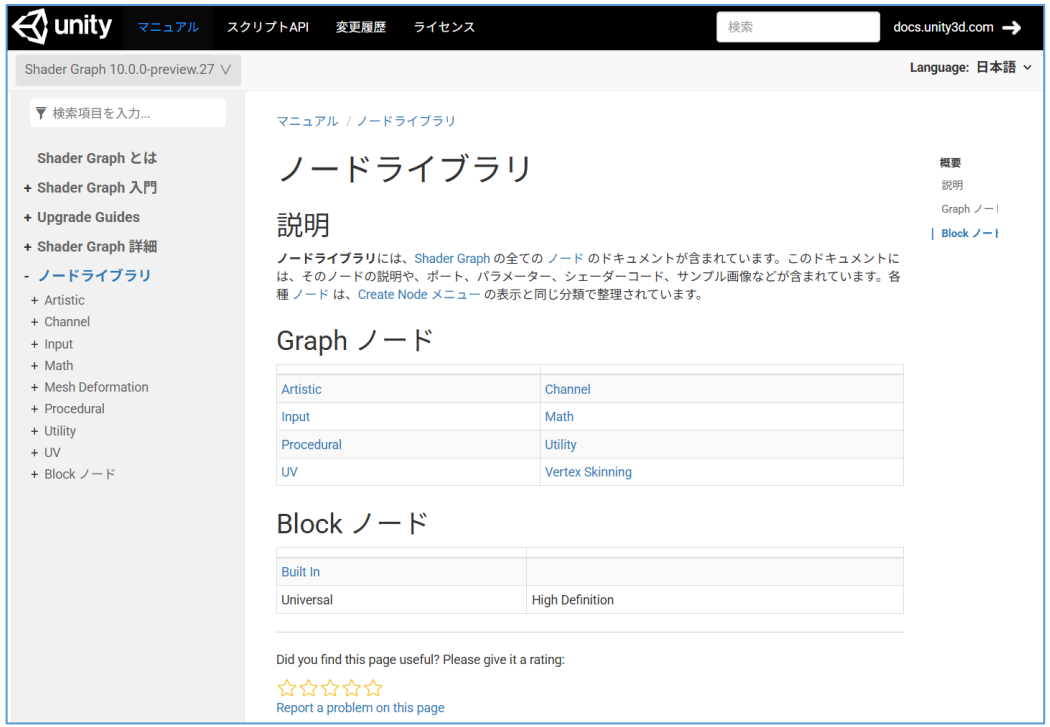
シェーダグラフ入門

- シェーダグラフの追加
- 色を変える
- ノードの追加
- ノードの種類
- Lit Shader Graph


Adjustment
Channel Mixer
Contrast
Hue
Invert Colors
Replace Color
Saturation
White Balance
Blend
Blend
Filter
Dither
Fade Transition
Mask
Channel Mask
Color Mask
Normal
Normal Blend
Normal From Height
Normal From Texture
Normal Reconstruct Z
Normal Strength
Normal Unpack
Utility
Colorspace Conversion
Channel
Append
Combine
Flip
Split
Swizzle
Custom Render Texture
Self
Size
Slice Index / Cubemap Face
Input
2D
Light Texture
Basic
Boolean
Color
Constant
Float
Integer
Slider
Time
Vector 2
Vector 3
Vector 4
Geometry
Bitangent Vector
Instance ID
Normal Vector
Position
Screen Position
Tangent Vector
UV
Vertex Color
Vertex ID
View Direction
View Vector
Gradient
Blackbody
Gradient
Sample Gradient
Lighting
Ambient
Baked GI
Main Light Direction
Reflection Probe
Matrix
Matrix 2x2
Matrix 3x3
Matrix 4x4
Transformation Matrix
Mesh Deformation
Compute Deformation
Linear Blend Skinning
Sprite Skinning
PBR
Dielectric Specular
Metal Reflectance
Scene
Camera
Eye Index
Fog
Object
Scene Color
Scene Depth
Scene Depth Difference
Screen
Texture
Calculate Level Of Detail Texture 2D
Cubemap Asset
Gather Texture 2D
Sample Cubemap
Sample Reflected Cubemap
Sample Texture 2D
Sample Texture 2D Array
Sample Texture 2D LOD
Sample Texture 3D
Sample Virtual Texture
Sampler State
Split Texture Transform
Texture 2D Array Asset
Texture 2D Asset
Texture 3D Asset
Texture Size
Universal
URP Sample Buffer
Math
Advanced
Absolute
Exponential
Length
Log
Modulo
Negate
Normalize
Posterize
Reciprocal
Reciprocal Square Root
Basic
Add
Divide
Multiply
Power
Square Root
Subtract
Derivative
DDX
DDXY
DDY
Interpolation
Inverse Lerp
Lerp
Smoothstep
Matrix
Matrix Construction
Matrix Determinant
Matrix Split
Matrix Transpose
Range
Clamp
Fraction
Maximum
Minimum
One Minus
Random Range
Remap
Saturate
Round
Ceiling
Floor
Round
Sign
Step
Truncate
Trigonometry
Arccosine
Arcsine
Arctangent
Arctangent2
Cosine
Degrees To Radians
Hyperbolic Cosine
Hyperbolic Sine
Hyperbolic Tangent
Radians To Degrees
Sine
Tangent
Vector
Cross Product
Distance
Dot Product
Fresnel Effect
Projection
Reflection
Refract
Rejection
Rotate About Axis
Sphere Mask
Transform
Wave
Noise Sine Wave
Sawtooth Wave
Square Wave
Triangle Wave
Procedural
Checkerboard
Noise
Gradient Noise
Simple Noise
Voronoi
Shape
Ellipse
Polygon
Rectangle
Rounded Polygon
Rounded Rectangle
Properties
Property: Albedo
SpeedTree
SpeedTree8Billboard
SpeedTree8CameraFacingLeafEffect
SpeedTree8ColorAlpha
SpeedTree8InterpolatedNormals
SpeedTree8LODTransition
SpeedTree8Wind
SpeedTree9CameraFacingLeafEffect
SpeedTree9Wind
Utility
Custom Function
Preview
Logic
All
And
Any
Branch
Comparison
Is Front Face
Is Infinite
Is NaN
Nand
Not
Or
UV
Flipbook
Parallax Mapping
Parallax Occlusion Mapping
Polar Coordinates
Radial Shear
Rotate
Spherize
Tiling And Offset
Triplanar
Twirl

Shader Graph 10.0.0-preview 27 マニュアル

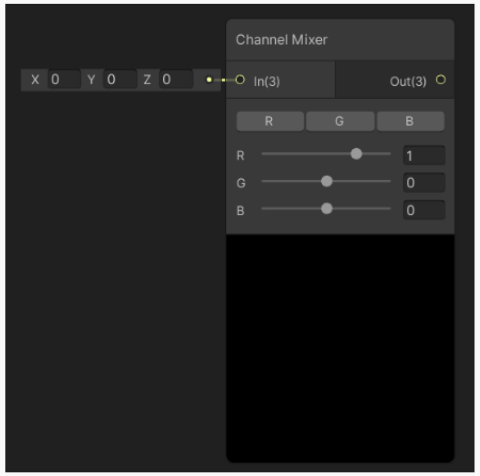
- https://docs.unity3d.com/ja/Packages/com.unity.shadergraph@10.0/manual/Node-Library.html
  - もう少し新しい日本語版のマニュアルが欲しいですね。。。
    - 英語も勉強しましょう



Artistic ノード

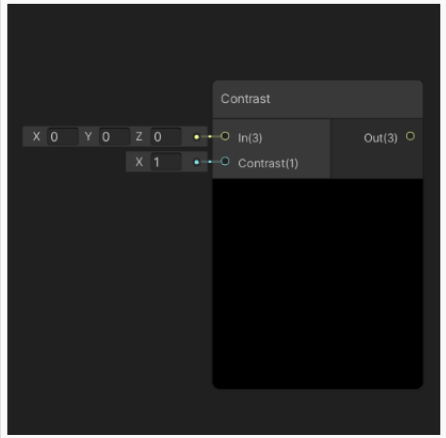
Adjustment

Channel Mixer



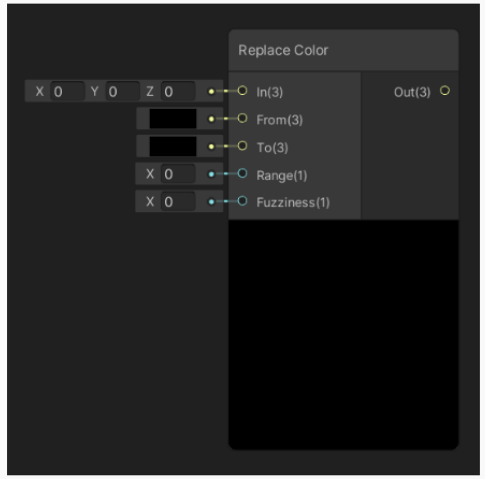
入力 In の各チャンネルが、各出力チャンネルに寄与する量を制御します。

Contrast



入力 In のコントラストを入力 Contrast の量によって調整します。

Replace Color

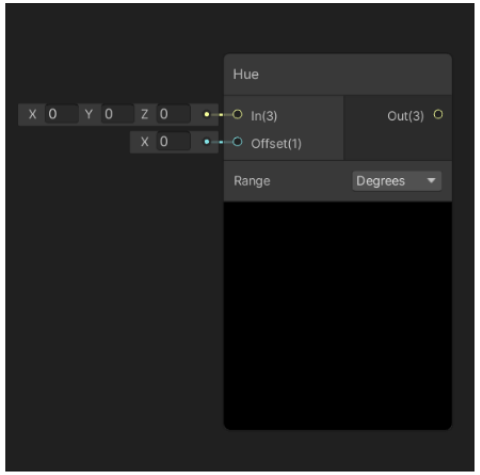


入力 From の値と等しい入力 In の値を、入力 To の値に置き換えます。

Saturation

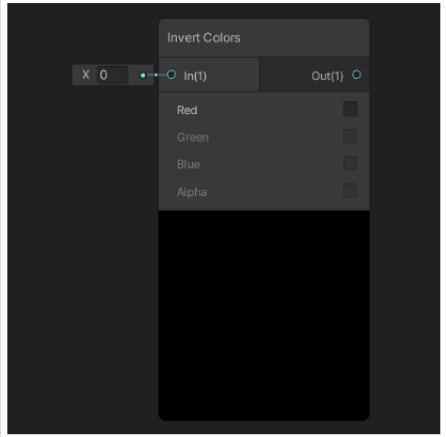
入力 In の彩度を入力 Saturation の値で調節します。

Hue



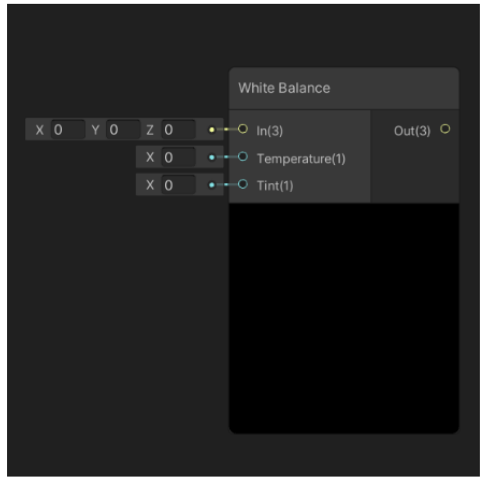
入力 In の色相を入力 Offset の量だけオフセットします。

Invert Colors



チャンネルごとに入力 In の色を反転します。

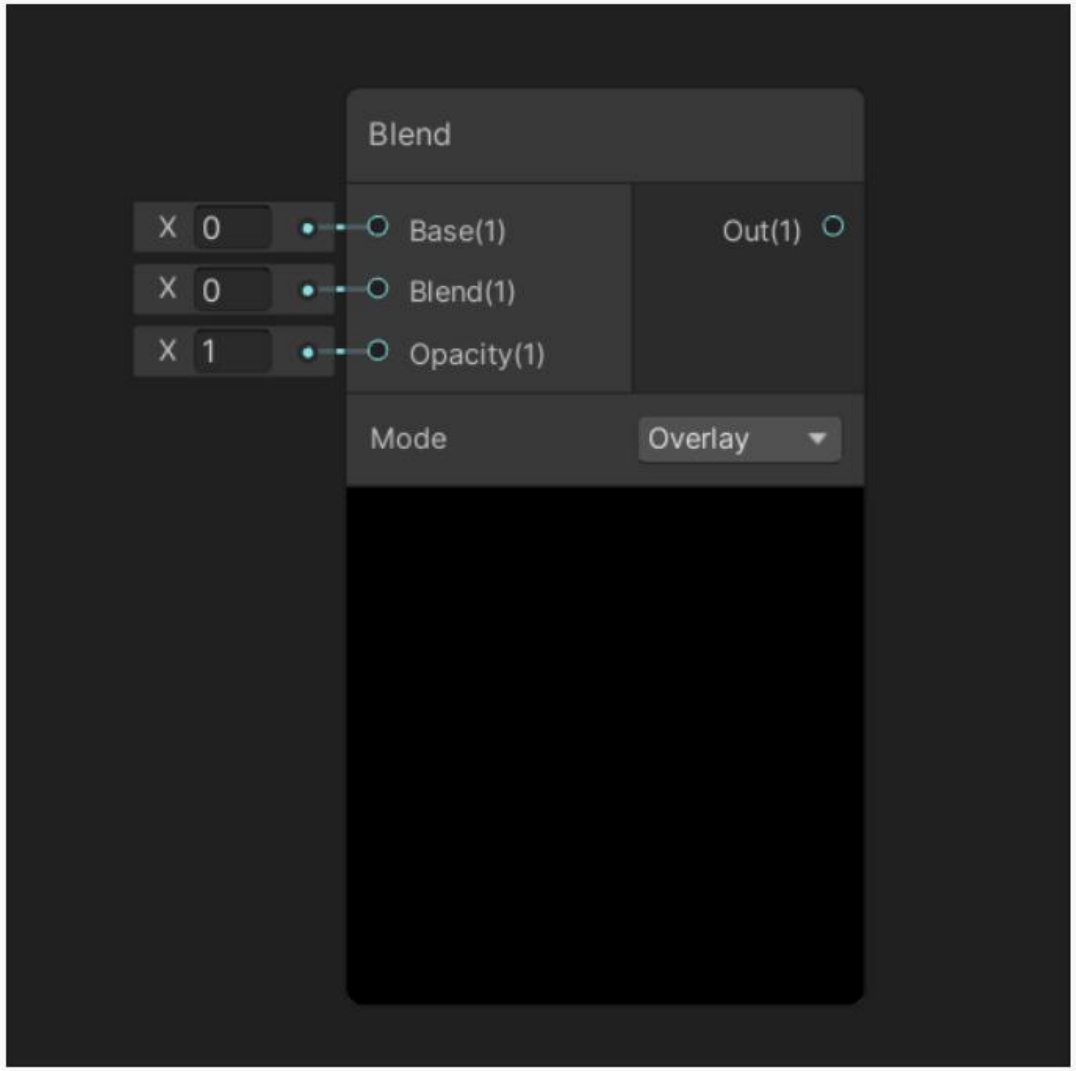
White Balance



入力 In の色温度と色合い (ティント) を入力 Temperature および入力 Tint の値で調節します。

Blend

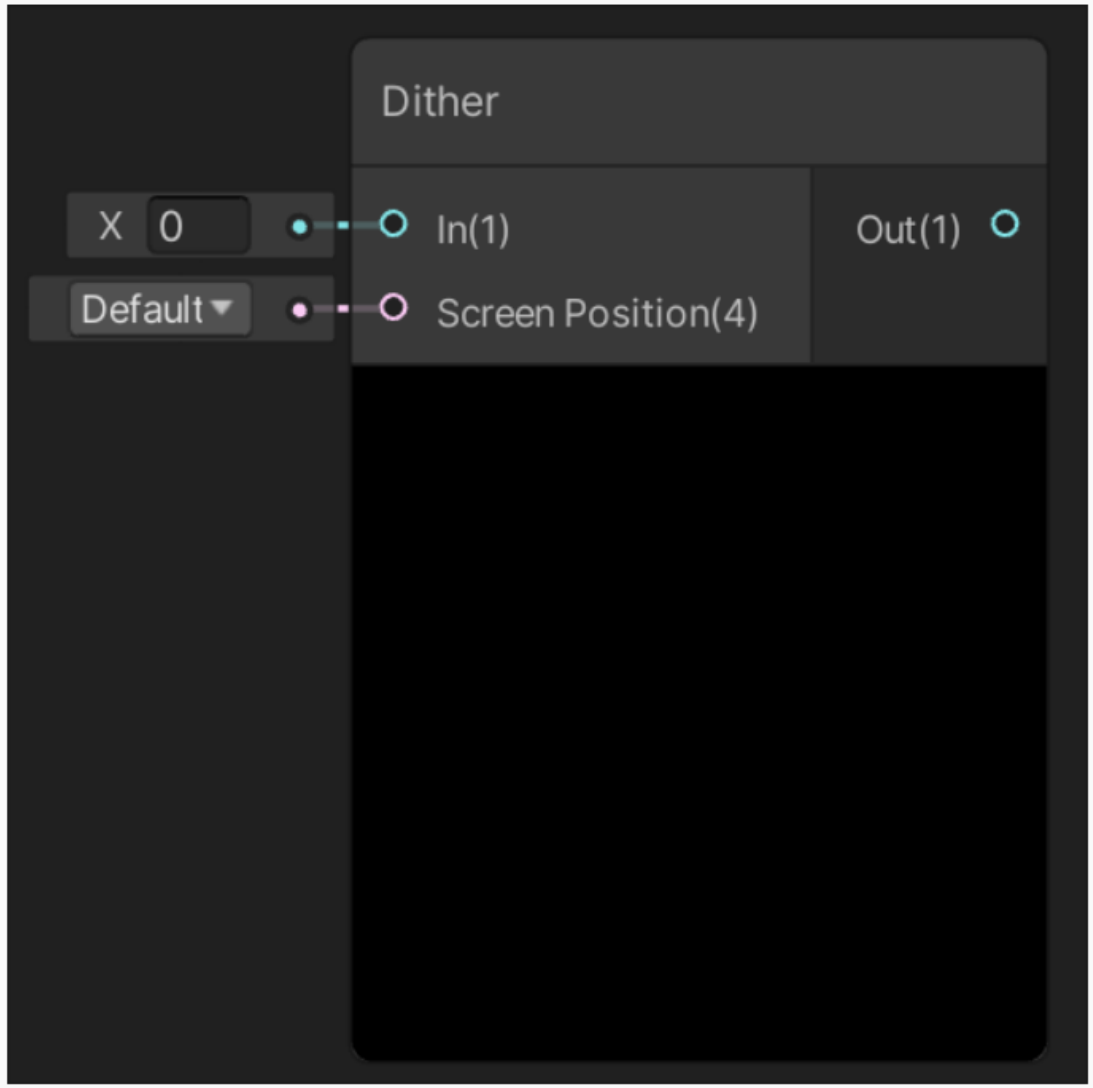
Blend



パラメーター Mode で定義されたブレンドモードを使用して、入力 Blend の値を入力 Base にブレンドします。

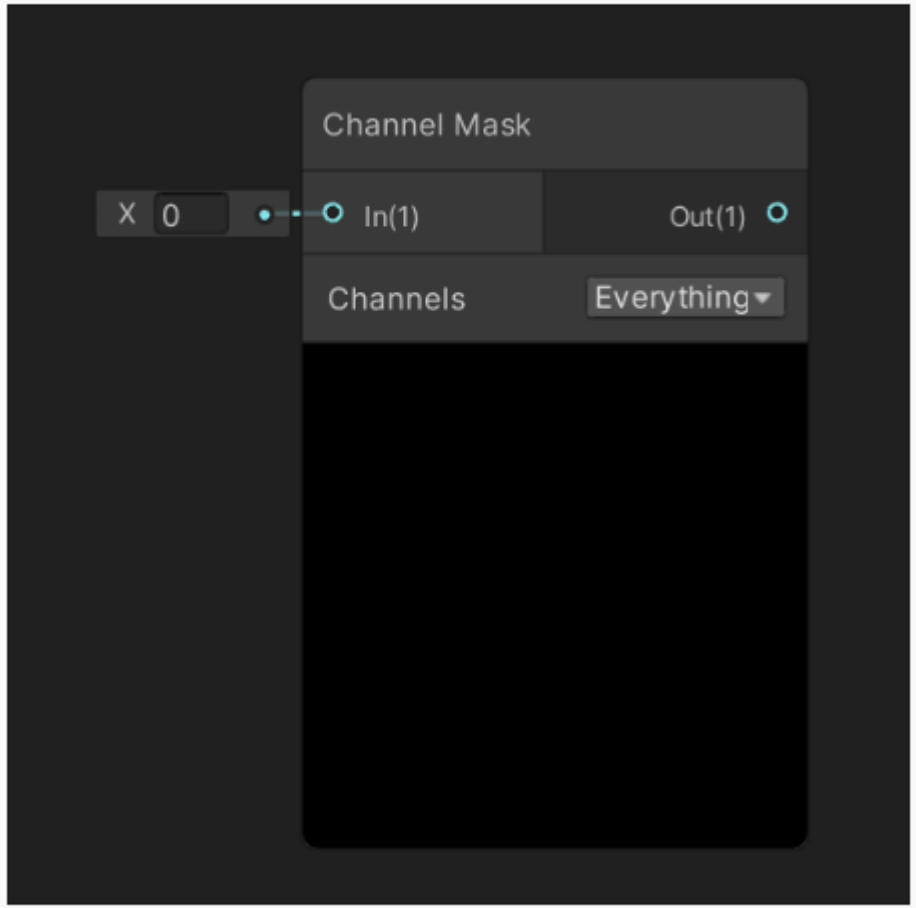
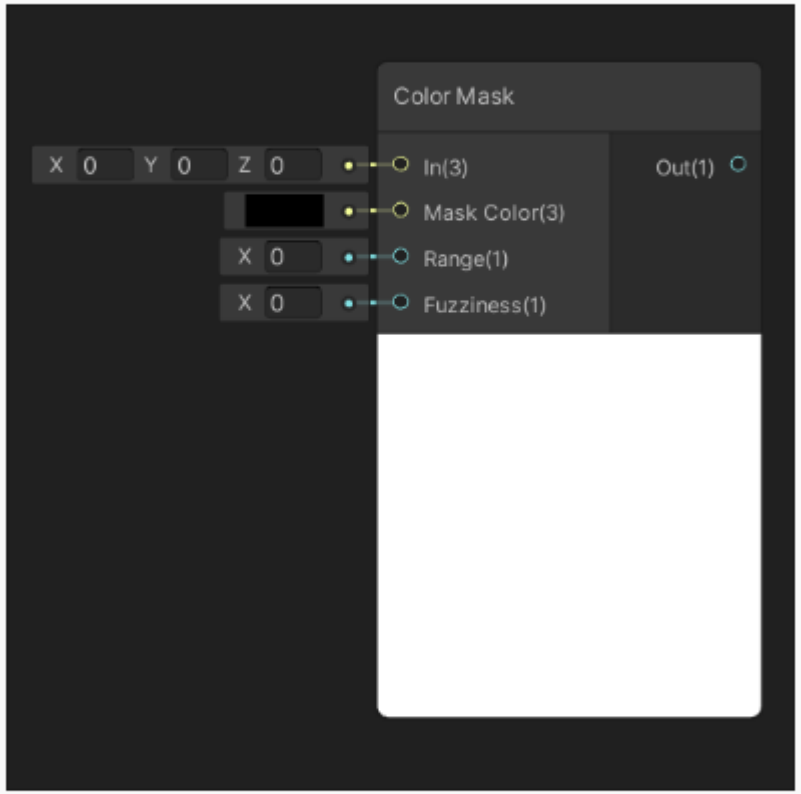
Filter

