

# ShaderGraph in RealityKit

Yasuhito Nagatomo Sep 27, 2023

# 内容

1. ShaderGraph とは
2. ShaderGraph と Swift code の例

Yasuhito Nagatomo, フリーランス iOS Developer

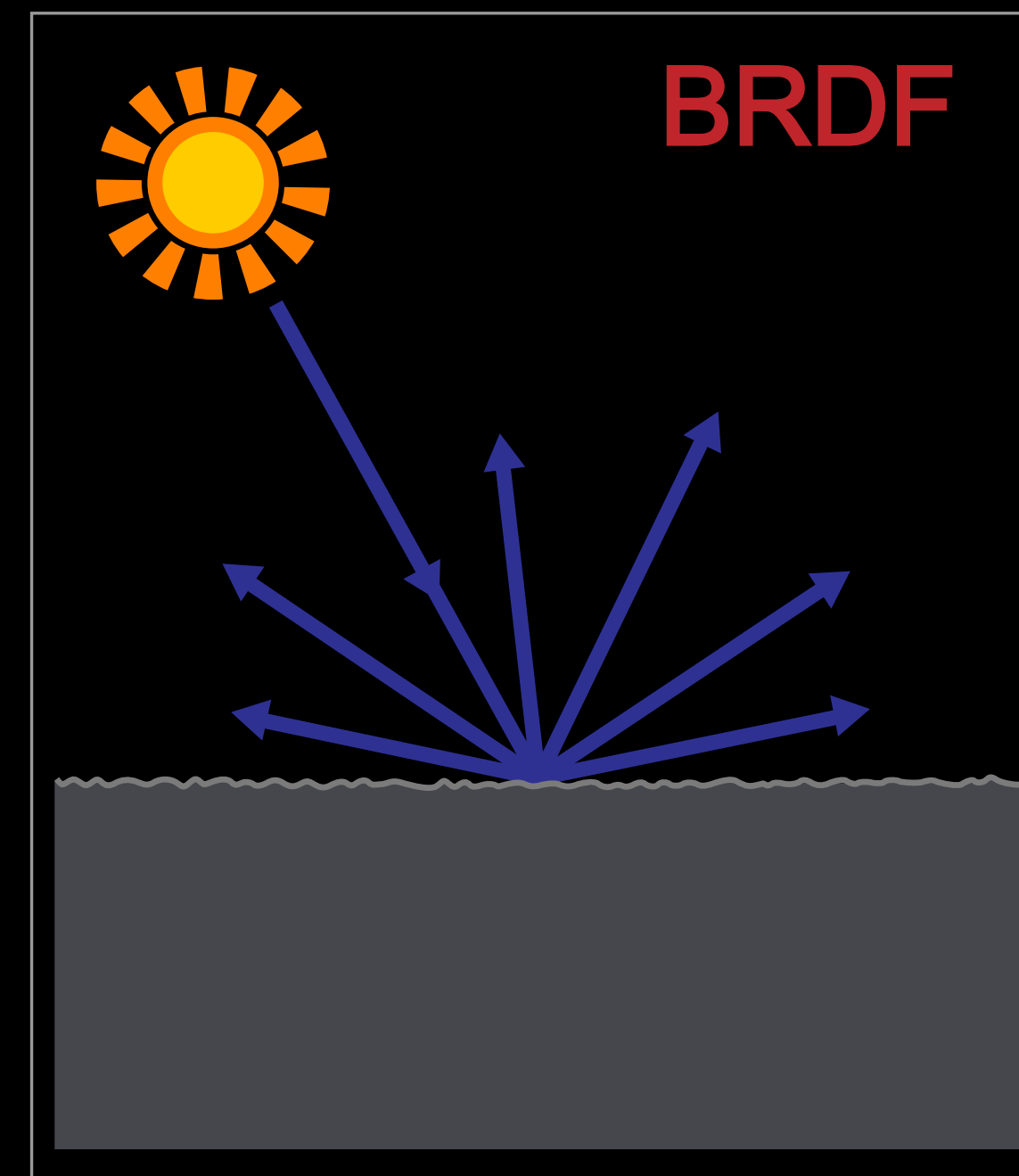


# ShaderGraph

## Shader とは

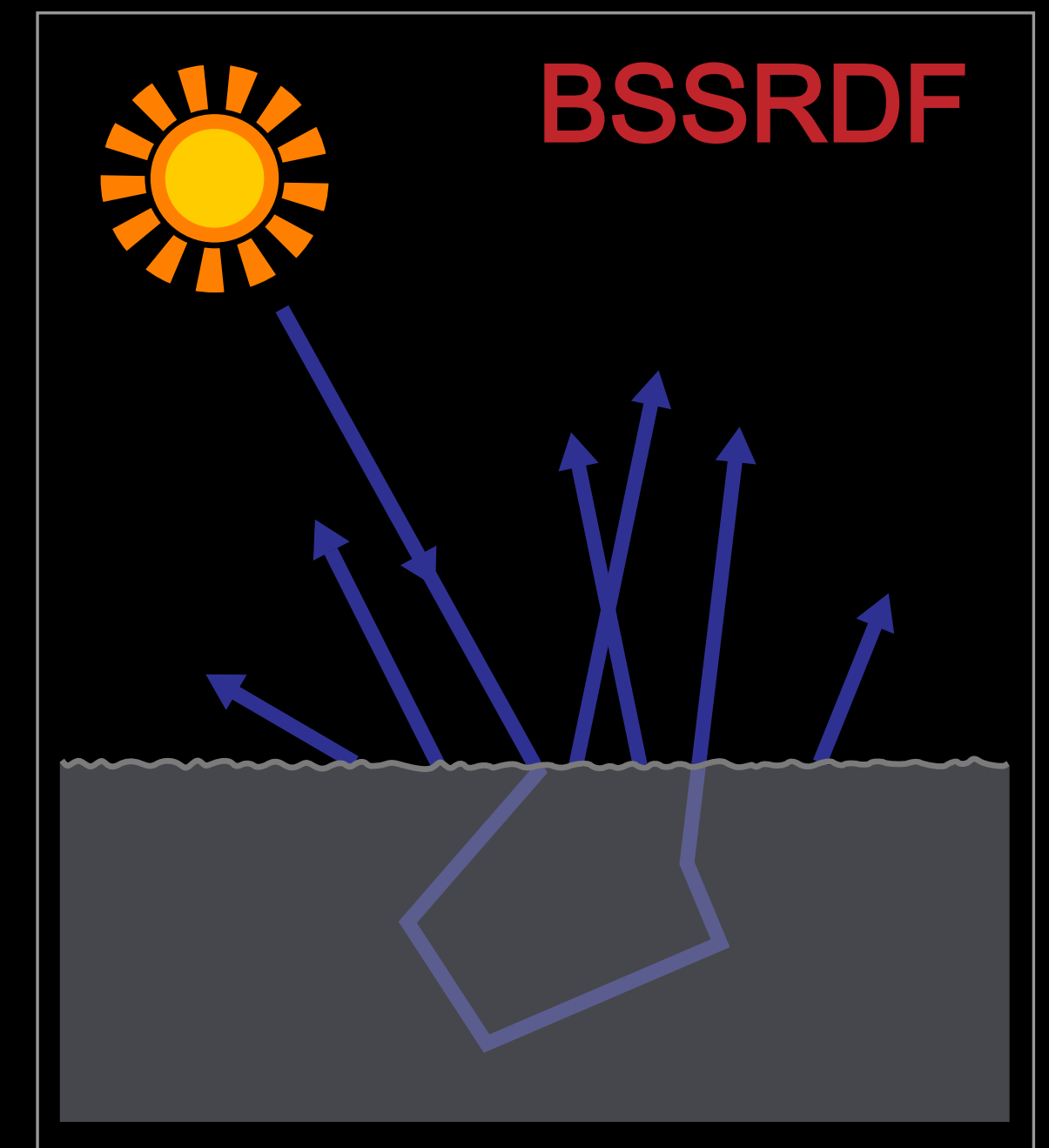
- レンダリング処理の中の陰影計算を行うプログラム（主にGPUで実行）
- 3D物体の位置と形状、表面材質を元に、物理法則をベースにした計算を行う（VertexShader, SegmentShader）
  - BaseColor（拡散反射色）, Roughness（粗さ）, Specular（鏡面反射量）などをパラメータとして拡散反射・鏡面反射などを計算する
- 形状を計算で生成したり（GeometryShader）、形状を細分化するような計算（Tessellation）もある

RealityKit



Game Engines

3D Modeling Tools



引用： Jurohi (original); Pbroks13 (redraw) - [http://en.wikipedia.org/wiki/Image:BSSDF01\\_400.png](http://en.wikipedia.org/wiki/Image:BSSDF01_400.png), CC 表示-継承 3.0

**BRDF** : Bidirectional Reflectance Distribution Function, **BSSRDF** : Bidirectional scattering-surface reflectance distribution function

