using System;

using UnityEditor;

using UnityEngine;

namespace UnityStandardAssets.ImageEffects

{

[CustomEditor (typeof(ColorCorrectionLookup))]

class ColorCorrectionLookupEditor : Editor

{

SerializedObject serObj;

void OnEnable () {

serObj = new SerializedObject (target);

}

private Texture2D tempClutTex2D;

public override void OnInspectorGUI () {

serObj.Update ();

EditorGUILayout.LabelField("Converts textures into color lookup volumes (for grading)", EditorStyles.miniLabel);

//EditorGUILayout.LabelField("Change Lookup Texture (LUT):");

//EditorGUILayout.BeginHorizontal ();

//Rect r = GUILayoutUtility.GetAspectRect(1.0ff);

Rect r; Texture2D t;

//EditorGUILayout.Space();

tempClutTex2D = EditorGUILayout.ObjectField (" Based on", tempClutTex2D, typeof(Texture2D), false) as Texture2D;

if (tempClutTex2D == null) {

t = AssetDatabase.LoadMainAssetAtPath(((ColorCorrectionLookup)target).basedOnTempTex) as Texture2D;

if (t) tempClutTex2D = t;

}

Texture2D tex = tempClutTex2D;

if (tex && (target as ColorCorrectionLookup).basedOnTempTex != AssetDatabase.GetAssetPath(tex))

{

EditorGUILayout.Separator();

if (!(target as ColorCorrectionLookup).ValidDimensions(tex))

{

EditorGUILayout.HelpBox ("Invalid texture dimensions!\nPick another texture or adjust dimension to e.g. 256x16.", MessageType.Warning);

}

else if (GUILayout.Button ("Convert and Apply"))

{

string path = AssetDatabase.GetAssetPath (tex);

TextureImporter textureImporter = AssetImporter.GetAtPath(path) as TextureImporter;

bool doImport = textureImporter.isReadable == false;

if (textureImporter.mipmapEnabled == true) {

doImport = true;

}

if (textureImporter.textureFormat != TextureImporterFormat.AutomaticTruecolor) {

doImport = true;

}

if (doImport)

{

textureImporter.isReadable = true;

textureImporter.mipmapEnabled = false;

textureImporter.textureFormat = TextureImporterFormat.AutomaticTruecolor;

AssetDatabase.ImportAsset (path, ImportAssetOptions.ForceUpdate);

//tex = AssetDatabase.LoadMainAssetAtPath(path);

}

(target as ColorCorrectionLookup).Convert(tex, path);

}

}

if ((target as ColorCorrectionLookup).basedOnTempTex != "")

{

EditorGUILayout.HelpBox("Using " + (target as ColorCorrectionLookup).basedOnTempTex, MessageType.Info);

t = AssetDatabase.LoadMainAssetAtPath(((ColorCorrectionLookup)target).basedOnTempTex) as Texture2D;

if (t) {

r = GUILayoutUtility.GetLastRect();

r = GUILayoutUtility.GetRect(r.width, 20);

r.x += r.width \* 0.05f/2.0f;

r.width \*= 0.95f;

GUI.DrawTexture (r, t);

GUILayoutUtility.GetRect(r.width, 4);

}

}

//EditorGUILayout.EndHorizontal ();

serObj.ApplyModifiedProperties();

}

}

}