**拖拽与坐标系转换**

Posted on 2013年06月17日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 45 次

Unity3D 中使用鼠标，或触屏设备手指拖拽物体移动，先介绍第一种方法：

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

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 |  |
| 03 | **using** System.Collections; |
| 04 |  |
| 05 | **public** **class** Drag : MonoBehaviour |
| 06 |  |
| 07 | { |
| 08 |  |
| 09 | *// Use this for initialization* |
| 10 |  |
| 11 | **void** Start() { |
| 12 |  |
| 13 | } |
| 14 |  |
| 15 | *// Update is called once per frame* |
| 16 |  |
| 17 | **void** Update() { |
| 18 |  |
| 19 | } |
| 20 |  |
| 21 | IEnumerator OnMouseDown() |
| 22 |  |
| 23 | { |
| 24 |  |
| 25 | *//将物体由世界坐标系转化为屏幕坐标系 ，由vector3 结构体变量ScreenSpace存储，以用来明确屏幕坐标系Z轴的位置* |
| 26 |  |
| 27 | Vector3 ScreenSpace = Camera.main.WorldToScreenPoint(transform.position); |
| 28 |  |
| 29 | *//完成了两个步骤，1.由于鼠标的坐标系是2维的，需要转化成3维的世界坐标系，2.只有三维的情况下才能来计算鼠标位置与物体的距离，offset即是距离。* |
| 30 |  |
| 31 | Vector3 offset = transform.position - Camera.main.ScreenToWorldPoint(new Vector3(Input.mousePosition.x, Input.mousePosition.y, ScreenSpace.z)); |
| 32 |  |
| 33 | Debug.Log(“down”); |
| 34 |  |
| 35 | *//当鼠标左键按下时* |
| 36 |  |
| 37 | **while** (Input.GetMouseButton(0)) |
| 38 |  |
| 39 | {&nbsp; <code lang="csharp">*//Unity3D教程手册* |

//得到现在鼠标的2维坐标系位置

Vector3 curScreenSpace = new Vector3(Input.mousePosition.x, Input.mousePosition.y, ScreenSpace.z);

//将当前鼠标的2维位置转化成三维的位置，再加上鼠标的移动量

Vector3 CurPosition = Camera.main.ScreenToWorldPoint(curScreenSpace) + offset;

//CurPosition就是物体应该的移动向量赋给transform的position属性

transform.position = CurPosition;

yield return new WaitForFixedUpdate();

}

}

}

使用这种方法成功运行，但是查看官方开发文档，如下：

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

|  |  |
| --- | --- |
| 1 | IMPORTANT: **This** function has no effect on iPhone. |

[](http://www.unitymanual.com/wp-content/uploads/2013/06/1113.jpg)

拖拽与坐标系转换

这个方法并不能用在iPhone中，下面介绍第二种方法，需要注意的是鼠标坐标需经过转换后才能使用。

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

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 |  |
| 03 | **using** System.Collections; |
| 04 |  |
| 05 | */// <summary>* |
| 06 |  |
| 07 | */// Common property of player* |
| 08 |  |
| 09 | */// </summary>* |
| 10 |  |
| 11 | **public** **class** PlayerBase : MonoBehaviour |
| 12 |  |
| 13 | { |
| 14 |  |
| 15 | *// Use this for initialization* |
| 16 |  |
| 17 | **void** Start() { |
| 18 |  |
| 19 | }*//Unity3D教程手册* |
| 20 |  |
| 21 | **bool** isBeDraged = **false**; |
| 22 |  |
| 23 | RaycastHit hit; |
| 24 |  |
| 25 | *// Update is called once per frame* |
| 26 |  |
| 27 | **void** Update() |
| 28 |  |
| 29 | { |
| 30 |  |
| 31 | **if** (Input.GetMouseButton(0)) |
| 32 |  |
| 33 | { |
| 34 |  |
| 35 | Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition); |
| 36 |  |
| 37 | **if** ((Physics.Raycast(ray, **out** hit)) && (**null** != hit.collider)) |
| 38 |  |
| 39 | isBeDraged = **true**; |
| 40 |  |
| 41 | **if** (isBeDraged) |
| 42 |  |
| 43 | { |
| 44 |  |
| 45 | *//Vector3 currentPostiont = Input.mousePosition;需要坐标系转换才能使用鼠标坐标，具体见本篇开头坐标系转换链接* |
| 46 |  |
| 47 | Vector3 currentPostiont = new Vector3(ray.origin.x, ray.origin.y, transform.position.z); |
| 48 |  |
| 49 | transform.position = currentPostiont; |
| 50 |  |
| 51 | } |
| 52 |  |
| 53 | } |
| 54 |  |
| 55 | **else** |
| 56 |  |
| 57 | isBeDraged = **false**; |
| 58 |  |
| 59 | } |
| 60 |  |
| 61 | } |