**触摸屏手势控制镜头旋转与缩放**

Posted on 2013年07月12日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script) /被围观 4 次

主要的实现代码：

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

|  |  |
| --- | --- |
| 001 | *//用于绑定参照物对象* |
| 002 | **var** target : Transform; |
| 003 | *//缩放系数* |
| 004 | **var** distance = 10.0; |
| 005 | *//左右滑动移动速度* |
| 006 | **var** xSpeed = 250.0; |
| 007 | **var** ySpeed = 120.0; |
| 008 | *//缩放限制系数* |
| 009 | **var** yMinLimit = -20; |
| 010 | **var** yMaxLimit = 80; |
| 011 | *//摄像头的位置* |
| 012 | **var** x = 0.0; |
| 013 | **var** y = 0.0; |
| 014 | *//记录上一次手机触摸位置判断用户是在左放大还是缩小手势* |
| 015 | **private** **var** oldPosition1 : Vector2; |
| 016 | **private** **var** oldPosition2 : Vector2; |
| 017 |  |
| 018 | *//初始化游戏信息设置* |
| 019 | function Start () { |
| 020 | **var** angles = transform.eulerAngles; |
| 021 | x = angles.y; |
| 022 | y = angles.x; |
| 023 |  |
| 024 | *// Make the rigid body not change rotation* |
| 025 | **if** (rigidbody) |
| 026 | rigidbody.freezeRotation = **true**; |
| 027 | } |
| 028 |  |
| 029 | function Update () |
| 030 | { |
| 031 | *//判断触摸数量为单点触摸* |
| 032 | **if**(Input.touchCount == 1) |
| 033 | { |
| 034 | *//触摸类型为移动触摸* |
| 035 | **if**(Input.GetTouch(0).phase==TouchPhase.Moved) |
| 036 | { |
| 037 | *//根据触摸点计算X与Y位置* |
| 038 | x += Input.GetAxis("Mouse X") \* xSpeed \* 0.02; |
| 039 | y -= Input.GetAxis("Mouse Y") \* ySpeed \* 0.02; |
| 040 |  |
| 041 | } |
| 042 | } |
| 043 |  |
| 044 | *//判断触摸数量为多点触摸* |
| 045 | **if**(Input.touchCount >1 ) |
| 046 | { |
| 047 | *//前两只手指触摸类型都为移动触摸* |
| 048 | **if**(Input.GetTouch(0).phase==TouchPhase.Moved??Input.GetTouch(1).phase==TouchPhase.Moved) |
| 049 | { *//Unity3D教程手册：www.unitymanual.com* |
| 050 | *//计算出当前两点触摸点的位置* |
| 051 | **var** tempPosition1 = Input.GetTouch(0).position; |
| 052 | **var** tempPosition2 = Input.GetTouch(1).position; |
| 053 | *//函数返回真为放大，返回假为缩小* |
| 054 | **if**(isEnlarge(oldPosition1,oldPosition2,tempPosition1,tempPosition2)) |
| 055 | { |
| 056 | *//放大系数超过3以后不允许继续放大* |
| 057 | *//这里的数据是根据我项目中的模型而调节的，大家可以自己任意修改* |
| 058 | **if**(distance > 3) |
| 059 | { |
| 060 | distance -= 0.5; |
| 061 | } |
| 062 | }**else** |
| 063 | { |
| 064 | *//缩小洗漱返回18.5后不允许继续缩小* |
| 065 | *//这里的数据是根据我项目中的模型而调节的，大家可以自己任意修改* |
| 066 | **if**(distance < 18.5) |
| 067 | { |
| 068 | distance += 0.5; |
| 069 | } |
| 070 | } |
| 071 | *//备份上一次触摸点的位置，用于对比* |
| 072 | oldPosition1=tempPosition1; |
| 073 | oldPosition2=tempPosition2; |
| 074 | } |
| 075 | } |
| 076 | } |
| 077 |  |
| 078 | *//函数返回真为放大，返回假为缩小* |
| 079 | function isEnlarge(oP1 : Vector2,oP2 : Vector2,nP1 : Vector2,nP2 : Vector2) : boolean |
| 080 | { |
| 081 | *//函数传入上一次触摸两点的位置与本次触摸两点的位置计算出用户的手势* |
| 082 | **var** leng1 =Mathf.Sqrt((oP1.x-oP2.x)\*(oP1.x-oP2.x)+(oP1.y-oP2.y)\*(oP1.y-oP2.y)); |
| 083 | **var** leng2 =Mathf.Sqrt((nP1.x-nP2.x)\*(nP1.x-nP2.x)+(nP1.y-nP2.y)\*(nP1.y-nP2.y)); |
| 084 | **if**(leng1<leng2) |
| 085 | { |
| 086 | *//放大手势* |
| 087 | **return** **true**; |
| 088 | }**else** |
| 089 | { |
| 090 | *//缩小手势* |
| 091 | **return** **false**; |
| 092 | } |
| 093 | } |
| 094 | *//Unity3D教程手册：www.unitymanual.com* |
| 095 | *//Update方法一旦调用结束以后进入这里算出重置摄像机的位置* |
| 096 | function LateUpdate () { |
| 097 |  |
| 098 | *//target为我们绑定的箱子变量，缩放旋转的参照物* |
| 099 | **if** (target) { |
| 100 |  |
| 101 | *//重置摄像机的位置* |
| 102 | y = ClampAngle(y, yMinLimit, yMaxLimit); |
| 103 | **var** rotation = Quaternion.Euler(y, x, 0); |
| 104 | **var** position = rotation \* Vector3(0.0, 0.0, -distance) + target.position; |
| 105 |  |
| 106 | transform.rotation = rotation; |
| 107 | transform.position = position; |
| 108 | } |
| 109 | } |
| 110 |  |
| 111 | **static** function ClampAngle (angle : **float**, min : **float**, max : **float**) { |
| 112 | **if** (angle < -360) |
| 113 | angle += 360; |
| 114 | **if** (angle > 360) |
| 115 | angle -= 360; |
| 116 | **return** Mathf.Clamp (angle, min, max); |
| 117 | } |

