using System;

using UnityEditor;

using UnityEngine;

namespace UnityStandardAssets.ImageEffects

{

[CustomEditor (typeof(DepthOfFieldDeprecated))]

class DepthOfFieldDeprecatedEditor : Editor

{

SerializedObject serObj;

SerializedProperty simpleTweakMode;

SerializedProperty focalPoint;

SerializedProperty smoothness;

SerializedProperty focalSize;

SerializedProperty focalZDistance;

SerializedProperty focalStartCurve;

SerializedProperty focalEndCurve;

SerializedProperty visualizeCoc;

SerializedProperty resolution;

SerializedProperty quality;

SerializedProperty objectFocus;

SerializedProperty bokeh;

SerializedProperty bokehScale;

SerializedProperty bokehIntensity;

SerializedProperty bokehThresholdLuminance;

SerializedProperty bokehThresholdContrast;

SerializedProperty bokehDownsample;

SerializedProperty bokehTexture;

SerializedProperty bokehDestination;

SerializedProperty bluriness;

SerializedProperty maxBlurSpread;

SerializedProperty foregroundBlurExtrude;

void OnEnable () {

serObj = new SerializedObject (target);

simpleTweakMode = serObj.FindProperty ("simpleTweakMode");

// simple tweak mode

focalPoint = serObj.FindProperty ("focalPoint");

smoothness = serObj.FindProperty ("smoothness");

// complex tweak mode

focalZDistance = serObj.FindProperty ("focalZDistance");

focalStartCurve = serObj.FindProperty ("focalZStartCurve");

focalEndCurve = serObj.FindProperty ("focalZEndCurve");

focalSize = serObj.FindProperty ("focalSize");

visualizeCoc = serObj.FindProperty ("visualize");

objectFocus = serObj.FindProperty ("objectFocus");

resolution = serObj.FindProperty ("resolution");

quality = serObj.FindProperty ("quality");

bokehThresholdContrast = serObj.FindProperty ("bokehThresholdContrast");

bokehThresholdLuminance = serObj.FindProperty ("bokehThresholdLuminance");

bokeh = serObj.FindProperty ("bokeh");

bokehScale = serObj.FindProperty ("bokehScale");

bokehIntensity = serObj.FindProperty ("bokehIntensity");

bokehDownsample = serObj.FindProperty ("bokehDownsample");

bokehTexture = serObj.FindProperty ("bokehTexture");

bokehDestination = serObj.FindProperty ("bokehDestination");

bluriness = serObj.FindProperty ("bluriness");

maxBlurSpread = serObj.FindProperty ("maxBlurSpread");

foregroundBlurExtrude = serObj.FindProperty ("foregroundBlurExtrude");

}

public override void OnInspectorGUI () {

serObj.Update ();

GameObject go = (target as DepthOfFieldDeprecated).gameObject;

if (!go)

return;

if (!go.GetComponent<Camera>())

return;

if (simpleTweakMode.boolValue)

GUILayout.Label ("Current: "+go.GetComponent<Camera>().name+", near "+go.GetComponent<Camera>().nearClipPlane+", far: "+go.GetComponent<Camera>().farClipPlane+", focal: "+focalPoint.floatValue, EditorStyles.miniBoldLabel);

else

GUILayout.Label ("Current: "+go.GetComponent<Camera>().name+", near "+go.GetComponent<Camera>().nearClipPlane+", far: "+go.GetComponent<Camera>().farClipPlane+", focal: "+focalZDistance.floatValue, EditorStyles.miniBoldLabel);

EditorGUILayout.PropertyField (resolution, new GUIContent("Resolution"));

EditorGUILayout.PropertyField (quality, new GUIContent("Quality"));

EditorGUILayout.PropertyField (simpleTweakMode, new GUIContent("Simple tweak"));

EditorGUILayout.PropertyField (visualizeCoc, new GUIContent("Visualize focus"));

EditorGUILayout.PropertyField (bokeh, new GUIContent("Enable bokeh"));

EditorGUILayout.Separator ();

GUILayout.Label ("Focal Settings", EditorStyles.boldLabel);

if (simpleTweakMode.boolValue) {

focalPoint.floatValue = EditorGUILayout.Slider ("Focal distance", focalPoint.floatValue, go.GetComponent<Camera>().nearClipPlane, go.GetComponent<Camera>().farClipPlane);

EditorGUILayout.PropertyField (objectFocus, new GUIContent("Transform"));

EditorGUILayout.PropertyField (smoothness, new GUIContent("Smoothness"));

focalSize.floatValue = EditorGUILayout.Slider ("Focal size", focalSize.floatValue, 0.0f, (go.GetComponent<Camera>().farClipPlane - go.GetComponent<Camera>().nearClipPlane));

}

else {

focalZDistance.floatValue = EditorGUILayout.Slider ("Distance", focalZDistance.floatValue, go.GetComponent<Camera>().nearClipPlane, go.GetComponent<Camera>().farClipPlane);

EditorGUILayout.PropertyField (objectFocus, new GUIContent("Transform"));

focalSize.floatValue = EditorGUILayout.Slider ("Size", focalSize.floatValue, 0.0f, (go.GetComponent<Camera>().farClipPlane - go.GetComponent<Camera>().nearClipPlane));

focalStartCurve.floatValue = EditorGUILayout.Slider ("Start curve", focalStartCurve.floatValue, 0.05f, 20.0f);

focalEndCurve.floatValue = EditorGUILayout.Slider ("End curve", focalEndCurve.floatValue, 0.05f, 20.0f);

}

EditorGUILayout.Separator ();

GUILayout.Label ("Blur (Fore- and Background)", EditorStyles.boldLabel);

EditorGUILayout.PropertyField (bluriness, new GUIContent("Blurriness"));

EditorGUILayout.PropertyField (maxBlurSpread, new GUIContent("Blur spread"));

if (quality.enumValueIndex > 0) {

EditorGUILayout.PropertyField (foregroundBlurExtrude, new GUIContent("Foreground size"));

}

EditorGUILayout.Separator ();

if (bokeh.boolValue) {

EditorGUILayout.Separator ();

GUILayout.Label ("Bokeh Settings", EditorStyles.boldLabel);

EditorGUILayout.PropertyField (bokehDestination, new GUIContent("Destination"));

bokehIntensity.floatValue = EditorGUILayout.Slider ("Intensity", bokehIntensity.floatValue, 0.0f, 1.0f);

bokehThresholdLuminance.floatValue = EditorGUILayout.Slider ("Min luminance", bokehThresholdLuminance.floatValue, 0.0f, 0.99f);

bokehThresholdContrast.floatValue = EditorGUILayout.Slider ("Min contrast", bokehThresholdContrast.floatValue, 0.0f, 0.25f);

bokehDownsample.intValue = EditorGUILayout.IntSlider ("Downsample", bokehDownsample.intValue, 1, 3);

bokehScale.floatValue = EditorGUILayout.Slider ("Size scale", bokehScale.floatValue, 0.0f, 20.0f);

EditorGUILayout.PropertyField (bokehTexture , new GUIContent("Texture mask"));

}

serObj.ApplyModifiedProperties();

}

}

}