Dither

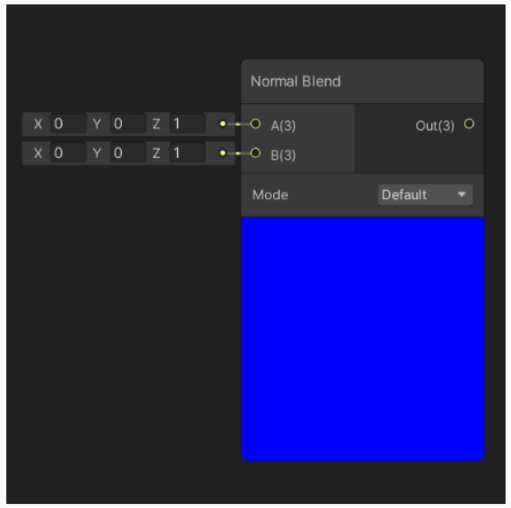
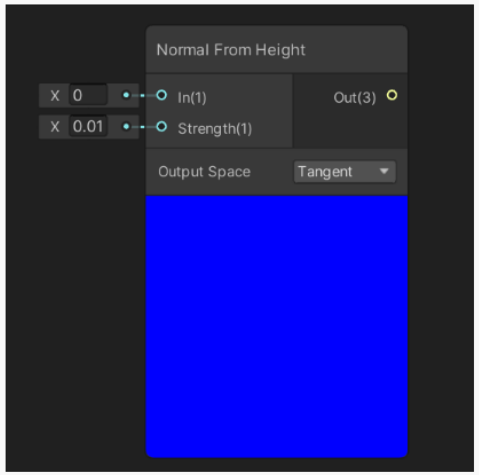
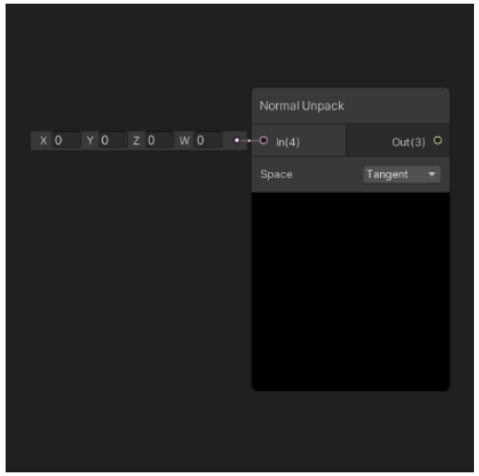


ディザーは量子化誤差をランダム化するために使用される意図的なノイズです。画像内におけるカラーバンディングなどの大規模なパターン (繰り返し模様) を防止するために使用されます。

Mask

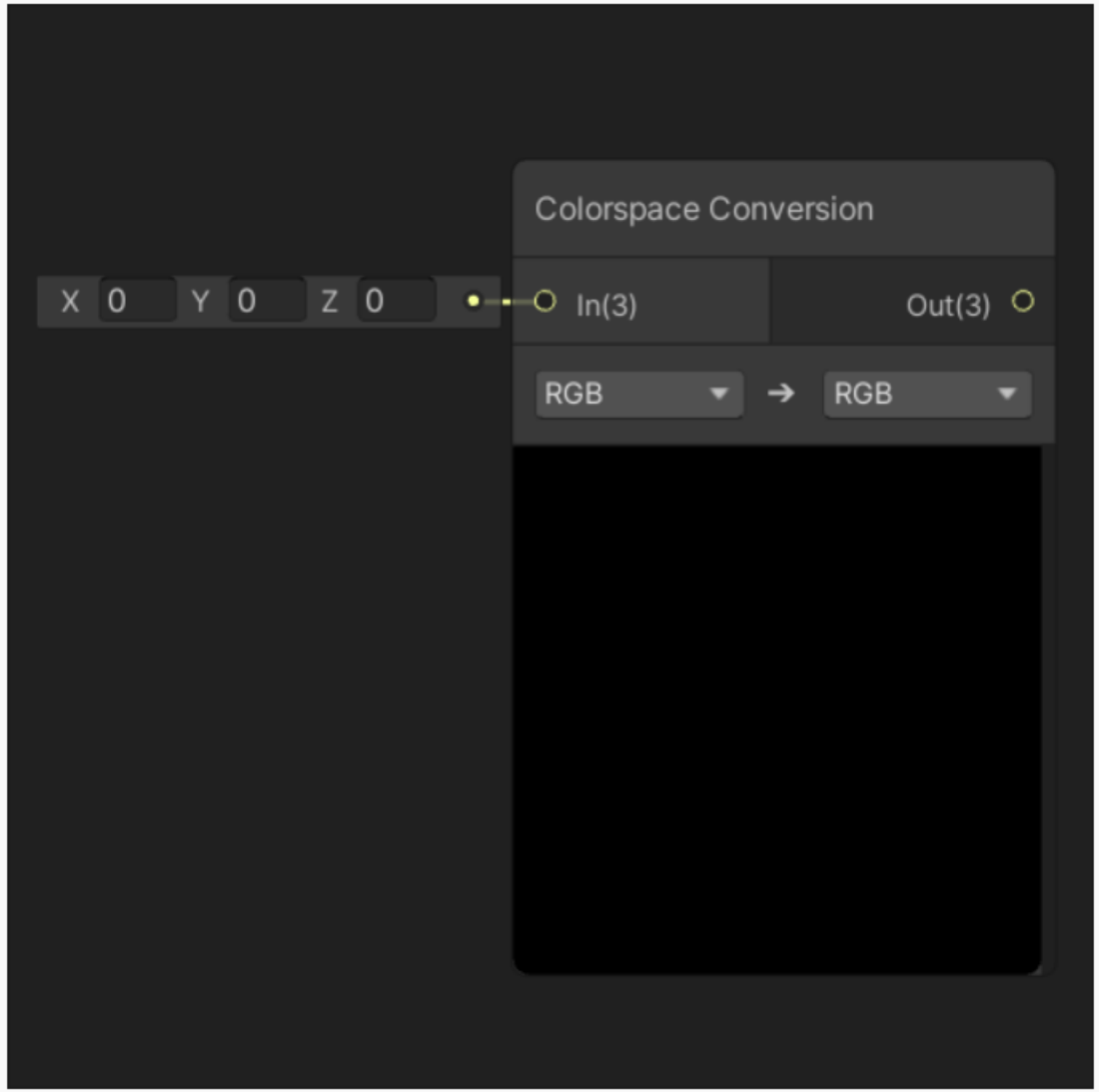
Channel Mask	Color Mask
	
Channels のドロップダウンメニューで選択されたチャンネル上で、入力 In の値をマスクします。	入力 Mask Color と等しい入力 In の値から、マスクを作成します。

Normal

Normal Blend	Normal From Height
	
入力 A で定義される法線マップと入力 B で定義される 2 つの法線マップをブレンドします。	入力 Texture で定義されるハイトマップから法線マップを作成します。
Normal Strength	**Normal Unpack**
	
入力 In で定義される法線マップの強度を入力 Strength の値で調節します。	入力 In で定義される法線マップを展開します。

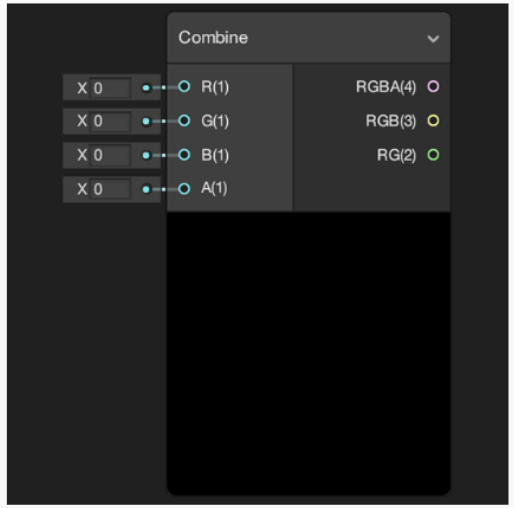
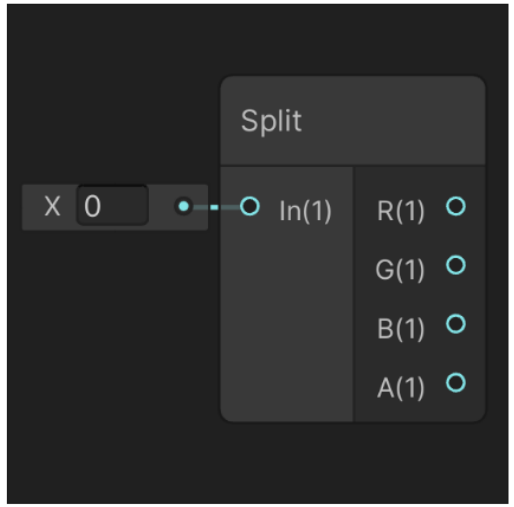
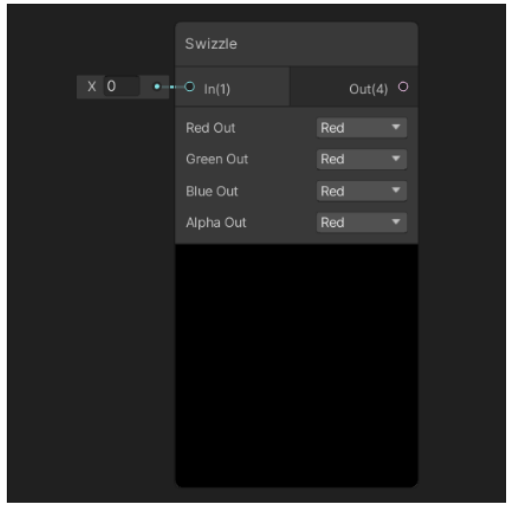
Utility

Colorspace Conversion



入力 In の値をある色空間から別の色空間へ変換した結果を返します。

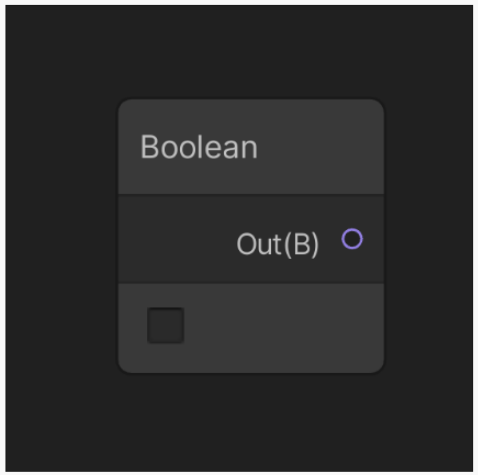
Channel ノード

Combine	Flip
	
入力 In の各チャンネルが、各出力チャンネルに寄与する量を制御します。	入力 In のコントラストを入力 Contrast の量によって調整します。
Split	**Swizzle**
	
入力 In の色相を入力 Offset の量だけオフセットします。	チャンネルごとに入力 In の色を反転します。

Input ノード

Basic (基本)

Boolean

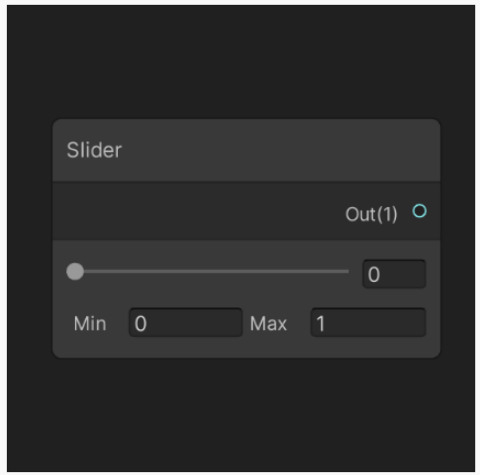


定数 Boolean の値 (ブール値) をシェーダー上で定義します。

Color

Color フィールドを使用して定数 Vector 4 の値をシェーダー上で定義します。

Slider

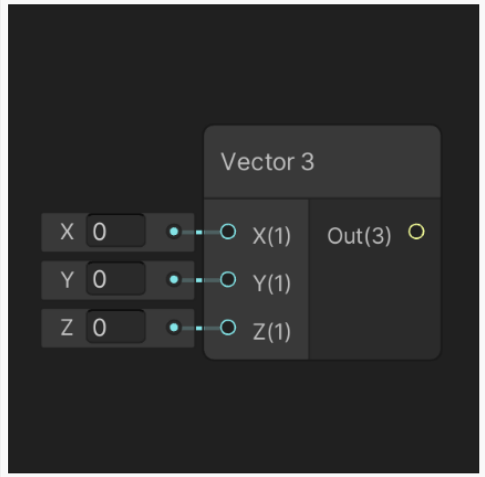


Slider フィールドを使用して定数 Vector 1 の値をシェーダー上で定義します。

Time

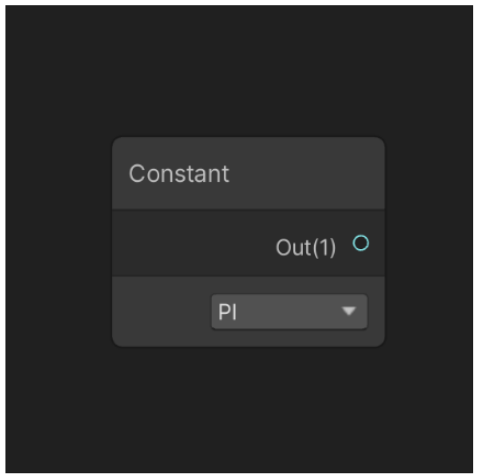
シェーダー上で各種の Time パラメーターを使用できます。

Vector 3



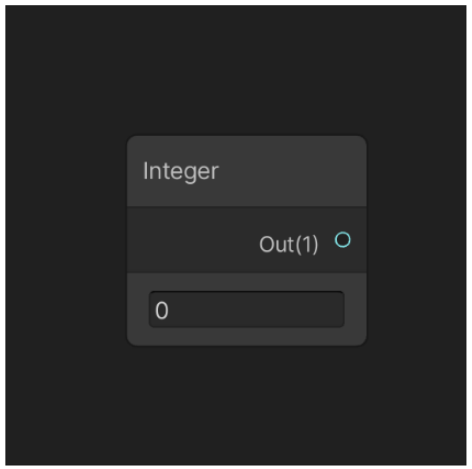
シェーダー上で Vector 3 値を定義します。

Constant



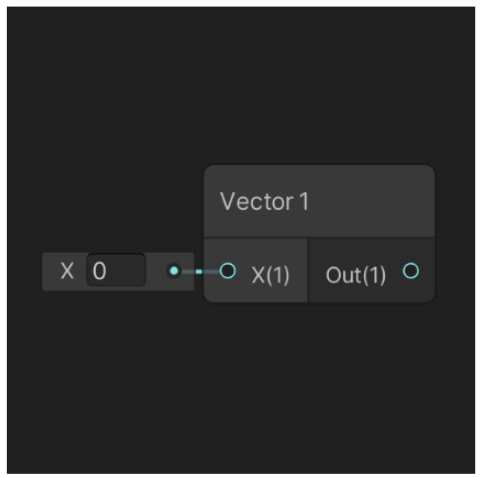
数学定数 Vector 1 の値をシェーダー上で定義します。

Integer



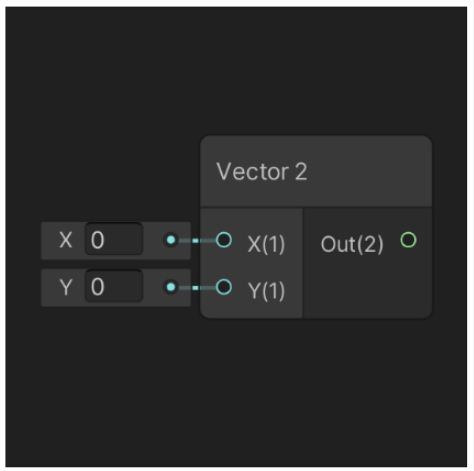
Integer フィールドを使用して定数 Vector 1 の値をシェーダー上で定義します。

Vector 1



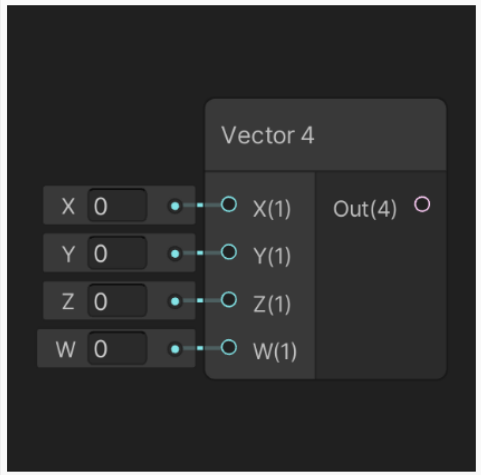
シェーダー上で Vector 1 の値を定義します。

Vector 2



シェーダー上で Vector 2 値を定義します。

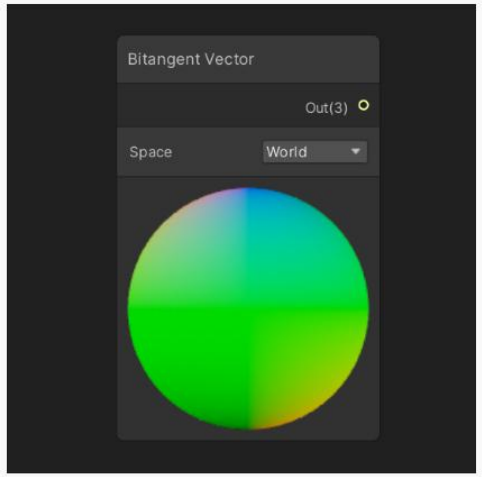
Vector 4



シェーダー上で Vector 4 値を定義します。

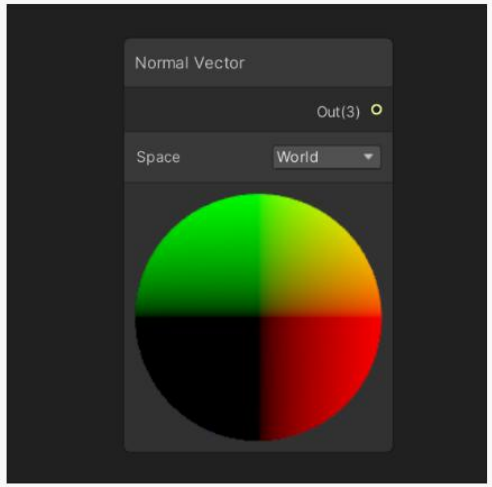
Geometry (ジオメトリ)

Bitangent Vector



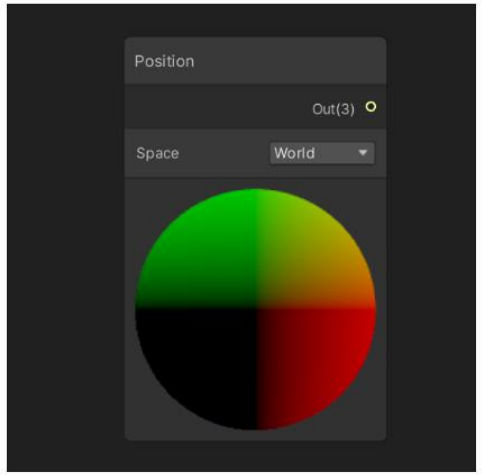
メッシュの頂点あるいはフラグメントの Bitangent Vector (従接線ベクトル) へのアクセスを提供します。

Normal Vector



メッシュの頂点あるいはフラグメントの Normal Vector (法線ベクトル) へのアクセスを提供します。

Position

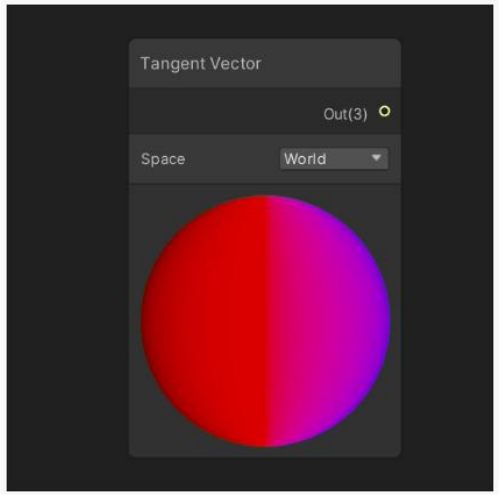


メッシュの頂点あるいはフラグメントの Position (位置) へのアクセスを提供します。

Screen Position

メッシュの頂点あるいはフラグメントの Screen Position (スクリーン座標位置) へのアクセスを提供します。

Tangent Vector

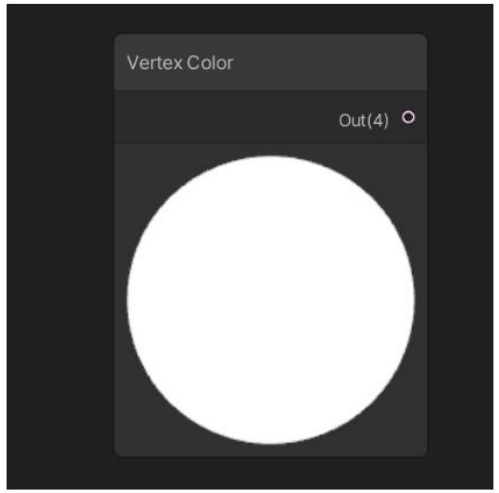


メッシュの頂点あるいはフラグメントの Tangent Vector (接線ベクトル) へのアクセスを提供します。

UV

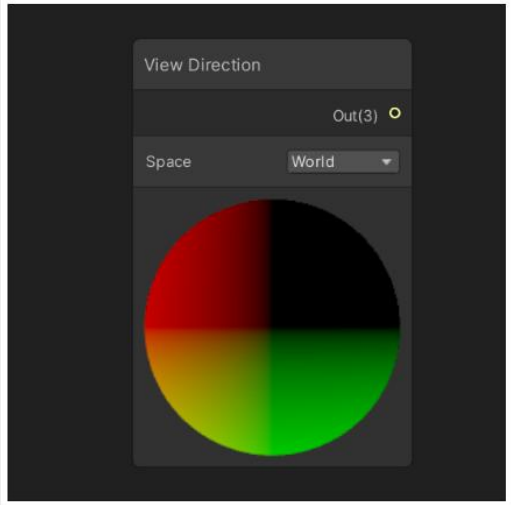
メッシュの頂点あるいはフラグメントの UV 座標へのアクセスを提供します。

Vertex Color



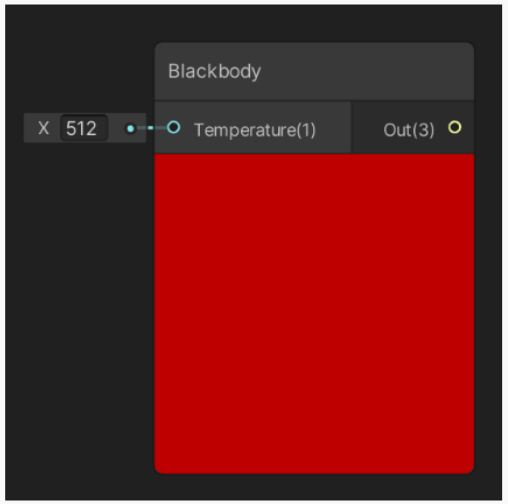
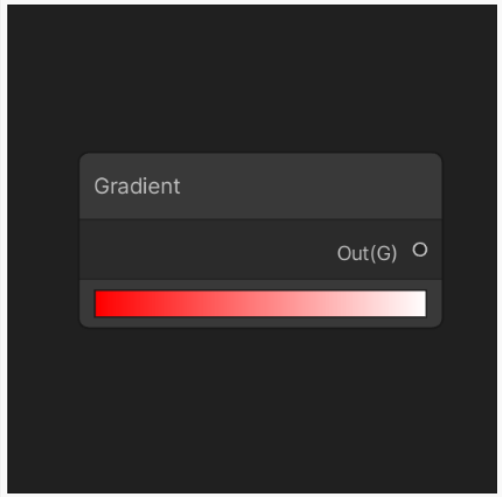
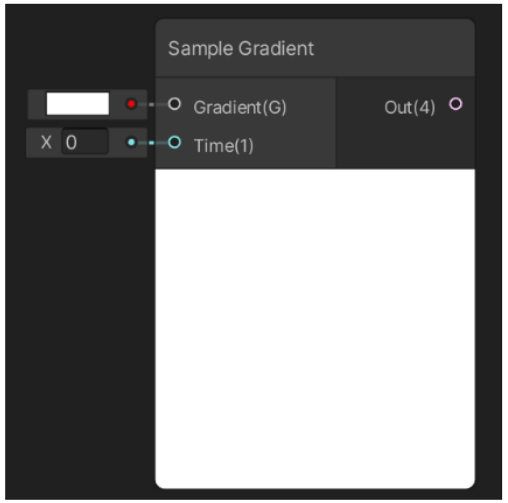
メッシュの頂点あるいはフラグメントの Vertex Color (頂点色) 値へのアクセスを提供します。

