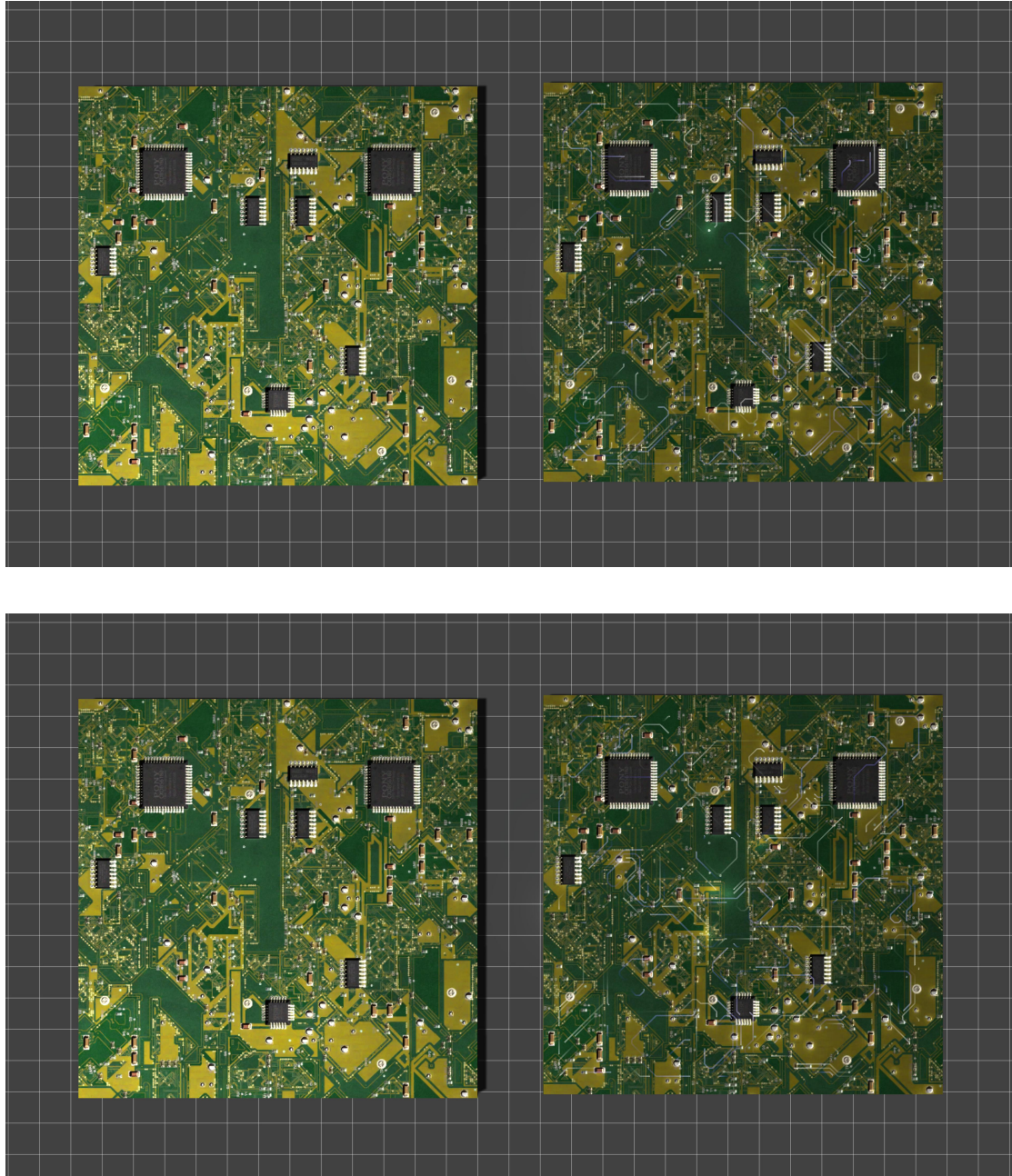


遮挡剔除与视锥剔除

深互动设计硕 20 班 沈琪川 2020214425

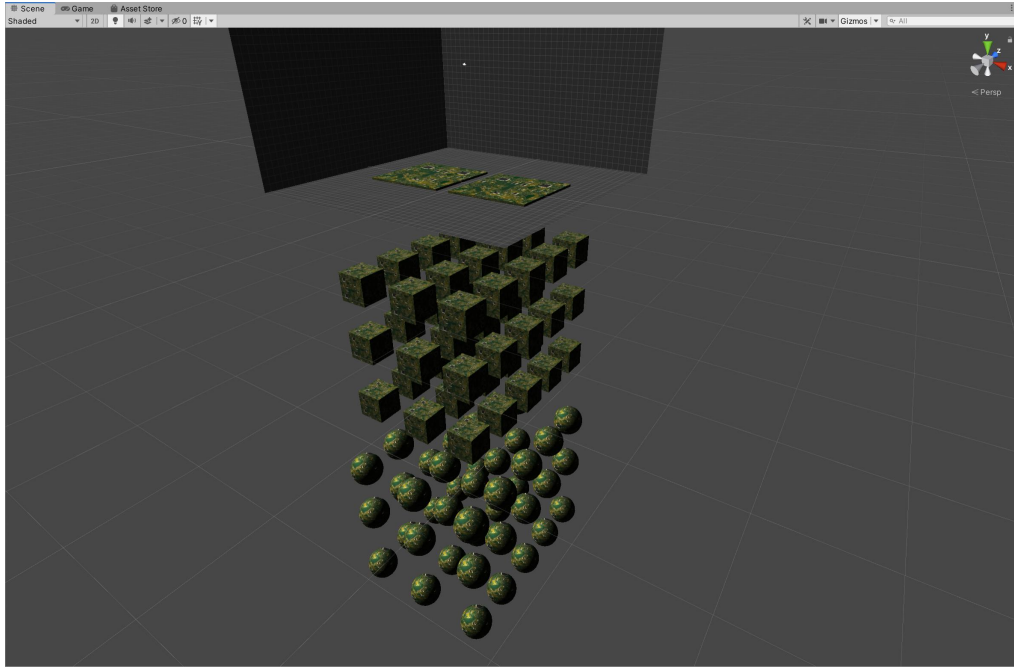
电路板及电路流动

完成了电路板材质（图左）与电路的流动效果（图右）。
详细效果见打包后的可执行文件。



