



## 图书馆代码系列教程

题目：HLSL 着色器头文件(. fxh) 教程

类型：

作者： 南宫萧

图书馆编号：

所属组(选填)：

完 成 日 期：

基 于 版 本：

联系方式（选填）：

B051813@163. com

# HLSL 着色器头文件(.fxh)教程

本篇仅对《钢铁雄心 4》着色器头文件原始数值做出解析以及做出修改说明

## 1.buttonstate.fxh(顶点着色器的头文件)

```
// 定义顶点着色器输入结构体, 包含顶点位置和纹理坐标
VertexStruct VS_INPUT
{
    float3 vPosition : POSITION; // 顶点位置 (模型空间坐标), 语义为 POSITION,
    用于顶点处理
    float2 vTexCoord : TEXCOORD0; // 纹理坐标 (UV 坐标), 语义为 TEXCOORD0, 用
    于采样纹理
};

// 声明一个常量缓冲区, 绑定到寄存器 0 的空间 0 (ConstantBuffer(寄存器, 空间))
ConstantBuffer(0, 0)
{
    float4x4 WorldViewProjectionMatrix; // 4x4 矩阵, 用于将顶点从模型空间变换到
    投影空间 (MVP 矩阵)
    float4 Color; // RGBA 颜色值, 可能用于顶点或像素着色器的颜
    色输出
    float2 Offset; // 二维偏移量, 可能用于纹理坐标偏移 (如 UI 动
    画)
    float2 NextOffset; // 下一个动画状态的偏移量, 用于插值过渡 (如
    按钮按下/悬停动画)
    float Time; // 全局时间变量, 可用于动态效果 (如随时间变化
    的动画)
    float AnimationTime; // 动画时间参数, 控制动画进度 (如按钮状态切
    换的持续时间)
```

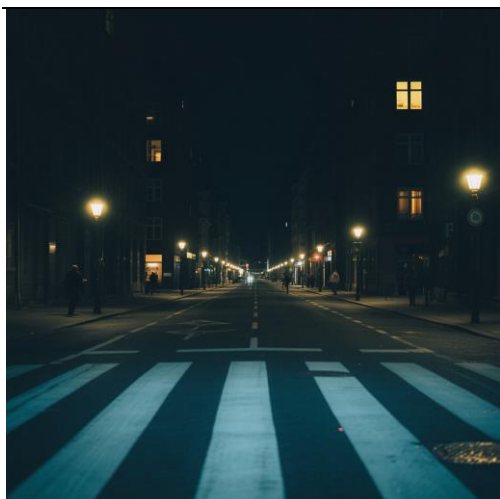
没啥好解释的, 就类似于一个初始化的文件

## 2.constants.fxh(着色器的常量集合)

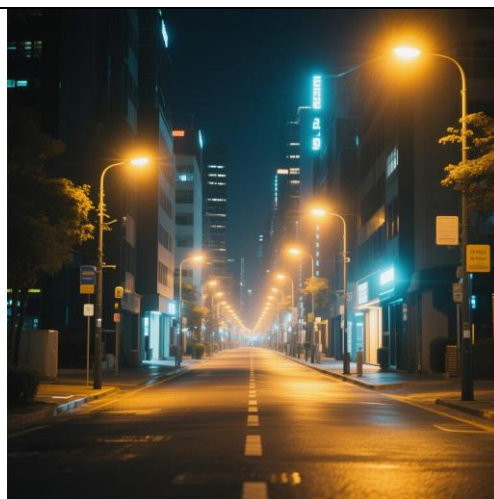
### 2.1 光照 (Light)

#### 2.1.1 夜间环境光增强系数

```
static const float NIGHT_AMBIENT_BOOST = 3.0f; // 夜间环境光增强系数, 用于提升
夜间场景的亮度
```



夜间环境光增强-关闭



夜间环境光增强-开启

## 2.1.2 白天各方向环境光颜色

### 2.1.2.1 白天-右上侧环境光 (X+方向)

```
static const float3 DayAmbientMapPosX = float3(0.1, 0.1, 0.05); // 右侧  
(X+方向) 环境光颜色
```

右侧 (X+方向) 的环境光, 低亮度 (0.05~0.1) 表示白天右侧环境光较柔和,

变量解释 (本篇通用):

