20200906

我们研究了unity自带的postprocessing系统，折腾了很长时间发现这个东西很难满足我们对部分特效进行强化渲染的需求。

首先这个功能本身是偏向于“作用于镜头”。说白了这个东西就是个滤镜，只能全镜头加效果，可认为它没法针对任何单独的东西进行单独处理。确实它可以对单独区域进行单独的特殊渲染但区域还必须是方体，这几乎就让这个局部渲染功能没啥意义。

1. post processing layer组件是挂在相机上的，直接体现出整个机制是作用于相机的特点。这个模块下面有volume blending这个部分，有两个待选选项，里面有“layer”的选择，然而这个选项的实际作用和我们直感的完全不同。整个系统并不是针对特定层的gameobject产生作用，而是对特定层的post processing volume起作用。

2. post processing volume 这个东西的翻译是实际渲染区域，如果is globle不是true的话那到底是哪块区域，竟是靠collider来协助规定。而如果是true，那么就相当于整个世界陷入了一个大果冻，那么在这个果冻里看的东西当然收到这个果冻的材质的影响了。单独一个处于所选postprocessing层的gameobject并不会在视觉上被特殊处理，整个系统真正进行了视觉处理的东西是post processing volume 。

3. 综上所述只要一个东西，不管gameobject还是UI，他在post processing volume 这个大果冻里的话就一定会受到特殊视觉处理。然而基于这些原理我们还真的想到一个办法让这种处理只针对特定层。

步骤如下：设置一个新相机，这个相机的depth比主相机还要低，而且clear flags为固定黑或天空盒，而主相机为更上一层。clearflags设置为depth only。新相机的culling mask为特殊渲染层postprocessing，然后设置一个globel的post processing volume ，从而，这个新相机由于depth最低并且只渲染postprocessing层的东西，所以肯定能把特效物体给特殊视觉化，而其他渲染层的东西渲染出来的时候，因为是后渲染，渲染在这个post层之上，那么就都没受大果冻影响。

但这个做法导致了postprocessing层的东西一定在最底层。比如某个角色发了个闪电拳那么这个闪电总在拳头后面挡着，这不是我们要的感觉。