<http://blog.csdn.net/honey199396/article/details/51423630>

<https://www.cnblogs.com/diamond/p/4168083.html>

<https://www.baidu.com/link?url=M2xiF0ozsSwrujp_S3QoPYjZp76YbxRdPU7T050ZfnritIOu-GYufiqUOh1P1O8xU6Y3o-6BqYJB5zC51aVF1Ir0LTb_RW1r7-rteHs4On_HoWT9jqwjtM_rpbK3-wxj&wd=&eqid=f657be880005caab000000035b4336a6>

使用Unity3D ScreenToWorldPoint(Vector3 position)时，如果将z坐标设置为零的话，那么转换的结果可能是出错，会变成摄像机的位置，所以想让结果正确要将position的z坐标设置为非零的值，比如100，或者是-100。

Screenspace is defined in pixels. The bottom-left of the screen is (0,0); the right-top is ([pixelWidth](https://docs.unity3d.com/ScriptReference/Camera-pixelWidth.html),[pixelHeight](https://docs.unity3d.com/ScriptReference/Camera-pixelHeight.html)). The z position is in world units from the camera.

camera = Camera.main;

Vector3 v1 = camera.ScreenToWorldPoint(new Vector3(20.0f, 30.0f, 10.0f));

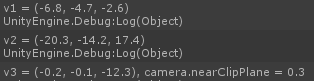
Vector3 v2 = camera.ScreenToWorldPoint(new Vector3(20.0f, 30.0f, 30.0f));

Vector3 v3 = camera.ScreenToWorldPoint(new Vector3(20.0f, 30.0f, camera.nearClipPlane));

Debug.Log("v1 = " + v1);

Debug.Log("v2 = " + v2);

Debug.Log("v3 = " + v3 + ", camera.nearClipPlane = " + camera.nearClipPlane);



其中得到的Ray中的

Ray.origin是摄像机与近平面的交点

Ray.direction是方向

实际上我们可以将屏幕上的点通过ScreenToWorldPoint（注意不要让z值等于0）转换为世界坐标系上的点

然后通过摄像机的起点与转换后的世界坐标系上的点得到方向，最终得到的方向与通过

Camera.main.ScreenPointToRay(Input.mousePoint)的结果一样

// Draw a yellow sphere in the scene view at the position

// on the near plane of the selected camera that is

// 100 pixels from lower-left.

//在所选相机的近裁剪面上，绘制一个黄色的球，在左下100像素的位置

function OnDrawGizmosSelected () {

var p : [Vector3](http://www.ceeger.com/Script/Vector3/Vector3.html) = camera.ScreenToWorldPoint ([Vector3](http://www.ceeger.com/Script/Vector3/Vector3.html) (100,100,camera.nearClipPlane));

[Gizmos.color](http://www.ceeger.com/Script/Gizmos/Gizmos.color.html) = [Color.yellow](http://www.ceeger.com/Script/Color/Color.yellow.html);

[Gizmos.DrawSphere](http://www.ceeger.com/Script/Gizmos/Gizmos.DrawSphere.html) (p, 0.1);

}