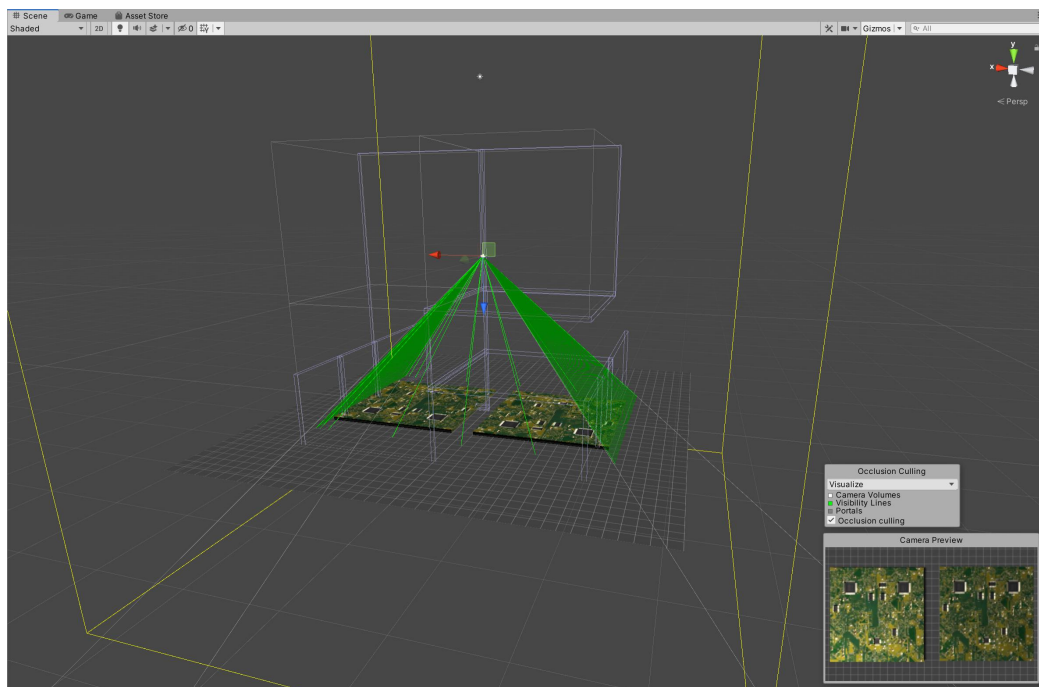
遮挡剔除

为了实验遮挡剔除，在场景中加了一堆正方体和球体，给予了一定的遮挡关系，如下图。



暂时关闭流动的电场效果，在检视面板标记其余所有需要应用遮挡剔除的物体为 Occlusion Static 和 Occludee Static，之后进行 Bake 开启遮挡剔除，在界面中勾选 Occlusion Culling。

