**漫游相机脚本**

Posted on 2013年07月15日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script) /被围观 28 次

1.修改unity3d的Mouse Look相机脚本：按住右键控制前进方向，方向键改变位置

2.给场景相机添加系统自带的fpswalker脚本

3.指定fpswalker里transform指定场景相机

PS：Mouse Look.cs修改部分：

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

|  |  |
| --- | --- |
| 001 | **void** Update () |
| 002 | { |
| 003 | **if** (Input.GetMouseButton(1) && axes == RotationAxes.MouseXAndY) |
| 004 | { |
| 005 | *// Read the mouse input axis* |
| 006 | rotationX += Input.GetAxis("Mouse X") \* sensitivityX; |
| 007 | rotationY += Input.GetAxis("Mouse Y") \* sensitivityY; |
| 008 |  |
| 009 | rotationX = ClampAngle(rotationX, minimumX, maximumX); |
| 010 | rotationY = ClampAngle(rotationY, minimumY, maximumY); |
| 011 |  |
| 012 | Quaternion xQuaternion = Quaternion.AngleAxis(rotationX, Vector3.up); |
| 013 | Quaternion yQuaternion = Quaternion.AngleAxis(rotationY, Vector3.left); |
| 014 |  |
| 015 | transform.localRotation = originalRotation \* xQuaternion \* yQuaternion; |
| 016 | } |
| 017 | **else** **if** (Input.GetMouseButton(1) && axes == RotationAxes.MouseX) |
| 018 | { |
| 019 | rotationX += Input.GetAxis("Mouse X") \* sensitivityX; |
| 020 | rotationX = ClampAngle(rotationX, minimumX, maximumX); |
| 021 |  |
| 022 | Quaternion xQuaternion = Quaternion.AngleAxis(rotationX, Vector3.up); |
| 023 | transform.localRotation = originalRotation \* xQuaternion; |
| 024 | } |
| 025 | } |
| 026 |  |
| 027 | 当然也可以，直接用以下脚本： |
| 028 | **using** UnityEngine; |
| 029 | **using** System.Collections; |
| 030 | **public** **class** MouseLook : MonoBehaviour { |
| 031 |  |
| 032 | **public** **enum** RotationAxes { MouseXAndY = 0, MouseX = 1, MouseY = 2 } |
| 033 | **public** RotationAxes axes = RotationAxes.MouseXAndY; |
| 034 |  |
| 035 | **public** **float** sensitivityX = 15F; |
| 036 | **public** **float** sensitivityY = 15F; |
| 037 |  |
| 038 | **public** **float** minimumX = -360F; |
| 039 | **public** **float** maximumX = 360F; |
| 040 |  |
| 041 | **public** **float** minimumY = -60F; |
| 042 | **public** **float** maximumY = 60F; |
| 043 |  |
| 044 | **float** moveSensitivity = 1.0f; |
| 045 | **public** **float** speed = 6.0f; |
| 046 |  |
| 047 | **float** rotationX = 0F; |
| 048 | **float** rotationY = 0F; |
| 049 |  |
| 050 | Quaternion originalRotation; |
| 051 |  |
| 052 | **void** Update () |
| 053 | {*//Unity3D教程手册：www.unitymanual.com* |
| 054 |  |
| 055 | CharacterController controller = gameObject.GetComponent<CharacterController>(); |
| 056 | **if** (Input.GetMouseButton(1) && axes == RotationAxes.MouseXAndY) |
| 057 | { |
| 058 | *// Read the mouse input axis* |
| 059 | rotationX += Input.GetAxis("Mouse X") \* sensitivityX; |
| 060 | rotationY += Input.GetAxis("Mouse Y") \* sensitivityY; |
| 061 |  |
| 062 | rotationX = ClampAngle(rotationX, minimumX, maximumX); |
| 063 | rotationY = ClampAngle(rotationY, minimumY, maximumY); |
| 064 |  |
| 065 | Quaternion xQuaternion = Quaternion.AngleAxis(rotationX, Vector3.up); |
| 066 | Quaternion yQuaternion = Quaternion.AngleAxis(rotationY, Vector3.left); |
| 067 |  |
| 068 | transform.localRotation = originalRotation \* xQuaternion \* yQuaternion; |
| 069 | } |
| 070 | **else** **if** (Input.GetMouseButton(1) && axes == RotationAxes.MouseX) |
| 071 | { |
| 072 | rotationX += Input.GetAxis("Mouse X") \* sensitivityX; |
| 073 | rotationX = ClampAngle(rotationX, minimumX, maximumX); |
| 074 |  |
| 075 | Quaternion xQuaternion = Quaternion.AngleAxis(rotationX, Vector3.up); |
| 076 | transform.localRotation = originalRotation \* xQuaternion; |
| 077 | } |
| 078 | *//Unity3D教程手册：www.unitymanual.com* |
| 079 | Vector3 moveDirection = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical")); |
| 080 | moveDirection = transform.TransformDirection(moveDirection); |
| 081 | moveDirection \*= speed; |
| 082 | controller.Move(moveDirection \* Time.deltaTime); |
| 083 | } |
| 084 | **void** Start () |
| 085 | { |
| 086 | *// Make the rigid body not change rotation* |
| 087 | **if** (rigidbody) |
| 088 | rigidbody.freezeRotation = **true**; |
| 089 | originalRotation = transform.localRotation; |
| 090 | } |
| 091 | **public** **static** **float** ClampAngle (**float** angle, **float** min, **float** max) |
| 092 | { |
| 093 | **if** (angle < -360F) |
| 094 | angle += 360F; |
| 095 | **if** (angle > 360F) |
| 096 | angle -= 360F; |
| 097 | **return** Mathf.Clamp (angle, min, max); |
| 098 | } |
| 099 | } |
| 100 |  |