View Direction

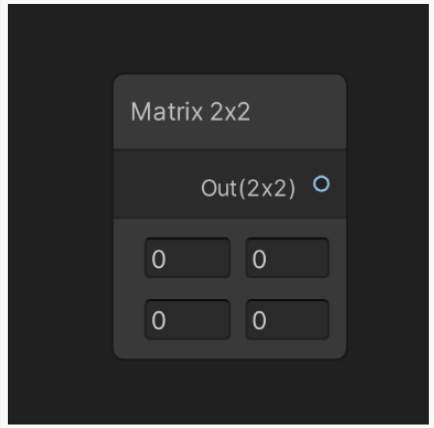
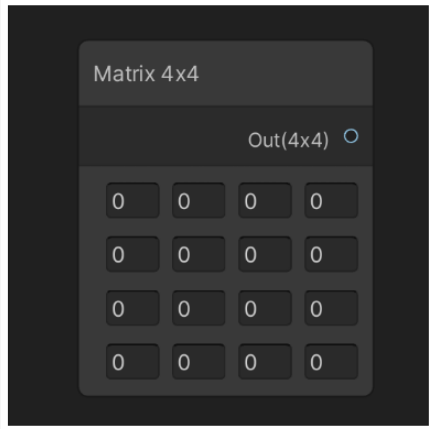
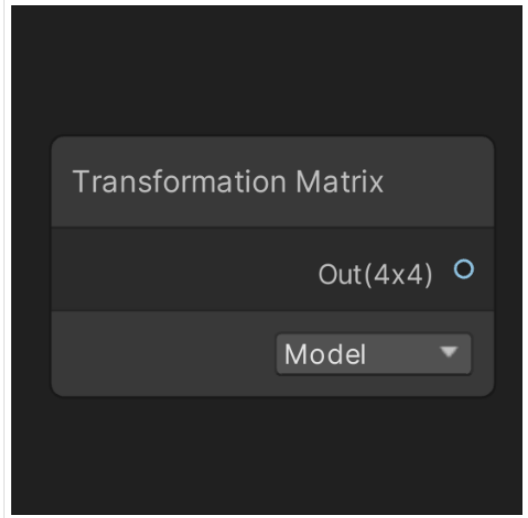


メッシュの頂点あるいはフラグメントの View Direction (ビュー方向) ベクトルへのアクセスを提供します。

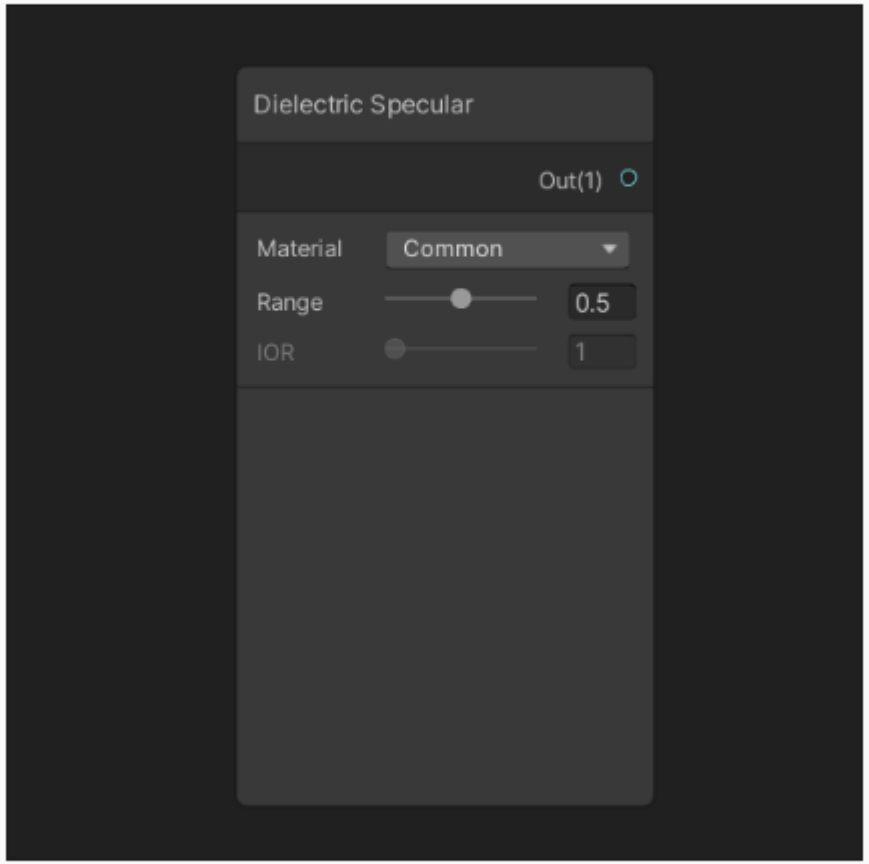
Gradient (グラデーション)

Blackbody	Gradient
	
温度 (単位: ケルビン) の入力から、放射によるグラデーションをサンプリングします。	定数 Gradient をシェーダー上で定義します。
Sample Gradient	
	
Time の入力に応じて Gradient (グラデーション) をサンプリングします。	

Matrix (行列)

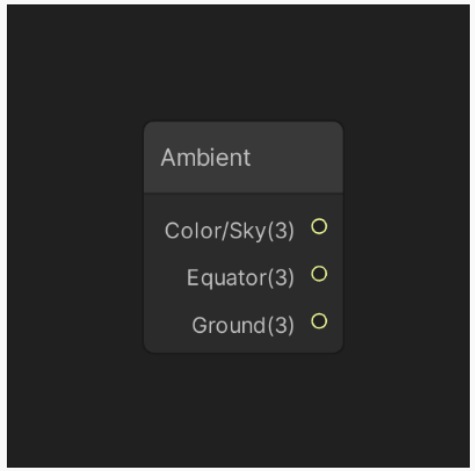
Matrix 2x2	Matrix 3x3
	
定数 Matrix 2x2 の値をシェーダー上で定義します。	定数 Matrix 3x3 の値をシェーダー上で定義します。
Matrix 4x4	**Transformation Matrix**
	
定数 Matrix 4x4 の値をシェーダー上で定義します。	デフォルトの Unity Transformation Matrix 用の定数 Matrix 4x4 の値をシェーダー上で定義します。

PBR (物理ベースレンダリング)

Dielectric Specular	Metal Reflectance
	
物理ベースのマテリアルの、Dielectric Specular (誘電体スペキュラー) の F0 値を返します。	物理ベースのマテリアルの、Metal Reflectance (金属の反射率) の値を返します。

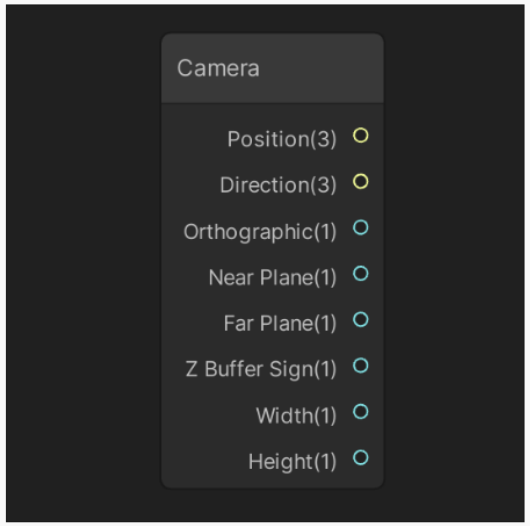
Scene (シーン)

Ambient



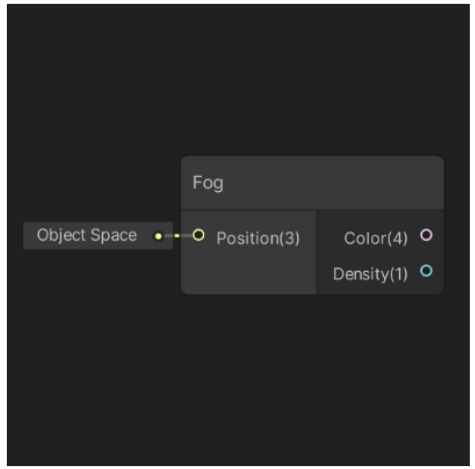
シーンの Ambient (環境) の色の値へのアクセスを提供します。

Camera



現在の Camera (カメラ) の各種パラメーターへのアクセスを提供します。

Fog



シーンの Fog (フォグ) パラメーターへのアクセスを提供します。

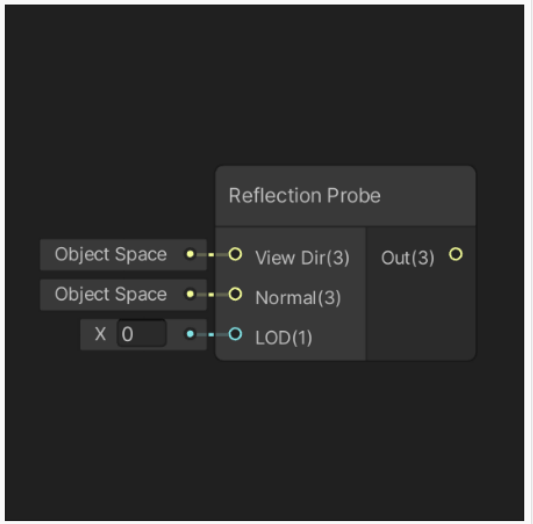
Baked GI

頂点あるいはフラグメントの位置における Baked GI の値へのアクセスを提供します。

Object

オブジェクト (Object) の各種パラメーターへのアクセスを提供します。

Reflection Probe

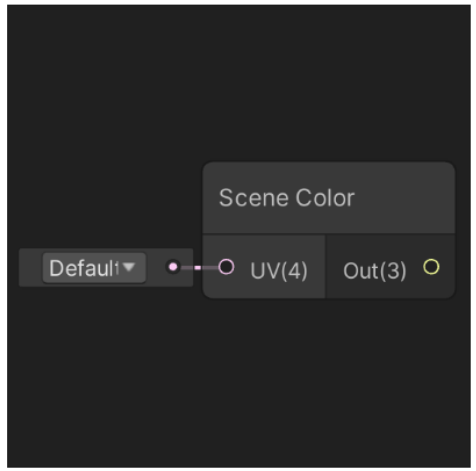


オブジェクトに最も近い Reflection Probe (リフレクションプローブ) へのアクセスを提供します。

Screen

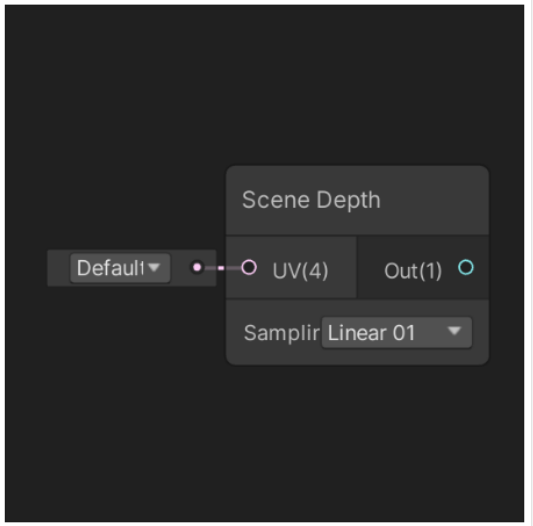
画面のパラメーターへのアクセスを提供します。

Scene Color



現在のカメラ (Camera) のカラーバッファへのアクセスを提供します。

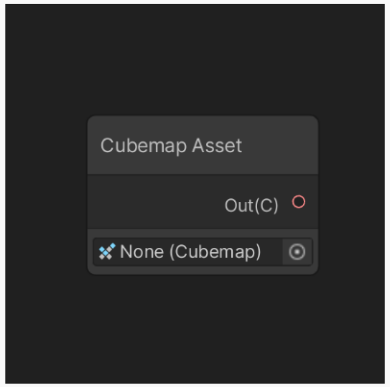
Scene Depth



現在の Camera (カメラ) の深度バッファへのアクセスを提供します。

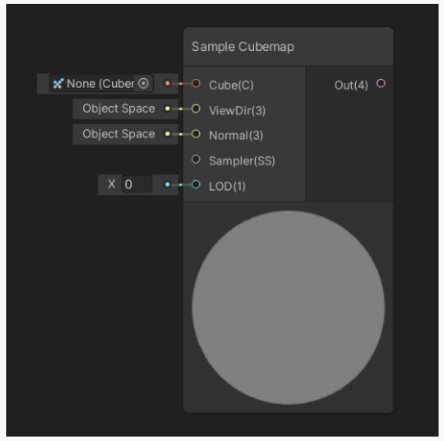
Texture (テクスチャー)

Cubemap Asset



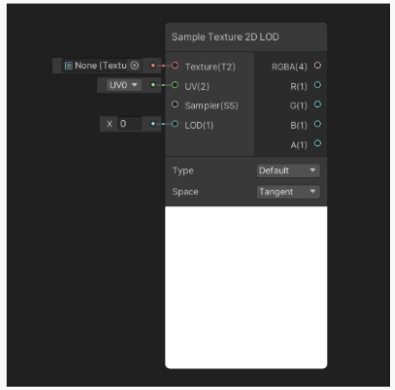
シェーダー内で使用する定数 Cubemap Asset を定義します。

Sample Cubemap



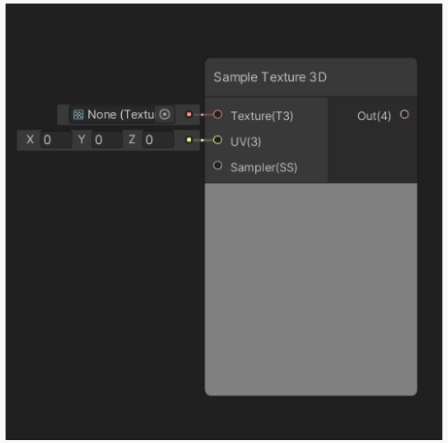
Cubemap (キューブマップ) をサンプリングし、シェーダー内で使用する Vector 4 の色の値を返します。

Sample Texture 2D LOD



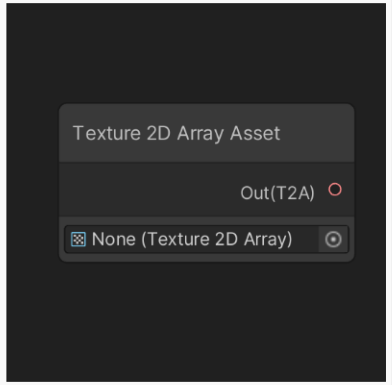
特定の LOD で Texture 2D をサンプリングし、シェーダー内で使用する色の値を返します。

Sample Texture 3D



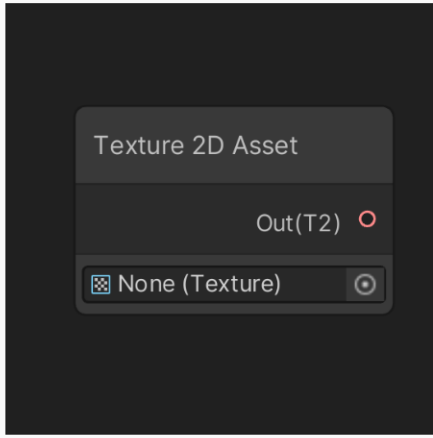
Texture 3D をサンプリングし、シェーダー内で使用する色の値を返します。

Texture 2D Array Asset



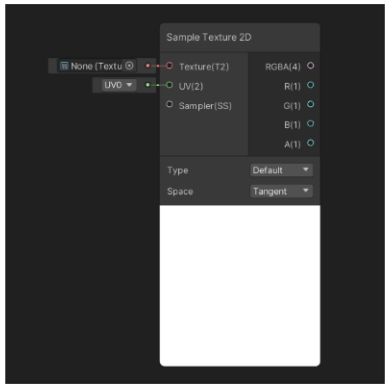
シェーダー内で使用する定数 Texture 2D Array Asset を定義します。

Texture 2D Asset



シェーダー内で使用する定数 Texture 2D Asset を定義します。

Sample Texture 2D

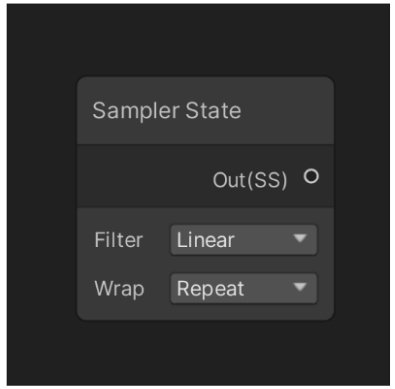


Texture 2D をサンプリングし、シェーダー内で使用する色の値を返します。

Sample Texture 2D Array

特定のインデックスで Texture 2D Array をサンプリングし、シェーダー内で使用する色の値を返します。

Sampler State

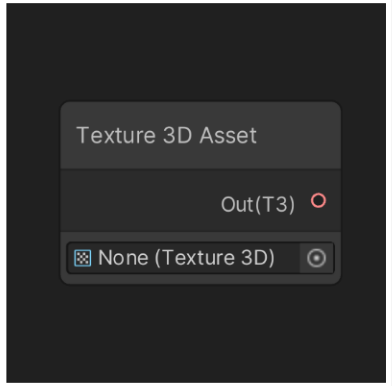


テクスチャーのサンプリング用に Sampler State (サンプラー状態) を定義します。

Texel Size

Texture 2D 入力のテクセルサイズの Width(幅) と Height (高さ) を返します。

Texture 3D Asset

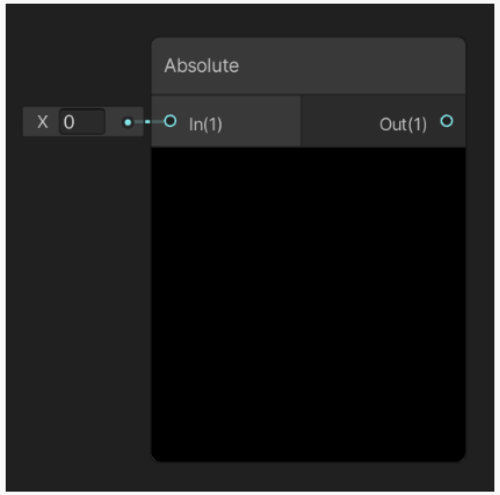


シェーダー内で使用する定数 Texture 3D Asset を定義します。

Math (数学) ノード

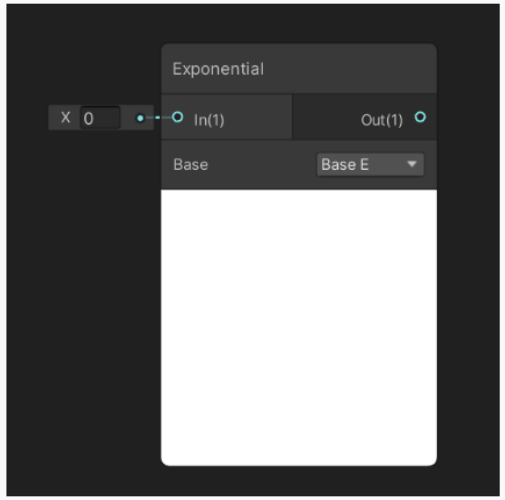
Advanced (算術関数)