# ShaderGraph

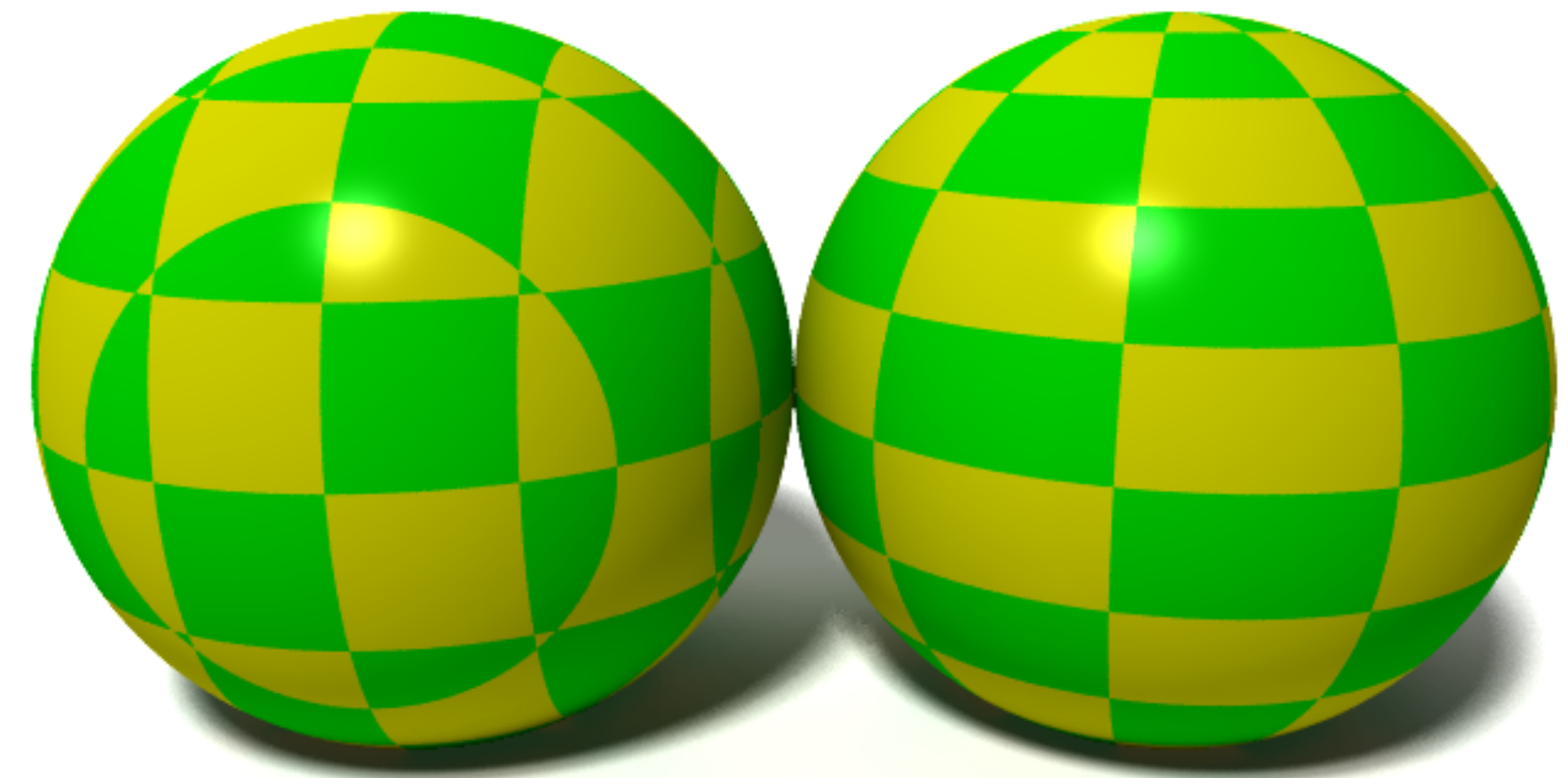
## Shaders in RealityKit

### 1. Vertex Shader (頂点シェーダ)

- 位置と形状に関する計算 (主に頂点の座標変換)
- RK では GeometryModifier と呼ぶ

### 2. Fragment Shader (フラグメントシェーダ)

- 表面材質を元に、照明・カメラの位置を考慮して、ピクセルの色を計算
- RK では SurfaceShader と呼ぶ



# ShaderGraph

## visionOSで Shader 記述が変わる

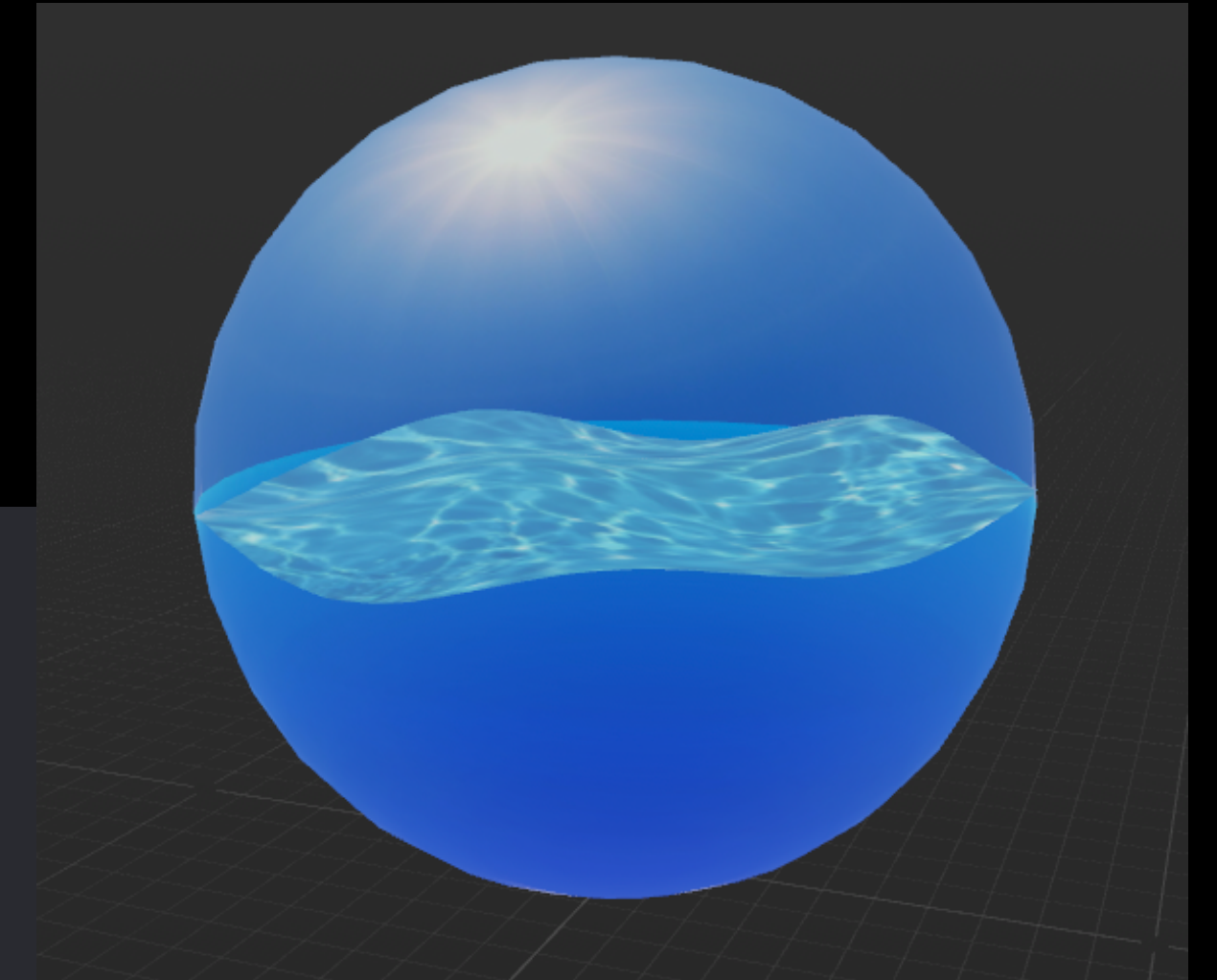
### A. RK in iOS

- MSL (Metal Shading Language): C++ 14 +/-

### B. RK in visionOS

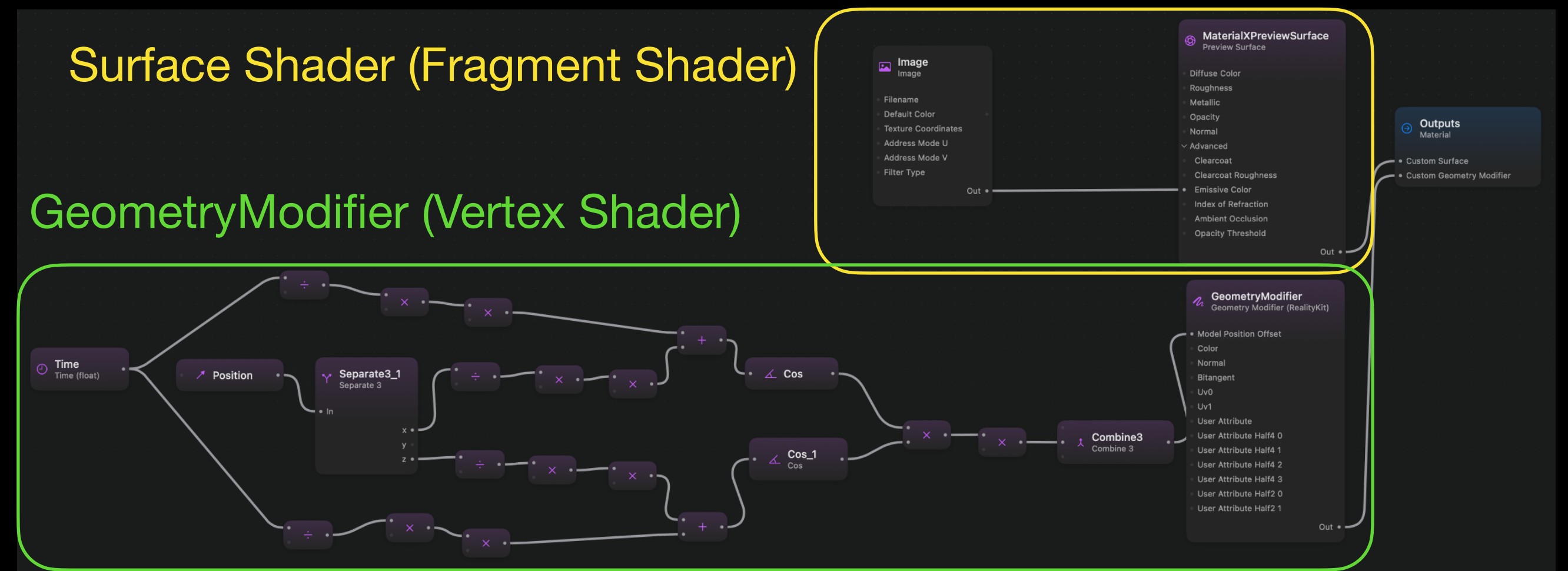
- MaterialX 標準
- Node Graph (ShaderGraph)
- GUI Node Editor (Reality Composer Pro)

```
8 #include <metal_stdlib>
9 #include <RealityKit/RealityKit.h>
10 using namespace metal;
11
12 [[visible]]
13 void waveGeometryModifier(realitykit::geometry_parameters params)
14 {
15     float3 pos = params.geometry().model_position();
16     // x axis: wave length = 0.2 [m], cycle = 8.0 [sec]
17     // z axis: wave length = 0.3 [m], cycle = 10.0 [sec]
18     // wave height = +/- 0.005 [m]
19     float3 offset = float3(0.0,
20                             cos( 3.14 * 2.0 * pos.x / 0.2 + 3.14 * 2.0 * params.uniforms().time() / 8.0 )
21                             * cos( 3.14 * 2.0 * pos.z / 0.3 + 3.14 * 2.0 * params.uniforms().time() / 10.0 ) * 0.005,
22                             0.0);
23     params.geometry().set_model_position_offset(offset);
24 }
```



