

Shader 实验报告

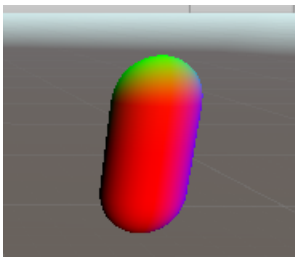
516030910435 汤志彪

基础功能：

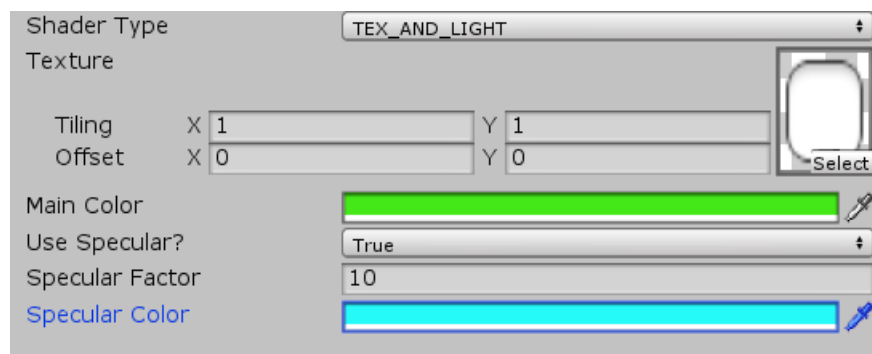
按照文档中的步骤实现了基础功能；

GUI 中设计了两种 shader 类型：显示法向量的 Shader 和纹理贴图+光照的 Shader

NORMAL_ONLY

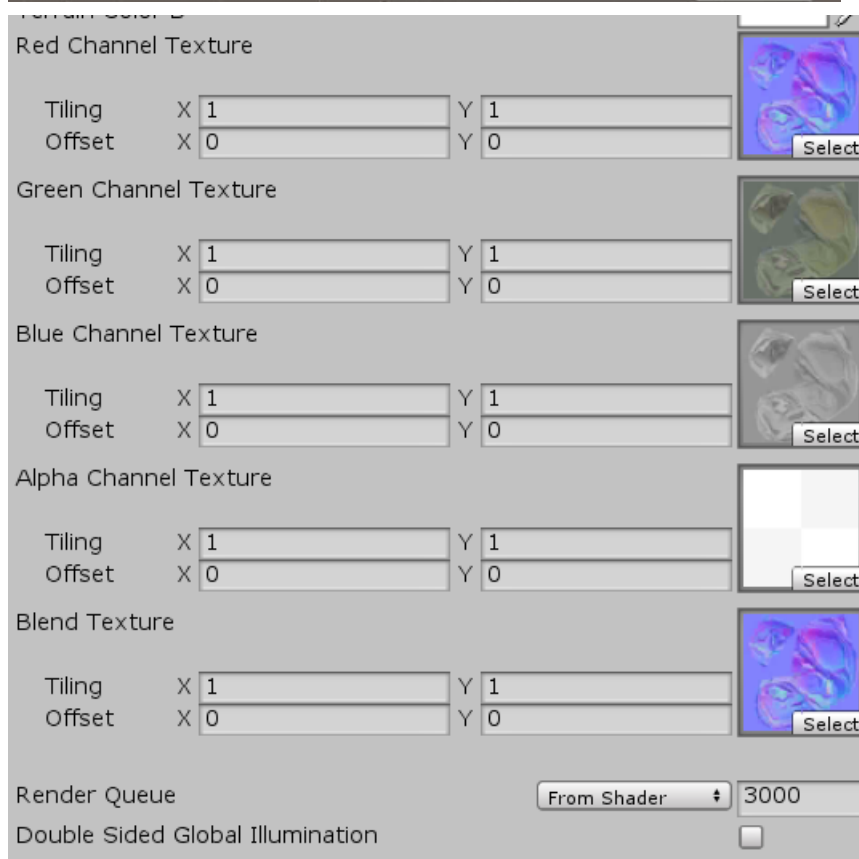


TEX_AND_LIGHT



额外内容：

混合纹理 Shader：

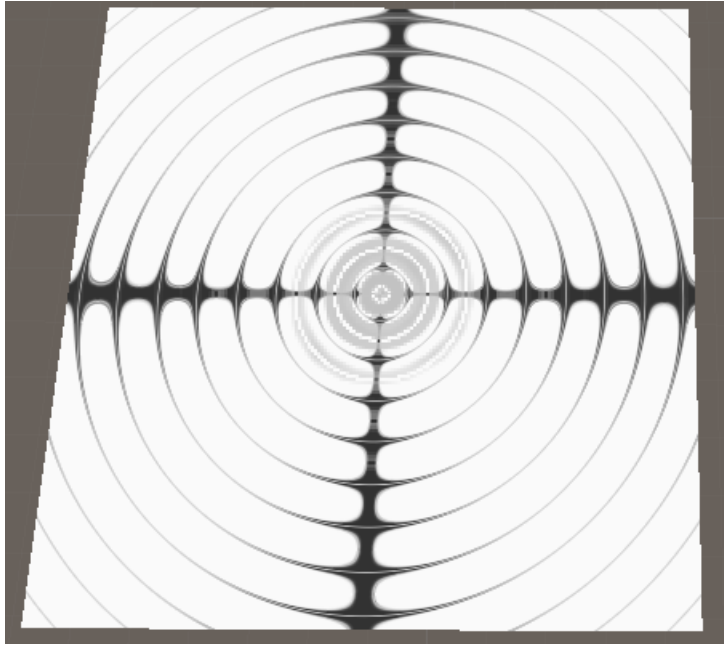


该 shader 把 RGBA 四个通道的纹理进行混合。混合原理是将四张纹理进行线性插值。Blend texture 储存需要混合的纹理的索引值，如

```
finalColor = lerp(rTexData, gTexData, blendData.g);
```

是将 BlendTexture 中的绿色通道当作 f 值，也就 BlendTexture 中越绿的地方，gTexture 的值越多，表现就是 gTexture 纹理越明显。剩下的也是同样的道理。

水波效果的 Shader（波源在平面中心）



```
//计算出fixed2(0.5, 0.5)到uv每个点的单位向量方向
fixed2 uvDir = normalize(IN.uv - fixed2(0.5, 0.5));
//计算出fixed2(0.5, 0.5)到uv每个点的距离
fixed dis = distance(IN.uv, fixed2(0.5, 0.5));

fixed2 uv = IN.uv + _WaveStrength * uvDir * sin(_Time.y * _TimeScale + dis * _WaveFactor);
return tex2D(_MainTex, uv);
```

`_WaveStrength * uvDir` 控制偏移量的大小范围和方向；

`sin` 函数的变量为 `_Time.y`，随着时间变化，`sin` 值会在 `(-1,1)` 之间变化，偏移量大小就会在 `(-_WaveStrength, WaveStrength)` 之间 变化；