using System;

using UnityEngine;

namespace UnityStandardAssets.Water

{

public enum WaterQuality

{

High = 2,

Medium = 1,

Low = 0,

}

[ExecuteInEditMode]

public class WaterBase : MonoBehaviour

{

public Material sharedMaterial;

public WaterQuality waterQuality = WaterQuality.High;

public bool edgeBlend = true;

public void UpdateShader()

{

if (waterQuality > WaterQuality.Medium)

{

sharedMaterial.shader.maximumLOD = 501;

}

else if (waterQuality > WaterQuality.Low)

{

sharedMaterial.shader.maximumLOD = 301;

}

else

{

sharedMaterial.shader.maximumLOD = 201;

}

// If the system does not support depth textures (ie. NaCl), turn off edge bleeding,

// as the shader will render everything as transparent if the depth texture is not valid.

if (!SystemInfo.SupportsRenderTextureFormat(RenderTextureFormat.Depth))

{

edgeBlend = false;

}

if (edgeBlend)

{

Shader.EnableKeyword("WATER\_EDGEBLEND\_ON");

Shader.DisableKeyword("WATER\_EDGEBLEND\_OFF");

// just to make sure (some peeps might forget to add a water tile to the patches)

if (Camera.main)

{

Camera.main.depthTextureMode |= DepthTextureMode.Depth;

}

}

else

{

Shader.EnableKeyword("WATER\_EDGEBLEND\_OFF");

Shader.DisableKeyword("WATER\_EDGEBLEND\_ON");

}

}

public void WaterTileBeingRendered(Transform tr, Camera currentCam)

{

if (currentCam && edgeBlend)

{

currentCam.depthTextureMode |= DepthTextureMode.Depth;

}

}

public void Update()

{

if (sharedMaterial)

{

UpdateShader();

}

}

}

}