### Surface Shader (Fragment Shader)

### GeometryModifier (Vertex Shader)





# ShaderGraph

## ShaderGraph を学ぶ

### 1. WWDC23

- Session: Explore materials in Reality Composer Pro (<https://developer.apple.com/videos/play/wwdc2023/10202/>)

### 2. Apple Documentations

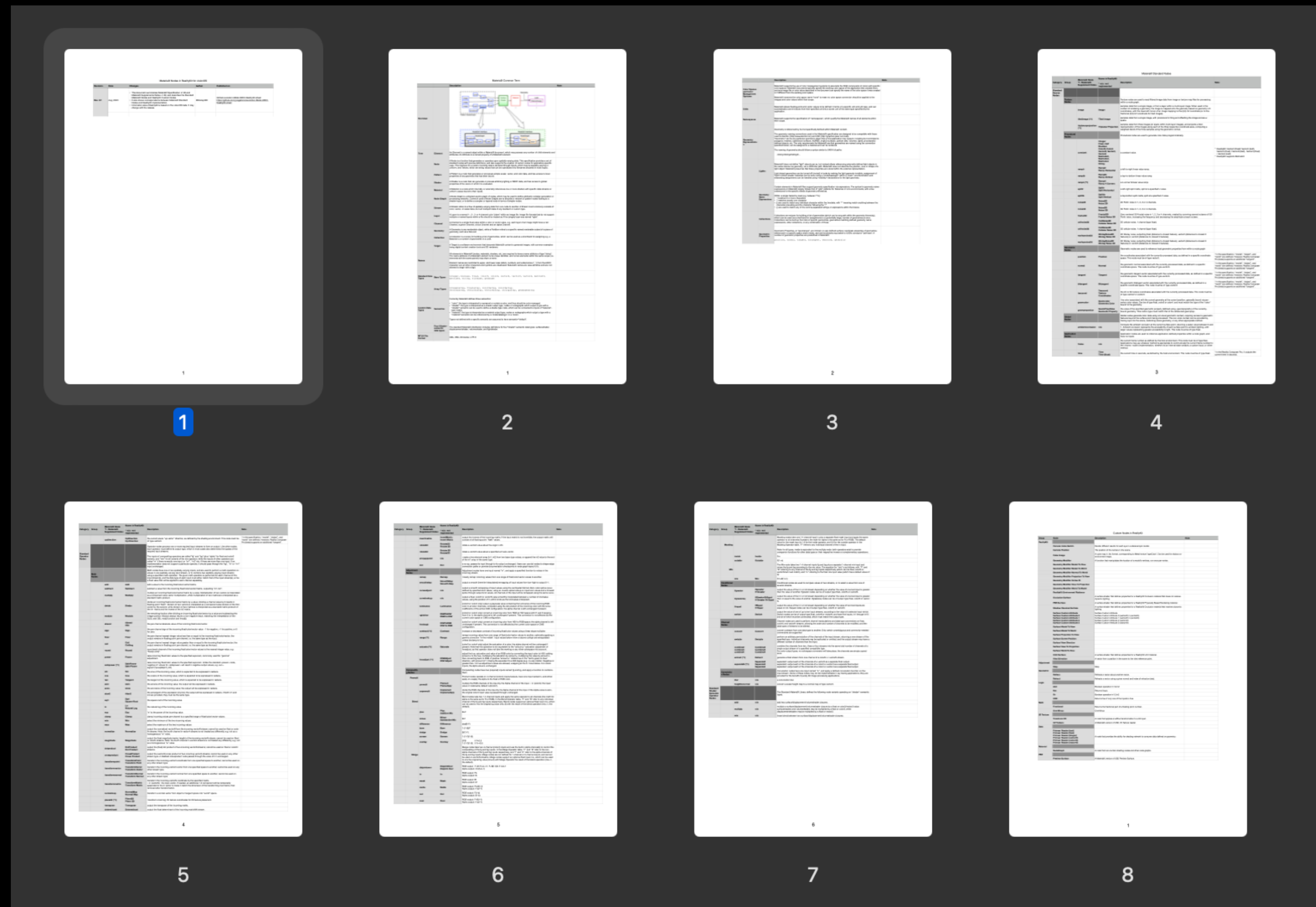
- Article: Designing RealityKit content with Reality Composer Pro (<https://developer.apple.com/documentation/visionOS/designing-realitykit-content-with-reality-composer-pro>)

