**Unity3D脚本：RPG的镜头跟随脚本**

Posted on 2013年02月16日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 402 次

[**Unity**](http://www.unitymanual.com)中copy另存为MouseFollow\_ForRPG\_CSharp.cs，拖给你的镜头即可。想跟随哪一个就把目标物体拖到Inspector里target那一栏。

自动跟随角色正后方，按住鼠标右键移动鼠标，围着角色上下左右旋转镜头，滚轮缩放，放开鼠标镜头自动平滑的转回后方跟随的角度，镜头高度不变，经典的魔兽世界控制方式。另外多一个魔兽没有的功能，左键按住移动鼠标可以冻结摄像机镜头上下左右拖动，因为只是演示控制，一般左键都是战斗相关的功能，不想要把那几行删掉就行。

因为要跟随，所以不管有没有输入每次update都要参考目标位置修正镜头位置的，否则就不叫跟随了。

using UnityEngine;  
using System.Collections;  
public class MouseFollow\_ForRPG\_CSharp : MonoBehaviour {  
public GameObject target;  
public float ZoomSpeed = 30;//镜头缩放速率  
public float MovingSpeed = 1;//镜头移动速率  
public float RotateSpeed = 1; //镜头旋转速率  
public float distance = 20;//设置距离角色的距离  
public float ViewAngle = 30;//设置镜头斜视的角度  
void Start () {  
if(target){  
transform.rotation = Quaternion.Euler(ViewAngle, target.transform.rotation.eulerAngles.y,0 );  
transform.position = transform.rotation \* new Vector3(0,0,-distance)+target.transform.position;  
}  
}  
void Update () {  
Quaternion rotation;  
Vector3 position;  
float delta\_x,delta\_y;  
float delta\_rotation\_x,delta\_rotation\_y;  
if(target){  
if(Input.GetMouseButton(0))  
{  
delta\_x = Input.GetAxis("Mouse X") \* MovingSpeed;  
delta\_y = Input.GetAxis("Mouse Y") \* MovingSpeed;  
rotation = Quaternion.Euler(0, transform.rotation.eulerAngles.y,0 );  
transform.position = rotation \* new Vector3(-delta\_x,0,-delta\_y)+ transform.position;  
}  
else{  
if(Input.GetAxis("Mouse ScrollWheel")!= 0){  
distance += -Input.GetAxis("Mouse ScrollWheel") \* ZoomSpeed;  
}  
if (Input.GetMouseButton(1)) {  
delta\_rotation\_x = Input.GetAxis("Mouse X") \* RotateSpeed;  
delta\_rotation\_y = -Input.GetAxis("Mouse Y") \* RotateSpeed;  
transform.Rotate(0,delta\_rotation\_x,0,Space.World);  
transform.Rotate(delta\_rotation\_y,0,0);  
}  
else {  
transform.rotation = Quaternion.Slerp(transform.rotation,  
Quaternion.Euler(transform.rotation.eulerAngles.x, target.transform.rotation.eulerAngles.y,0  
),Time.deltaTime\*2);  
}  
transform.position = transform.rotation\* new Vector3(0,0,-distance)+ target.transform.position;  
}  
}  
}  
}