using System;

using UnityEngine;

namespace UnityStandardAssets.ImageEffects

{

[ExecuteInEditMode]

[RequireComponent (typeof(Camera))]

[AddComponentMenu ("Image Effects/Edge Detection/Crease Shading")]

class CreaseShading : PostEffectsBase

{

public float intensity = 0.5f;

public int softness = 1;

public float spread = 1.0f;

public Shader blurShader = null;

private Material blurMaterial = null;

public Shader depthFetchShader = null;

private Material depthFetchMaterial = null;

public Shader creaseApplyShader = null;

private Material creaseApplyMaterial = null;

public override bool CheckResources ()

{

CheckSupport (true);

blurMaterial = CheckShaderAndCreateMaterial (blurShader, blurMaterial);

depthFetchMaterial = CheckShaderAndCreateMaterial (depthFetchShader, depthFetchMaterial);

creaseApplyMaterial = CheckShaderAndCreateMaterial (creaseApplyShader, creaseApplyMaterial);

if (!isSupported)

ReportAutoDisable ();

return isSupported;

}

void OnRenderImage (RenderTexture source, RenderTexture destination)

{

if (CheckResources()==false)

{

Graphics.Blit (source, destination);

return;

}

int rtW = source.width;

int rtH = source.height;

float widthOverHeight = (1.0f \* rtW) / (1.0f \* rtH);

float oneOverBaseSize = 1.0f / 512.0f;

RenderTexture hrTex = RenderTexture.GetTemporary (rtW, rtH, 0);

RenderTexture lrTex1 = RenderTexture.GetTemporary (rtW/2, rtH/2, 0);

Graphics.Blit (source,hrTex, depthFetchMaterial);

Graphics.Blit (hrTex, lrTex1);

for(int i = 0; i < softness; i++)

{

RenderTexture lrTex2 = RenderTexture.GetTemporary (rtW/2, rtH/2, 0);

blurMaterial.SetVector ("offsets", new Vector4 (0.0f, spread \* oneOverBaseSize, 0.0f, 0.0f));

Graphics.Blit (lrTex1, lrTex2, blurMaterial);

RenderTexture.ReleaseTemporary (lrTex1);

lrTex1 = lrTex2;

lrTex2 = RenderTexture.GetTemporary (rtW/2, rtH/2, 0);

blurMaterial.SetVector ("offsets", new Vector4 (spread \* oneOverBaseSize / widthOverHeight, 0.0f, 0.0f, 0.0f));

Graphics.Blit (lrTex1, lrTex2, blurMaterial);

RenderTexture.ReleaseTemporary (lrTex1);

lrTex1 = lrTex2;

}

creaseApplyMaterial.SetTexture ("\_HrDepthTex", hrTex);

creaseApplyMaterial.SetTexture ("\_LrDepthTex", lrTex1);

creaseApplyMaterial.SetFloat ("intensity", intensity);

Graphics.Blit (source,destination, creaseApplyMaterial);

RenderTexture.ReleaseTemporary (hrTex);

RenderTexture.ReleaseTemporary (lrTex1);

}

}

}