

ダンカグライクにShader/PostEffectを追加する方法

著者:Wicky 翻訳:たらていー

Step 1:[Unityのウェブサイト](#)から **Unity Editor 2021.3** をダウンロードしてインストールします。

警告: 2021.x 以外のバージョンをインストールするとクラッシュの原因になる可能性があるので注意！

+まだエディターをインストールしていない場合、以下をインストールしてください:

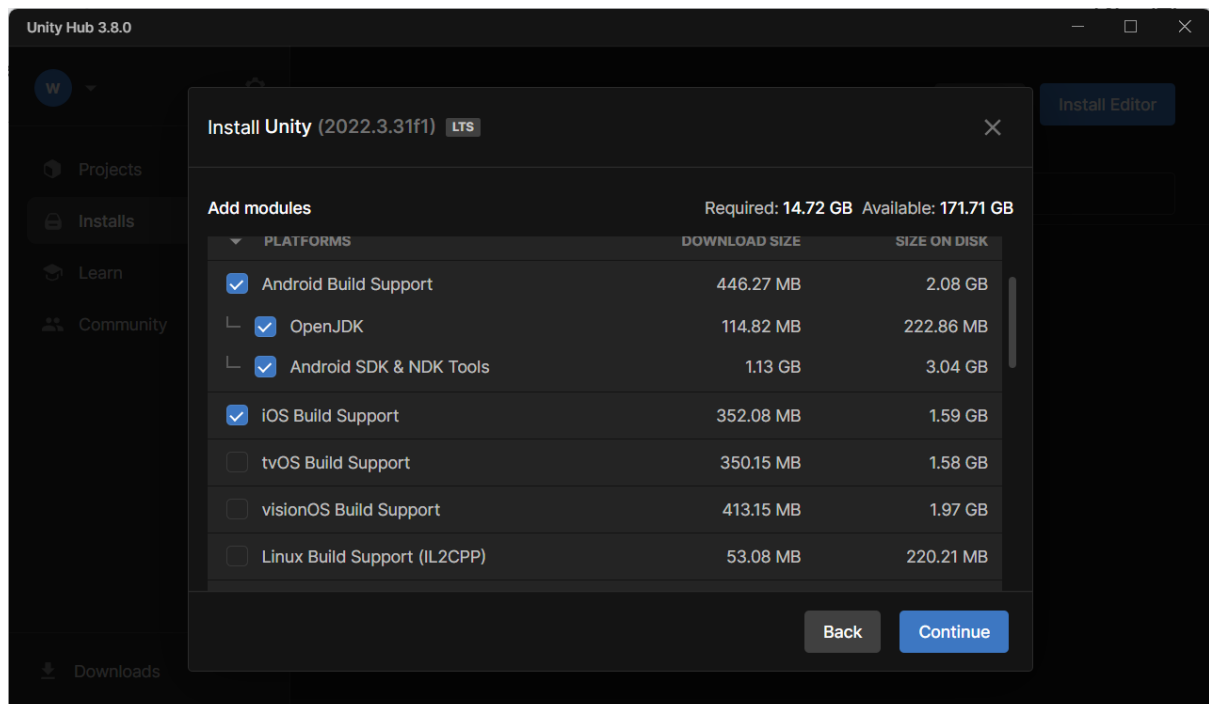
- Android Build Support
- iOS Build Support
- Universal Platform Build Support
- Mac Build Support (Mono)

+既にエディタをインストール済みの場合:

- Unity Hubを開く
- 「Installs」タブでインストール済みのバージョンの右側にある歯車アイコンをクリック
- 「Add modules」をクリックして必要なビルドサポートをインストール