### 3. MaterialX

- MaterialX Spec v1.38 (<https://materialx.org/>)

### 4. Note

- MaterialX Nodes in RealityKit (<https://github.com/ynagatomo/evolution-Metal-ARKit-RealityKit-sheet>)
- MaterialX + RealityKit 拡張 : 約 140 個の Nodes



MaterialX Nodes in RealityKit  
(<https://github.com/ynagatomo/evolution-Metal-ARKit-RealityKit-sheet>)

# Example

## Skybox Animation

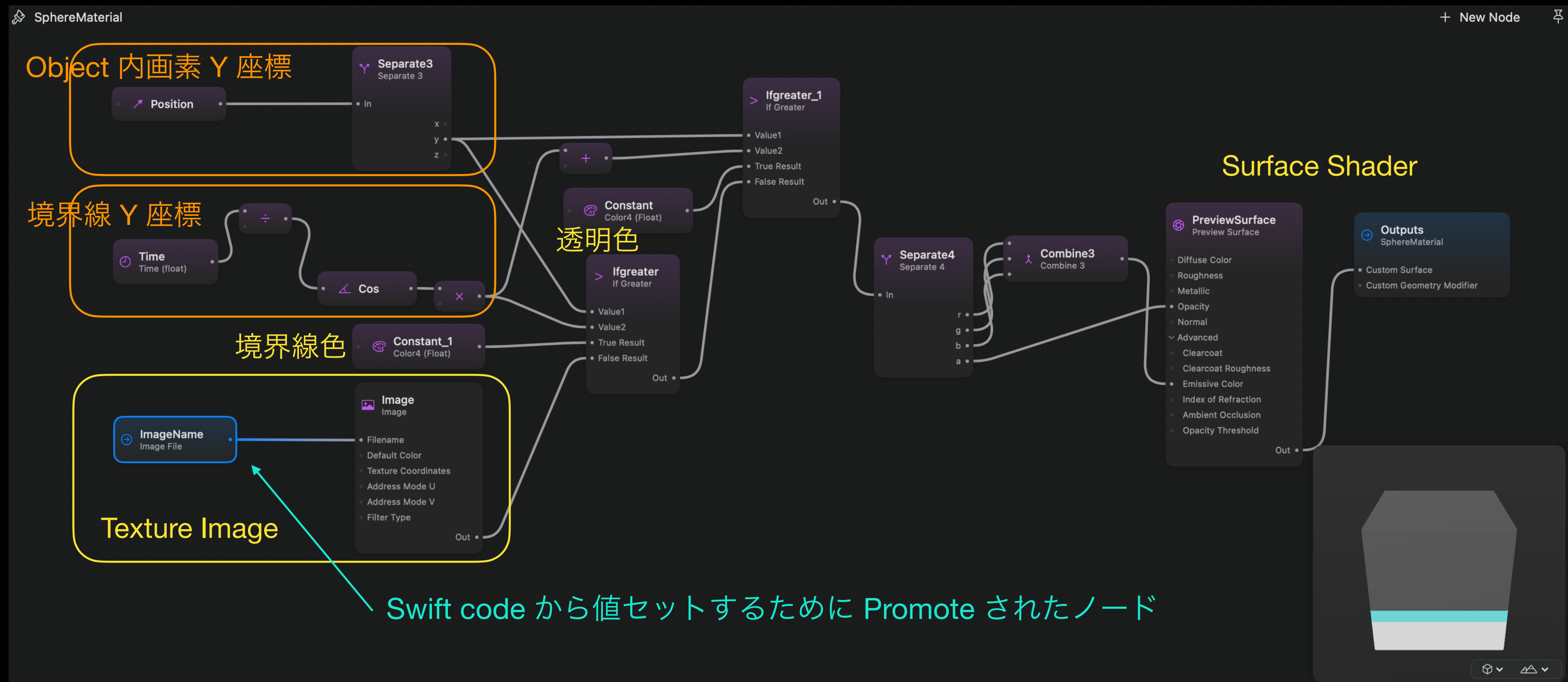
- 球体 (Sphere) の内側にテクスチャ画像 (全天球イメージ equirectangular) を貼る
- 時間経過に合わせて下から上、上から下に全天球イメージが現れる (通常は半透明Gray色)
- テクスチャ画像は Swift Code で切り替えることができる



# Example

## Skybox Animation: ShaderGraph

- ShaderGraph Material は Reality Composer Pro を使って作成し、シーン作成のための Material として使用できます。現時点では Particle System では使えない
- Swift code にて、ShaderGraph Material を読み込み使用することもできます（本 Example でサンプル提示）