**Day:** 表示白天的环境光配置

**AmbientMap:** 环境光贴图或环境光参数

**PosX:** 表示 X+方向, 即为右侧

`float3(0.1, 0.1, 0.05)`: 表示 RGB 通道的三个颜色的权重系数



白天-右上侧环境光

### 2.1.2.2 白天-左上侧环境光（X-方向）

```
static const float3 DayAmbientMapNegX = float3(0.15, 0.15, 0.15); // 左侧  
(X-方向) 环境光颜色
```

左侧（X-方向）环境光，均匀的 0.15 表示中性灰，避免左右两侧颜色差异过大。



白天-左上侧环境光

### 2.1.2.3 白天-前/后方向环境光（Y+方向）

```
static const float3 DayAmbientMapPosY = float3(0.03, 0.03, 0.06); // 前/后  
(Y+方向) 环境光颜色，可能用于抑制某些颜色
```

后方向（Y+）的环境光极，低亮度（0.03）和较高蓝色（0.06）可能用于消除复杂光照干扰



白天-后方向环境光

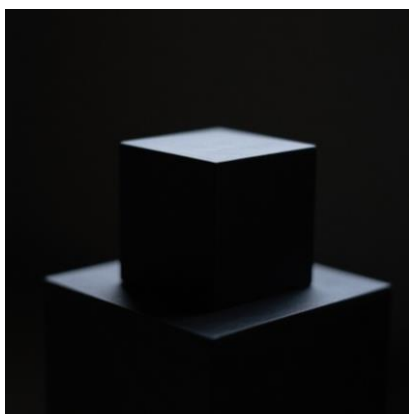


白天-前方向环境光

#### 2.1.2.4 白天-下方向环境光(下方向直射光源) (Y-方向)

```
static const float3 DayAmbientMapNegY = float3(0.0, 0.0, 0.0);    // 方(Y-  
方向) 环境光颜色 (通常无光照)
```

下方 (Y-方向) 环境光, 通常无实际光源 (如地面以下), 全 0 表示无任何光照



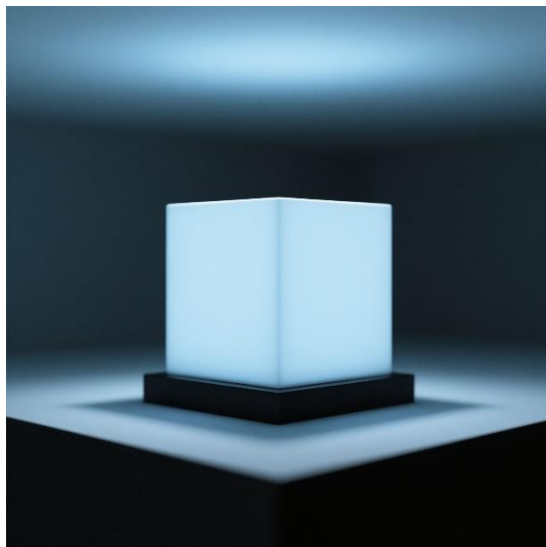
白天-下方向环境光(下方向直射光源)

#### 2.1.2.5 白天-上方环境光颜色 (Z+方向)

```
static const float3 DayAmbientMapPosZ = float3(0.0502, 0.05023, 0.1023); //  
/ 上方 (Z+方向) 环境光颜色 (如天空光)
```

---

上方 (Z+方向, 如天空) 的环境光, 模拟自然天光, 轻微偏蓝 (Z 分量 0.1023) 以匹配天空颜色



白天-上方环境光颜色

#### 2.1.2.5 白天-下方环境光颜色 (Z-方向) (模拟漫反射)

```
static const float3 DayAmbientMapNegZ = float3(0.03, 0.033, 0.033);  
// 下方 (Z-方向) 环境光颜色 (如地面反射) 下方 (Z-方向, 如地面反射) 的环境光,  
低亮度中性灰, 避免地面反光过强
```