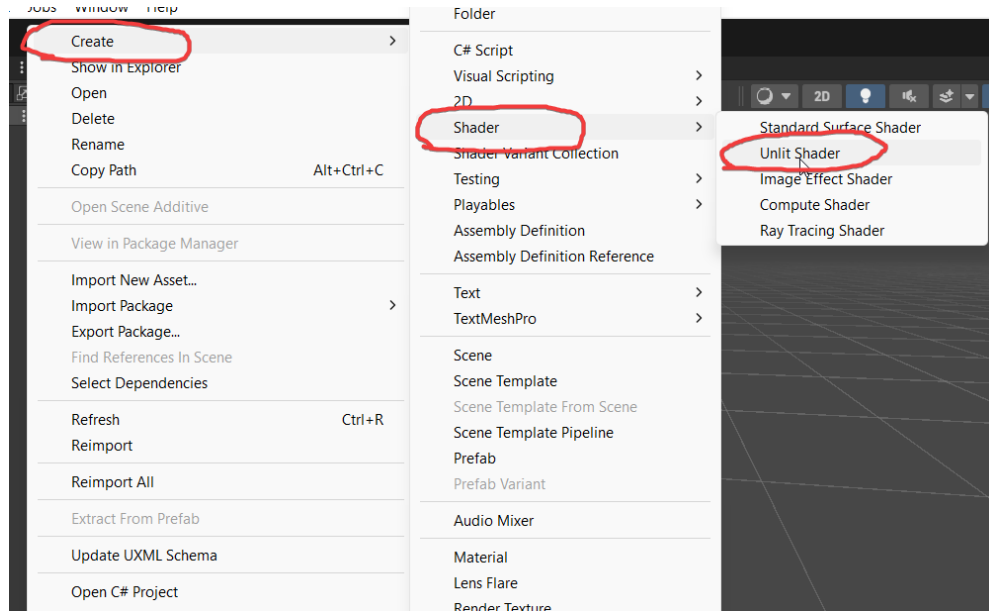
注意: エディターをインストールすると10GB以上のストレージを消費する可能性があります。

メモ: Visual Studio を使用しない場合はチェックを外してください

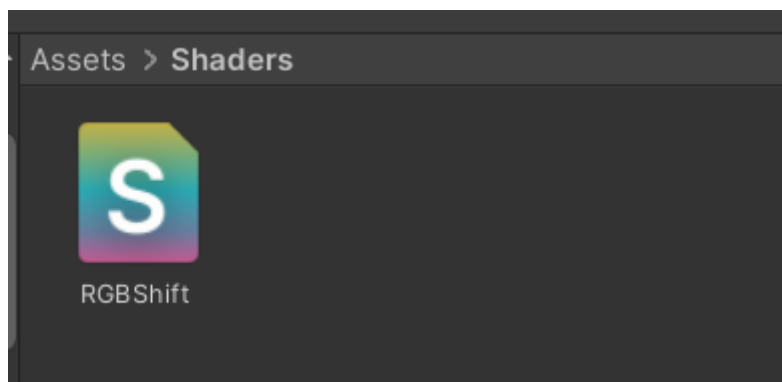


Step 2: 新しいシェーダーを作成する

1. Unityを開き、新しいプロジェクトを作成します。
2. プロジェクト内で、Assetsウィンドウ(画面下部)を右クリックし、「Create」→「Shader」→「Unlit Shader」を選択します。



Step 3: このシェーダーに任意の名前を付けます。(例: "RGBShift")

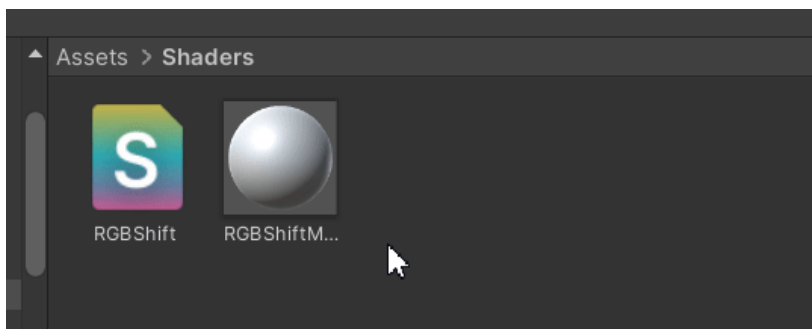


Step 4: マテリアルを作成する

1. 右クリックし、「Create」→「Material」を選択します。
2. 作成したマテリアルに任意の名前を付けます。(例: "RGBShiftMat")

