



# 大家好

欢迎加入LightDir (光向) 研习社  
欢迎大家一同探索开源共享的知识分享模式

# 今日内容



■ 符文·增广·面板

■ 符文·增广·参数

■ 符文·增广·IVO

■ 符文·化神·方法

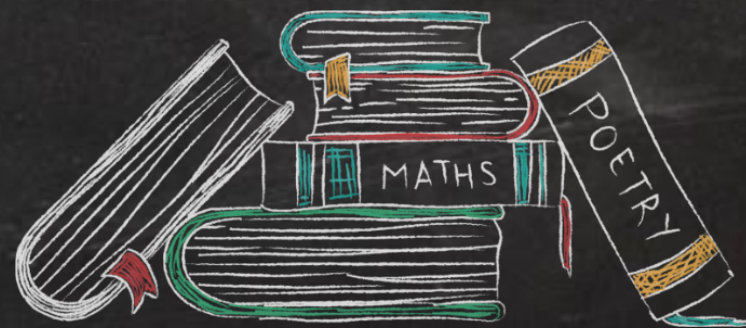
■ 符文·化神·库

■ 委托任务

$$\begin{array}{r} 24 \times 5 \\ \hline 33 \end{array}$$

1

# 符文·增广·面板





# 01 面板参数声明格式

数值, 范围:

- `_Name ( "标签名" , float) = defaultVal`
- `_Name ( "标签名" , range(min, max)) = defaultVal`
- `_Name ( "标签名" , int) = defaultVal`

位置, 向量, 颜色:

- `_Name ( "标签名" , vector) = (xVal, yVal, zVal, wVal)`
- `_Name ( "标签名" , color) = (rVal, gVal, bVal, aVal)`

2D, 3D纹理, 环境球:

- `_Name ( "标签名" , 2d) = "defaultTex" {}`
- `_Name ( "标签名" , 3d) = "defaultTex" {}`
- `_Name ( "标签名" , cube) = "defaultTex" {}`

# 02 参数属性

## [HideInInspect]

- 用途：在面板上隐藏该参数；
- 可用于：任何参数；
- 例：[HideInInspect] \_FakeLightDir ( “伪光方向” , vector) = (0.0, 1.0, 0.0, 1.0)

## [NoScaleOffset]

- 用途：禁用纹理的TilingOffset面板；不需要做TilingOffset的纹理，比如大部分的角色纹理，防止美术误设置；
- 可用于：纹理参数；
- 例：[NoScaleOffset] \_MainTex ( “主贴图” , 2d) = “white” {}

## [Normal]

- 用途：标示该纹理参数为法线贴图，以激活相关自检功能；
- 可用于：2D纹理参数；
- 例：[Normal] \_NormTex ( “法线贴图” , 2d) = “bump” {}

## [HDR]

- 用途：用于设置高动态范围颜色值；如：灯光颜色，自发光颜色等；
- 可用于：颜色参数；
- 例：[HDR] \_EmitCol ( “自发光颜色” , color) = (1.0, 1.0, 1.0, 1.0)

示范

# 03 参数属性

## [Gamma]

- 用途：用于颜色参数的色彩空间的转换；一般用于色彩空间为Linear的项目；
- 可用于：颜色参数；
- 例：[Gamma]\_EmitCol ( “自发光颜色” , color) = (1.0, 1.0, 1.0, 1.0)

示范

## [PowerSlider(value)]

- 用途：对范围参数做Power处理后再传入Shader；纠正部分参数调节手感；
- 可用于：范围参数；
- 例：[PowerSlider(0.5)]\_SpecPow ( “高光次幂” , range(1, 90)) = 30

## [Header(Label)]

- 用途：标签，用于排版；
- 可用于：单独使用；
- 例：[Header(Texture)]

## [Space(value)]

- 用途：空行，用于排版；
- 可用于：单独使用；
- 例：[Space(50)]

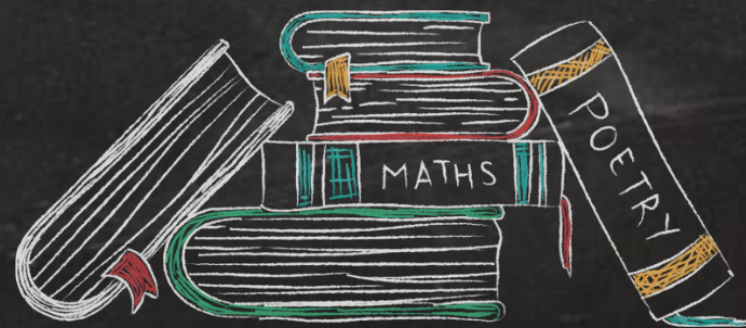
其他：[Toggle] [Enum] [Keyword] 配合宏使用，暂时不用知道；自定义Drawer需要一定C#能力，暂时不用知道；