白天-下方环境光颜色

### 2.1.3 夜间各方向环境光颜色(整体示意图参考 2.1.1 部分)

#### 2.1.3.1 夜间-右侧环境光 (X+方向)

```
static const float3 NightAmbientMapPosX = float3(0.2, 0.2, 0.2); // 右侧  
(X+方向)
```

夜间右侧 (X+方向) 环境光, 可能模拟月光或人工光源, 配合 NIGHT\_MBIENT\_BOOST 提升可见性。

#### 2.1.3.2 夜间-左侧环境光 (X-方向)

```
static const float3 NightAmbientMapNegX = float3(0.0, 0.0, 0.0); // 左侧  
(X-方向)
```

左侧 (X-方向) 完全无光照, 增强夜间对比度, 由于此处无灯光, 暂无灯光示意图

#### 2.1.3.3 夜间-前/后环境光 (Y+方向)

```
static const float3 NightAmbientMapPosY = float3(0.01, 0.01, 0.01); // 前/  
后 (Y+方向)
```

前/后方向 (Y+) 极弱环境光, 可能用于抑制远处细节

#### 2.1.3.4 夜间-下方环境光 (Y-方向)(模拟镜面反射)

```
static const float3 NightAmbientMapNegY = float3(0.0, 0.0, 0.1); // 下方  
(Y-方向, 可能模拟微弱的水面反光)
```

下方 (Y-方向) 微弱蓝光, 可能模拟水面对月光的反射

#### 2.1.3.5 夜间-上方环境光 (Z+方向)

```
static const float3 NightAmbientMapPosZ = float3(0.06, 0.1, 0.15); // 上方  
(Z+方向, 如月光)
```

上方 (Z+方向) 月光颜色, 偏冷色调 (蓝绿色)

#### 2.1.3.6 夜间-下方环境光 (Z-方向)(模拟漫反射)

```
static const float3 NightAmbientMapNegZ = float3(0.14, 0.14, 0.2); // 下方  
(Z-方向, 可能模拟干燥地面反射)
```

下方 (Z-方向) 地面反光, 较高亮度 (0.2) 模拟湿润地面反射

## 2.1.4 夜间单位环境光



本篇介绍代码控制的光源示意图



本篇介绍代码控制的光源示意图

### 2.1.4.1 右侧单位光源(X+方向)

```
static const float3 NightAmbientPosX = float3(2.0, 2.0, 2.0); // 右侧 (X+  
方向) 高亮度 (可能用于月光或人工光源)
```

右侧 (X+) 强光源 (如路灯等单位光直射), 高亮度 (2.0) 直接照亮特定区域

### 2.1.4.2 左侧单位光源(X-方向)

```
static const float3 NightAmbientNegX = float3(0.2, 0.2, 0.2); // 左侧 (X-  
方向)
```

左侧 (X-) 微弱光源, 避免完全黑暗

### 2.1.4.3 前/后方向单位光源(Y方向)

```
static const float3 NightAmbientPosY = float3(0.1, 0.1, 0.1); // 前/后 (Y+  
方向)  
static const float3 NightAmbientNegY = float3(0.0, 0.0, 0.0); // 下方 (Y-  
方向)
```

前/后方向基础照明, 避免角色或物体完全剪影

### 2.1.4.5 上方单位光源(Z+方向)



---

```
static const float3 NightAmbientPosZ = float3(3.0, 3.0, 3.0); // 上方 (Z+ 方向, 强光源)
```

上方 (Z+) 强光, 亮度极高 (3.0), 主导夜间场景

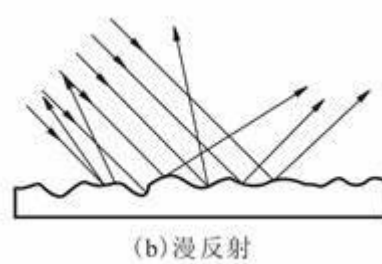
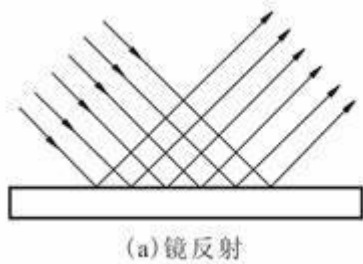
#### 2.1.4.6 下方单位光源(Z-方向)(地面单位光反射)

```
static const float3 NightAmbientNegZ = float3(0.8, 0.8, 0.8); // 下方 (Z- 方向)
```

