using System;

using UnityEngine;

namespace UnityStandardAssets.ImageEffects

{

[ExecuteInEditMode]

[RequireComponent (typeof(Camera))]

[AddComponentMenu ("Image Effects/Displacement/Fisheye")]

class Fisheye : PostEffectsBase

{

public float strengthX = 0.05f;

public float strengthY = 0.05f;

public Shader fishEyeShader = null;

private Material fisheyeMaterial = null;

public override bool CheckResources ()

{

CheckSupport (false);

fisheyeMaterial = CheckShaderAndCreateMaterial(fishEyeShader,fisheyeMaterial);

if (!isSupported)

ReportAutoDisable ();

return isSupported;

}

void OnRenderImage (RenderTexture source, RenderTexture destination)

{

if (CheckResources()==false)

{

Graphics.Blit (source, destination);

return;

}

float oneOverBaseSize = 80.0f / 512.0f; // to keep values more like in the old version of fisheye

float ar = (source.width \* 1.0f) / (source.height \* 1.0f);

fisheyeMaterial.SetVector ("intensity", new Vector4 (strengthX \* ar \* oneOverBaseSize, strengthY \* oneOverBaseSize, strengthX \* ar \* oneOverBaseSize, strengthY \* oneOverBaseSize));

Graphics.Blit (source, destination, fisheyeMaterial);

}

}

}