$$\frac{24 \times 5}{33}$$

# 2

## 符文·增广·参数



# 01 ShaderLab中的参数类型

- fixed: 11位定点数,  $-2.0 \sim 2.0$ , 精度 $1/256$ ;
- half: 16位浮点数,  $-60000 \sim 60000$ , 精度约3位小数;
- float: 32位浮点数,  $-3.4E38 \sim 3.4E38$ , 精度约6, 7位小数;
- int: 32位整形数, 较少使用;
- bool: 布尔型数, 较少使用;
- 矩阵:
  - float2x2, float3x3, float4x4, float2x3 诸如此类格式;
  - half2x2, half3x3, half4x4, half2x3 诸如此类格式;
- 纹理对象:
  - sampler2D: 2D纹理
  - sampler3D: 3D纹理
  - samplerCUBE: Cube纹理

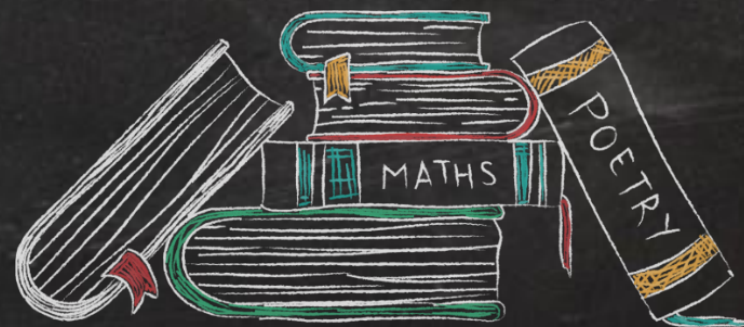


## 02 参数的精度选择

- 原则上优先使用精度最低的数据类型；
- 经验：
  - 世界空间位置和UV坐标，使用float；
  - 向量，HDR颜色，使用half；视情况升到float；
  - LDR颜色，简单乘子，可使用fixed；
- 但是：
  - 不同平台对数据类型的支持情况不同；一般会自动转换，极少数情况自动转换会带来问题；
  - 部分平台上，数据类型精度转换消耗也不小；所以fixed也是慎用；
  - 多和你们的图形开发商量；

$$\begin{array}{r} 24 \times 5 \\ \hline 33 \end{array}$$

# 3 符文·增广·IVO



# 01 可访问的顶点Input数据

- |             |       |                      |
|-------------|-------|----------------------|
| • POSITION  | 顶点位置  | float3 float4        |
| • TEXCOORD0 | UV通道1 | float2 float3 float4 |
| • TEXCOORD1 | UV通道2 | float2 float3 float4 |
| • TEXCOORD2 | UV通道3 | float2 float3 float4 |
| • TEXCOORD3 | UV通道4 | float2 float3 float4 |
| • NORMAL    | 法线方向  | float3               |
| • TANGENT   | 切线方向  | float4               |
| • COLOR     | 顶点色   | float4               |



# 02 常用的顶点Output数据

• pos	顶点位置CS	float4
• uv0	一般纹理UV	float2
• uv1	LighmapUV	float2
• posWS	顶点位置WS	float3
• nDirWS	法线方向WS	half3
• tDirWS	切线方向WS	half3
• bDirWS	副切线方向WS	half3
• color	顶点色	fixed4

# 03 常用顶点Shader操作

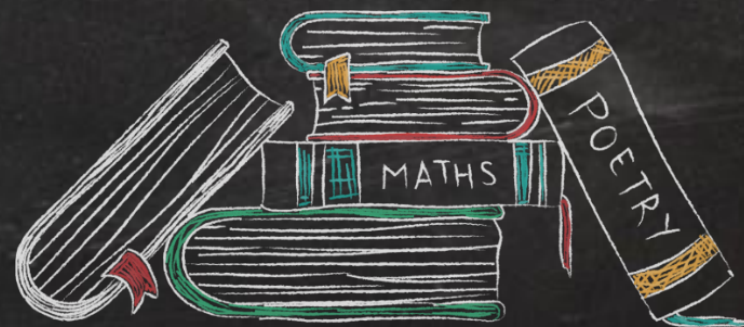
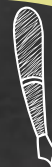
注：Unity2019.3.2f1版本