# Example

## Skybox Animation: Swift Code

- Promote した Node の値を Swift Code で設定・変更することができます
- TextureResource の値設定は WWDC/Article/Sample で示されていないので本 Example で提示します

```
12 struct ImmersiveView: View {
13     var body: some View {
14         RealityView { content in
15             if let scene = try? await Entity(named: "Immersive", in: realityKitContentBundle) {
16                 if let sphere = scene.findEntity(named: "Sphere_Right") as? ModelEntity {
17                     if var sphereMaterial = sphere.model?.materials.first as? ShaderGraphMaterial { ← 1) ShaderGraph Material 取得
18                         if let textureResource = try? await TextureResource(named: "town360") { ← 2) Image Texture 作成
19                             try? sphereMaterial.setParameter(name: "ImageName", value: .textureResource(textureResource))
20                             // Attach the material to a sphere.
21                             let entity = Entity()
22                             entity.components.set(ModelComponent(
23                                 mesh: .generateSphere(radius: 2),
24                                 materials: [sphereMaterial] ← 4) ShaderGraph Material を貼った球体 作成
25                             ))
26                             // Ensure the texture image points inward at the viewer.
27                             entity.scale *= .init(x: -1, y: 1, z: 1) ← 5) 球体の内側表面に Material を適用
28                             entity.transform.translation = SIMD3<Float>(0, 1, 0)
29                             content.add(entity) ← 6) content に追加

```

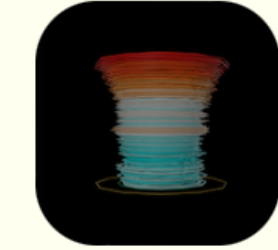
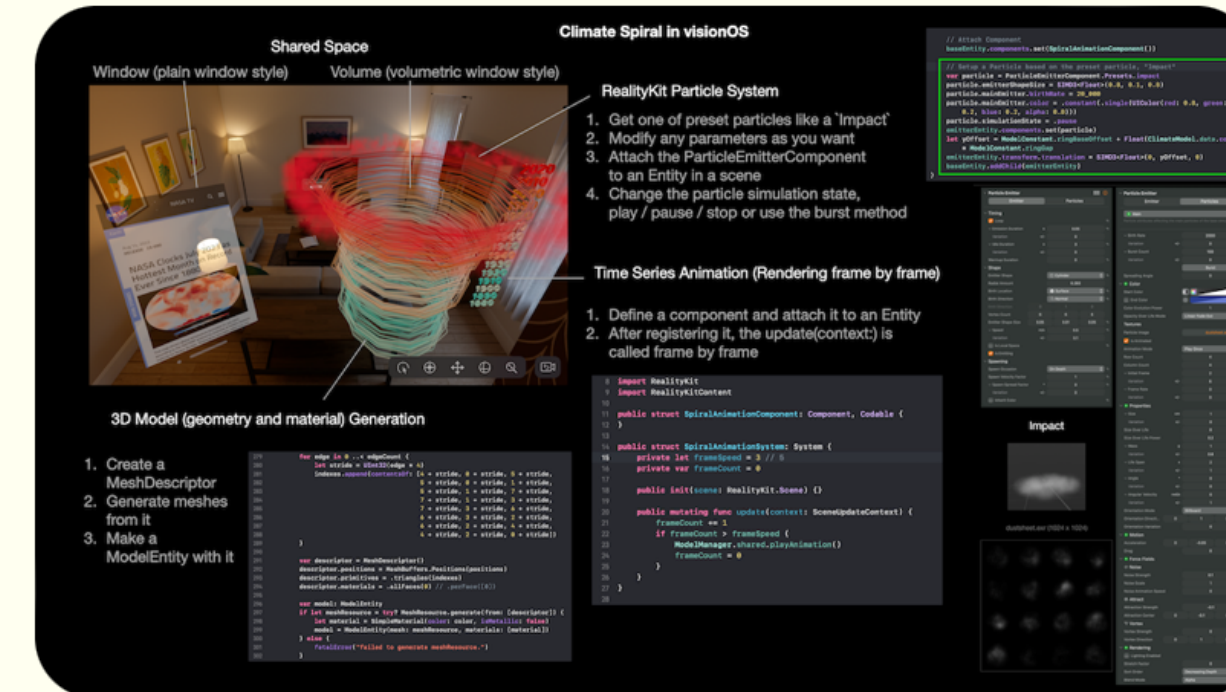
↑  
3) ImageName ノードに値をセット