Absolute



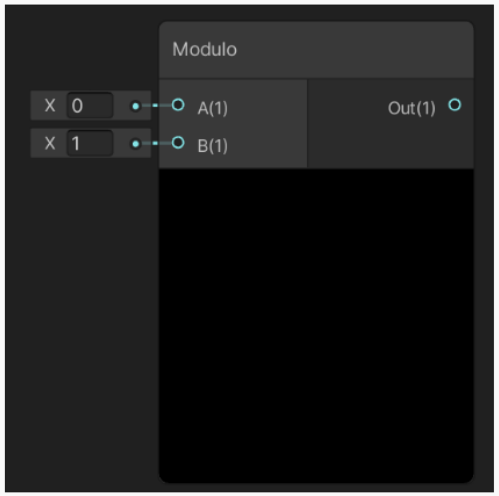
入力 In の絶対値を返します。

Exponential



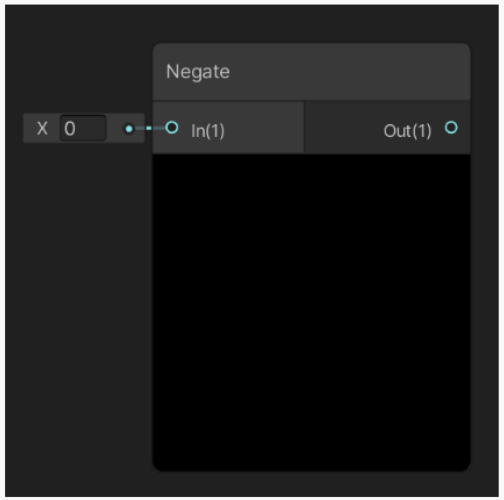
入力 In の指数値を返します。

Modulo



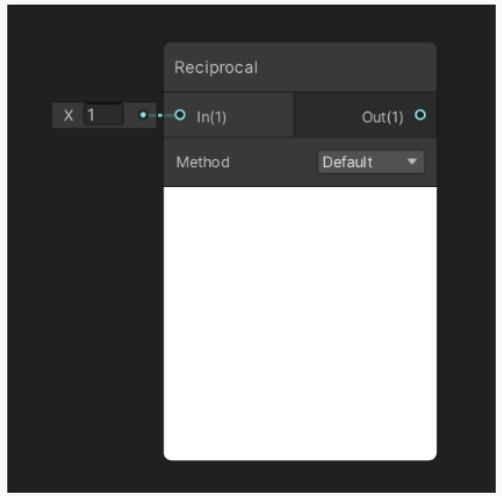
入力 A を入力 B で割った余りを返します。

Negate



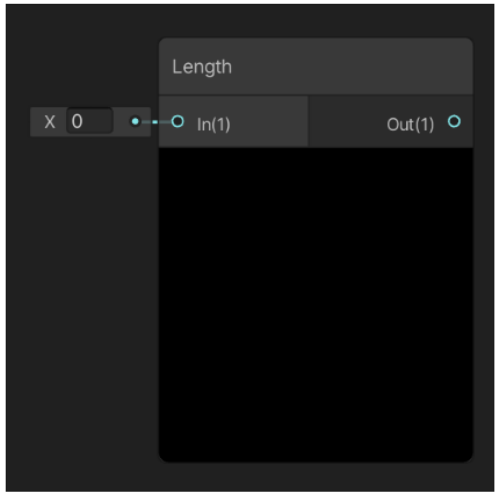
入力 In の逆数を返します。

Reciprocal



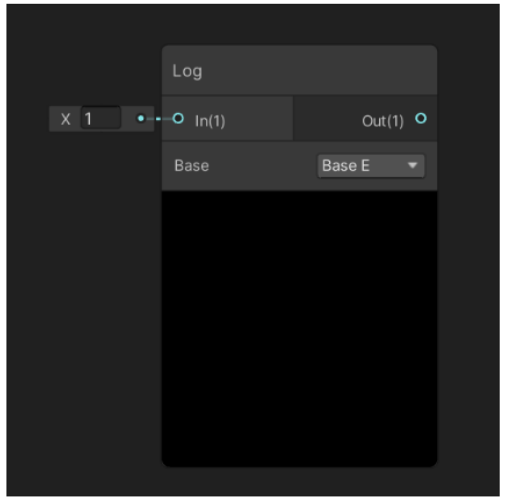
1 を入力 In で割った結果を返します。

Length



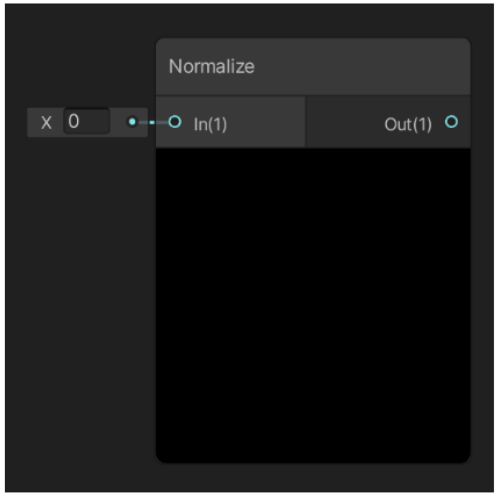
入力 In の長さを返します。

Log



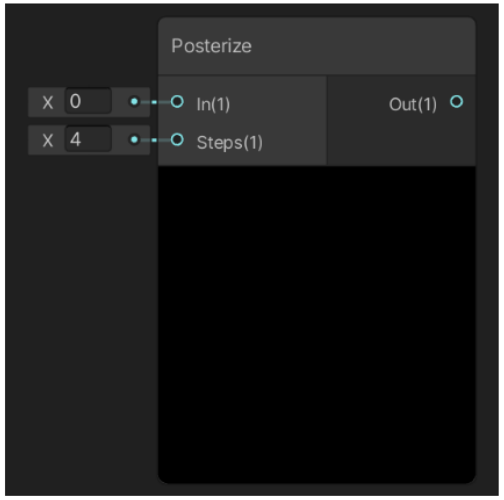
入力 In の対数を返します。

Normalize



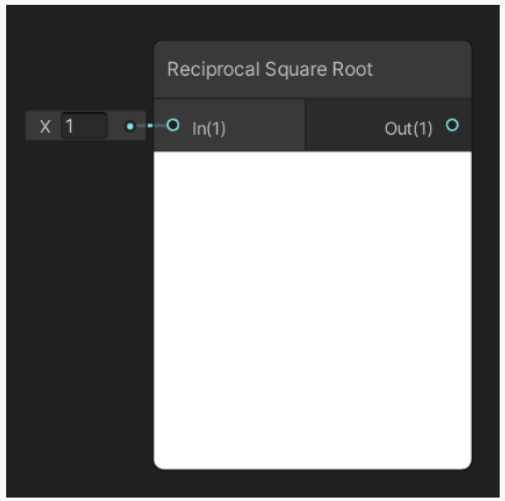
入力 In の正規化ベクトルを返します。

Posterize



入力 In を、入力 Steps の値で定義された値の数に変換した結果を返します。

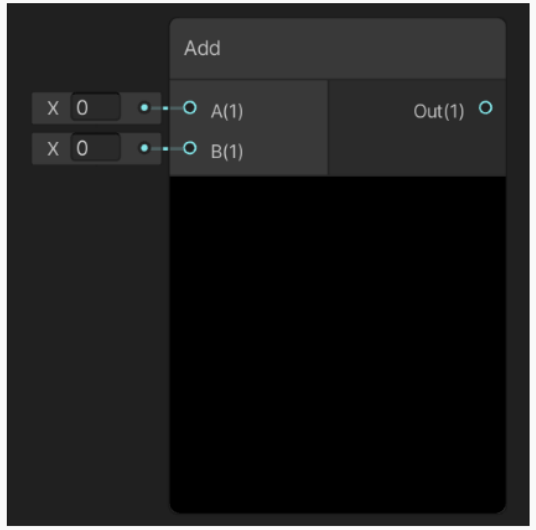
Reciprocal Square Root



1 を入力 In の平方根で割った結果を返します。

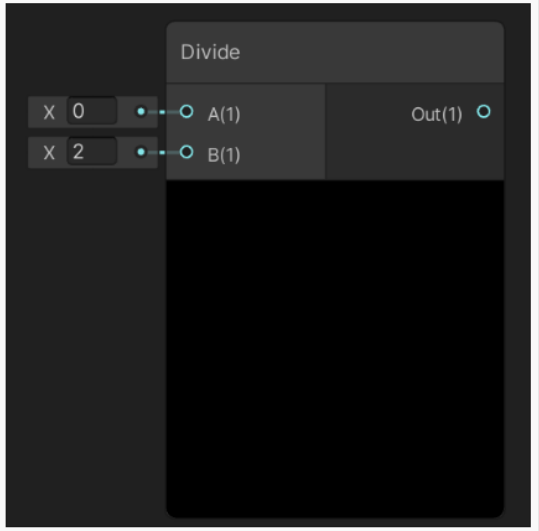
Basic (基本算術関数)

Add



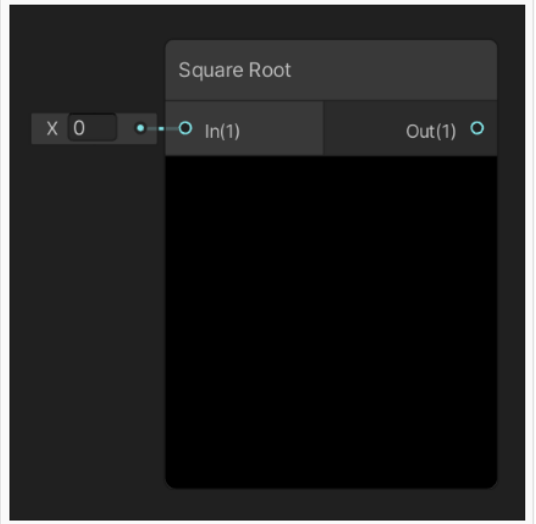
2 つの入力値の合計を返します。

Divide



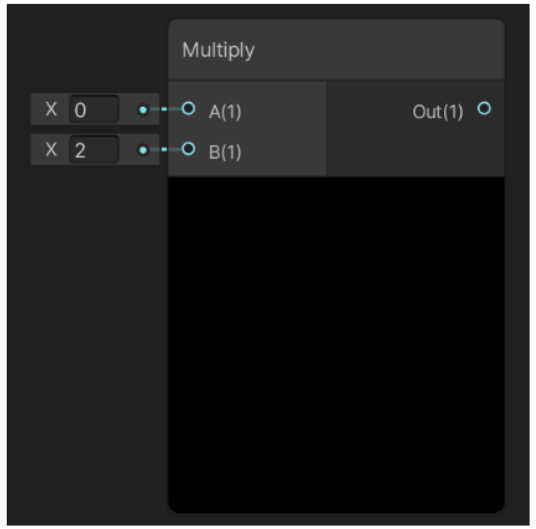
入力 A を入力 B で割った結果を返します。

Square Root



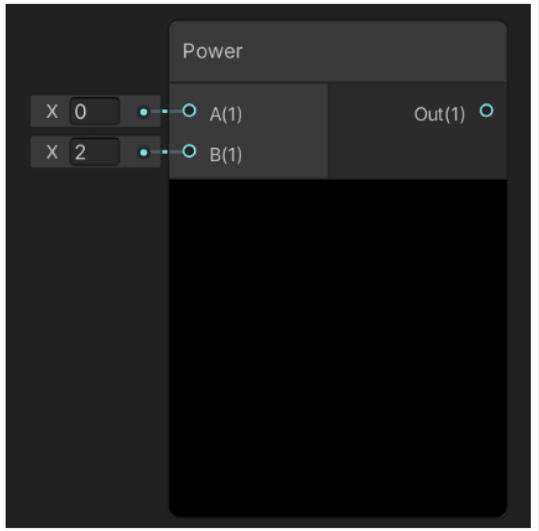
入力 In の平方根を返します。

Multiply



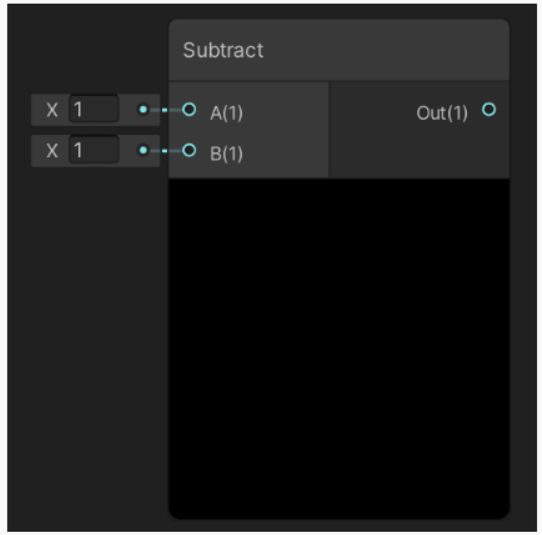
入力 A に入力 B を乗じた結果を返します。

Power



入力 A の入力 B 乗の結果を返します。

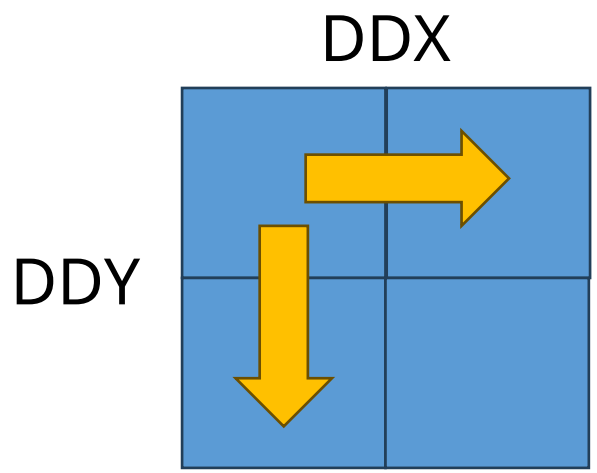
Subtract



入力 A から入力 B を引いた結果を返します。

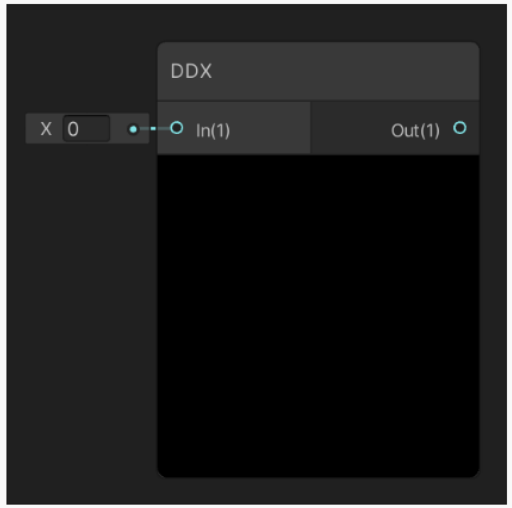
微分

- 実際は差分：隣接ピクセルとの差
- 2x2ピクセルで共通の値
- この演算を考慮して、GPUでは、必ず2x2の単位でレンダリングされる



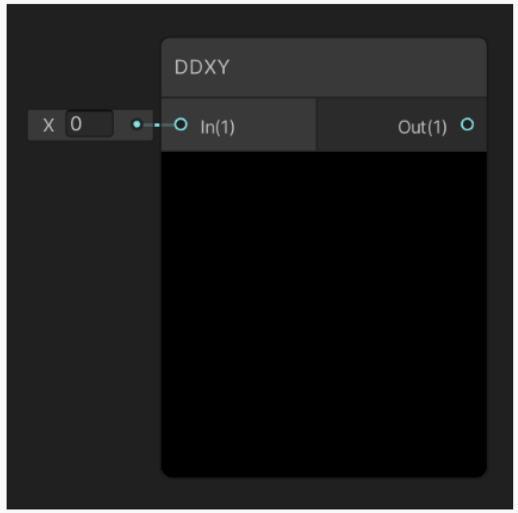
Derivative (微分)

DDX



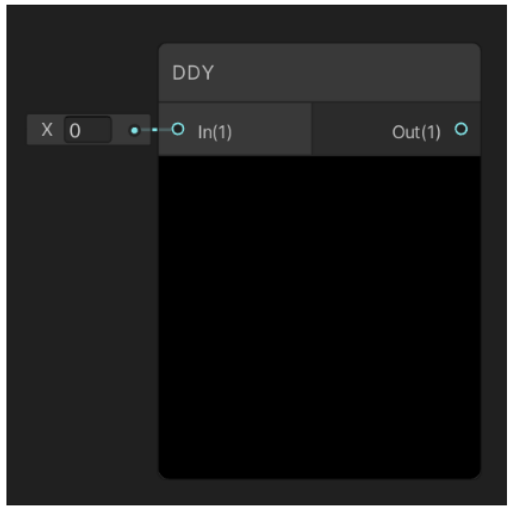
スクリーンスペース x 座標に対する微分値を返します。

DDXY



両方の微分値の和を返します。

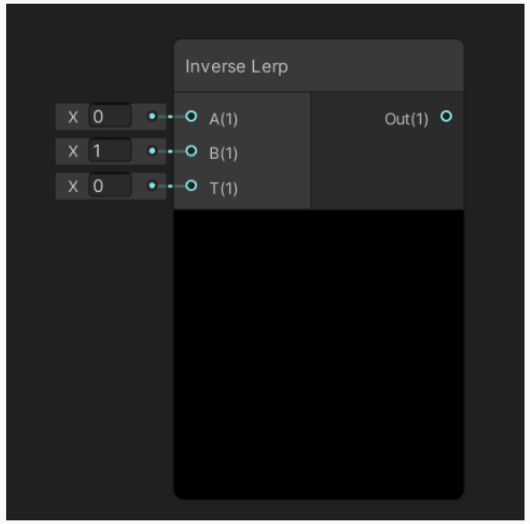
DDY



スクリーンスペース y 座標に対する微分値を返します。

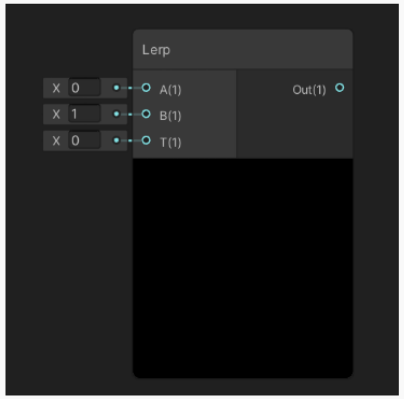
Interpolation (補間)

Inverse Lerp



入力 A から入力 B までの範囲の入力 T によって定義される補間を生成するパラメーターを返します。

Lerp

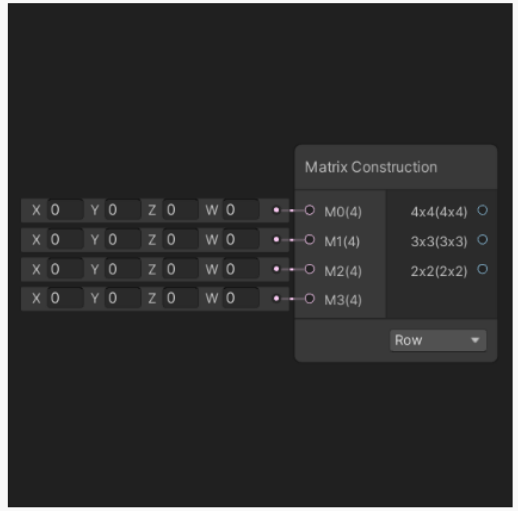
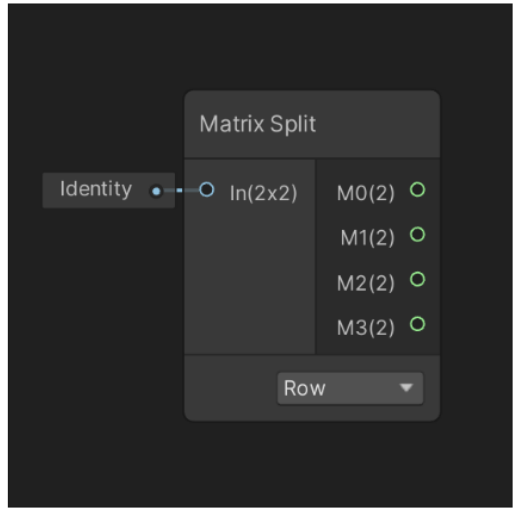
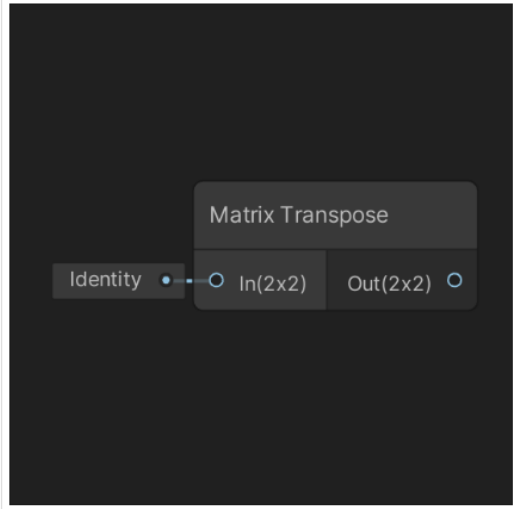


入力 A と入力 B の間を入力 T で線形補間した結果を返します。

Smoothstep

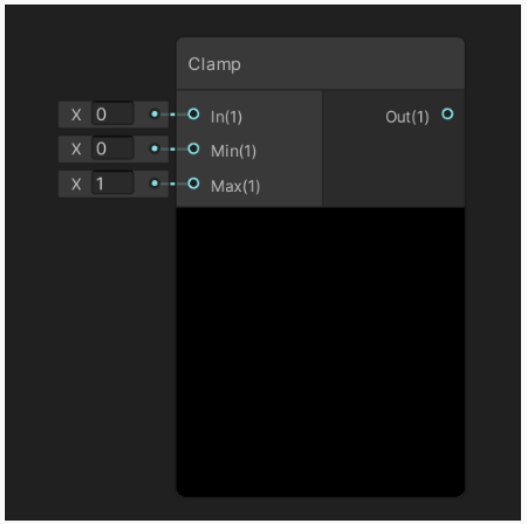
入力 In が入力 Edge1 と Edge2 の間にある場合に、0 と 1 の間の滑らかなエルミート補間の結果を返します。

Matrix (行列)

Matrix Construction	Matrix Determinant
	
M0、M1、M2、M3 の 4 つの入力ベクトルから正方行列を構築します。	入力 In で定義される行列の行列式を返します。
Matrix Split	**Matrix Transpose**
	
入力 In で定義される正方行列をベクトルに分割します。	入力 In で定義される行列の、入れ替えた値を返します。

Range (範囲)

Clamp

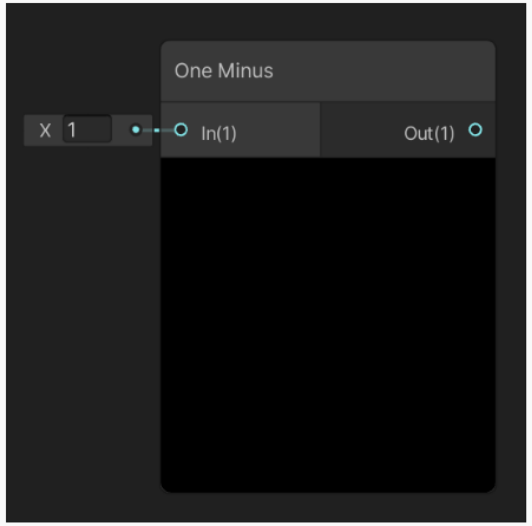


入力 In を、入力 Min で定義される最小値から入力 Max で定義される最大値までの範囲に固定して返します。

Fraction

入力 In の小数部分 (0 以上 1 未満の値) を返します。

One Minus

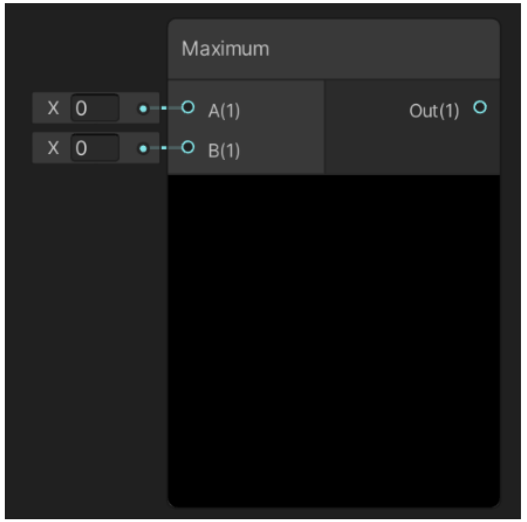


1 から入力 In を引いた結果を返します。

Random Range

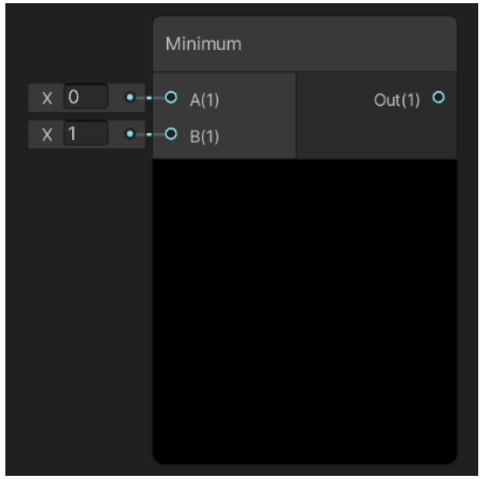
入力 Min で定義される最小値から入力 Max で定義される最大値までの範囲の擬似乱数を返します。

Maximum



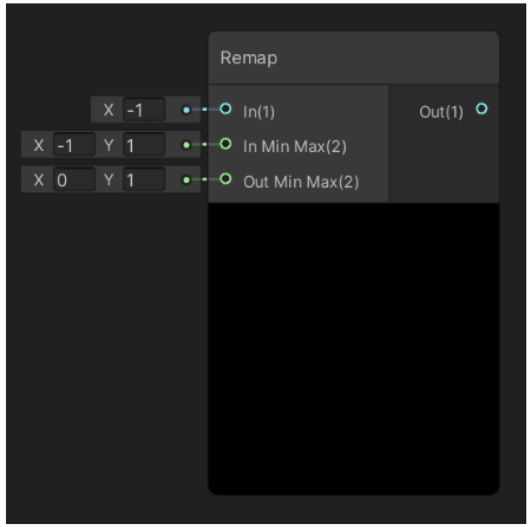
A と B の 2 つの入力値のうちの最大値を返します。

Minimum



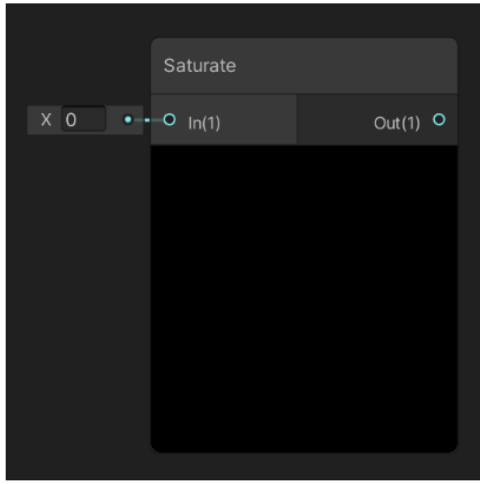
A と B の 2 つの入力値のうちの最小値を返します。

Remap



入力 Out Min Max の範囲から入力 In Min Max の範囲へ、入力 In の値を再マップします。

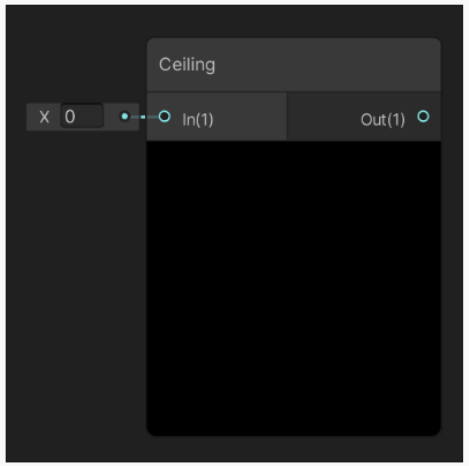
Saturate



入力 In の値を 0 と 1 の間に固定して返します。

Round (端数処理)