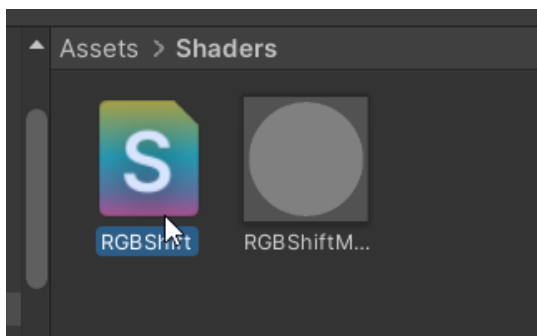


Step 5:シェーダーを作成したマテリアルにドラッグ & ドロップします。



Step 6:作成したシェーダー（例: "RGBShift"）をダブルクリックして開きます。インストールされているIDEが開きます。

メモ: IDEがインストールされていない場合、Notepadが開きますが、プログラミングに興味がある場合はIDEをインストールすることをお勧めします。



Step 7: シェーダーにコードを書き込む

シェーダーを使うには自分でソースコードを書き込む必要があります。Unityでは、Shaderlabという宣言型言語を使用して、Rendererのverticesや色を操作します。

しかし、プログラミングやUnityに詳しくない人には難しいので、既存のシェーダーコードを使用することをオススメします。

このチュートリアルでは、以下のリンクからRGBShiftコードを借用しています。(作成者に感謝します)

[RGBShiftシェーダーコード](#)

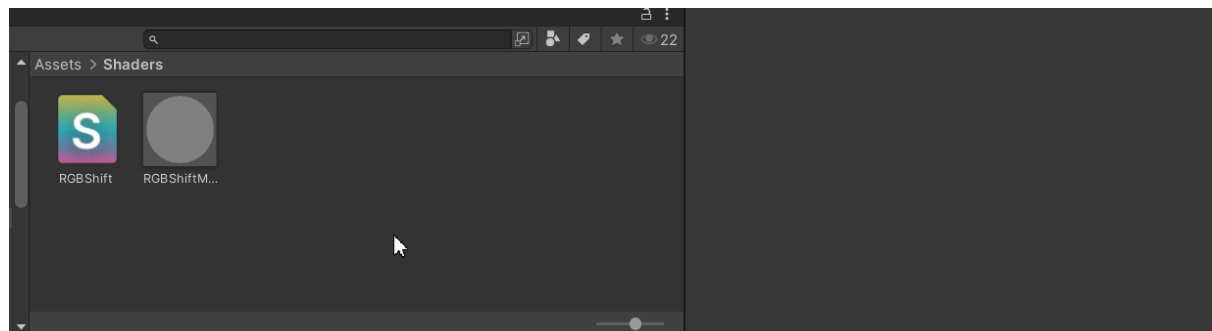
[Githubリポジトリ](#)には他にも役立つシェーダーがたくさんあります。

また、AI(ChatGPTなど)もシンプルなShaderlabコードを生成する手助けをしてくれますが、エラーが発生する可能性があるため、推奨しません。

```
RGBShift.shader X
D: > Projects > DKLike > Assets > Shaders > RGBShift.shader
1  Shader "Custom/Shader" {
2      Properties {
3          _MainTex ("MainTex", 2D) = "" {}
4          _Amount ("Amount", Range(0, 0.1)) = 0.005
5          _Speed ("Speed", Range(0, 20)) = 10
6      }
7
8
9      SubShader {
10         Pass {
11             CGPROGRAM
12             #include "UnityCG.cginc"
13             #pragma vertex vert_img
```