# まとめ

## ShaderGraph in RK

- visionOS にShaderGraphが導入された
- ShaderGraph と Swift Code は連携できる

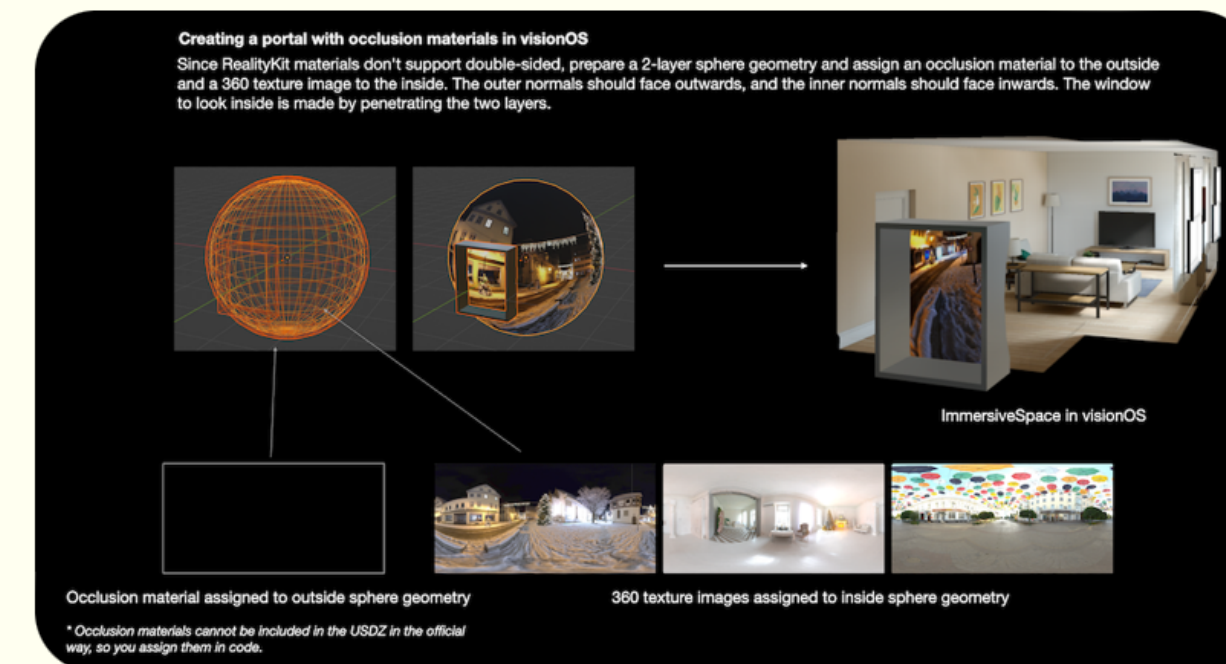
- X: @AtarayoSD
- GitHub: ynagatomo (<https://github.com/ynagatomo>)



Climate Spiral in visionOS [GitHub](#)

A simple visionOS app that displays the Climate Spiral

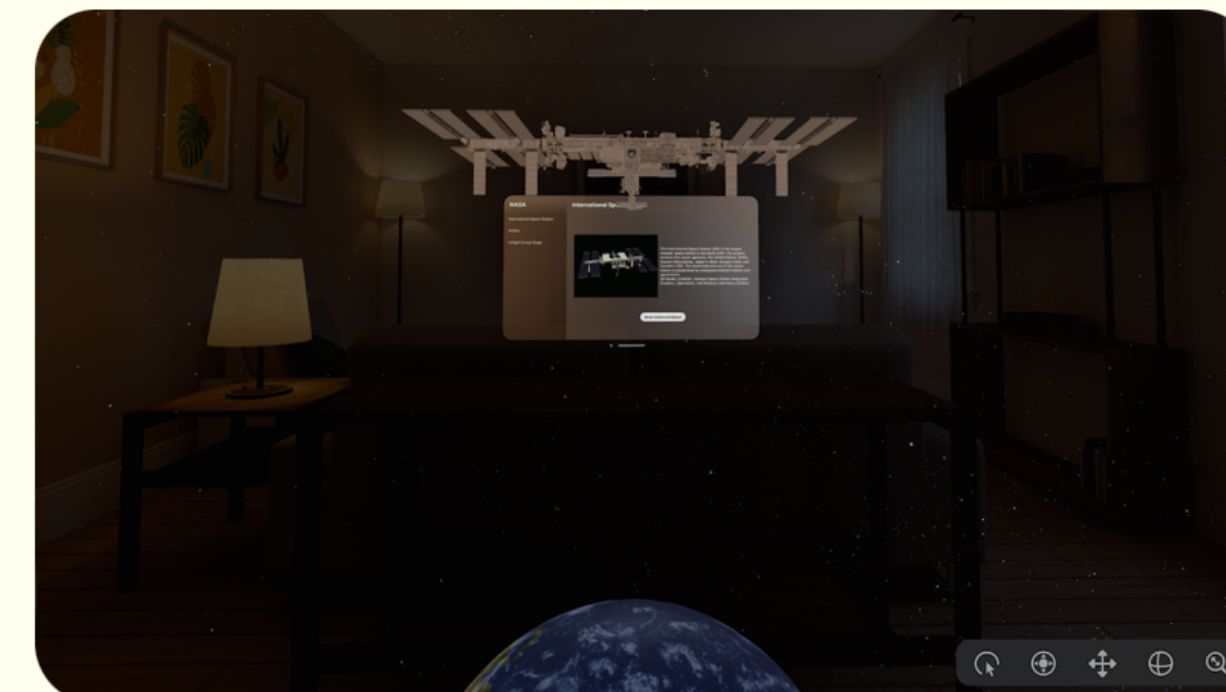
visionOS, SwiftUI, RealityKit



Portal with Occlusion Material in visionOS [GitHub](#)

A simple visionOS app that displays a portal with an Occlusion Material

visionOS, SwiftUI, RealityKit



ISS in your room in visionOS [GitHub](#)

A simple visionOS app that displays the ISS in your room

visionOS, SwiftUI, RealityKit

Projects for visionOS on GitHub