Ceiling

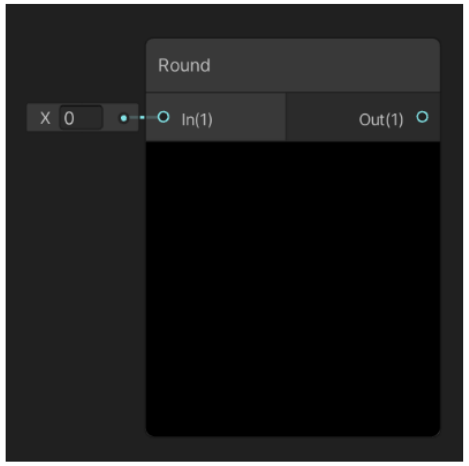


入力 In の値以上の、最小整数値を返します。

Floor

入力 In の値以下の、最大整数値を返します。

Round

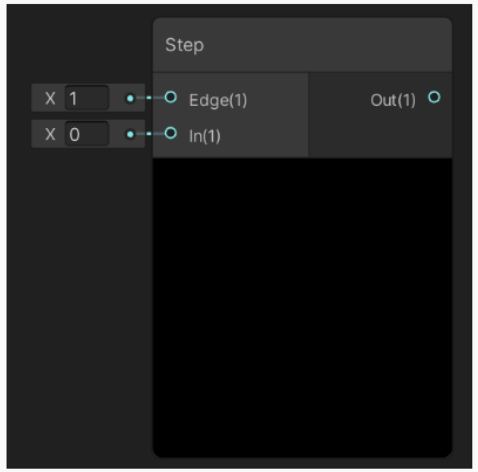


入力 In の値を、最も近い整数に四捨五入して返します。

Sign

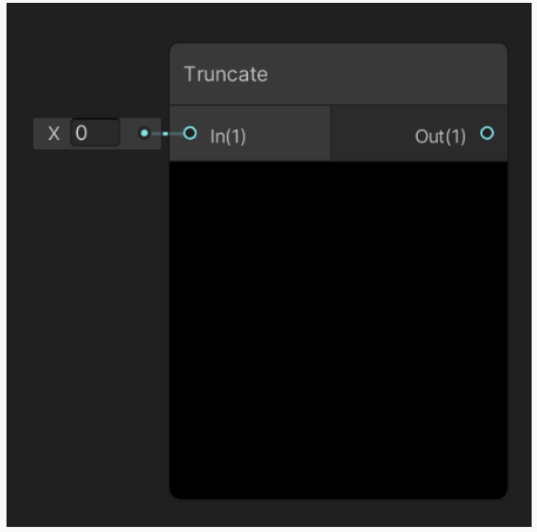
入力 In の値が 0 未満の場合は -1 を、0 に等しい場合は 0 を、0 より大きい場合は 1 を返します。

Step



入力 In の値が入力 Edge の値以上の場合は 1 を、それ以外の場合は 0 を返します。

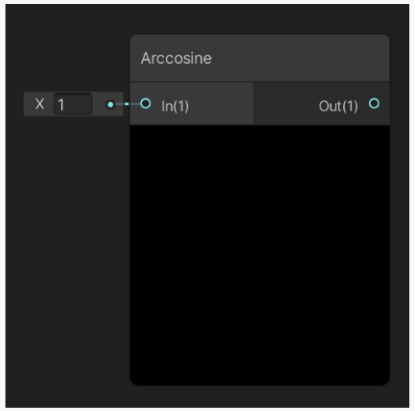
Truncate



入力 In の値の整数部分を返します。

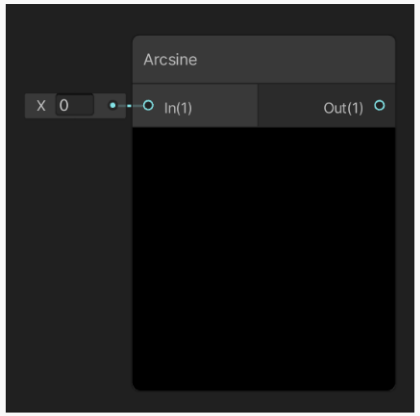
Trigonometry (三角関数)

Arccosine



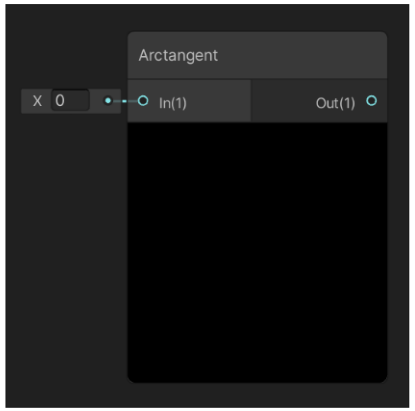
入力 In の各成分の逆余弦を、同じ長さのベクトルとして返します。

Arcsine



入力 In の各成分の逆正弦を、同じ長さのベクトルとして返します。

Arctangent

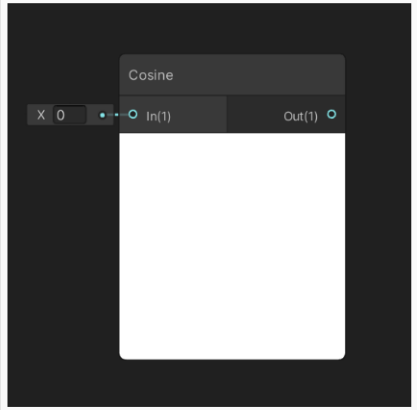


入力 In の値の逆正接を返します。各成分は -Pi/2 から Pi/2 の範囲内になります。

Arctangent2

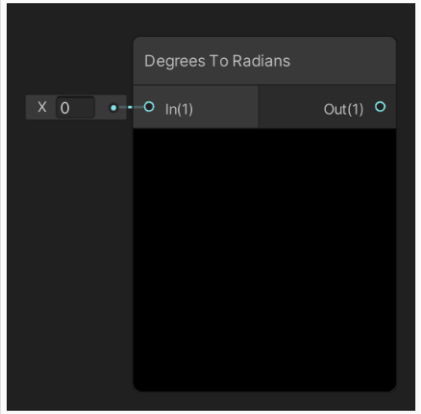
入力 A と入力 B の両方の値の逆正接を返します。

Cosine



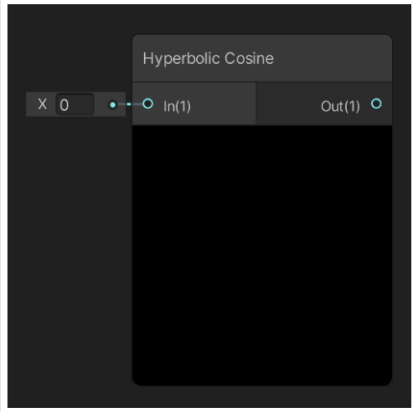
入力 In の値の余弦を返します。

Degrees to Radians



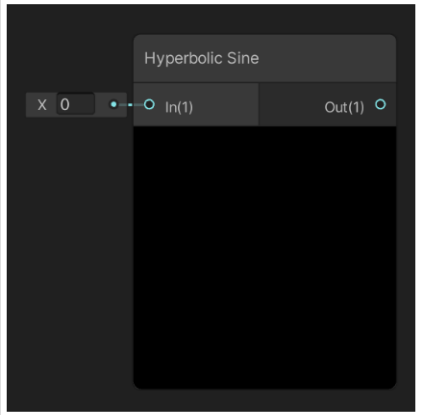
入力 In の値を度数からラジアンに変換して返します。

Hyperbolic Cosine



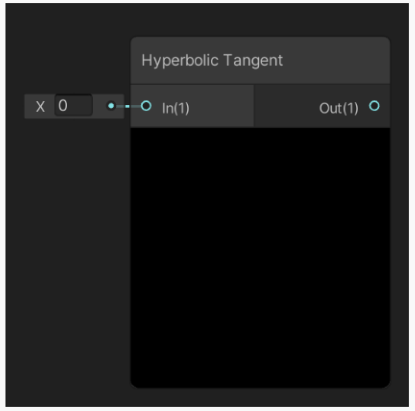
入力 In の双曲線余弦を返します。

Hyperbolic Sine



入力 In の値の双曲線正弦を返します。

Hyperbolic Tangent

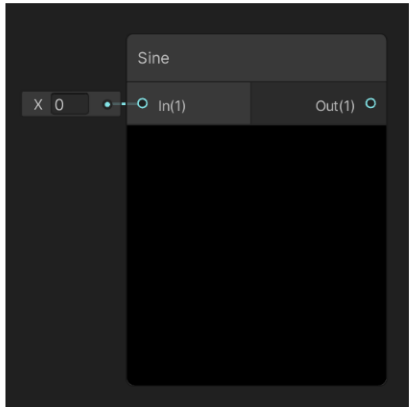


入力 In の値の双曲線正接を返します。

Radians to Degrees

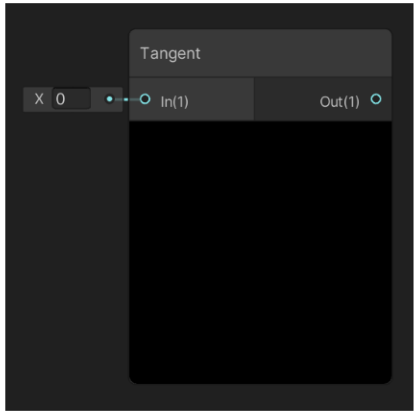
入力 In の値をラジアンから度数に変換して返します。

Sine



入力 In の値の正弦を返します。

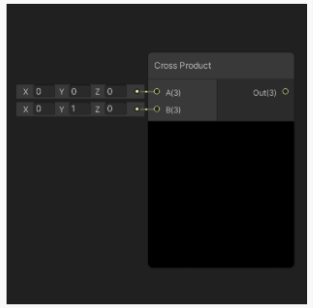
Tangent



入力 In の値の正接 (接線) を返します。

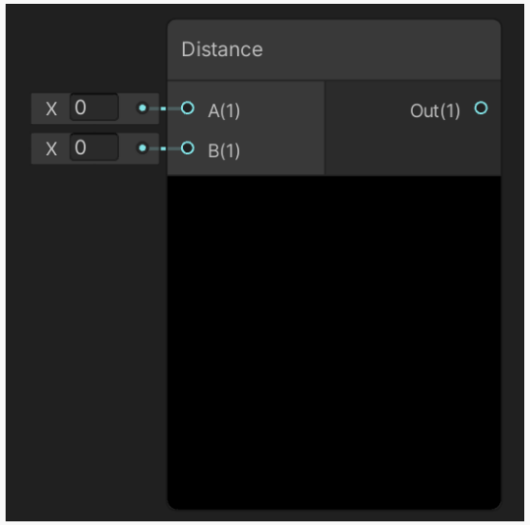
Vector (ベクトル)

Cross Product



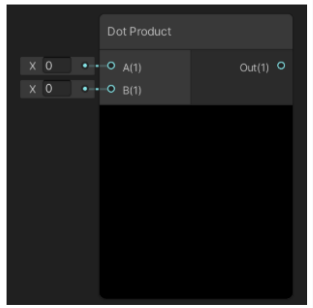
入力 A と B の値の外積を返します。

Distance



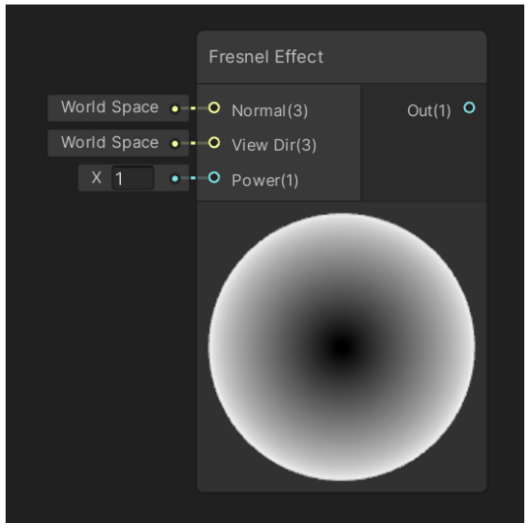
入力 A と B の値の間のユークリッド距離を返します。

Dot Product



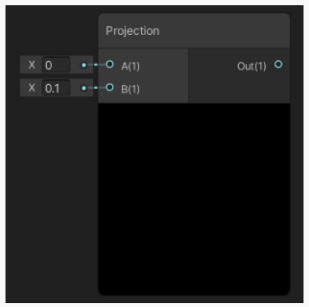
入力 A と B の値のドット積 (点乗積)、あるいはスカラー積 (内積) を返します。

Fresnel Effect



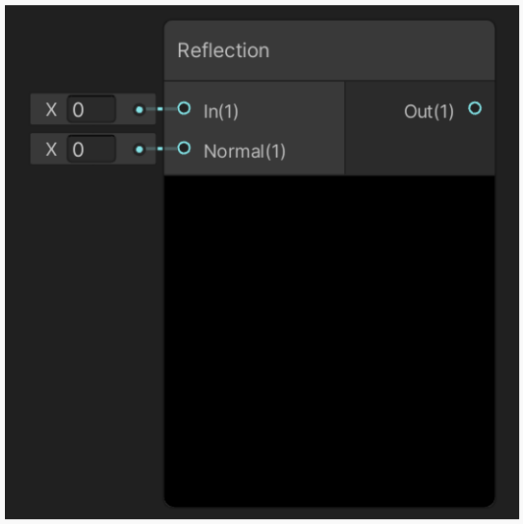
フレネル効果 (Fresnel Effect) は、見る角度によってサーフェス (表面) 上の反射率が変わるエフェクトで、グレージング角に近付くほど多くの光が反射されます。

Projection



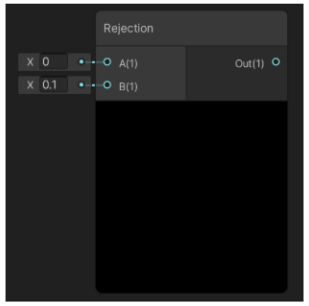
入力 A の値を入力 B の値に平行する直線に投射した結果を返します。

Reflection



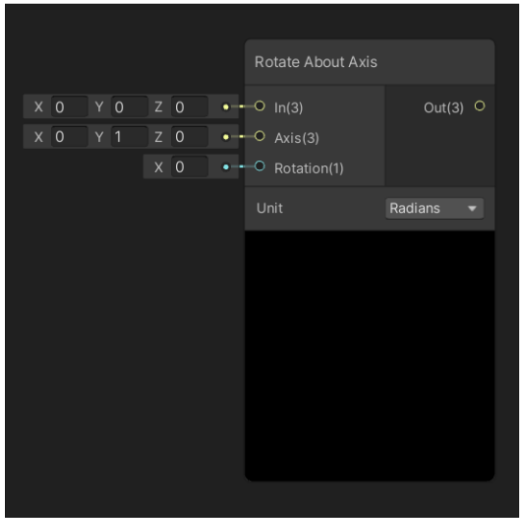
入力 In と面法線 Normal を使用して反射のベクトルを返します。

Rejection



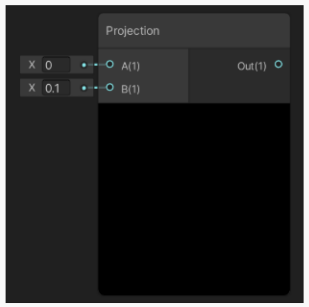
入力 A の値を、入力 B の値と直角を成す平面上に投射した結果を返します。

Rotate About Axis



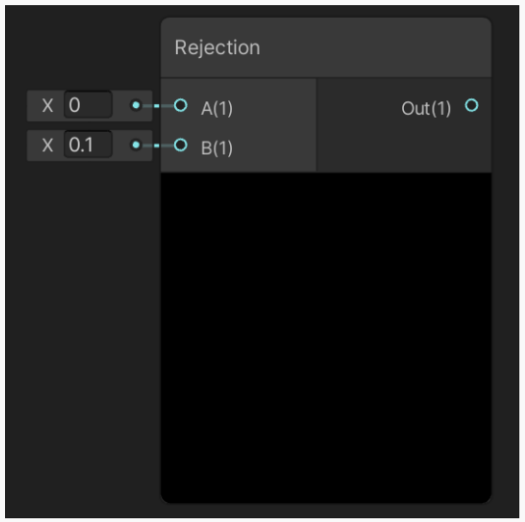
入力ベクトル In を、Rotation の値の分だけ、軸 Axis 周りに回転させます。

Projection



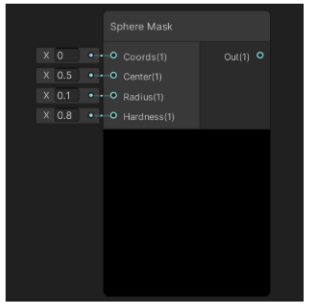
入力 A の値を入力 B の値に平行する直線に投射した結果を返します。

Rejection



入力 A の値を、入力 B の値と直角を成す平面上に投射した結果を返します。

Sphere Mask



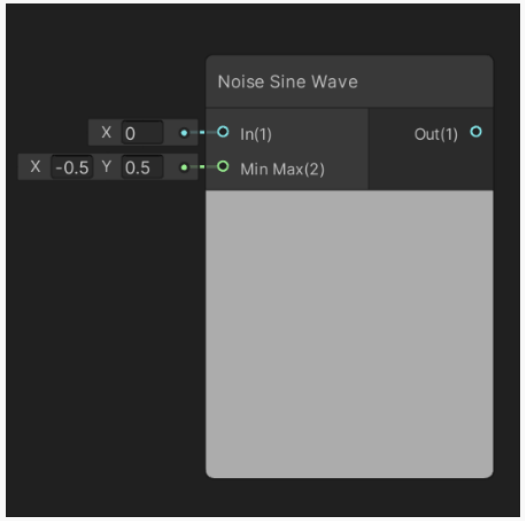
入力 Center を中心に球状マスクを作成します。

Transform

入力 In の値を 1 つの座標空間から別の座標空間へ変換した結果を返します。

Wave (波形)

Noise Sine Wave

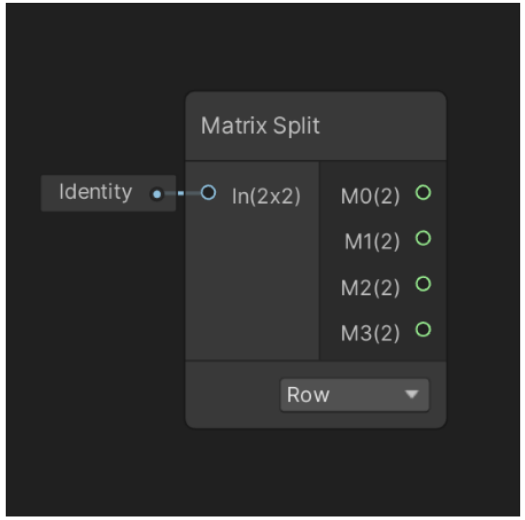


入力 In の値の正弦を返します。変化をつけるために正弦波の振幅にランダムなノイズが追加されます。

Sawtooth Wave

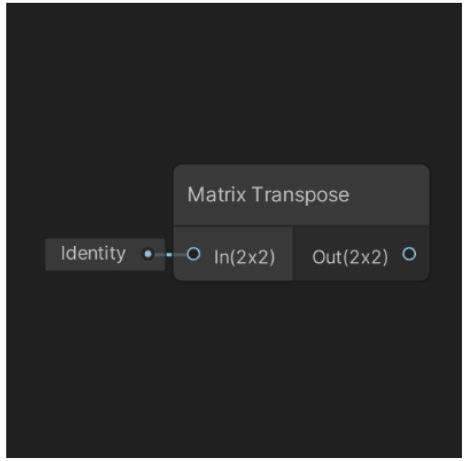
入力 In の値からのこぎり波を返します。

Matrix Split



入力 In で定義される正方行列をベクトルに分割します。

Matrix Transpose



入力 In で定義される行列の、入れ替えた値を返します。

Mesh Deformationノード

Compute Deformation ノード

説明

このノードを使用すると、コンピュートシェーダーで変形された頂点データを頂点シェーダーに渡すことができます。このノードは DOTS Hybrid Renderer との併用によってのみ機能します。 `DeformedVertexData` を `_DeformedMeshData` バッファ内に供給する必要があります。このノードは、 `_ComputeMeshIndex` プロパティーを使用して、現在のメッシュに関連する `DeformedVertexData` が `_DeformedMeshData` バッファ内のどこにあるかを計算します。データを出力するには、DOTS Hybrid Renderer パッケージと DOTS Animation パッケージの両方をインストールするか、カスタムのソリューションを使用する必要があります。

ポート

Name	Direction	タイプ	ステージ	説明
Position	出力	Vector3	Vertex	変形された頂点の位置を出力します。
Normal	出力	Vector3	Vertex	変形された頂点の法線を出力します。
Tangent	出力	Vector3	Vertex	変形された頂点の接線を出力します。

Linear Blend Skinning ノード

説明

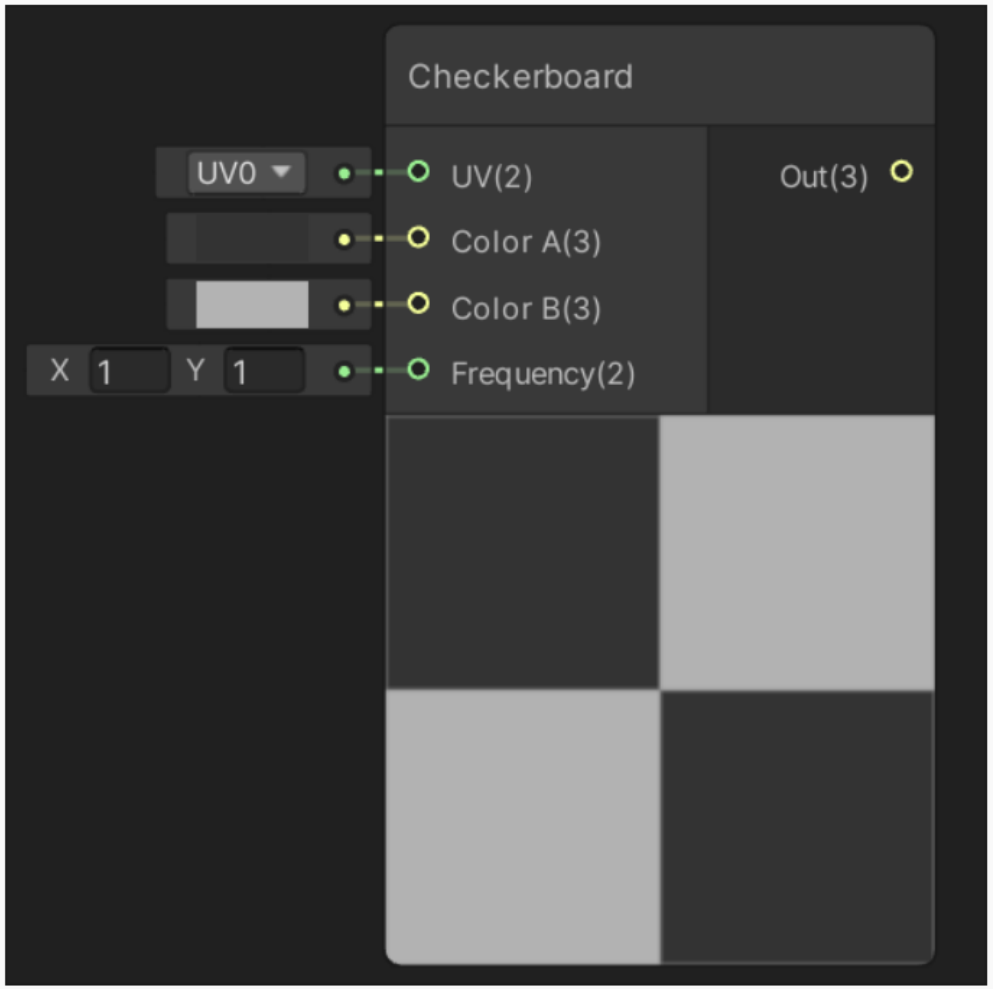
このノードを使用すると、線形ブレンド頂点スキニング (Linear Blend Vertex Skinning) が適用できます。このノードは DOTS Hybrid Renderer との併用によってのみ機能します。 `_SkinMatrices` バッファ内にスキニング行列を供給する必要があります。このノードは、 `_SkinMatrixIndex` プロパティーを使用して、現在のメッシュに関連する行列が `_SkinMatrices` バッファ内のどこにあるかを計算します。

