**Unity3D脚本：Unity3D闪烁灯光脚本**

Posted on 2013年01月27日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 350 次

using UnityEngine;  
using System.Collections;  
[RequireComponent( typeof( Light ) )]  
public class flickeringLight : MonoBehaviour  
{  
// Flickering Styles  
public enum flickerinLightStyles { CampFire = 0, Fluorescent = 1 };  
public flickerinLightStyles flickeringLightStyle = flickerinLightStyles.CampFire;  
// Campfire Methods  
public enum campfireMethods { Intensity = 0, Range = 1, Both = 2 };  
public campfireMethods campfireMethod = campfireMethods.Intensity;  
// Intensity Styles  
public enum campfireIntesityStyles { Sine = 0, Random = 1 };  
public campfireIntesityStyles campfireIntesityStyle = campfireIntesityStyles.Random;  
// Range Styles  
public enum campfireRangeStyles { Sine = 0, Random = 1 };  
public campfireRangeStyles campfireRangeStyle = campfireRangeStyles.Random;  
// Base Intensity Value  
public float CampfireIntensityBaseValue = 0.5f;  
// Intensity Flickering Power  
public float CampfireIntensityFlickerValue = 0.1f;  
// Base Range Value  
public float CampfireRangeBaseValue = 10.0f;  
// Range Flickering Power  
public float CampfireRangeFlickerValue = 2.0f;

// If Style is Sine  
private float CampfireSineCycleIntensity = 0.0f;  
private float CampfireSineCycleRange = 0.0f;  
// "Glow" Speeds  
public float CampfireSineCycleIntensitySpeed = 5.0f;  
public float CampfireSineCycleRangeSpeed = 5.0f;  
public float FluorescentFlickerMin = 0.4f;  
public float FluorescentFlickerMax = 0.5f;  
public float FluorescentFlicerPercent = 0.95f;  
// NOT IMPLEMENTED YET !!!!  
public bool FluorescentFlickerPlaySound = false;  
public AudioClip FluorescentFlickerAudioClip;  
// ------------------------  
// Use this for initialization  
void Start () {

}

// Update is called once per frame  
void Update () {  
switch( flickeringLightStyle )  
{  
// If Flickering Style is Campfire  
case flickerinLightStyles.CampFire:  
// If campfire method is Intesity OR Both  
if( campfireMethod == campfireMethods.Intensity || campfireMethod == campfireMethods.Both )  
{  
// If Intensity style is Sine  
if( campfireIntesityStyle == campfireIntesityStyles.Sine )  
{  
// Cycle the Campfire angle  
CampfireSineCycleIntensity += CampfireSineCycleIntensitySpeed;  
if( CampfireSineCycleIntensity > 360.0f ) CampfireSineCycleIntensity = 0.0f;  
// Base + Values  
light.intensity = CampfireIntensityBaseValue + ( ( Mathf.Sin( CampfireSineCycleIntensity \* Mathf.Deg2Rad ) \* ( CampfireIntensityFlickerValue / 2.0f ) ) + ( CampfireIntensityFlickerValue / 2.0f ) );  
}  
else light.intensity = CampfireIntensityBaseValue + Random.Range( 0.0f, CampfireIntensityFlickerValue );  
}  
// If campfire method is Range OR Both  
if( campfireMethod == campfireMethods.Range || campfireMethod == campfireMethods.Both )  
{  
// If Range style is Sine  
if( campfireRangeStyle == campfireRangeStyles.Sine )  
{  
// Cycle the Campfire angle  
CampfireSineCycleRange += CampfireSineCycleRangeSpeed;  
if( CampfireSineCycleRange > 360.0f ) CampfireSineCycleRange = 0.0f;  
// Base + Values  
light.range = CampfireRangeBaseValue + ( ( Mathf.Sin( CampfireSineCycleRange \* Mathf.Deg2Rad ) \* ( CampfireSineCycleRange / 2.0f ) ) + ( CampfireSineCycleRange / 2.0f ) );  
}  
else light.range = CampfireRangeBaseValue + Random.Range( 0.0f, CampfireRangeFlickerValue );  
}  
break;  
// If Flickering Style is Fluorescent  
case flickerinLightStyles.Fluorescent:  
if( Random.Range( 0.0f, 1.0f ) > FluorescentFlicerPercent )  
{  
light.intensity = FluorescentFlickerMin;  
// Check Audio - NOT IMPLEMENTED YET  
if( FluorescentFlickerPlaySound )  
{  
}  
}  
else light.intensity = FluorescentFlickerMax;  
break;  
default:  
// You should not be here.  
break;  
}

}  
}