Contents

1 Camera 相关

1.1 Camera Shake in Unity

- http://www.voidcn.com/article/p-yfmjbkxr-pu.html

```
var originPosition:Vector3;
var originRotation:Quaternion;
var shake_decay: float;
var shake_intensity: float;;
function OnGUI () {
    if (GUI.Button (Rect (20,40,80,20), "Shake")) {
        Shake();
    }
}
function Update(){
    if(shake_intensity > 0){
        transform.position = originPosition + Random.insideUnitSphere * shake_intensity;
        transform.rotation =  Quaternion(
                        originRotation.x + Random.Range(-shake_intensity,shake_intensity)*.2,
                        originRotation.y + Random.Range(-shake_intensity,shake_intensity)*.2,
                        originRotation.z + Random.Range(-shake_intensity,shake_intensity)*.2,
                        originRotation.w + Random.Range(-shake_intensity,shake_intensity)*.2);
        shake_intensity -= shake_decay;
    }
}
function Shake(){
    originPosition = transform.position;
    originRotation = transform.rotation;
    shake_intensity = .3;
    shake_decay = 0.002;
}
```

1.2 Unity 中做放大镜效果

- http://www.voidcn.com/article/p-qyltseeo-ry.html
- 其实和小地图都差不多了。还是要借助另一个相机
- 目的: 这篇文章的主要目的是要给你一个想法如何做放大境效果。
- 在 unity 中可以简单的实现放大镜效果啊. 那么现在就来一步一步实现这个:
- 创建一个摄像机对象，设置 projection 类型为 perspective 或者 orthographic.
- 设置相机的 orthographicSize 或者 fieldOfView (依赖于相机的 projection 类型).
- 设置其 pixelrect . 例如如果您想要在你鼠标位置显示放大境和其大小是 100 x 100 , 然后设置 pixelrect 为:

```
magnifyCamera.pixelRect = new Rect (Input.mousePosition.x - 100f / 2.0f, Input.mousePosition.y
```