ポート

Name	Direction	タイプ	ステージ	説明
Position	入力	Vector3	Vertex	オブジェクト空間における頂点の位置
Normal	入力	Vector3	Vertex	オブジェクト空間における頂点の法線
Tangent	入力	Vector3	Vertex	オブジェクト空間における頂点の接線
Position	出力	Vector3	Vertex	スキニングされた頂点の位置を出力します。
Normal	出力	Vector3	Vertex	スキニングされた頂点の法線を出力します。
Tangent	出力	Vector3	Vertex	スキニングされた頂点の接線を出力します。

Procedural ノード

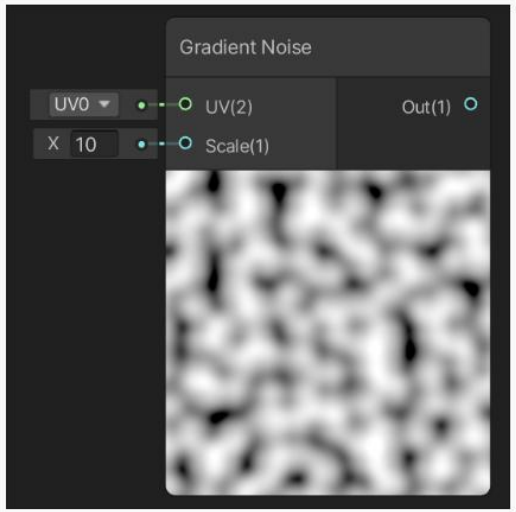
Checkerboard



入力 UV に基づいて、入力 Color A の色と入力 Color B の色で構成されたチェッカーボードを生成します。

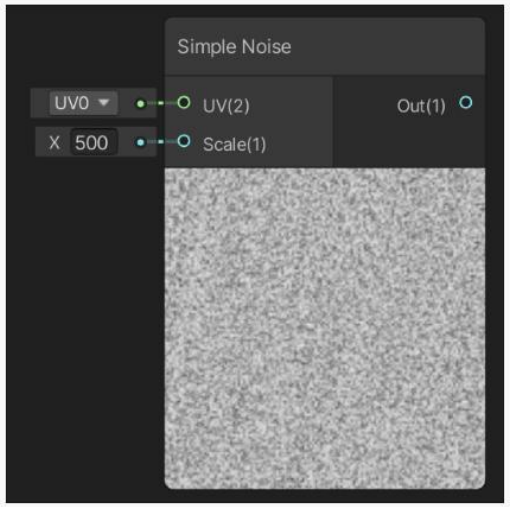
Noise (ノイズ)

Gradient Noise



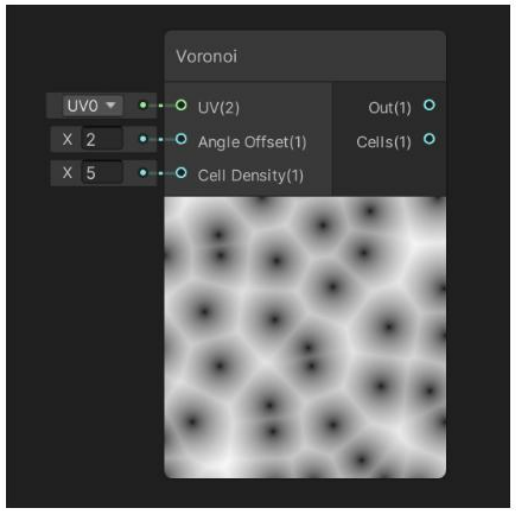
入力 UV に基づいて、グラデーション (パーリン) ノイズを生成します。

Simple Noise



入力 UV を基に、単純なノイズ (バリューノイズ) を生成します。

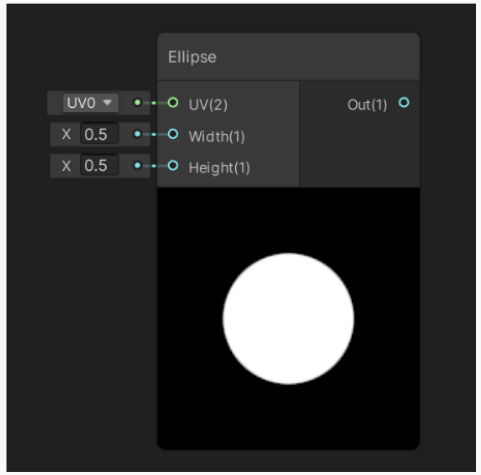
Voronoi



入力 UV に基づいて、ボロノイノイズ (Worley ノイズ) を生成します。

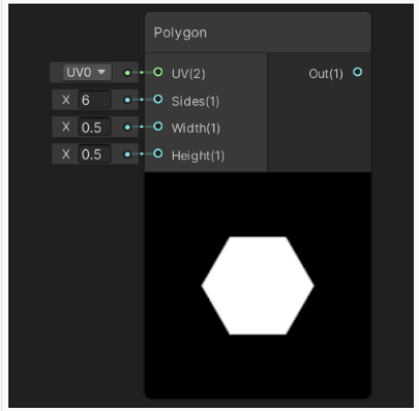
Shape (シェイプ)

Ellipse



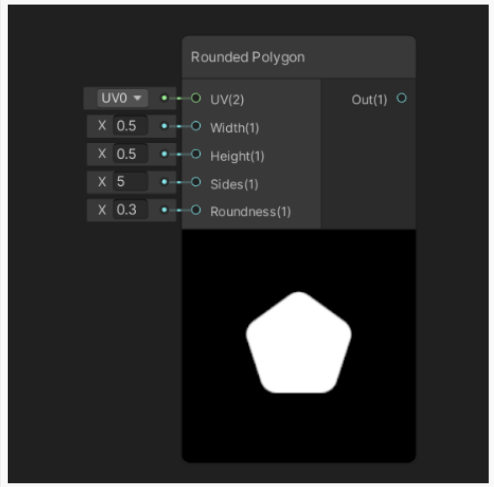
入力 UV に基づいて、入力 Width と 入力 Height に指定されたサイズの楕円形を生成します。

Polygon



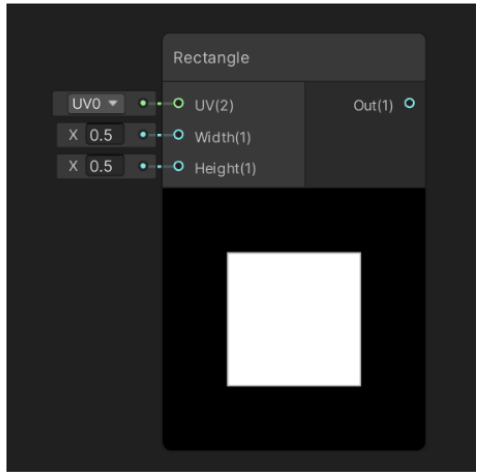
入力 UV に基づいて、入力 Width および Height に指定されたサイズの通常のポリゴンシェイプを生成します。ポリゴンの辺の数は入力 Sides によって指定されます。

Rounded Polygon



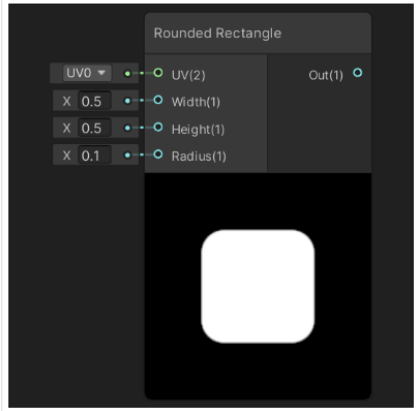
入力 UV に基づいて、入力 Width および Height に指定されたサイズの、角を丸めたポリゴンシェイプを生成します。入力 Sides は辺の数を指定し、入力 Roundness は各角の丸みを指定します。

Rectangle



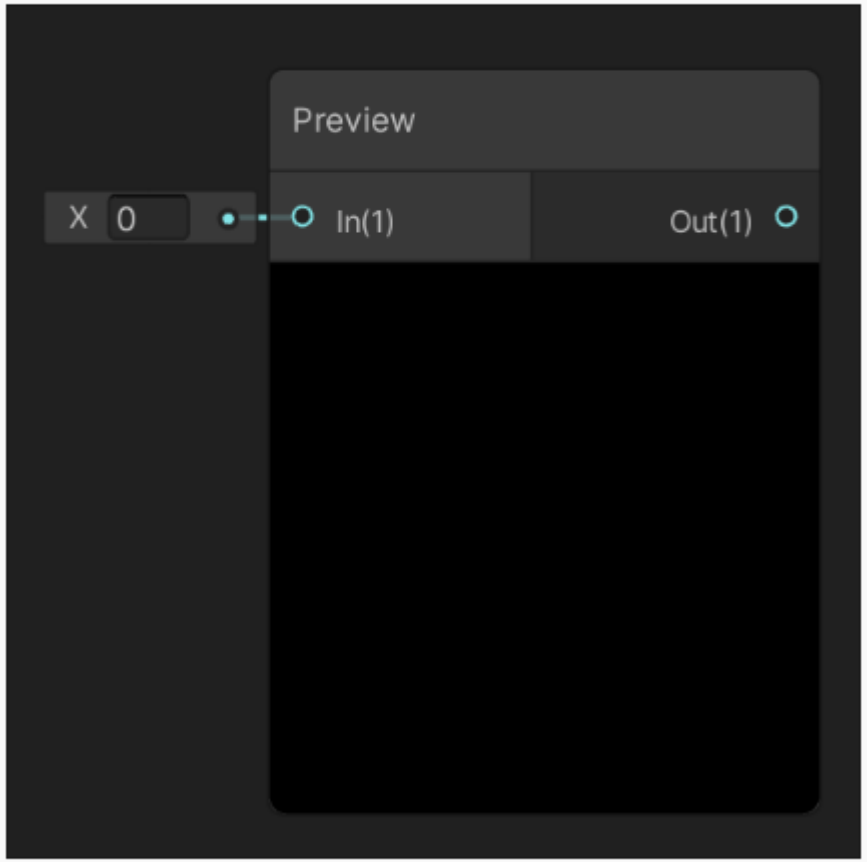
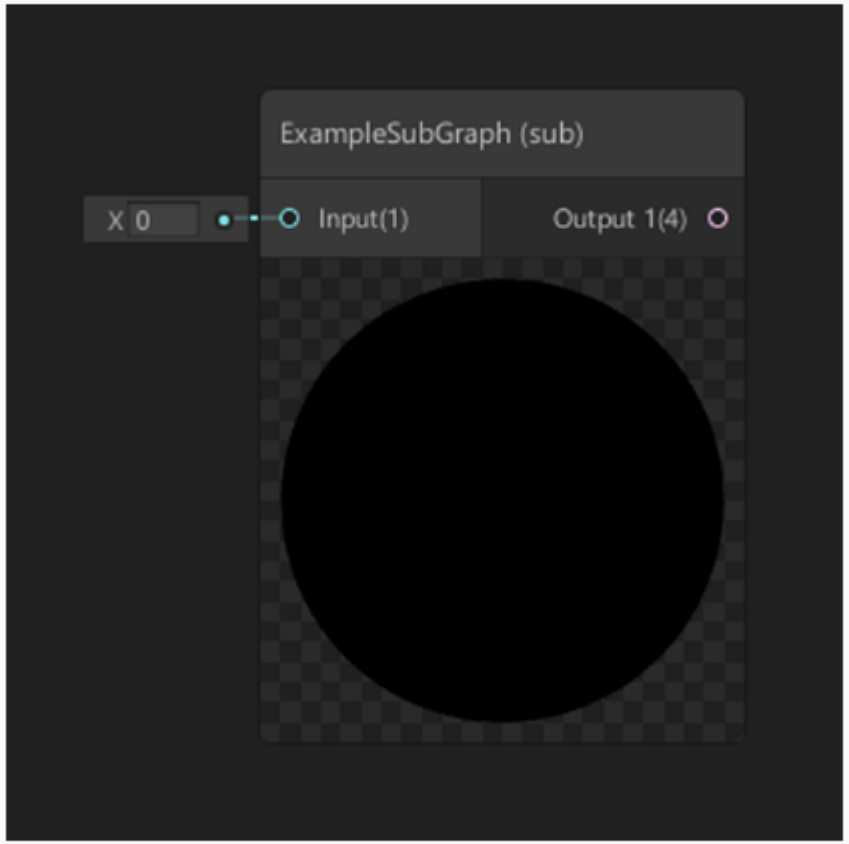
入力 UV に基づいて、入力 Width および Height に指定されたサイズの四角形を生成します。

Rounded Rectangle



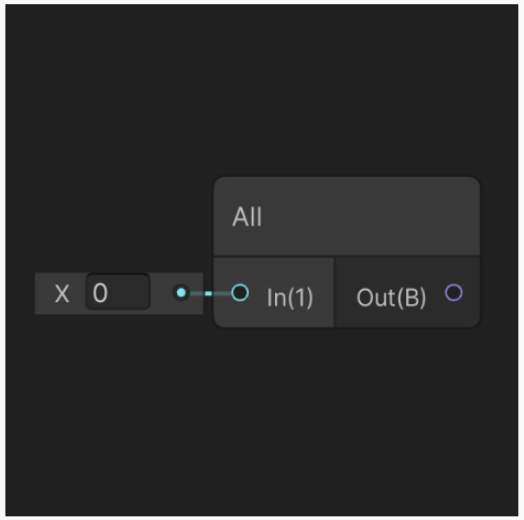
入力 UV に基づいて、入力 Width および Height に指定されたサイズの、角を丸めた四角形を生成します。各角の半径は入力 Radius によって定義されます。

Utility ノード

Preview	Sub-Graph
	
プレビューウィンドウを提供し、入力値を修正せずに出力します。	Sub Graph アセットへの参照を提供します。

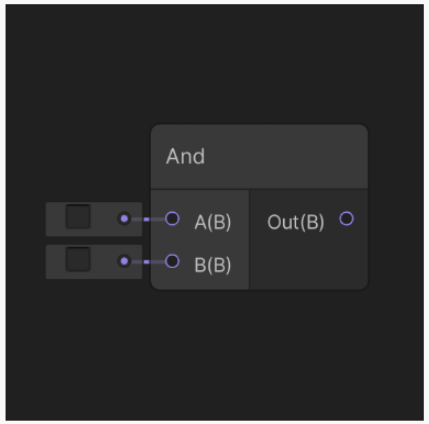
Logic (ロジック)

All



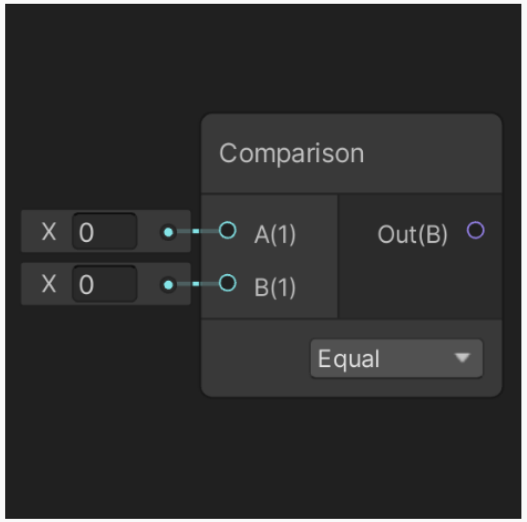
入力 In の全ての成分が 0 以外の場合に true を返します。

And



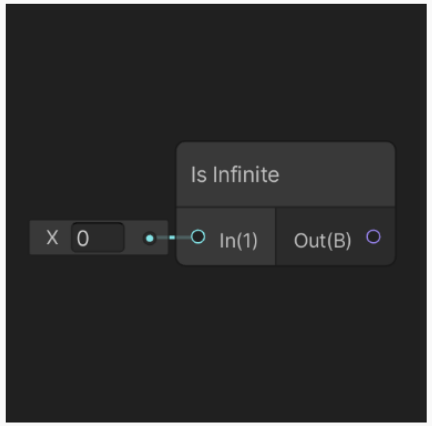
入力 A と入力 B の両方が true の場合に true を返します。

Comparison



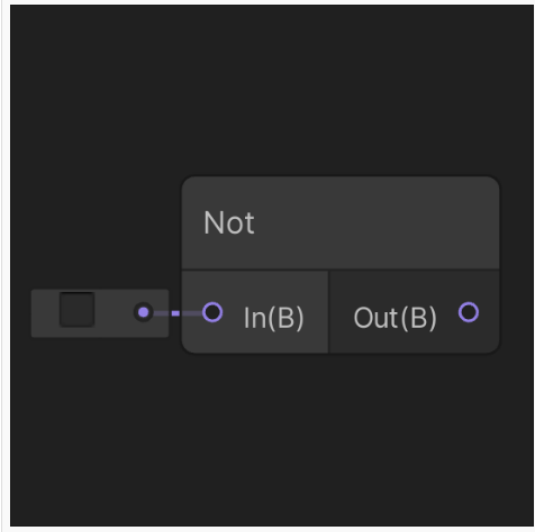
ドロップダウンメニューで選択された条件に基づいて、A と B の 2 つの入力値を比較します。

Is Infinite



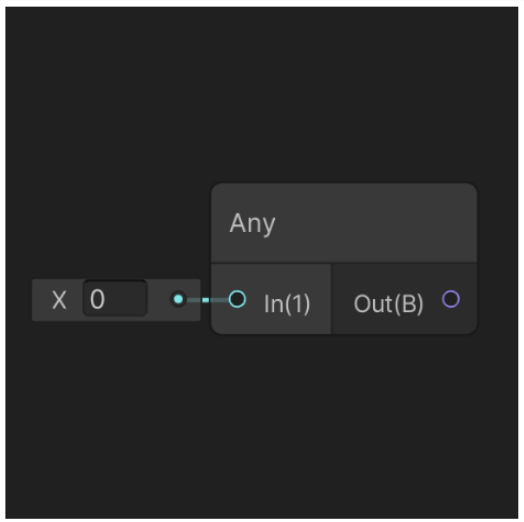
入力 In の成分のどれかが無限大の値である場合に true を返します。

Not



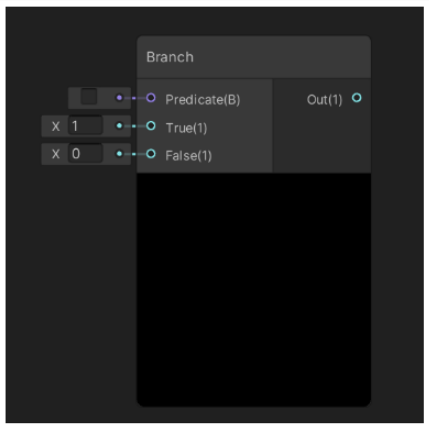
入力 In の反対を返します。In が true の場合は出力は false になり、そうでない場合は true になります。

Any



入力 In の成分のどれかが 0 以外である場合に true を返します。

Branch

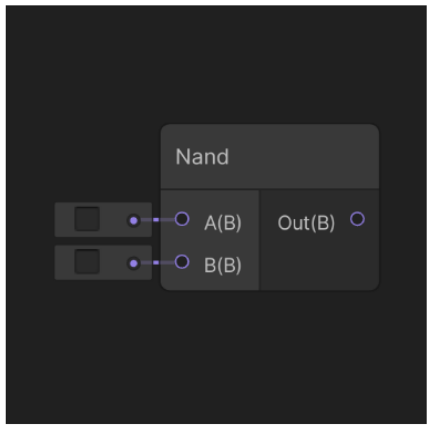


シェーダーに動的ブランチを提供します。

Is NaN

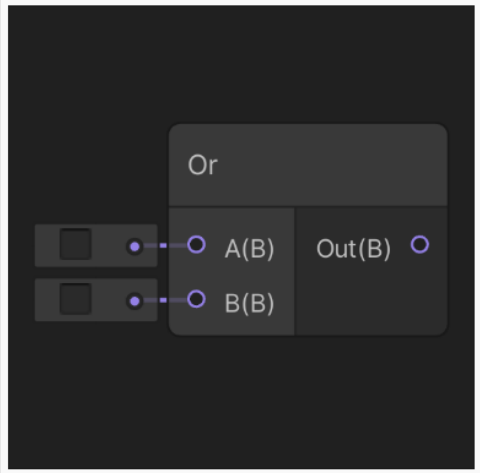
入力 In の成分のどれかが数値でない (NaN) 場合に true を返します。

Nand



入力 A と入力 B の両方が false である場合に true を返します。

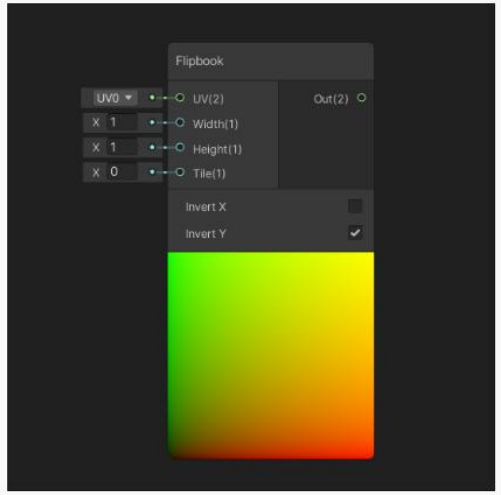
Or



入力 A と入力 B のどちらかが true である場合に true を返します。

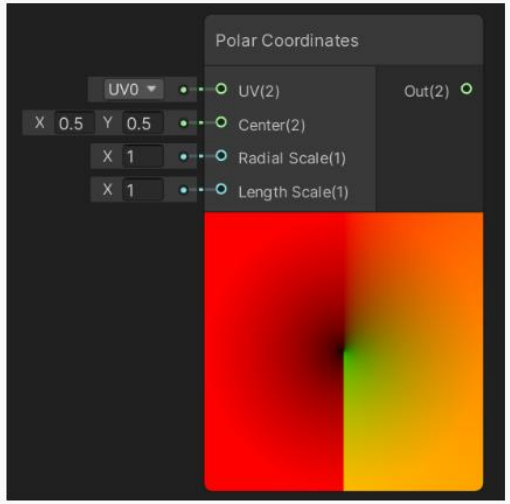
UV ノード

Flipbook



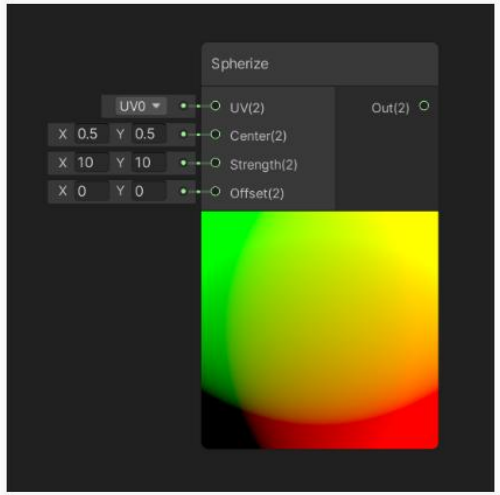
入力 In に供給する UV のフリップブック、または、テクスチャシートアニメーションを作成します。