- pos            o.pos = UnityObjectToClipPos(v.vertex);
- uv0            o.uv0 = v.uv0;   o.uv0 = TRANSFORM\_TEX(v.uv0, \_MainTex);
- uv1            o.uv1 = v.uv1;   o.uv1 = v.uv1 \* unity\_LightmapST.xy + unity\_LightmapST.zw;
- posWS          o.posWS = mul(unity\_ObjectToWorld, v.vertex);
- nDirWS          o.nDirWS = UnityObjectToWorldNormal(v.normal);
- tDirWS          o.tDirWS = normalize(mul(unity\_ObjectToWorld, float4(v.tangent.xyz, 0.0)).xyz);
- bDirWS          o.bDirWS = normalize(cross(o.nDirWS, o.tDirWS) \* v.tangent.w);
- color           o.color = v.color;

$$\begin{array}{r} 24 \times 5 \\ \hline 33 \end{array}$$

3

符文·化神·法





# 01 3ColAmbient方法

优点:

- 功能模块化;
- 代码复用;
- 积累自己的方法库;

示范

原写法:

```
float upMask = max(0.0, nDirWS.g); // 获取朝上部分遮罩
float downMask = max(0.0, -nDirWS.g); // 获取朝下部分遮罩
float sideMask = 1.0 - upMask - downMask; // 获取侧面部分遮罩
float3 envCol = _EnvUpCol * upMask +
               _EnvSideCol * sideMask +
               _EnvDownCol * downMask; // 混合环境色
```

归纳为方法后的写法:

```
// 3Col环境色方法
float3 TriColAmbient (float3 n, float3 uCol, float3 sCol, float3 dCol) {
    float uMask = max(0.0, n.g); // 获取朝上部分遮罩
    float dMask = max(0.0, -n.g); // 获取朝下部分遮罩
    float sMask = 1.0 - uMask - dMask; // 获取侧面部分遮罩
    float3 envCol = uCol * uMask +
                   sCol * sMask +
                   dCol * dMask; // 混合环境色

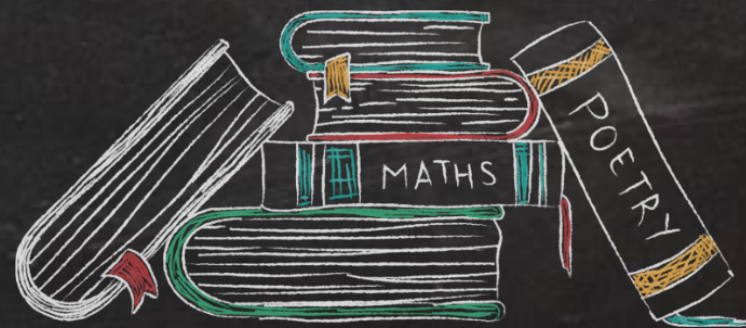
    return envCol;
}
```

```
// 使用3Col环境色方法
float3 envCol = TriColAmbient(nDirWS, _EnvUpCol, _EnvSideCol, _EnvDownCol);
```

$$\begin{array}{r} 24 \times 5 \\ \hline 33 \end{array}$$

4

符文·化神·库



# 01 3ColAmbient方法归库

- 宏 防止重复包含;
- 库内容;

示范

```
#ifndef MYCGINC
#define MYCGINC

// 3Col环境色方法
float3 TriColAmbient (float3 n, float3 uCol, float3 sCol, float3 dCol)
{
    float uMask = max(0.0, n.g);           // 获取朝上部分遮罩
    float dMask = max(0.0, -n.g);          // 获取朝下部分遮罩
    float sMask = 1.0 - uMask - dMask;     // 获取侧面部分遮罩
    float3 envCol = uCol * uMask +
                  sCol * sMask +
                  dCol * dMask;            // 混合环境色
    return envCol;
}

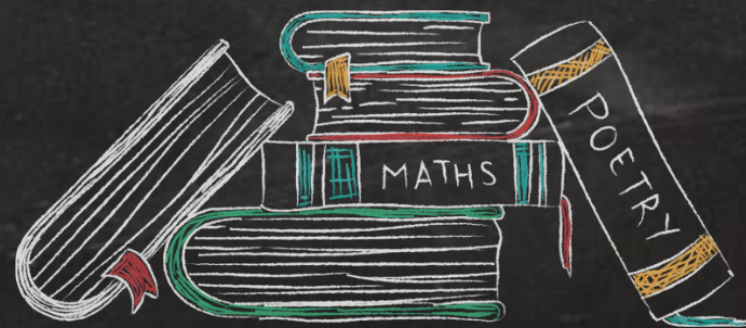
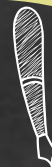
#endif
```



$$\begin{array}{r} 24 \times 5 \\ \hline 33 \end{array}$$

4

## 委托任务



# 01 大作业

假设制作人或主美，看到了这个帖子

<https://support.steampowered.com/kb/3081-QUXN-6209/dota-2-workshop-item-shader-masks>

说咱试试这个；你当如何？

- 自行查找资料及模型资源，可找组内同学讨论合作；
- 不要到网上找现成的代码，重在过程；
- 可用连连看作为辅助，作业只收代码；





# Thanks