using System;

using UnityEngine;

namespace UnityStandardAssets.ImageEffects

{

[RequireComponent (typeof(Camera))]

[AddComponentMenu ("Image Effects/Camera/Tilt Shift (Lens Blur)")]

class TiltShift : PostEffectsBase {

public enum TiltShiftMode

{

TiltShiftMode,

IrisMode,

}

public enum TiltShiftQuality

{

Preview,

Normal,

High,

}

public TiltShiftMode mode = TiltShiftMode.TiltShiftMode;

public TiltShiftQuality quality = TiltShiftQuality.Normal;

[Range(0.0f, 15.0f)]

public float blurArea = 1.0f;

[Range(0.0f, 25.0f)]

public float maxBlurSize = 5.0f;

[Range(0, 1)]

public int downsample = 0;

public Shader tiltShiftShader = null;

private Material tiltShiftMaterial = null;

public override bool CheckResources () {

CheckSupport (true);

tiltShiftMaterial = CheckShaderAndCreateMaterial (tiltShiftShader, tiltShiftMaterial);

if (!isSupported)

ReportAutoDisable ();

return isSupported;

}

void OnRenderImage (RenderTexture source, RenderTexture destination) {

if (CheckResources() == false) {

Graphics.Blit (source, destination);

return;

}

tiltShiftMaterial.SetFloat("\_BlurSize", maxBlurSize < 0.0f ? 0.0f : maxBlurSize);

tiltShiftMaterial.SetFloat("\_BlurArea", blurArea);

source.filterMode = FilterMode.Bilinear;

RenderTexture rt = destination;

if (downsample > 0f) {

rt = RenderTexture.GetTemporary (source.width>>downsample, source.height>>downsample, 0, source.format);

rt.filterMode = FilterMode.Bilinear;

}

int basePassNr = (int) quality; basePassNr \*= 2;

Graphics.Blit (source, rt, tiltShiftMaterial, mode == TiltShiftMode.TiltShiftMode ? basePassNr : basePassNr + 1);

if (downsample > 0) {

tiltShiftMaterial.SetTexture ("\_Blurred", rt);

Graphics.Blit (source, destination, tiltShiftMaterial, 6);

}

if (rt != destination)

RenderTexture.ReleaseTemporary (rt);

}

}

}