下方 (Z-) 地面反光, 中等亮度模拟湿润或雪地反射

## 2.2 高光 (Specular)

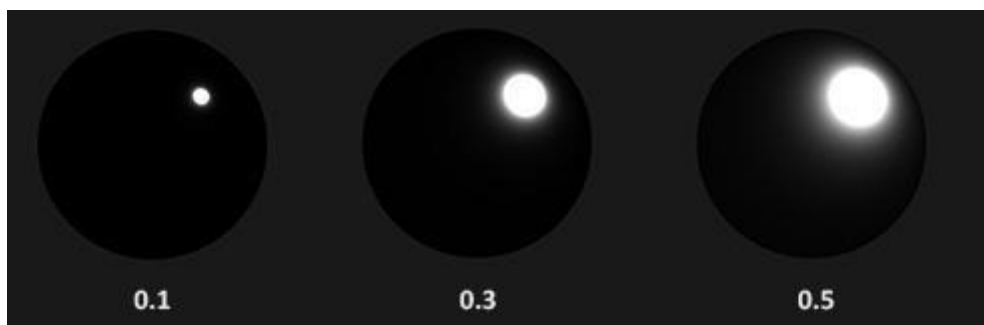
了解该部分之前先来了解一下光照的基本反射方式



### 2.2.1 高光反射的扩散范围

```
static const float SPECULAR_WIDTH = 15.0; // 高光宽度 (控制高光反射的扩散范围)
```

控制高光反射的扩散范围, 值越大高光区域越宽



高光扩散范围示意图

### 2.2.1 高光强度全局缩放

---

```
static const float SPECULAR_MULTIPLIER = 1.0; // 高光强度乘数（整体缩放高光亮度）
```

这个没啥好解释的，理解为： $SPECULAR\_WIDTH * SPECULAR\_MULTIPLIER$ ，用在当某些区域的光照太强或太弱时使用

### 2.2.1 地图元素高光宽度

```
static const float MAP_SPECULAR_WIDTH = 15.0; // 地图材质的高光宽度（可能用于地形或UI元素）
```

地图元素（如 UI 图标）的高光宽度，与场景物体一致，数值建议与  $SPECULAR\_WIDTH$  一致

## 2.3 地形 (Terrain)

### 2.3.1 城市灯光参数

首先认识一下《钢铁雄心 4》中的城市灯光



高纹理-城市灯光



低纹理-城市灯光

#### 2.3.1.1 城市灯光纹理的平铺密度

```
static const float CITY_LIGHTS_TILING = 0.09103; // 城市灯光纹理的平铺频率（值越小，重复次数越多）
```

控制城市灯光纹理的平铺密度，值越小纹理重复次数越多，适合中距离视角下的城市灯光分布



纹理示意图 CITY\_LIGHTS\_TILING 值越大，那么在一定范围内的纹理单位越多

### 2.3.1.2 城市灯光亮度

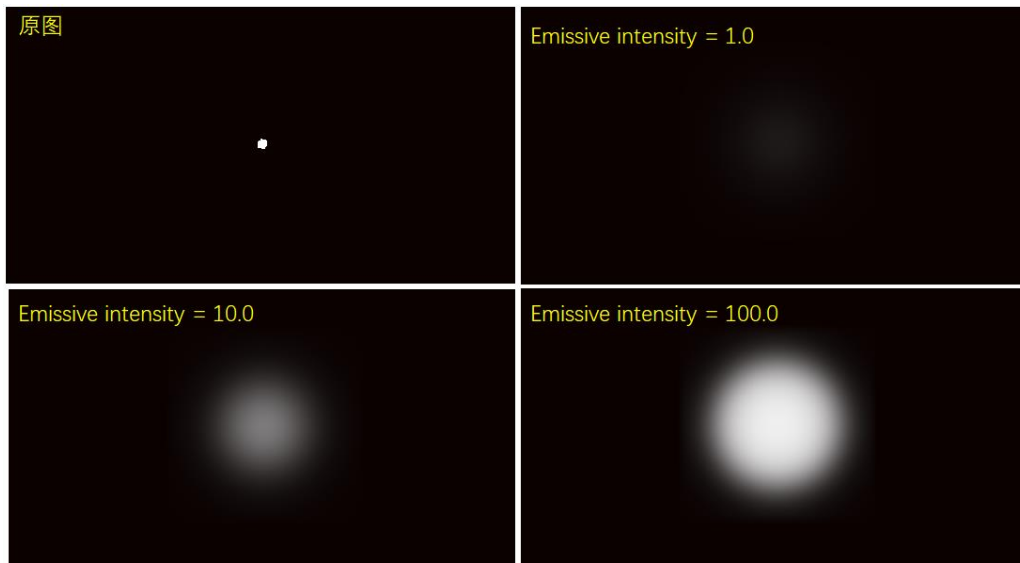
```
static const float CITY_LIGHTS_INTENSITY = 5.5;           // 城市灯光亮度
```