- 设置相机的位置。例如如果你想在你的鼠标位置显示放大镜效果，那么设置相机的位置为 mousePosition 世界点。
- 下面的 C# 脚本将创建一个 MagnifyGlass，并将它移动到 mousePosition 位置。
- MagnifyGlass 脚本: 添加到一个空的游戏对象。

```
using UnityEngine;
using System.Collections;

public class MagnifyGlass : MonoBehaviour {
    private Camera magnifyCamera;
    private GameObject magnifyBorders;
    private LineRenderer LeftBorder, RightBorder, TopBorder, BottomBorder; // Reference fo
    private float MGOX, MGOY; // Magnify Glass Origin X and Y position
    private float MGWidth = Screen.width/5f, MGHeight = Screen.width/5f; // Magnify glass
    private Vector3 mousePos;

    void Start () {
        createMagnifyGlass ();
    }

    void Update () {
// Following lines set the camera's pixelRect and camera position at mouse position
        magnifyCamera.pixelRect = new Rect (Input.mousePosition.x - MGWidth / 2.0f, Input.
        mousePos = getWorldPosition (Input.mousePosition);
        magnifyCamera.transform.position = mousePos;
        mousePos.z = 0;
        magnifyBorders.transform.position = mousePos;
    }

// Following method creates MagnifyGlass
    private void createMagnifyGlass() {
        GameObject camera = new GameObject("MagnifyCamera");
        MGOX = Screen.width / 2f - MGWidth/2f;
        MGOY = Screen.height / 2f - MGHeight/2f;
        magnifyCamera = camera.AddComponent<Camera>();
        magnifyCamera.pixelRect = new Rect(MGOX, MGOY, MGWidth, MGHeight);
        magnifyCamera.transform.position = new Vector3(0,0,0);
        if (Camera.main.isOrthoGraphic) {
            magnifyCamera.orthographic = true;
            magnifyCamera.orthographicSize = Camera.main.orthographicSize / 5.0f;//+ 1.0f;
            createBordersForMagniyGlass ();
        } else {
            magnifyCamera.orthographic = false;
            magnifyCamera.fieldOfView = Camera.main.fieldOfView / 10.0f;//3.0f;
        }
    }

// Following method sets border of MagnifyGlass
private void createBordersForMagniyGlass() {
magnifyBorders = new GameObject ();
LeftBorder = getLine ();
LeftBorder.SetVertexCount(2);
LeftBorder.SetPosition(0,
                       new Vector3(getWorldPosition(new Vector3(MGOX,MGOY,0)).x,
                                   getWorldPosition(new Vector3(MGOX,MGOY,0)).y-0.1f,
                                   -1));
LeftBorder.SetPosition(1,
                       new Vector3(getWorldPosition(new Vector3(MGOX,MGOY+MGHeight,0)).x,
                               getWorldPosition(new Vector3(MGOX,MGOY+MGHeight,0)).y+0.1f,
                                   -1));
LeftBorder.transform.parent = magnifyBorders.transform;
TopBorder = getLine ();
TopBorder.SetVertexCount(2);
TopBorder.SetPosition(0,
                      new Vector3(getWorldPosition(new Vector3(MGOX,MGOY+MGHeight,0)).x,
                                  getWorldPosition(new Vector3(MGOX,MGOY+MGHeight,0)).y,
                                  -1));
TopBorder.SetPosition(1,
                  new Vector3(getWorldPosition(new Vector3(MGOX+MGWidth,MGOY+MGHeight,0)).
                          getWorldPosition(new Vector3(MGOX+MGWidth,MGOY+MGHeight,0)).y,
                          -1));
TopBorder.transform.parent = magnifyBorders.transform;
RightBorder = getLine ();
RightBorder.SetVertexCount(2);
RightBorder.SetPosition(0,
                new Vector3(getWorldPosition(new Vector3(MGOX+MGWidth,MGOY+MGWidth,0)).x,
                       getWorldPosition(new Vector3(MGOX+MGWidth,MGOY+MGWidth,0)).y+0.1f,
                                    -1));
RightBorder.SetPosition(1,
                      new Vector3(getWorldPosition(new Vector3(MGOX+MGWidth,MGOY,0)).x,
                                getWorldPosition(new Vector3(MGOX+MGWidth,MGOY,0)).y-0.1f,
                                    -1));
RightBorder.transform.parent = magnifyBorders.transform;
BottomBorder = getLine ();
BottomBorder.SetVertexCount(2);
BottomBorder.SetPosition(0,
                         new Vector3(getWorldPosition(new Vector3(MGOX+MGWidth,MGOY,0)).x,
                                     getWorldPosition(new Vector3(MGOX+MGWidth,MGOY,0)).y,
                                     -1));
BottomBorder.SetPosition(1,
                         new Vector3(getWorldPosition(new Vector3(MGOX,MGOY,0)).x,
                                     getWorldPosition(new Vector3(MGOX,MGOY,0)).y,
                                     -1));
BottomBorder.transform.parent = magnifyBorders.transform;
}

// Following method creates new line for MagnifyGlass's border
    private LineRenderer getLine() {
        LineRenderer line = new GameObject("Line").AddComponent<LineRenderer>();
        line.material = new Material(Shader.Find("Diffuse"));
        line.SetVertexCount(2);
        line.SetWidth(0.2f, 0.2f);
        line.SetColors(Color.black,  Color.black);
        line.useWorldSpace = false;
        return line;
    }
    private void setLine(LineRenderer line) {
        line.material = new Material(Shader.Find("Diffuse"));
        line.SetVertexCount(2);
        line.SetWidth(0.2f, 0.2f);
        line.SetColors(Color.black, Color.black);
        line.useWorldSpace = false;
    }

// Following method calculates world's point from screen point as per camera's projection
    public Vector3 getWorldPosition(Vector3 screenPos) {
        Vector3 worldPos;
        if(Camera.main.isOrthoGraphic) {
            worldPos = Camera.main.ScreenToWorldPoint (screenPos);
            worldPos.z = Camera.main.transform.position.z;
        } else {
            worldPos = Camera.main.ScreenToWorldPoint (new Vector3 (screenPos.x, screenPos
            worldPos.x *= -1;
            worldPos.y *= -1;
        }
        return worldPos;
    }
}
```

1.3 Unity 3D– 摄像机 Clear Flags 和 Culling Mask 属性用途详解

- http://www.voidcn.com/article/p-pwbltogg-pu.html

1.4 [Unity 基础] 对 Camera 组件属性的一些理解

- http://www.voidcn.com/article/p-gtfuejmb-uz.html

1.5 Unity 的 camera 组件

- http://www.voidcn.com/article/p-mcxkifby-cb.html

1.6 unity Camera 属性详解

- http://www.voidcn.com/article/p-enwwkrfr-hh.html

1.7