# Unity3D之人物头顶名称与血条更新与绘制

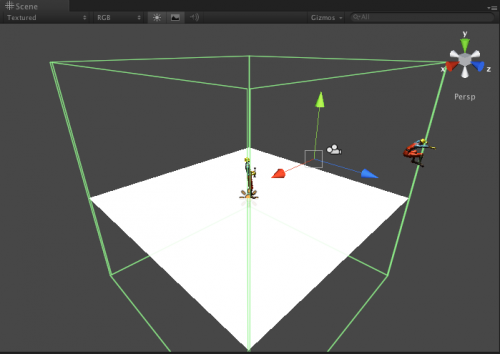
时间:12-07-18 栏目:[Unity3D教程](http://unity3d.9ria.com/?cat=1)作者:liuxiaoni 评论:3

人物的名称与血条的绘制方法很简单，但是我们需要解决的问题是如何在3D世界中寻找合适的坐标。因为3D世界中的人物是会移动的，它是在3D世界中移动，并不是在2D平面中移动，但是我们需要将3D的人物坐标换算成2D平面中的坐标，继而找到人物头顶在屏幕中的2D坐标最后使用GUI将名称与血条绘制出来。

首先学习本文的重点内容，如何将游戏世界中任意3D坐标转换成屏幕中的2D坐标。根据这个方法计算出的2D坐标屏幕左下角的点为0.0 ，屏幕右上角的坐标为1.1 所以真实的2D坐标还得通过Screen.height 与Screen.width计算一下才行。

Vector2 position = camera.WorldToScreenPoint (worldPosition);

在Unity工程导入角色控制器组件，不知道角色控制器的朋友请阅读我之前的文章哈。创建一个Plane做为游戏的地面，然后利用角色控制器组件创建两个模型，一个做为主角，一个作为NPC，主角可以通过控制来移动从四周来观察NPC对象。由于地面的面积比较小移动主角时为了避免主角越界掉下去，我们做一个边界的物理层。物理层其实很简单，就是给平面四周放置四个平面在四周将平面包围着，给四周的四个平面绑定上Box Collider组件，这样主角就不会越界掉下去啦。因为没有给贴图所以效果上看不到这四个对象。 如下图所示，在场景是途中主角被四个平面包围这，即时它拼命的想往外条但是还是跳不出去。

[](http://unity3d.9ria.com/?attachment_id=1060)

创建脚本NPC.cs 然后把脚本挂在NPC对象身上，在脚本中我们绘制主角的血条以及名称。

NPC.cs

using UnityEngine;

using System.Collections;

public class NPC : MonoBehaviour {

//主摄像机对象

private Camera camera;

//NPC名称

private string name = "我是雨松MOMO";

//主角对象

GameObject hero;

//NPC模型高度

float npcHeight;

//红色血条贴图

public Texture2D blood\_red;

//黑色血条贴图

public Texture2D blood\_black;

//默认NPC血值

private int HP = 100;

void Start ()

{

//根据Tag得到主角对象

hero = GameObject.FindGameObjectWithTag("Player");

//得到摄像机对象

camera = Camera.main;

//注解1

//得到模型原始高度

float size\_y = collider.bounds.size.y;

//得到模型缩放比例

float scal\_y = transform.localScale.y;

//它们的乘积就是高度

npcHeight = (size\_y \*scal\_y) ;

}

void Update ()

{

//保持NPC一直面朝主角

transform.LookAt(hero.transform);

}

void OnGUI()

{

//得到NPC头顶在3D世界中的坐标

//默认NPC坐标点在脚底下，所以这里加上npcHeight它模型的高度即可

Vector3 worldPosition = new Vector3 (transform.position.x , transform.position.y + npcHeight,transform.position.z);

//根据NPC头顶的3D坐标换算成它在2D屏幕中的坐标

Vector2 position = camera.WorldToScreenPoint (worldPosition);

//得到真实NPC头顶的2D坐标

position = new Vector2 (position.x, Screen.height - position.y);

//注解2

//计算出血条的宽高

Vector2 bloodSize = GUI.skin.label.CalcSize (new GUIContent(blood\_red));

//通过血值计算红色血条显示区域

int blood\_width = blood\_red.width \* HP/100;

//先绘制黑色血条

GUI.DrawTexture(new Rect(position.x - (bloodSize.x/2),position.y - bloodSize.y ,bloodSize.x,bloodSize.y),blood\_black);

//在绘制红色血条

GUI.DrawTexture(new Rect(position.x - (bloodSize.x/2),position.y - bloodSize.y ,blood\_width,bloodSize.y),blood\_red);

//注解3

//计算NPC名称的宽高

Vector2 nameSize = GUI.skin.label.CalcSize (new GUIContent(name));

//设置显示颜色为黄色

GUI.color = Color.yellow;

//绘制NPC名称

GUI.Label(new Rect(position.x - (nameSize.x/2),position.y - nameSize.y - bloodSize.y ,nameSize.x,nameSize.y), name);

}

//下面是经典鼠标点击对象的事件，大家看一下就应该知道是什么意思啦。

void OnMouseDrag ()

{

Debug.Log("鼠标拖动该模型区域时");

}

void OnMouseDown()

{

Debug.Log("鼠标按下时");

if(HP >0)

{

HP -=5 ;

}

}

void OnMouseUp()

{

Debug.Log("鼠标抬起时");

}

void OnMouseEnter()

{

Debug.Log("鼠标进入该对象区域时");

}

void OnMouseExit()

{

Debug.Log("鼠标离开该模型区域时");

}

void OnMouseOver()

{

Debug.Log("鼠标停留在该对象区域时");

}

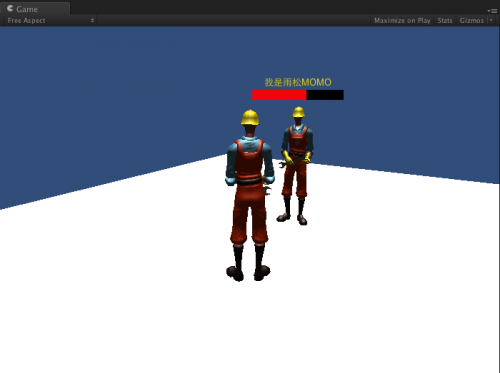
}

注解1:通过collider.bounds.size 可以拿到模型对应三个轴向的高度，但是模型是可以缩放的，所以真实的模型高度应当是原始高度乘以缩放系数才行。 transform.localScale可以拿到模型对应三个轴向的缩放系数，因为这里我们需要模型的高度，所以忽略X轴与Z轴。

注解2：在这里我们计算血条的宽度，GUI.skin.label.Calcsize()这个方法是以默认的皮肤对象Label对象去参数对象的宽高。参数是new GUIContent(blood\_Red)意思是拿红色血条的贴图的宽高，它将保存在返回的size中。最后以宽高将血条绘制在屏幕中，我们的血条采取两层。背景是黑色的，前面是红色的，当人物费血时红色血条减少。

注解3： 这里通过字符串来获取它整体的宽度与高度，因为NPC的名称是可变的，所以我们需要动态的获取整体的显示区域。同样是以GUI.skin.label对象去调用CalcSize。

如下图所示，当使用鼠标点击NPC对象时，NPC头顶的血条将开始发生减血。这个例子我使用OnGUI绘制当然大家也可以在Hierarchy 视图中的创建GUI Texture 或者GUI Text对象 来实现，不过原理都是这样的 大家可以试试。

[](http://unity3d.9ria.com/?attachment_id=1061)