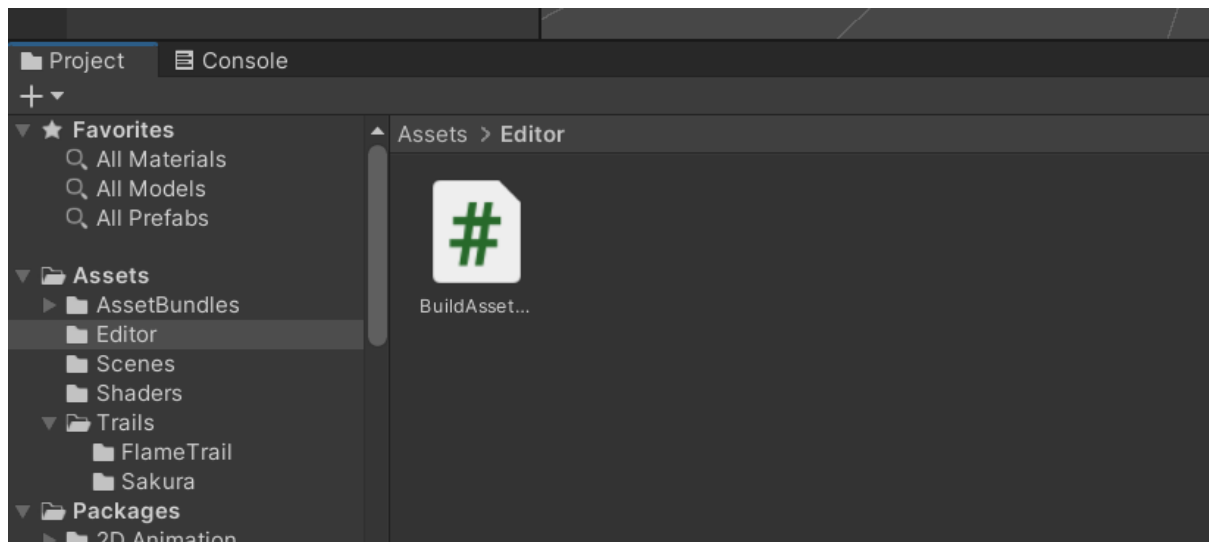
Step 8:アセットバンドルグループを作成し、任意の名前を付けます。(例: "rgbshift")

上記の2つのオブジェクトが「rgbshift」アセットバンドルに含まれていることを確認します。



Step 9:"Assets/Editor" ディレクトリに "BuildAssetBundle" という名前のスクリプトを作成します。(このディレクトリがない場合は作成してください)

作成できたら、スクリプトが実装されるまで待ちます。



Step 10: BuildAssetBundle.cs を開き、この[リンク](#)からスクリプトをコピーして貼り付けます。

Step 11: アセットバンドルをビルドする

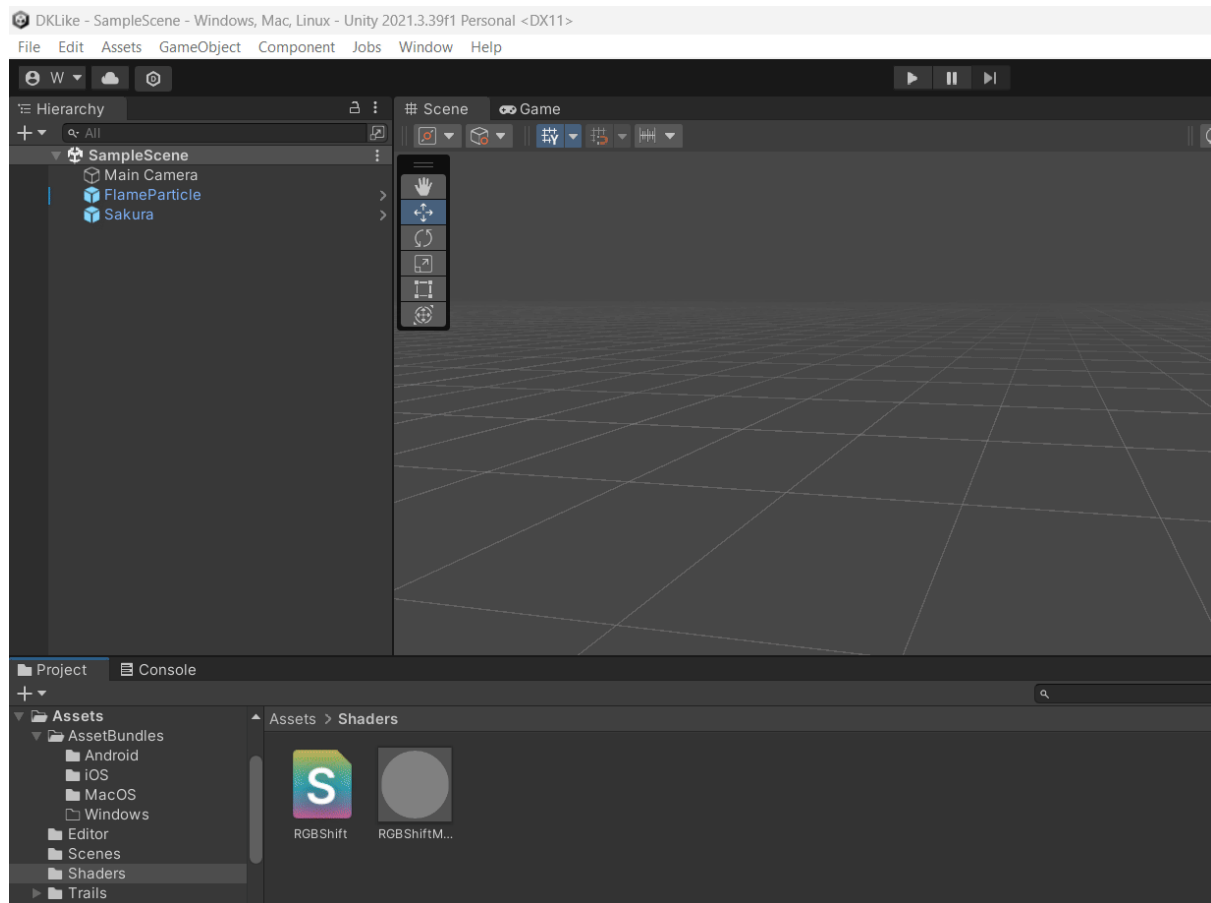
上記の2つのオブジェクトを選択し、以下のメニュー項目を実行します（初回の実行にはかなり時間がかかる場合があります）:

