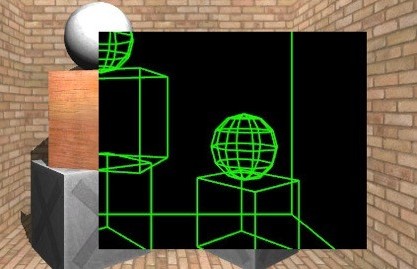
**Unity3D中用Vectrosity插件画直线、画点、画曲线、画方框**

Posted on 2013年03月29日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b)/被围观 453 次

[**Unity3D**](http://www.unitymanual.com)中用Vectrosity插件画直线、画点、画曲线、画方框 。Vectrosity插件是[**Unity3D**](http://www.unitymanual.com/category/manual)目前发现的一个画线最好的工具插件。

效果图如下：

[](http://www.unitymanual.com/wp-content/uploads/2013/03/QQ截图20130329110846.jpg)

Unity3D中用Vectrosity插件画直线、画点、画曲线、画方框

**画直线：**

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | *// Make Vector2 array; in this case we just use 2 elements...* |
| 02 |  |
| 03 | **var** linePoints = [Vector2(0, Random.Range(0, Screen.height)), *// ...one on the left side of the screen somewhere* |
| 04 |  |
| 05 | Vector2(Screen.width-1, Random.Range(0, Screen.height))]; *// ...and one on the right* |
| 06 |  |
| 07 | *// Make a VectorLine object using the above points and the default material, with a width of 2 pixels* |
| 08 |  |
| 09 | **var** line = new VectorLine("Line", linePoints, **null**, 2.0); |
| 10 |  |
| 11 | *// Draw the line* |
| 12 |  |
| 13 | Vector.DrawLine(line); |

画线我们肯定需要LinePoints，注意：Vector2是Screen以像素为单位的点，如果用Vector3的话那么就是world Space里面的点画线。

画直线用Vector.DrawLine，如果给了材质和贴图，可以Vector.SetTextureScale(line, textureScale)设置一下图片的scale，

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 1 | *// Draw a line from the lower-left corner to the upper-right corner* |
| 2 |  |
| 3 | Vector.SetLine (Color.white, Vector2(0, 0), Vector2(Screen.width-1, Screen.height-1)); |

以上可以简单画一条线,有点像Debug.DrawLine()那样就画一条细细的线。同理Debug.DrawRay()也在这里有Vector.SetRay()也可以，但是SetLine可以用Vector2和Vector3的点，但是SetRay只能是Vector3世界中的点。

**ADD+ Real 3D Lines**

为什么会有这个真正的3D线？当我们用SetRay在世界中画线的时候，这个线在3D物体前面。

Vector.SetRay3D()

用以上画即可。

**ADD+Update & Timing**

如果在Update里面不断调用SetRay()和SetLine之类的生成线方法，那么要注意了。这些线会一直存在！

VectorLine SetLine (Color color, float time, params Vector2[] points)

以上方法第二个参数可以传一个time进去，多少s会消失掉。如果要创建1次，然后在update里面不断更新这个线，怎么做？

SetLine和Set……是创建一个VectorLine的Object对象，我们在update（最好在LateUpate）里面动态改变这个返回值里面的参数，然后调用一下drawLine即可！

PS:在LateUpate里面每一帧更新会比在Update里面好。

**ADD+active**

相比destroy这个对象，我们喜欢关闭和显示可以直接用。

myLine.active = false;

myLine.active = true;

那么关闭时候，使用drawLine将没有作用。

**ADD+MakeLine**

Vector.MakeLine这类方法相当于复制一个快捷方式，我们可以重复制作一样属性的Line，

在执行MakeLine前面需要执行Vector.SetLineParameters（）设置一下和初始属性不一样的线，相当于制作一个快捷方式，以后用MakeLine将都一样。

**画点**：

和VectorLines一样，可以new VectorPoints（）创建一个点集对象，和VL一样，可以设置颜色，属性选项等。

然后用Vector.DrawPoints(myPoints);就可以画出来了，基本上和vl的概念是一样的。

**画曲线，画方框**：

**ADD+Utilities**

在Vector类里面，提供了很多额外的实用工具方法给我们，一般有Set和Get以及Make开头。

例如SetColor和SetColors……等。

**ADD+VectorManager**

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | *// Make a Vector3 array that contains points for a cube that's 1 unit in size* |
| 02 |  |
| 03 | **var** cubePoints = [Vector3(-0.5, -0.5, 0.5), Vector3(0.5, -0.5, 0.5), Vector3(-0.5, 0.5, 0.5), Vector3(-0.5, -0.5, 0.5), Vector3(0.5, -0.5, 0.5), Vector3(0.5, 0.5, 0.5), Vector3(0.5, 0.5, 0.5), Vector3(-0.5, 0.5, 0.5), Vector3(-0.5, 0.5, -0.5), Vector3(-0.5, 0.5, 0.5), Vector3(0.5, 0.5, 0.5), Vector3(0.5, 0.5, -0.5), Vector3(0.5, 0.5, -0.5), Vector3(-0.5, 0.5, -0.5), Vector3(-0.5, -0.5, -0.5), Vector3(-0.5, 0.5, -0.5), Vector3(0.5, 0.5, -0.5), Vector3(0.5, -0.5, -0.5), Vector3(0.5, -0.5, -0.5), Vector3(-0.5, -0.5, -0.5), Vector3(-0.5, -0.5, 0.5), Vector3(-0.5, -0.5, -0.5), Vector3(0.5, -0.5, -0.5), Vector3(0.5, -0.5, 0.5)]; |
| 04 |  |
| 05 | *// Make a line using the above points and material, with a width of 2 pixels* |
| 06 |  |
| 07 | **var** line = new VectorLine("Cube", cubePoints, Color.white, lineMaterial, 2.0); |
| 08 |  |
| 09 | *// Make this transform have the vector line object that's defined above* |
| 10 |  |
| 11 | *// This object is a rigidbody, so the vector object will do exactly what this object does* |
| 12 |  |
| 13 | VectorManager.ObjectSetup (gameObject, line, Visibility.**Dynamic**, Brightness.None); |

以上是实现构造一个3DObject的形状，需要点集，需要根据这些点构造VL线段，然后再通过VectorManager.ObjectSetup（）创建即可。