using System;

using UnityEditor;

using UnityEngine;

namespace UnityStandardAssets.ImageEffects

{

[CustomEditor (typeof(DepthOfField))]

class DepthOfFieldEditor : Editor

{

SerializedObject serObj;

SerializedProperty visualizeFocus;

SerializedProperty focalLength;

SerializedProperty focalSize;

SerializedProperty aperture;

SerializedProperty focalTransform;

SerializedProperty maxBlurSize;

SerializedProperty highResolution;

SerializedProperty blurType;

SerializedProperty blurSampleCount;

SerializedProperty nearBlur;

SerializedProperty foregroundOverlap;

SerializedProperty dx11BokehThreshold;

SerializedProperty dx11SpawnHeuristic;

SerializedProperty dx11BokehTexture;

SerializedProperty dx11BokehScale;

SerializedProperty dx11BokehIntensity;

void OnEnable () {

serObj = new SerializedObject (target);

visualizeFocus = serObj.FindProperty ("visualizeFocus");

focalLength = serObj.FindProperty ("focalLength");

focalSize = serObj.FindProperty ("focalSize");

aperture = serObj.FindProperty ("aperture");

focalTransform = serObj.FindProperty ("focalTransform");

maxBlurSize = serObj.FindProperty ("maxBlurSize");

highResolution = serObj.FindProperty ("highResolution");

blurType = serObj.FindProperty ("blurType");

blurSampleCount = serObj.FindProperty ("blurSampleCount");

nearBlur = serObj.FindProperty ("nearBlur");

foregroundOverlap = serObj.FindProperty ("foregroundOverlap");

dx11BokehThreshold = serObj.FindProperty ("dx11BokehThreshold");

dx11SpawnHeuristic = serObj.FindProperty ("dx11SpawnHeuristic");

dx11BokehTexture = serObj.FindProperty ("dx11BokehTexture");

dx11BokehScale = serObj.FindProperty ("dx11BokehScale");

dx11BokehIntensity = serObj.FindProperty ("dx11BokehIntensity");

}

public override void OnInspectorGUI () {

serObj.Update ();

EditorGUILayout.LabelField("Simulates camera lens defocus", EditorStyles.miniLabel);

GUILayout.Label ("Focal Settings");

EditorGUILayout.PropertyField (visualizeFocus, new GUIContent(" Visualize"));

EditorGUILayout.PropertyField (focalLength, new GUIContent(" Focal Distance"));

EditorGUILayout.PropertyField (focalSize, new GUIContent(" Focal Size"));

EditorGUILayout.PropertyField (focalTransform, new GUIContent(" Focus on Transform"));

EditorGUILayout.PropertyField (aperture, new GUIContent(" Aperture"));

EditorGUILayout.Separator ();

EditorGUILayout.PropertyField (blurType, new GUIContent("Defocus Type"));

if (!(target as DepthOfField).Dx11Support() && blurType.enumValueIndex>0) {

EditorGUILayout.HelpBox("DX11 mode not supported (need shader model 5)", MessageType.Info);

}

if (blurType.enumValueIndex<1)

EditorGUILayout.PropertyField (blurSampleCount, new GUIContent(" Sample Count"));

EditorGUILayout.PropertyField (maxBlurSize, new GUIContent(" Max Blur Distance"));

EditorGUILayout.PropertyField (highResolution, new GUIContent(" High Resolution"));

EditorGUILayout.Separator ();

EditorGUILayout.PropertyField (nearBlur, new GUIContent("Near Blur"));

EditorGUILayout.PropertyField (foregroundOverlap, new GUIContent(" Overlap Size"));

EditorGUILayout.Separator ();

if (blurType.enumValueIndex>0) {

GUILayout.Label ("DX11 Bokeh Settings");

EditorGUILayout.PropertyField (dx11BokehTexture, new GUIContent(" Bokeh Texture"));

EditorGUILayout.PropertyField (dx11BokehScale, new GUIContent(" Bokeh Scale"));

EditorGUILayout.PropertyField (dx11BokehIntensity, new GUIContent(" Bokeh Intensity"));

EditorGUILayout.PropertyField (dx11BokehThreshold, new GUIContent(" Min Luminance"));

EditorGUILayout.PropertyField (dx11SpawnHeuristic, new GUIContent(" Spawn Heuristic"));

}

serObj.ApplyModifiedProperties();

}

}

}