灯光亮度，值越大越明亮，CITY\_LIGHTS\_INTENSITY 值 5.5 使灯光在夜间显著可见，但不过曝

### 2.3.1.3 泛光 (Bloom) 效果强度

```
static const float CITY_LIGHTS_BLOOM_FACTOR = 0.3;       // 城市灯光的泛光  
(Bloom) 效果强度
```

泛光（Bloom）效果强度，值越大光晕越明显

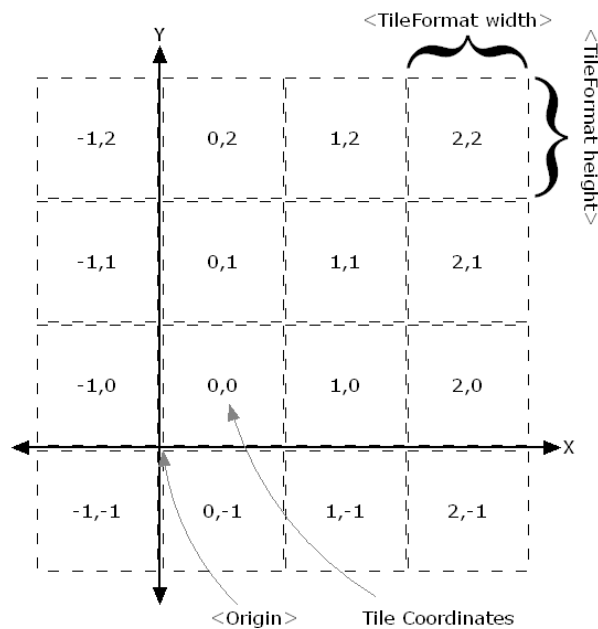


泛光（Bloom）效果强度示意图

## 2.3.2 地形分块参数

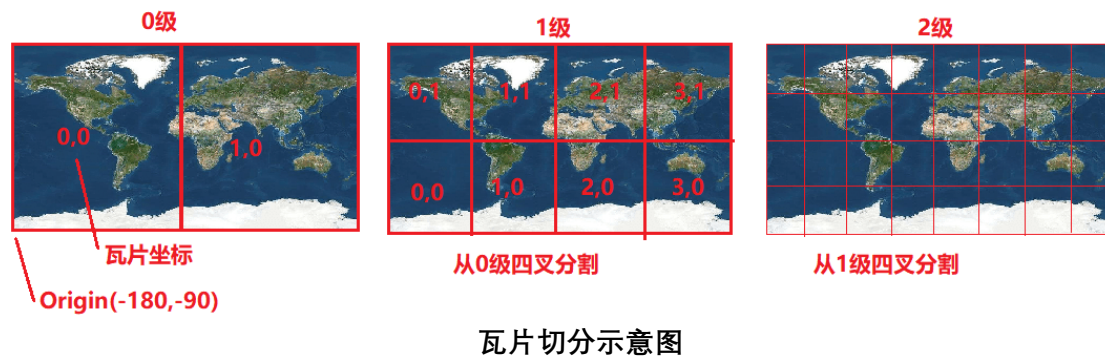
### 2.3.2.1 地图瓦片

地图瓦片是包含了一系列比例尺、一定地图范围内的地图切片文件。



地图瓦片解释图

瓦片地图(TileMap)由一组具有不同比例尺瓦片集(TileSet)组成，每个瓦片由相同大小格式的规则瓦片平铺而成。下一级的瓦片集由上一级的四叉分割而来



### 2.3.2.2 地形瓦片平铺频率

```
static const float TERRAIN_TILE_FREQ = 128.0f; // 地形块的平铺频率
```

地形瓦片的平铺频率，值越大瓦片越小，同时意味着在相同地图大小下瓦片越小数量越多，显存占用越大，128.0 匹配中等分辨率地形纹理

### 2.3.2.2 地图瓦片可见数量

```
static const float MAP_NUM_TILES = 4.0f; // 地图中可见的瓦片数量
```

地图中可见的瓦片数量，控制渲染范围，4.0 用于表示 4x4 瓦片网格

### 2.3.2.3 瓦片的纹理像素数

```
static const float TEXELS_PER_TILE = 512.0f; // 每个瓦片的纹理像素数 (控制纹理分辨率)
```

每个瓦片的纹理像素数，决定纹理细节，512x512 像素为常见中高分辨率。

### 2.3.2.4 地形图集尺寸

```
static const float ATLAS_TEXEL_POW2_EXPONENT = 11.0f;
```

地形图集尺寸为  $2^{11}=2048 \times 2048$  像素，优化显存占用

### 2.3.2.5 水面裁剪

水面裁剪是 3D 渲染中的一种技术，用于动态控制水面（如海洋、湖泊）的显示范围，通常基于**地形高度**(低于某个阈值的地形区域会被视为水域，自动渲染水面)和**摄像机距离**(根据摄像机远近调整水面细节的显示或隐藏，以优化性能)

```
