**物品展示必备代码（旋转与缩放代码）**

Posted on 2013年07月04日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script) /被围观 85 次

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

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 |  |
| 03 | **public** **class** MouseFollowRotation : MonoBehaviour { |
| 04 |  |
| 05 | **public** Transform target; |
| 06 | **public** **float** xSpeed=200, ySpeed=200, mSpeed=10; |
| 07 | **public** **float** yMinLimit=-50, yMaxLimit=50; |
| 08 | **public** **float** distance=7, minDistance=2, maxDistance=30; |
| 09 |  |
| 10 | *//bool needDamping = false;* |
| 11 | **public** **bool** needDamping =**true**; |
| 12 | **float** damping = 5.0f; |
| 13 |  |
| 14 | **public** **float** x = 0.0f; |
| 15 | **public** **float** y = 0.0f; |
| 16 |  |
| 17 |  |
| 18 | **public** **void** SetTarget( GameObject go ) |
| 19 |  |
| 20 |  |
| 21 | { |
| 22 | target = go.transform; |
| 23 | } |
| 24 | *// Use this for initialization* |
| 25 | **void** Start () { |
| 26 | Vector3 angles = transform.eulerAngles; |
| 27 | x = angles.y; |
| 28 | y = angles.x; |
| 29 | } |
| 30 |  |
| 31 | *// Update is called once per frame* |
| 32 | **void** LateUpdate () |
| 33 | { |
| 34 |  |
| 35 |  |
| 36 | **if** (target) |
| 37 | { |
| 38 | *//use the light button of mouse to rotate the camera* |
| 39 | **if**( Input.GetMouseButton(0) ) |
| 40 | { |
| 41 | x += Input.GetAxis("Mouse X") \* xSpeed \* 0.02f; |
| 42 | y -= Input.GetAxis("Mouse Y") \* ySpeed \* 0.02f; |
| 43 |  |
| 44 | y = ClampAngle(y, yMinLimit, yMaxLimit); |
| 45 |  |
| 46 | *//print(Input.GetAxis("Mouse X"));* |
| 47 | *//print( Input.GetAxis("Mouse Y"));* |
| 48 | *//print(x);* |
| 49 | *//print(y);* |
| 50 |  |
| 51 | *//Unity3D教程手册：www.unitymanual.com* |
| 52 |  |
| 53 | } |
| 54 |  |
| 55 |  |
| 56 | distance -= Input.GetAxis("Mouse ScrollWheel")\*mSpeed; |
| 57 | distance = Mathf.Clamp(distance, minDistance, maxDistance); |
| 58 |  |
| 59 |  |
| 60 | Quaternion rotation = Quaternion.Euler(y, x, 0.0f); |
| 61 | Vector3 disVector = new Vector3( 0.0f, 0.0f, -distance ); |
| 62 | Vector3 position = rotation \* disVector + target.position; |
| 63 | *//adjust the camera* |
| 64 | **if**( needDamping ) |
| 65 | { |
| 66 | transform.rotation = Quaternion.Lerp(transform.rotation, rotation, Time.deltaTime\*damping); |
| 67 | transform.position = Vector3.Lerp(transform.position, position, Time.deltaTime\*damping); |
| 68 | } |
| 69 | **else** |
| 70 | { |
| 71 | transform.rotation = rotation; |
| 72 | transform.position = position; |
| 73 | } |
| 74 | *//Unity3D教程手册：www.unitymanual.com* |
| 75 |  |
| 76 | } |
| 77 | } |
| 78 |  |
| 79 | **static** **float** ClampAngle (**float** angle, **float** min, **float** max) |
| 80 | { |
| 81 | **if** (angle < -360) |
| 82 | angle += 360; |
| 83 | **if** (angle > 360) |
| 84 | angle -= 360; |
| 85 | **return** Mathf.Clamp (angle, min, max); |
| 86 | } |
| 87 | } |
| 88 |  |
| 89 |  |