using System;

using UnityEngine;

using Object = UnityEngine.Object;

namespace UnityStandardAssets.ImageEffects

{

class Triangles

{

private static Mesh[] meshes;

private static int currentTris = 0;

static bool HasMeshes()

{

if (meshes == null)

return false;

for (int i = 0; i < meshes.Length; i++)

if (null == meshes[i])

return false;

return true;

}

static void Cleanup()

{

if (meshes == null)

return;

for (int i = 0; i < meshes.Length; i++)

{

if (null != meshes[i])

{

Object.DestroyImmediate(meshes[i]);

meshes[i] = null;

}

}

meshes = null;

}

static Mesh[] GetMeshes(int totalWidth, int totalHeight)

{

if (HasMeshes() && (currentTris == (totalWidth \* totalHeight)))

{

return meshes;

}

int maxTris = 65000 / 3;

int totalTris = totalWidth \* totalHeight;

currentTris = totalTris;

int meshCount = Mathf.CeilToInt((1.0f \* totalTris) / (1.0f \* maxTris));

meshes = new Mesh[meshCount];

int i = 0;

int index = 0;

for (i = 0; i < totalTris; i += maxTris)

{

int tris = Mathf.FloorToInt(Mathf.Clamp((totalTris - i), 0, maxTris));

meshes[index] = GetMesh(tris, i, totalWidth, totalHeight);

index++;

}

return meshes;

}

static Mesh GetMesh(int triCount, int triOffset, int totalWidth, int totalHeight)

{

var mesh = new Mesh();

mesh.hideFlags = HideFlags.DontSave;

var verts = new Vector3[triCount \* 3];

var uvs = new Vector2[triCount \* 3];

var uvs2 = new Vector2[triCount \* 3];

var tris = new int[triCount \* 3];

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

{

int i3 = i \* 3;

int vertexWithOffset = triOffset + i;

float x = Mathf.Floor(vertexWithOffset % totalWidth) / totalWidth;

float y = Mathf.Floor(vertexWithOffset / totalWidth) / totalHeight;

Vector3 position = new Vector3(x \* 2 - 1, y \* 2 - 1, 1.0f);

verts[i3 + 0] = position;

verts[i3 + 1] = position;

verts[i3 + 2] = position;

uvs[i3 + 0] = new Vector2(0.0f, 0.0f);

uvs[i3 + 1] = new Vector2(1.0f, 0.0f);

uvs[i3 + 2] = new Vector2(0.0f, 1.0f);

uvs2[i3 + 0] = new Vector2(x, y);

uvs2[i3 + 1] = new Vector2(x, y);

uvs2[i3 + 2] = new Vector2(x, y);

tris[i3 + 0] = i3 + 0;

tris[i3 + 1] = i3 + 1;

tris[i3 + 2] = i3 + 2;

}

mesh.vertices = verts;

mesh.triangles = tris;

mesh.uv = uvs;

mesh.uv2 = uvs2;

return mesh;

}

}

}