可以发现，被灰色板子遮挡而看不见的物体（下面所有的方块和小球），就不会进行渲染，如下图所示。



接下来比较一下开启视锥剔除前后的 DC 值，左图是开启前，右图是开启后。

Statistics		Statistics	
Audio:		Audio:	
Level: -74.8 dB	DSP load: 0.1%	Level: -74.8 dB	DSP load: 0.1%
Clipping: 0.0%	Stream load: 0.0%	Clipping: 0.0%	Stream load: 0.0%
Graphics:		Graphics:	
2407.5 FPS (0.4ms)		1951.1 FPS (0.5ms)	
CPU: main 0.4ms render thread 0.2ms		CPU: main 0.5ms render thread 0.2ms	
Batches: 12	Saved by batching: 178	Batches: 7	Saved by batching: 3
Tris: 70.3k	Verts: 48.7k	Tris: 80	Verts: 160
Screen: 1581x995 - 18.0 MB		Screen: 1581x995 - 18.0 MB	
SetPass calls: 12	Shadow casters: 2	SetPass calls: 7	Shadow casters: 2
Visible skinned meshes: 0	Animations: 0	Visible skinned meshes: 0	Animations: 0

可以看出，Batched Draw Calls 在开启遮挡剔除后明显变小，减少了绘制调用的次数，提高了游戏性能。

视锥剔除

由 Unity 官方文档可知，Unity 自动开启了视锥剔除，如下图。

裁剪面 (Clip Planes)

Near 和 Far Clip Plane 属性可确定摄像机视图的开始和结束位置。这些平面垂直于摄像机的方向，并从摄像机位置算起。__近平面__是要渲染的最近位置，而__远平面__是最远位置。

裁剪面还将确定深度缓冲区精度在场景上的分布情况。一般情况下，为了获得更好的精度，应将__近平面__尽可能移到远处。

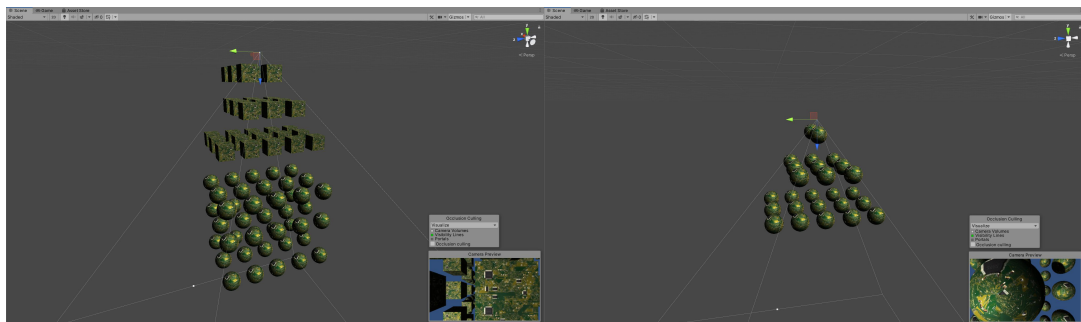
注意，近裁剪面和远裁剪面以及由摄像机可视角度定义的平面一起描述了通常所说的摄像机_视锥体_。Unity 确保在渲染对象时不会显示完全位于此视锥体之外的对象。这就是所谓的视锥体剔除 (Frustum Culling)。无论您是否在游戏中使用遮挡剔除 (Occlusion Culling)，都会发生视锥体剔除。

出于性能原因，您可能希望更早剔除小对象。例如，使小石块和碎片不可见的距离可远小于使大型建筑物不可见的距离。为此，可将小对象放入单独一层，并使用 Camera.LayerCullDistances 脚本函数来设置每层剔除距离。

“Unity 确保在渲染对象时不会显示完全位于此视锥体之外的对象。这就是所谓的视锥体剔除 (Frustum Culling)。无论您是否在游戏中使用遮挡剔除 (Occlusion Culling)，都会发生视锥体剔除。”