C#版本并加上y限制。

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

|  |  |
| --- | --- |
| 001 | **using** UnityEngine; |
| 002 | **using** System.Collections; |
| 003 |  |
| 004 | **public** **class** Move : MonoBehaviour { |
| 005 | **public** GameObject target; |
| 006 | **public** **float** distance = 10.0f; |
| 007 |  |
| 008 | **public** **float** xSpeed = 250.0f; |
| 009 | **public** **float** ySpeed = 120.0f; |
| 010 | Vector3 tmp; |
| 011 | **public** **float** yMinLimit = -20; |
| 012 | **public** **float** yMaxLimit = 80; |
| 013 |  |
| 014 | **public** **float** x = 0.0f; |
| 015 | **public** **float** y = 0.0f; |
| 016 |  |
| 017 | **private** Vector2 oldPosition1; |
| 018 | **private** Vector2 oldPosition2; |
| 019 |  |
| 020 | **void** Start (){ |
| 021 | Vector2 angles= transform.eulerAngles; |
| 022 | x = angles.y; |
| 023 | y = angles.x; |
| 024 |  |
| 025 | *// Make the rigid body not change rotation* |
| 026 | **if** (rigidbody) |
| 027 | rigidbody.freezeRotation = **true**; |
| 028 | } |
| 029 | **void** Update (){ |
| 030 |  |
| 031 | **if**(Input.touchCount == 1) |
| 032 | { |
| 033 |  |
| 034 | **if**(Input.GetTouch(0).phase==TouchPhase.Moved) |
| 035 | { |
| 036 |  |
| 037 | x += Input.GetAxis("Mouse X") \* xSpeed \* 0.02f; |
| 038 | y -= Input.GetAxis("Mouse Y") \* ySpeed \* 0.02f; |
| 039 | y=ClampAngle(y,yMinLimit,yMaxLimit); |
| 040 |  |
| 041 | } |
| 042 | } |
| 043 |  |
| 044 |  |
| 045 | **if**(Input.touchCount >1 ) |
| 046 | { |
| 047 |  |
| 048 | **if**(Input.GetTouch(0).phase==TouchPhase.Moved||Input.GetTouch(1).phase==TouchPhase.Moved) |
| 049 | { |
| 050 |  |
| 051 | Vector3 tempPosition1= Input.GetTouch(0).position; |
| 052 | Vector3 tempPosition2= Input.GetTouch(1).position; |
| 053 | **if**(isEnlarge(oldPosition1,oldPosition2,tempPosition1,tempPosition2)) |
| 054 | { |
| 055 | **if**(distance > 3) |
| 056 | { |
| 057 | distance -= 0.5f; |
| 058 | } *//Unity3D教程手册：www.unitymanual.com* |
| 059 | }**else** |
| 060 | { |
| 061 | **if**(distance < 18.5f) |
| 062 | { |
| 063 | distance += 0.5f; |
| 064 | } |
| 065 | } |
| 066 | oldPosition1=tempPosition1; |
| 067 | oldPosition2=tempPosition2; |
| 068 | } |
| 069 | } |
| 070 | } |
| 071 |  |
| 072 | **bool** isEnlarge ( Vector2 oP1 , Vector2 oP2 , Vector2 nP1 , Vector2 nP2 ) |
| 073 | { |
| 074 | **float** leng1=Mathf.Sqrt((oP1.x-oP2.x)\*(oP1.x-oP2.x)+(oP1.y-oP2.y)\*(oP1.y-oP2.y)); |
| 075 | **float** leng2=Mathf.Sqrt((nP1.x-nP2.x)\*(nP1.x-nP2.x)+(nP1.y-nP2.y)\*(nP1.y-nP2.y)); |
| 076 | **if**(leng1<leng2) |
| 077 | { |
| 078 | **return** **true**; |
| 079 | }**else** |
| 080 | { |
| 081 | **return** **false**; |
| 082 | } |
| 083 | } *//Unity3D教程手册：www.unitymanual.com* |
| 084 | **public** **void** LateUpdate (){ |
| 085 | **if** (target) { |
| 086 |  |
| 087 | ClampAngle(y, yMinLimit, yMaxLimit); |
| 088 | Quaternion rotation= Quaternion.Euler(y, x, 0); |
| 089 |  |
| 090 | tmp.**Set**(0.0f, 0.0f, (-1)\*distance); |
| 091 | Vector3 position= rotation \* tmp + target.transform.position; |
| 092 |  |
| 093 | transform.rotation = rotation; |
| 094 | transform.position = position; |
| 095 | } |
| 096 | } |
| 097 | **static** **float** ClampAngle ( **float** angle , **float** min , **float** max ){ |
| 098 | **if** (angle < -360) |
| 099 | angle += 360; |
| 100 | **if** (angle > 360) |
| 101 | angle -= 360; |
| 102 | **return** Mathf.Clamp (angle, min, max); |
| 103 | } |
| 104 |  |
| 105 | } |