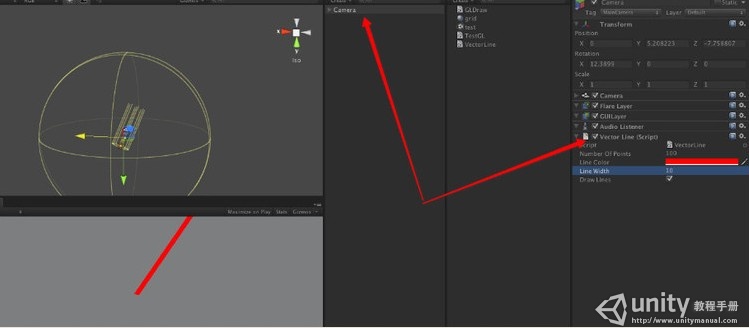
**Unity3D GL方式画线**

Posted on 2013年03月29日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 447 次

[**Unity3D**](http://www.unitymanual.com) GL方式画线如下图：

相关文章：[**Unity3D LineRenderer方式画线**](http://www.unitymanual.com/3400.html)

[](http://www.unitymanual.com/wp-content/uploads/2013/03/51.jpg)

Unity3D GL方式画线

把如下脚本放在摄像机上：

using UnityEngine;

using System.Collections;

[RequireComponent(typeof (Camera))]

public class VectorLine : MonoBehaviour

{

public int numberOfPoints = 2;

public Color lineColor = Color.red;

public int lineWidth = 3;

public bool drawLines = true;

private Material lineMaterial;

private Vector2[] linePoints;

private Camera cam;

void Awake()

{

lineMaterial = new Material( "Shader \"Lines/Colored Blended\" {" +

"SubShader { Pass {" +

" BindChannels { Bind \"Color\",color }" +

" Blend SrcAlpha OneMinusSrcAlpha" +

" ZWrite Off Cull Off Fog { Mode Off }" +

"} } }");

lineMaterial.hideFlags = HideFlags.HideAndDontSave;

lineMaterial.shader.hideFlags = HideFlags.HideAndDontSave;

cam = camera;

}

// Creates a simple two point line

void Start()

{

linePoints = new Vector2[2];

}

// Sets line endpoints to center of screen and mouse position

void Update()

{

linePoints[0] = new Vector2(0.5f, 0.5f);

linePoints[1] = new Vector2(Input.mousePosition.x/Screen.width, Input.mousePosition.y/Screen.height);

}

void OnPostRender()

{

if (!drawLines || linePoints == null || linePoints.Length < 2)

return;

float nearClip = cam.nearClipPlane + 0.00001f;

int end = linePoints.Length - 1;

float thisWidth = 1f/Screen.width \* lineWidth \* 0.5f;

lineMaterial.SetPass(0);

GL.Color(lineColor);

if (lineWidth == 1)

{

GL.Begin(GL.LINES);

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

{

GL.Vertex(cam.ViewportToWorldPoint(new Vector3(linePoints[i].x, linePoints[i].y, nearClip)));

GL.Vertex(cam.ViewportToWorldPoint(new Vector3(linePoints[i+1].x, linePoints[i+1].y, nearClip)));

}

}

else

{

GL.Begin(GL.QUADS);

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

{

Vector3 perpendicular = (new Vector3(linePoints[i+1].y, linePoints[i].x, nearClip) -

new Vector3(linePoints[i].y, linePoints[i+1].x, nearClip)).normalized \* thisWidth;

Vector3 v1 = new Vector3(linePoints[i].x, linePoints[i].y, nearClip);

Vector3 v2 = new Vector3(linePoints[i+1].x, linePoints[i+1].y, nearClip);

GL.Vertex(cam.ViewportToWorldPoint(v1 - perpendicular));

GL.Vertex(cam.ViewportToWorldPoint(v1 + perpendicular));

GL.Vertex(cam.ViewportToWorldPoint(v2 + perpendicular));

GL.Vertex(cam.ViewportToWorldPoint(v2 - perpendicular));

}

}

GL.End();

}

void OnApplicationQuit()

{

DestroyImmediate(lineMaterial);

}

}