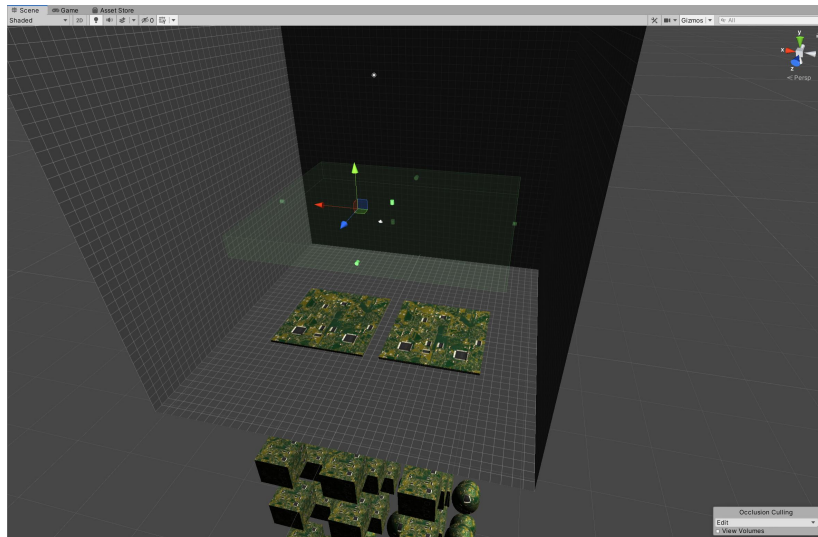
参考文档：<https://docs.unity3d.com/cn/current/Manual/class-Camera.html>

在 Unity 中进行实操如下，把遮挡剔除的效果先取消勾选，只看视锥剔除的效果，可以看到在视锥体以外的物体不会被渲染，如下图所示。

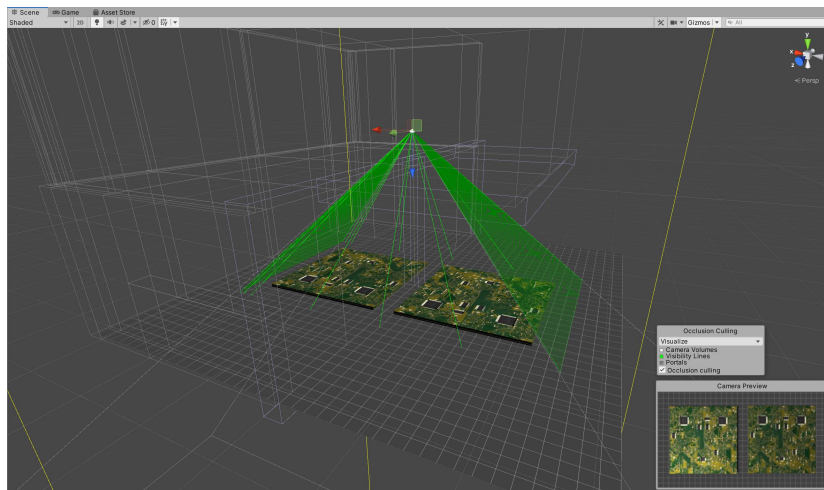


布置 Occlusion Area

由于只在电路板的正上方观察，所以相机移动的位置相对固定，布置 Occlusion Area 如下图。



然后进行烘焙，开启遮挡剔除。



在布置恰当的 Occlusion Area 之后，在 Frame Debug 中各项事件次数都有所下降，烘焙数据减小。

Frame Debug	
Disable	Editor
▼ Camera.Render	12
▼ UpdateDepthTexture	3
Clear (color+Z+stencil)	
▼ DepthPass.Job	2
Static Batch	
Draw Mesh bottom	
▼ Drawing	7
▼ Render.OpaqueGeometry	7
▼ RenderForwardOpaque.Render	7
▼ Shadows.RenderShadowMap	2
Clear (color+Z+stencil)	
▼ Shadows.RenderJob	1
▼ Shadows.RenderJobDir	1
Static Batch	
▼ RenderForwardOpaque.CollectShad	2
▼ Shadows.CollectShadows	2
Clear (color)	
Draw GL	
▼ Clear	1
Clear (color+Z+stencil)	
▼ RenderForward.RenderLoop.Job	2
Static Batch	
Draw Mesh bottom	
▼ Camera.ImageEffects	2
▼ RenderTexture.ResolveAA	1
Resolve Color	
Draw Dynamic	