Polar Coordinates



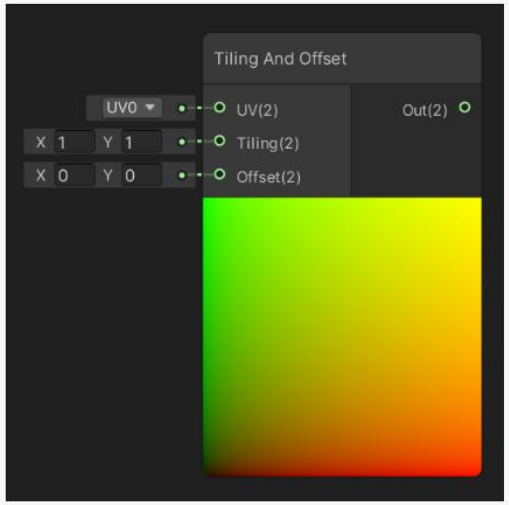
入力 UV の値を極座標に変換します。

Spherize



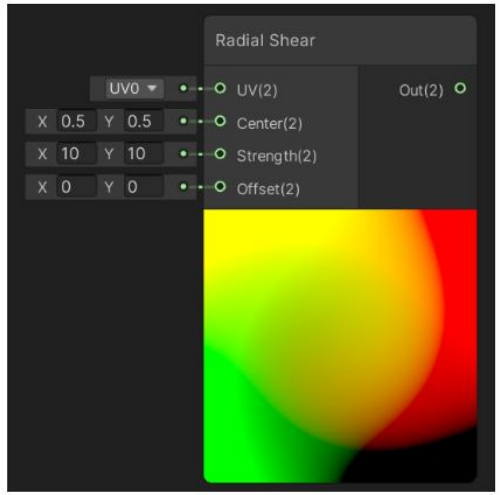
入力 UV の値に、カメラの魚眼レンズのようなスフィア状のワープ効果を適用します。

Tiling and Offset



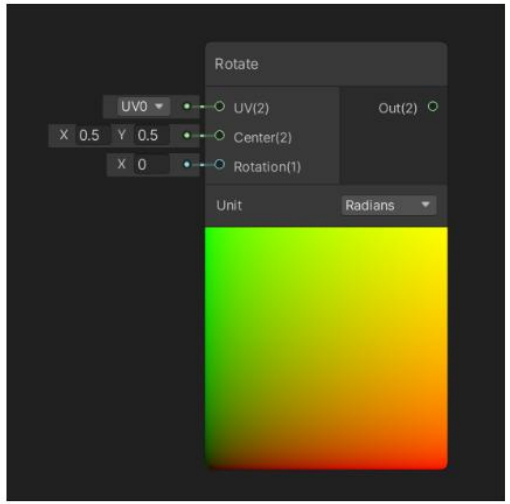
入力 UV の値を、入力 Tiling に基づいてタイル化し、入力 Offset に基づいてオフセットします。

Radial Shear



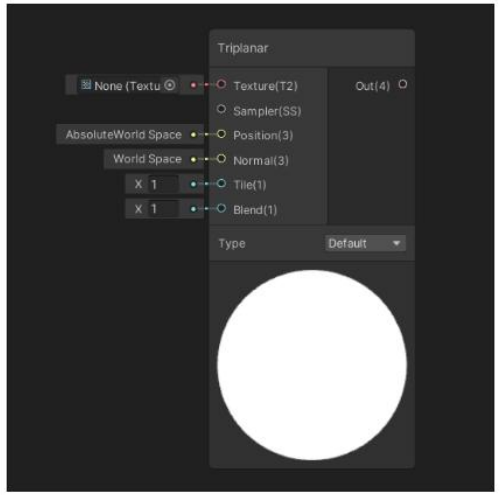
入力 UV の値に、放射状シアーの波のようなワープ (ゆがみ) 効果を適用します。

Rotate



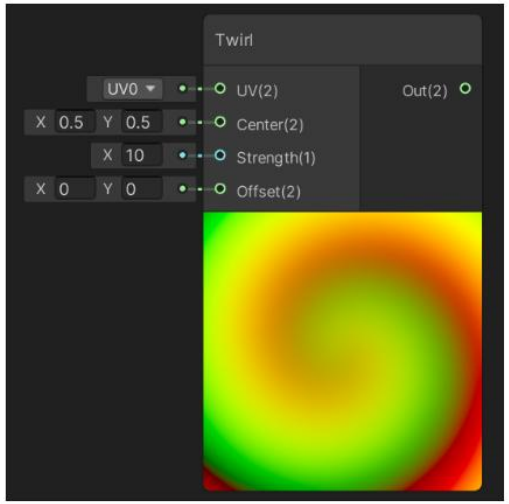
入力 UV の値を、入力 Center で定義される基準点を中心に、入力 Rotation の量だけ回転させます。

Triplanar



ワールド空間への投影によって UV 生成とテクスチャサンプリングを行う方法です。

Twirl



入力 UV の値に、ブラックホールのような渦巻き状のワープ (ゆがみ) 効果を適用します。

Block ノード

説明

Block (ブロック) は、Master Stack (マスタースタック) の特殊なタイプのノードです。各ブロックは、シェーダーの最終出力で使用される 1 つのフラグメント (または頂点) の説明データを表します。常に使用可能な Block ノードもありますが、特定のパイプラインでしか使用できない Block ノードもあります。

一部のブロックは特定の Graph Settings (グラフ設定) でしか使用できません。ブロックはグラフ設定に応じてアクティブまたは非アクティブになる場合があります。

ブロックの切り取り、コピー、貼り付けは行えません。

Block ノードの追加と削除

新しい Block ノードをマスタースタック内のコンテキストに追加するには、コンテキスト内の何もない場所にマウスオーバーし、スペースキーを押すか、右クリックして “Create Node” を選択します。これにより Create Node メニューが表示されます。

このメニューには、そのコンテキストに有効な Block ノードのみが含まれています。Fragment コンテキストの Create Node メニューには Vertex ブロックは表示されません。

このメニューから特定の Block ノードを選択すると、そのノードがコンテキストに追加されます。

コンテキスト内で特定の Block ノードを選択し、"Delete" を押すか、右クリックして "Delete" を選択すると、コンテキストからそのブロックが削除されます。

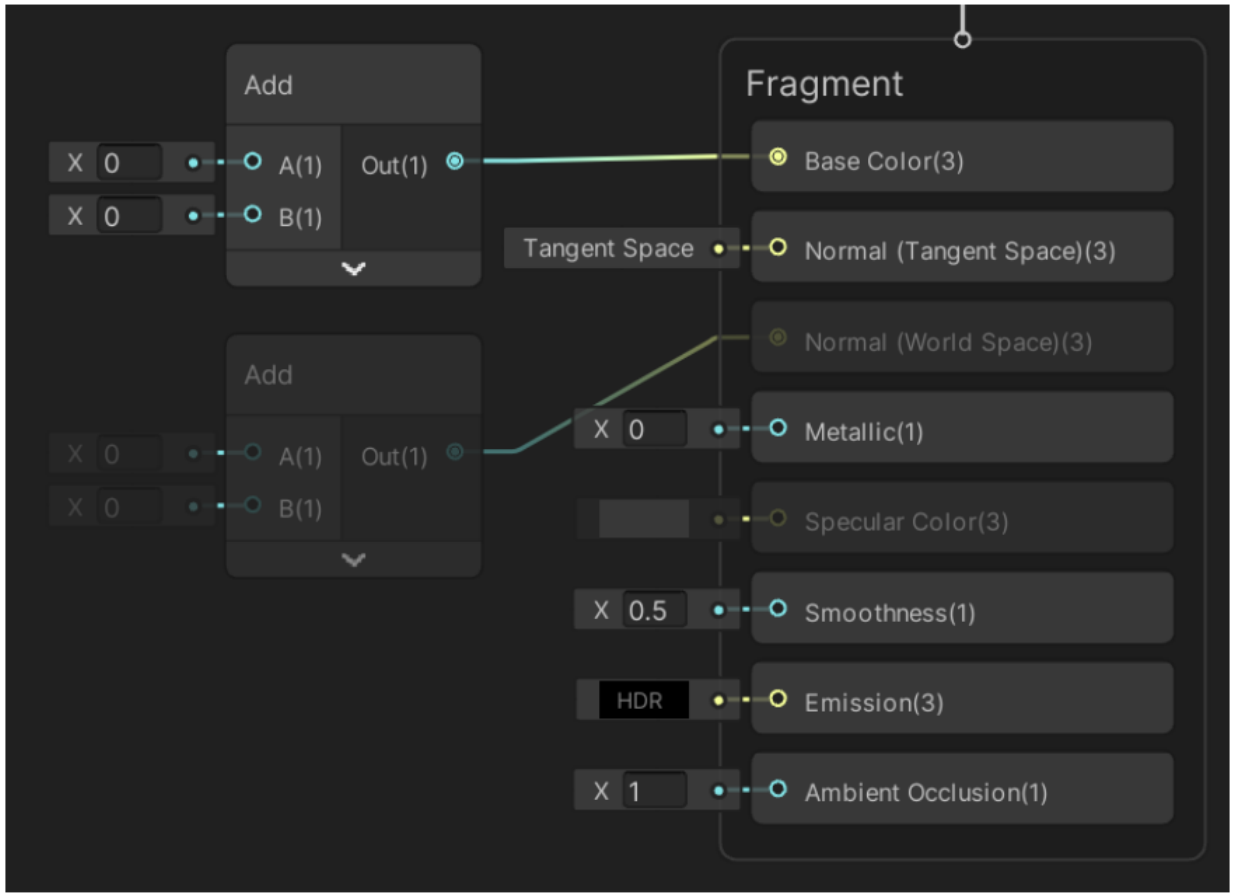
ブロックの自動的な追加と削除

ユーザーの Shader Graph 環境設定に応じて自動的にブロックをコンテキストに追加したり、削除したりすることもできます。`Automatically Add or Remove Blocks` をオンにすると、特定のターゲットやマテリアルのタイプに必要な Block ノードが自動的に追加されます。接続されておらずデフォルト値を持たない、適合しない Block ノードは、全て自動的にコンテキストから削除されます。

`Automatically Add or Remove Blocks` をオフにすると、Block ノードの自動追加や削除は一切行われません。全ての必要な Block ノードは、選択されたターゲットとマテリアルの設定に基づいて、ユーザーが手作業で追加する必要があります。

アクティブなブロックと非アクティブなブロック

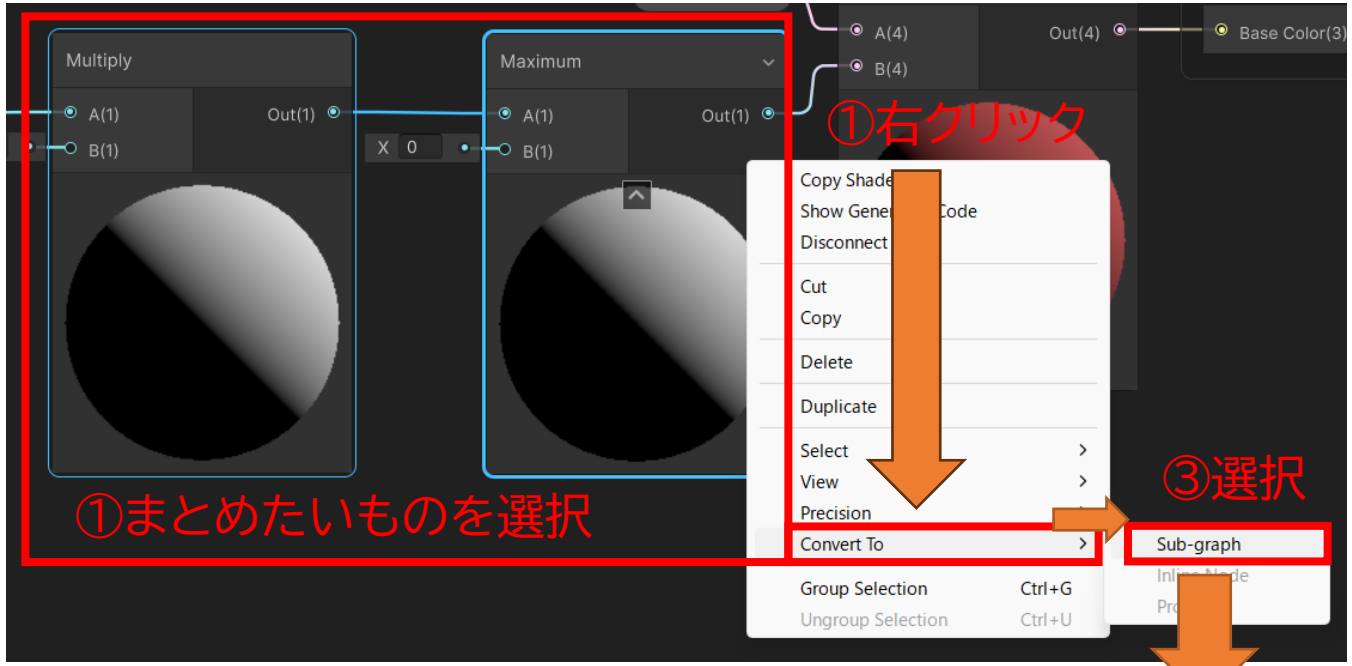
アクティブな Block ノードとは、生成されており、かつ、最終的なシェーダーに寄与しているブロックです。非アクティブな Block ノードとは、Shader Graph 内に存在しているが、生成されていない (あるいは最終的なシェーダーに寄与していない) ブロックです。

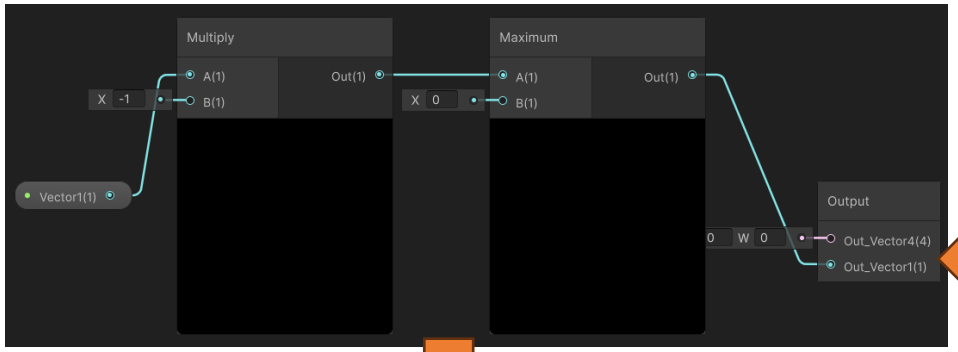


グラフ設定によっては、ブロックがアクティブあるいは非アクティブになる場合があります。この状態は、"非アクティブな Block ノード" と "非アクティブな Block ノードに のみ に接続されている全てのノードストリーム" をグレーアウト表示すれば確認できます。

Sub Graph

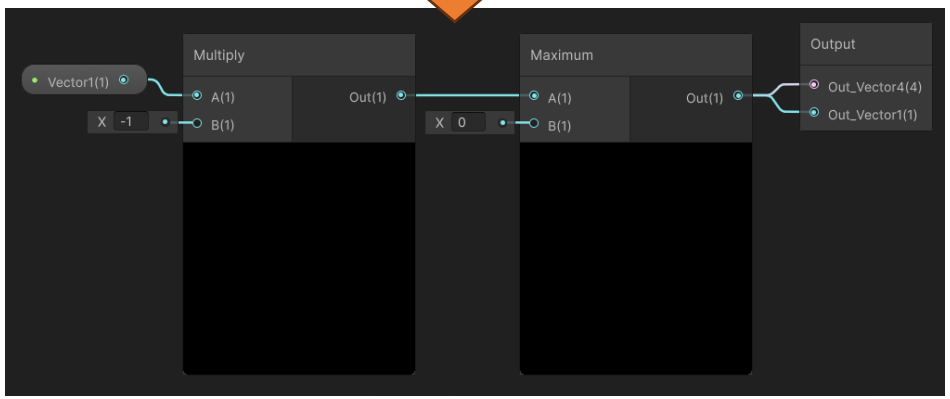
- ノードをまとめる

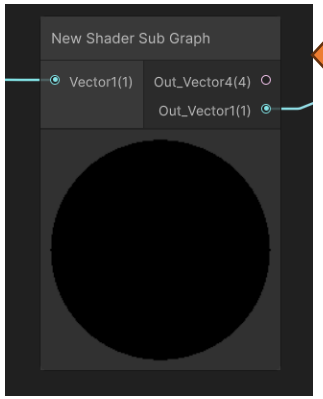


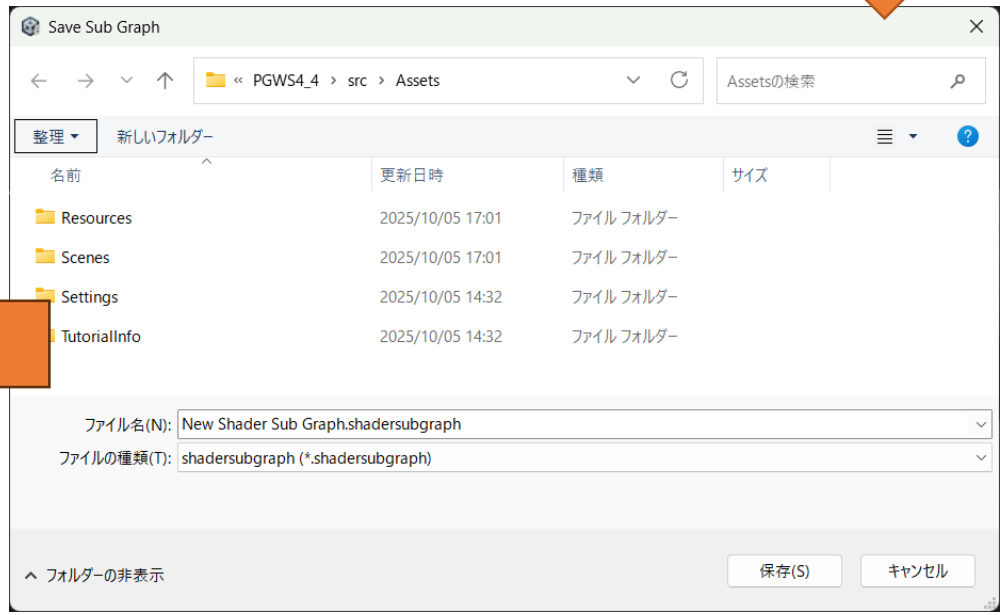


④ 選択

⑤ 必要に応じて編集





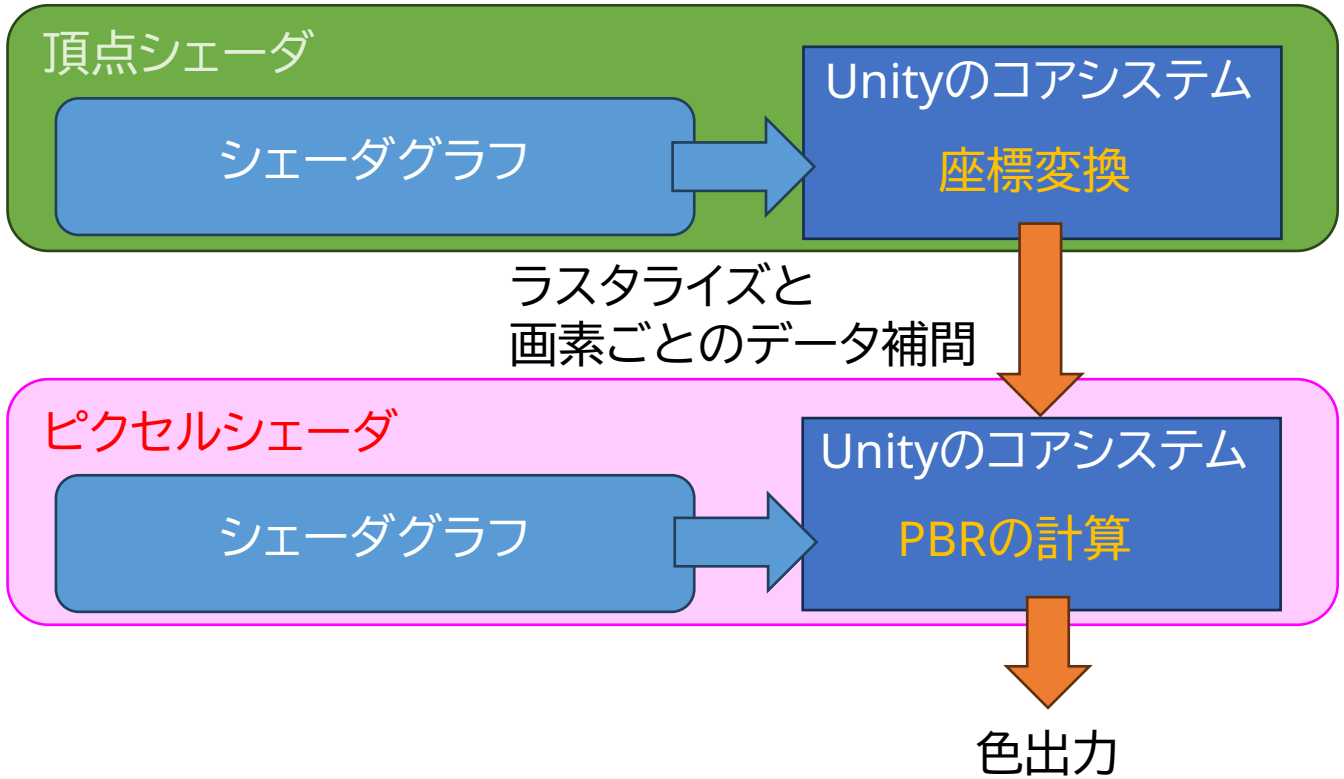


シェーダグラフ入門

- シェーダグラフの追加
- 色を変える
- ノードの追加
- ノードの種類
- Lit Shader Graph

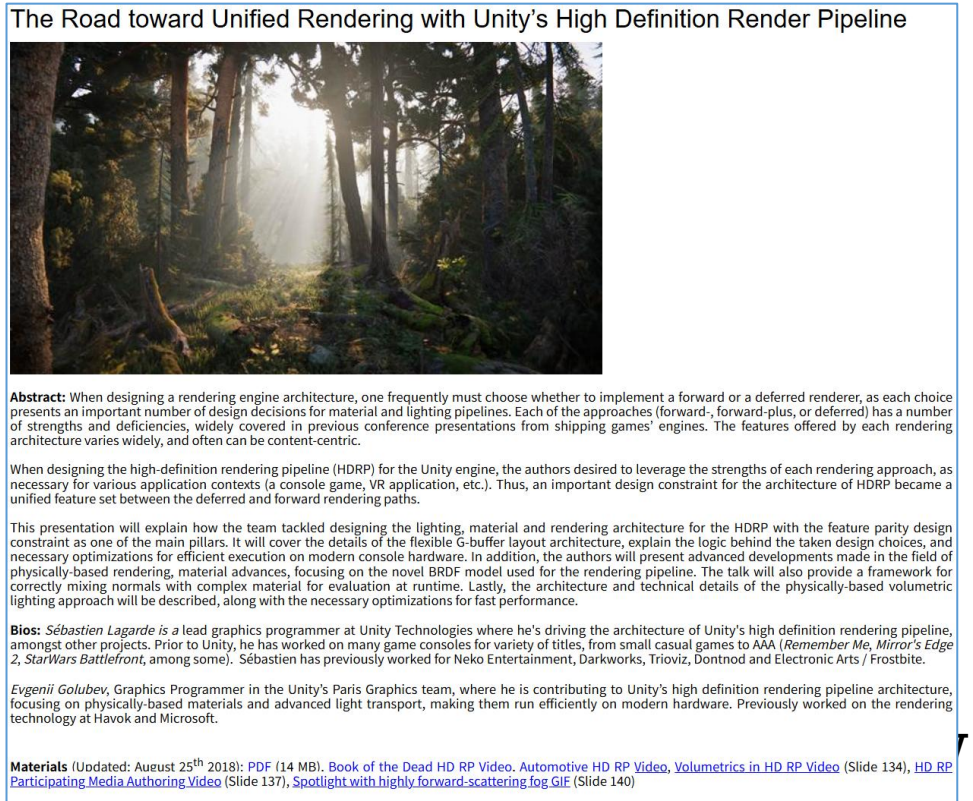
Lit Shader Graph

- Uniy側で標準的な照明計算をしてくれるシェーダ
- PBRに対して必要なパラメータを渡していく
  - 実現できない表現は？
    - Unlit Shader Graphで作る



Lit Shader Graphの中身

- The Road toward Unified Rendering with Unity’s High Definition Render Pipeline
  - https://advances.realtimerendering.com/s2018/
  - 拡散項:Disneyの拡散モデル 等
  - 鏡面反射:多重散乱等方GGX 等



The Road toward Unified Rendering with Unity’s High Definition Render Pipeline

Abstract: When designing a rendering engine architecture, one frequently must choose whether to implement a forward or a deferred renderer, as each choice presents an important number of design decisions for material and lighting pipelines. Each of the approaches (forward-, forward-plus, or deferred) has a number of strengths and deficiencies, widely covered in previous conference presentations from shipping games' engines. The features offered by each rendering architecture varies widely, and often can be content-centric.

