**正交相机**

在unity中的正交相机中，有一个size的参数

这个参数实际上是设置的渲染高度，确定了裁剪区域（渲染高度就是裁剪高度？）

Size = 渲染高度 / pixel per unit / 2

也就是说渲染

渲染高度 = size \* 2 \* pixel per unit

渲染宽度 = 渲染高度 \* 屏幕宽度 / 屏幕高度

*Debug*.*Log*("---Screen.width = " + *Screen*.*width* + ", Screen.height = " + *Screen*.*height*);

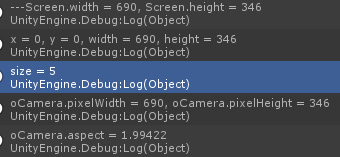
*Rect* rect = oCamera.*pixelRect*;

*Debug*.*Log*("x = " + rect.*x* + ", y = " + rect.*y* + ", width = " + rect.*width* + ", height = " + rect.*height*);

*Debug*.*Log*("size = " + oCamera.*orthographicSize*);

*Debug*.*Log*("oCamera.pixelWidth = " + oCamera.*pixelWidth* + ", oCamera.pixelHeight = " + oCamera.*pixelHeight*);

*Debug*.*Log*("oCamera.aspect = " + oCamera.*aspect*);

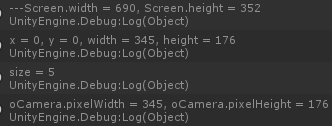


可以看到正交相机的宽高裁剪也与屏幕宽高有关系，aspect也会变化

注意：pixelWidth与pixelHeight会根据viewport发生变化

Camera设置：



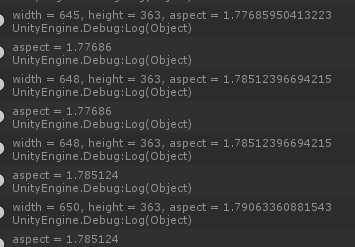


**透视相机**

string text = string.*Format*("width = {0}, height = {1}, aspect = {2}", *Screen*.*width*, *Screen*.*height*, *Convert*.*ToDouble*(*Screen*.*width*) / *Screen*.*height*);

*Debug*.*Log*(text);

*Debug*.*Log*("aspect = " + camera.*aspect*);



可以看出透视相机的初始化aspect就是屏幕的宽高比