using System;

using UnityEditor;

using UnityEngine;

namespace UnityStandardAssets.ImageEffects

{

[CustomEditor (typeof(ColorCorrectionCurves))]

class ColorCorrectionCurvesEditor : Editor {

SerializedObject serObj;

SerializedProperty mode;

SerializedProperty redChannel;

SerializedProperty greenChannel;

SerializedProperty blueChannel;

SerializedProperty useDepthCorrection;

SerializedProperty depthRedChannel;

SerializedProperty depthGreenChannel;

SerializedProperty depthBlueChannel;

SerializedProperty zCurveChannel;

SerializedProperty saturation;

SerializedProperty selectiveCc;

SerializedProperty selectiveFromColor;

SerializedProperty selectiveToColor;

private bool applyCurveChanges = false;

void OnEnable () {

serObj = new SerializedObject (target);

mode = serObj.FindProperty ("mode");

saturation = serObj.FindProperty ("saturation");

redChannel = serObj.FindProperty ("redChannel");

greenChannel = serObj.FindProperty ("greenChannel");

blueChannel = serObj.FindProperty ("blueChannel");

useDepthCorrection = serObj.FindProperty ("useDepthCorrection");

zCurveChannel = serObj.FindProperty ("zCurve");

depthRedChannel = serObj.FindProperty ("depthRedChannel");

depthGreenChannel = serObj.FindProperty ("depthGreenChannel");

depthBlueChannel = serObj.FindProperty ("depthBlueChannel");

serObj.ApplyModifiedProperties ();

selectiveCc = serObj.FindProperty ("selectiveCc");

selectiveFromColor = serObj.FindProperty ("selectiveFromColor");

selectiveToColor = serObj.FindProperty ("selectiveToColor");

}

void CurveGui ( string name, SerializedProperty animationCurve, Color color) {

// @NOTE: EditorGUILayout.CurveField is buggy and flickers, using PropertyField for now

//animationCurve.animationCurveValue = EditorGUILayout.CurveField (GUIContent (name), animationCurve.animationCurveValue, color, Rect (0.0f,0.0f,1.0f,1.0f));

EditorGUILayout.PropertyField (animationCurve, new GUIContent (name));

if (GUI.changed)

applyCurveChanges = true;

}

void BeginCurves () {

applyCurveChanges = false;

}

void ApplyCurves () {

if (applyCurveChanges) {

serObj.ApplyModifiedProperties ();

(serObj.targetObject as ColorCorrectionCurves).gameObject.SendMessage ("UpdateTextures");

}

}

public override void OnInspectorGUI () {

serObj.Update ();

GUILayout.Label ("Use curves to tweak RGB channel colors", EditorStyles.miniBoldLabel);

saturation.floatValue = EditorGUILayout.Slider( "Saturation", saturation.floatValue, 0.0f, 5.0f);

EditorGUILayout.PropertyField (mode, new GUIContent ("Mode"));

EditorGUILayout.Separator ();

BeginCurves ();

CurveGui (" Red", redChannel, Color.red);

CurveGui (" Green", greenChannel, Color.green);

CurveGui (" Blue", blueChannel, Color.blue);

EditorGUILayout.Separator ();

if (mode.intValue > 0)

useDepthCorrection.boolValue = true;

else

useDepthCorrection.boolValue = false;

if (useDepthCorrection.boolValue) {

CurveGui (" Red (depth)", depthRedChannel, Color.red);

CurveGui (" Green (depth)", depthGreenChannel, Color.green);

CurveGui (" Blue (depth)", depthBlueChannel, Color.blue);

EditorGUILayout.Separator ();

CurveGui (" Blend Curve", zCurveChannel, Color.grey);

}

EditorGUILayout.Separator ();

EditorGUILayout.PropertyField (selectiveCc, new GUIContent ("Selective"));

if (selectiveCc.boolValue) {

EditorGUILayout.PropertyField (selectiveFromColor, new GUIContent (" Key"));

EditorGUILayout.PropertyField (selectiveToColor, new GUIContent (" Target"));

}

ApplyCurves ();

if (!applyCurveChanges)

serObj.ApplyModifiedProperties ();

}

}

}