When designing the high-definition rendering pipeline (HDRP) for the Unity engine, the authors desired to leverage the strengths of each rendering approach, as necessary for various application contexts (a console game, VR application, etc.). Thus, an important design constraint for the architecture of HDRP became a unified feature set between the deferred and forward rendering paths.

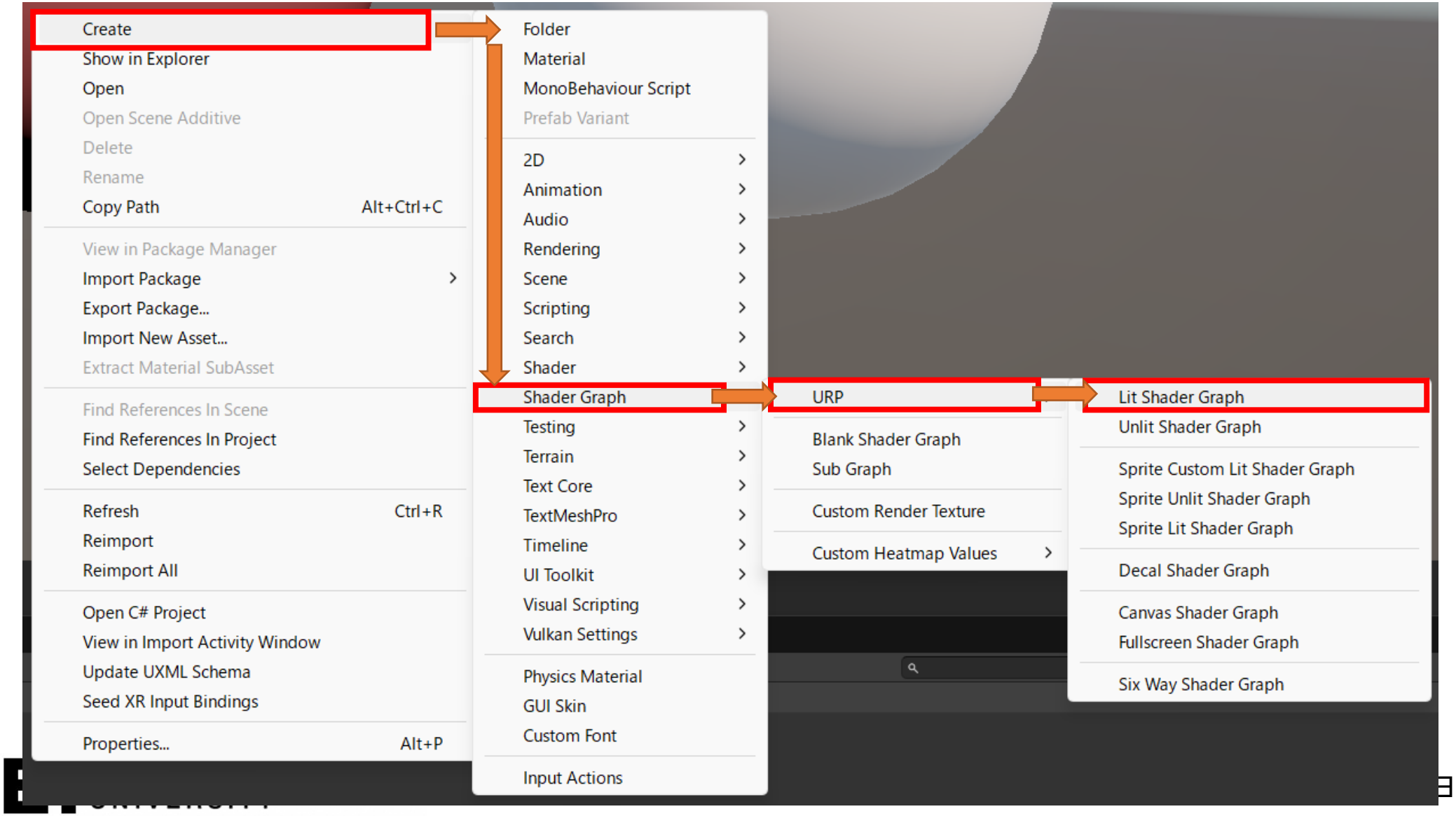
This presentation will explain how the team tackled designing the lighting, material and rendering architecture for the HDRP with the feature parity design constraint as one of the main pillars. It will cover the details of the flexible G-buffer layout architecture, explain the logic behind the taken design choices, and necessary optimizations for efficient execution on modern console hardware. In addition, the authors will present advanced developments made in the field of physically-based rendering, material advances, focusing on the novel BRDF model used for the rendering pipeline. The talk will also provide a framework for correctly mixing normals with complex material for evaluation at runtime. Lastly, the architecture and technical details of the physically-based volumetric lighting approach will be described, along with the necessary optimizations for fast performance.

Bios: *Sébastien Lagarde is a* lead graphics programmer at Unity Technologies where he's driving the architecture of Unity's high definition rendering pipeline, amongst other projects. Prior to Unity, he has worked on many game consoles for variety of titles, from small casual games to AAA (*Remember Me, Mirror's Edge 2, StarWars Battlefront*, among some). Sébastien has previously worked for Neko Entertainment, Darkworks, Trioviz, Dontnod and Electronic Arts / Frostbite.

Evgenii Golubev, Graphics Programmer in the Unity's Paris Graphics team, where he is contributing to Unity's high definition rendering pipeline architecture, focusing on physically-based materials and advanced light transport, making them run efficiently on modern hardware. Previously worked on the rendering technology at Havok and Microsoft.

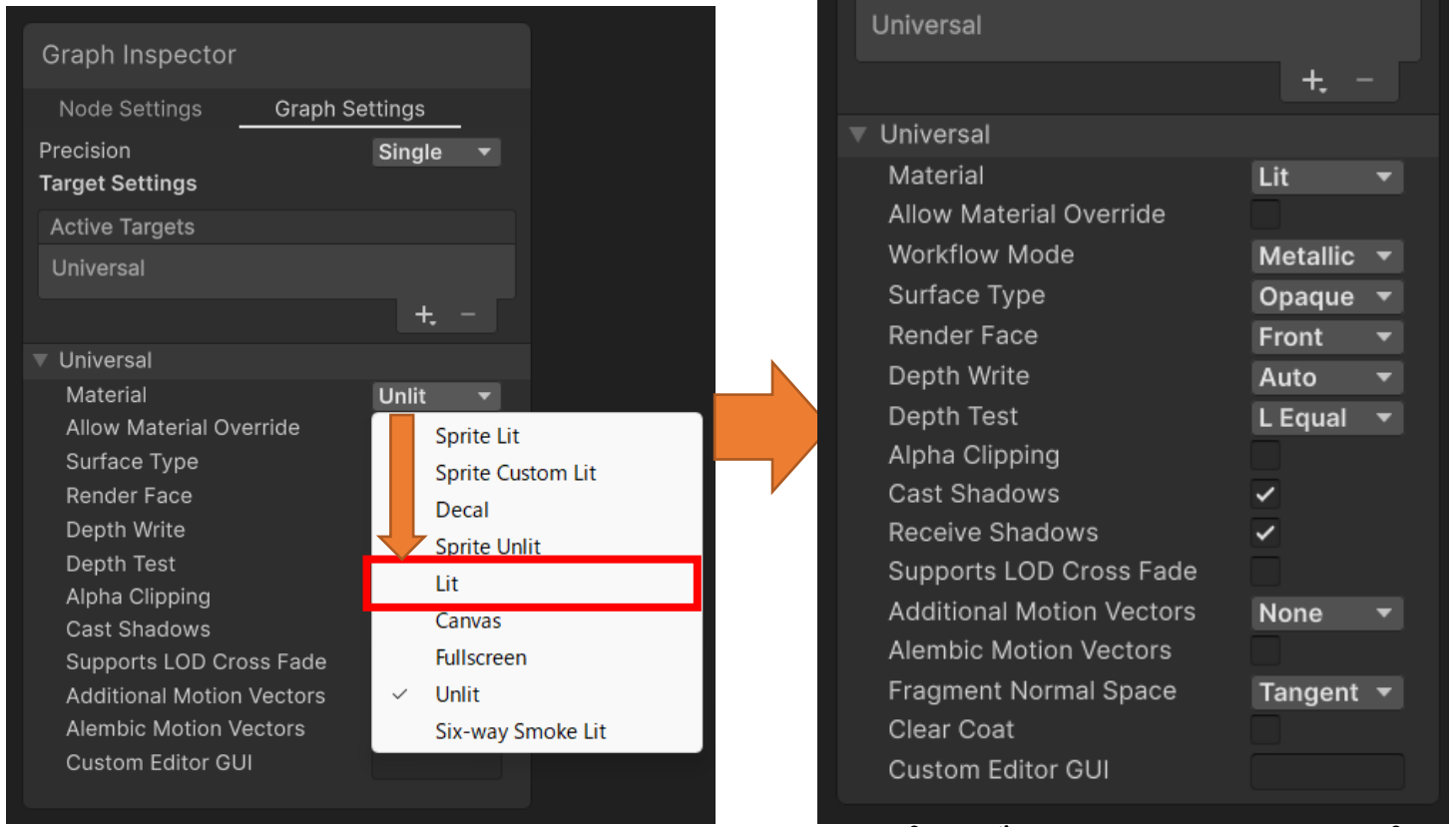
Materials (Updated: August 25th 2018): PDF (14 MB). Book of the Dead HD RP Video. Automotive HD RP Video, Volumetrics in HD RP Video (Slide 134), HD RP Participating Media Authoring Video (Slide 137), Spotlight with highly forward-scattering fog GIF (Slide 140)

Lit Shader Graphの作成



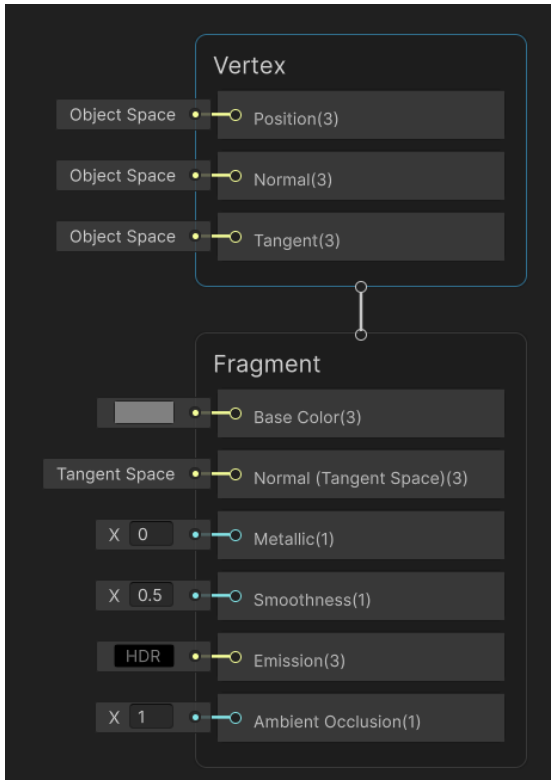
Unlit Shader Graph から切り替え

- 「Graph Inspector」の「Graph Settings」の「Universal」の「Material」を変更
  - 「Unlit」から「Lit」へ



パラメータを変更してみよう

- 実際の計算は？
  - だいたい前回の計算（AOはそのうち…）
  - 使っている数式は、常に改善されているはずなので、思い込みも危険



ビルトインのブロック

Vertex ブロック

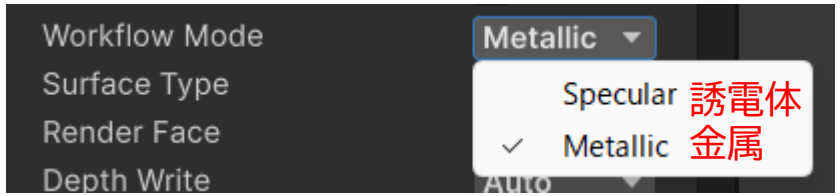
	Name	タイプ	バインディング	説明
Object Space Position(3)	Position	Vector 3	オブジェクト空間位置	絶対オブジェクト空間の頂点位置を、頂点ごとに定義します。
Object Space Normal(3)	Normal	Vector 3	オブジェクト空間法線	絶対オブジェクト空間の頂点法線を、頂点ごとに定義します。
Object Space Tangent(3)	Tangent	Vector 3	オブジェクト空間接線	絶対オブジェクト空間の頂点接線を、頂点ごとに定義します。

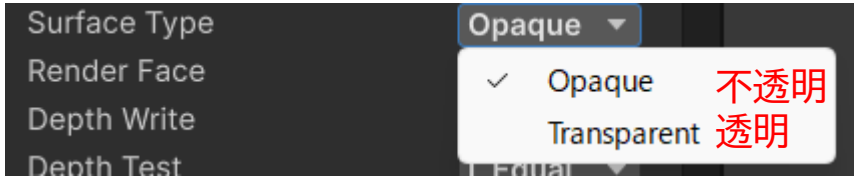
Fragment ブロック

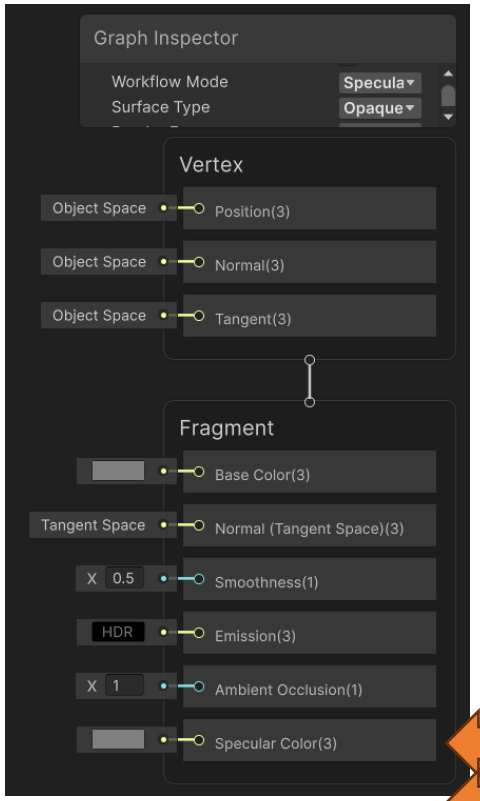
	Name	タイプ	バインディング	説明
Base Color(3)	Base Color	Vector 3	なし	マテリアルの基本色の値を定義します。想定範囲は 0 から 1 です。
Tangent Space Normal (Tangent Space)(3)	Normal (Tangent Space)	Vector 3	接線空間法線	接線空間におけるマテリアルの法線の値を定義します。
Object Space Normal (Object Space)(3)	Normal (Object Space)	Vector 3	オブジェクト空間法線	オブジェクト空間におけるマテリアルの法線の値を定義します。
World Space Normal (World Space)(3)	Normal (World Space)	Vector 3	ワールド空間法線	ワールド空間におけるマテリアルの法線の値を定義します。
HDR Emission(3)	Emission	Vector 3	なし	マテリアルの放射光の色の値を定義します。正の値が想定されます。
X 0 Metallic(1)	Metallic	Vector 1	なし	マテリアルのメタリック値を定義します。0 は非メタリック、1 はメタリックです。
Specular Color(3)	Specular	Vector 3	なし	マテリアルのスペキュラー色の値を定義します。想定範囲は 0 から 1 です。
X 0.5 Smoothness(1)	Smoothness	Vector 1	なし	マテリアルのスムースネス値を定義します。想定範囲は 0 から 1 です。
X 1 Ambient Occlusion(1)	Ambient Occlusion	Vector 1	なし	マテリアルのアンビエントオクルージョン値を定義します。想定範囲は 0 から 1 です。
X 1 Alpha(1)	Alpha	Vector 1	なし	マテリアルのアルファ値を定義します。透明性やアルファクリップに使用されます。想定範囲は 0 から 1 です。
X 0.5 Alpha Clip Threshold(1)	Alpha Clip Threshold	Vector 1	なし	アルファ値がこの設定値を下回るフラグメントが破棄されます。想定範囲は 0 から 1 です。

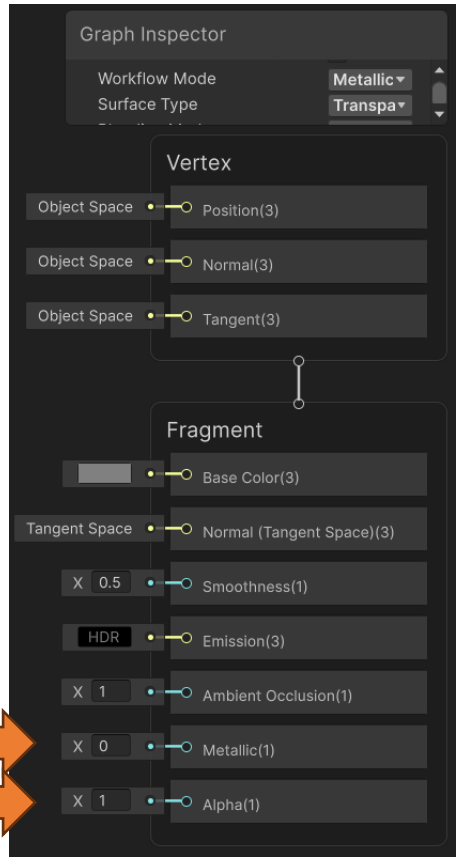
ないパラメータは？

- グラフ設定を変更







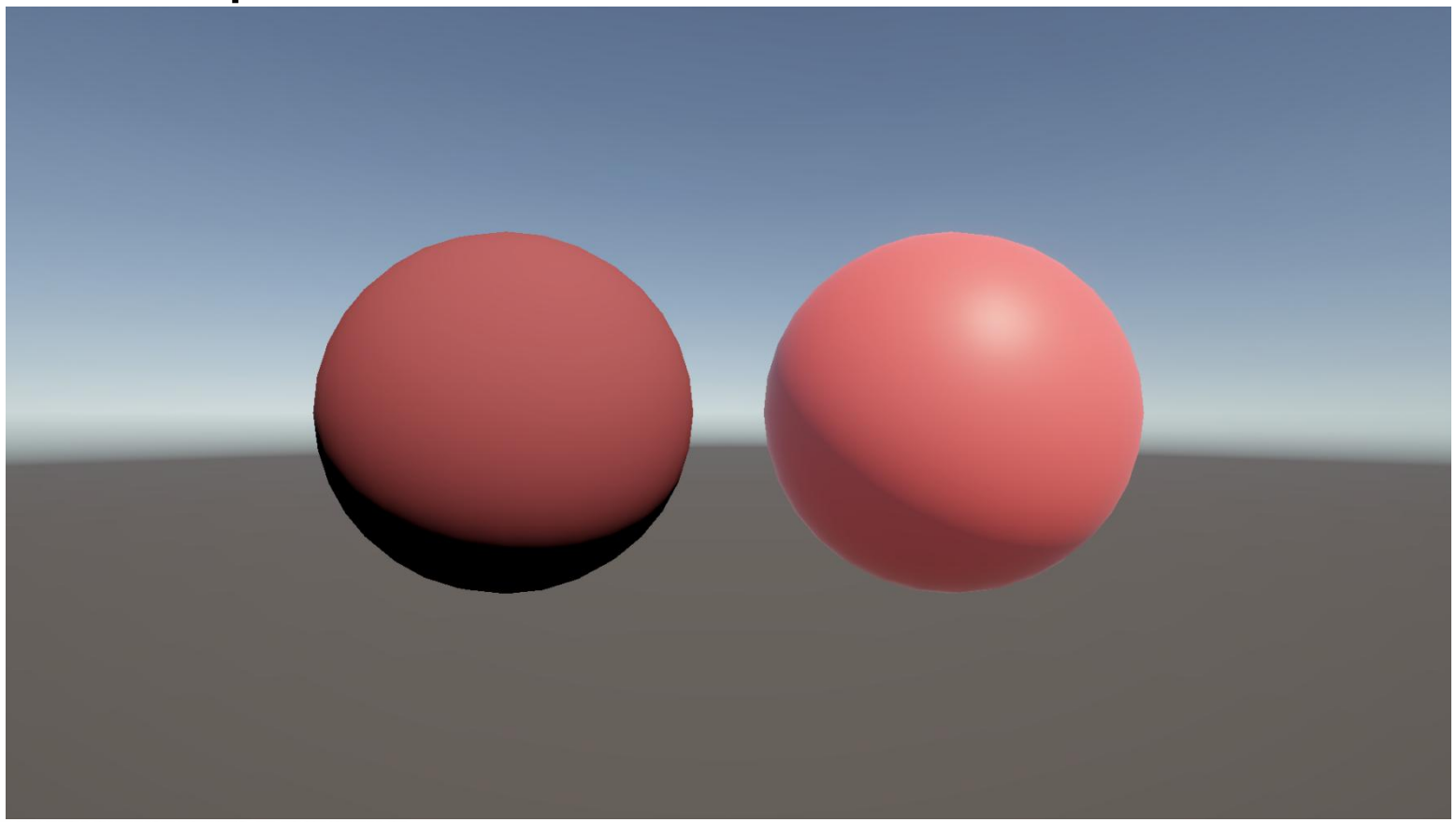


スペキュラ　メタリック

Opaque（不透明）　Transparent（半透明）

パラメータを変更してみよう

- Lit Shader Graphを使って描画してみよう



まとめ

- シェーダグラフ入門
  - シェーダグラフの追加
  - 色を変える
  - ノードの追加
  - ノードの種類
  - Lit Shader Graph