using System;

using UnityEditor;

using UnityEngine;

namespace UnityStandardAssets.ImageEffects

{

[CustomEditor(typeof (Antialiasing))]

public class AntialiasingEditor : Editor

{

private SerializedObject serObj;

private SerializedProperty mode;

private SerializedProperty showGeneratedNormals;

private SerializedProperty offsetScale;

private SerializedProperty blurRadius;

private SerializedProperty dlaaSharp;

private SerializedProperty edgeThresholdMin;

private SerializedProperty edgeThreshold;

private SerializedProperty edgeSharpness;

private void OnEnable()

{

serObj = new SerializedObject(target);

mode = serObj.FindProperty("mode");

showGeneratedNormals = serObj.FindProperty("showGeneratedNormals");

offsetScale = serObj.FindProperty("offsetScale");

blurRadius = serObj.FindProperty("blurRadius");

dlaaSharp = serObj.FindProperty("dlaaSharp");

edgeThresholdMin = serObj.FindProperty("edgeThresholdMin");

edgeThreshold = serObj.FindProperty("edgeThreshold");

edgeSharpness = serObj.FindProperty("edgeSharpness");

}

public override void OnInspectorGUI()

{

serObj.Update();

GUILayout.Label("Luminance based fullscreen antialiasing", EditorStyles.miniBoldLabel);

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

Material mat = (target as Antialiasing).CurrentAAMaterial();

if (null == mat && (target as Antialiasing).enabled)

{

EditorGUILayout.HelpBox("This AA technique is currently not supported. Choose a different technique or disable the effect and use MSAA instead.", MessageType.Warning);

}

if (mode.enumValueIndex == (int) AAMode.NFAA)

{

EditorGUILayout.PropertyField(offsetScale, new GUIContent("Edge Detect Ofs"));

EditorGUILayout.PropertyField(blurRadius, new GUIContent("Blur Radius"));

EditorGUILayout.PropertyField(showGeneratedNormals, new GUIContent("Show Normals"));

}

else if (mode.enumValueIndex == (int) AAMode.DLAA)

{

EditorGUILayout.PropertyField(dlaaSharp, new GUIContent("Sharp"));

}

else if (mode.enumValueIndex == (int) AAMode.FXAA3Console)

{

EditorGUILayout.PropertyField(edgeThresholdMin, new GUIContent("Edge Min Threshhold"));

EditorGUILayout.PropertyField(edgeThreshold, new GUIContent("Edge Threshhold"));

EditorGUILayout.PropertyField(edgeSharpness, new GUIContent("Edge Sharpness"));

}

serObj.ApplyModifiedProperties();

}

}

}