<https://www.jianshu.com/p/bdee9d90a120>

<https://www.cnblogs.com/liaoguipeng/p/5132775.html>

Flexible 意味着保持原有像素,不进行缩放.分辨率的变化不影响UI的像素.

在正交相机中有个size参数，等于相机高度的一半

注意一下，unity中的单位和像素之间有一个转换关系，叫做Pixels To Units



protected virtual void Awake () { mTrans = transform; }

protected virtual void OnEnable () { list.Add(this); }

protected virtual void OnDisable () { list.Remove(this); }

protected virtual void Start ()

{

UIOrthoCamera oc = GetComponentInChildren<UIOrthoCamera>();

if (oc != null)

{

Debug.LogWarning("UIRoot should not be active at the same time as UIOrthoCamera. Disabling UIOrthoCamera.", oc);

Camera cam = oc.gameObject.GetComponent<Camera>();

oc.enabled = false;

**if (cam != null) cam.orthographicSize = 1f;** **// 设置正交相机的size为1**

}

else UpdateScale(false);

}

void Update ()

{

#if UNITY\_EDITOR

if (!Application.isPlaying && gameObject.layer != 0)

UnityEditor.EditorPrefs.SetInt("NGUI Layer", gameObject.layer);

#endif

UpdateScale();

}

实际上每帧都会调用UpdateScale

public void UpdateScale (bool updateAnchors = true)

{

if (mTrans != null)

{

float calcActiveHeight = activeHeight;

if (calcActiveHeight > 0f)

{

float size = 2f / calcActiveHeight;

Vector3 ls = mTrans.localScale;

if (!(Mathf.Abs(ls.x - size) <= float.Epsilon) ||

!(Mathf.Abs(ls.y - size) <= float.Epsilon) ||

!(Mathf.Abs(ls.z - size) <= float.Epsilon))

{

mTrans.localScale = new Vector3(size, size, size);

if (updateAnchors) BroadcastMessage("UpdateAnchors", SendMessageOptions.DontRequireReceiver);

}

}

}

}

这里的camera.size=1，而size是屏幕逻辑高度的一半，因此，屏幕的逻辑高度为2

同时每一帧都会调用子节点的UpdateAnchors函数