Assets -> Build AB Windows

Assets -> Build AB Android

Assets -> Build AB iOS

Assets -> Build AB MacOS



Step 12: アセットバンドルを移動する

Assets/AssetBundles フォルダに4つのプラットフォームに対応するフォルダが作成されます。

これらのフォルダをチャートフォルダ内の「modules」フォルダに移動します。任意のフォルダ名に変更できます。

Step 13: 譜面フォルダ内に "script.lua" という名前のLuaスクリプトを作成します。

Step 14: Luaスクリプトにコードを書き込む

以下のコードをコピーして "script.lua" ファイルに貼り付けます。

```
local function GetPlatformPath()
    local platform = APPMAN:GetPlatformInt()
    if (platform == 1) then
        return "Windows"
    end
    if (platform == 2) then
```

```

        return "MacOS"
    end
    if (platform == 3) then
        return "Android"
    end
    if (platform == 4) then
        return "iOS"
    end
    return "Other/"
end

function onloaded()

    local platform = GetPlatformPath()
    local assetbundleHash = ASSETMAN:LoadAssetBundle("modules/"..
platform .. "/rgbshift") --name of asset bundle group
    _mosaicMat = ASSETMAN:LoadMaterial(assetbundleHash, "RGBShiftMat")
--name of material in Unity

    local _postEffect = CAMERAMAN:GetPostEffect()
    _postEffect:SetMaterial(_mosaicMat)
    _postEffect:SetEnable(true)
end

```

このスクリプトは、チャートの読み込み時にエフェクトが「rgbshift」からハッシュ値を取得し、そのハッシュ値を使ってRGBShiftMatオブジェクトを取得します。その後、カメラにそのマテリアルオブジェクトをロードします。

(ハッシュ値というのはデータから算出される暗号のようなものです。ITのセキュリティ分野とかに興味がある人は調べてみてください。よくわからなかったら無視でいいです。)

Step 15:オプションでカスタマイズする

オプションで設定をカスタマイズする場合、以下の2行を追加します。

```

3  _mosaicMat = ASSETMAN:LoadMaterial(assetBundleName, "Mosaic")
4
5  local _postEffect = CAMERAMAN:GetPostEffect()
6  _postEffect:SetMaterial(_mosaicMat)
7
8  --Custom properties
9  _postEffect:SetFloat('_Amount', 0.01)
10 _postEffect:SetFloat('_Speed', 10)
11
12 _postEffect:SetEnable(true)
13 end

```

プロパティはシェーダーファイルに記載されたShaderlabコードから指定します。

```

D: > Projects > DKLike > Assets > Shaders > RGBShift.shader
1  Shader "Custom/Shader" {
2    Properties {
3      _MainTex ("MainTex", 2D) = "" {}
4      _Amount ("Amount", Range(0, 0.1)) = 0.005
5      _Speed ("Speed", Range(0, 20)) = 10
6    }
7
8

```

Step 16:完成 !