static const float TERRAIN_WATER_CLIP_HEIGHT = 3.0f; // 地形水面裁剪高度  
                (低于此高度的区域视为水域)  
static const float TERRAIN_WATER_CLIP_CAM_HI = 700.0f; // 高视角下水面裁剪的摄像机高度阈值  
static const float TERRAIN_WATER_CLIP_CAM_LO = 50.0f; // 低视角下水面裁剪的摄像机高度阈值
```

3.0f 为水面裁剪高度，低于此值的地形视为水域，3.0f 单位高度匹配游戏世界比例，700f 启用水面裁剪，50f 禁用水面裁剪，700f~50f 之间保持上一次的裁剪状态

### 2.3.3 泥泞参数

控制泥泞材质在场景中的视觉突出程度

```
static const float MUD_TILING = 0.09; // 泥泞纹理的平铺频率  
static const float MUD_NORMAL_CUTOFF = 10.982; // 泥泞法线贴图的截止值  
static const float MUD_STRENGTHEN = 1.0; // 泥泞效果的强度
```

MUD\_STRENGTHEN 所控制的效果强度会综合影响以下多个渲染方面

视觉元素	描述	强度影响
颜色混合	泥泞颜色与基础地形的混合程度	值越大，泥泞色越覆盖原始地表颜色
纹理细节	泥泞特有的污渍、脚印、车辙等细节纹理的可见度	值越大，细节越清晰
法线变形	模拟泥泞表面的凹凸不平（通过法线贴图实现）	值越大，凹凸感越强

高光反射	湿润泥泞表面的反光特性	值越大，潮湿反光效果越明显
高度混合	泥泞区域与干燥区域的过渡平滑度	值越大，过渡边界越锐利
深度效果	模拟陷入泥泞的凹陷感	值越大，“下陷”视觉效果越深

MUD\_TILING 控制泥泞纹理的重复密度，值越小单个泥泞斑块越大，值越大泥泞纹理更密集

MUD\_NORMAL\_CUTOFF 控制法线变形的敏感度，值越大只有更陡峭的地形才显示泥泞法线效果

MUD\_STRENGTHEN 所有效果的全局放大器，增强时同时放大颜色混合度，纹理细节，法线变形，高光反射

### 2.3.4 雪地透明度参数

```
static const float SNOW_OPACITY_MIN = 0.95f; // 雪的最小透明度（完全不透明时接近1.0）
```

```
static const float SNOW_OPACITY_MAX = 0.2f; // 雪的最大透明度（值越小越透明）
```

控制积雪最厚区域的视觉不透明度，0.0（完全透明）到 1.0（完全不透明），同时也可表示为雪地厚度，越不透明雪地越厚

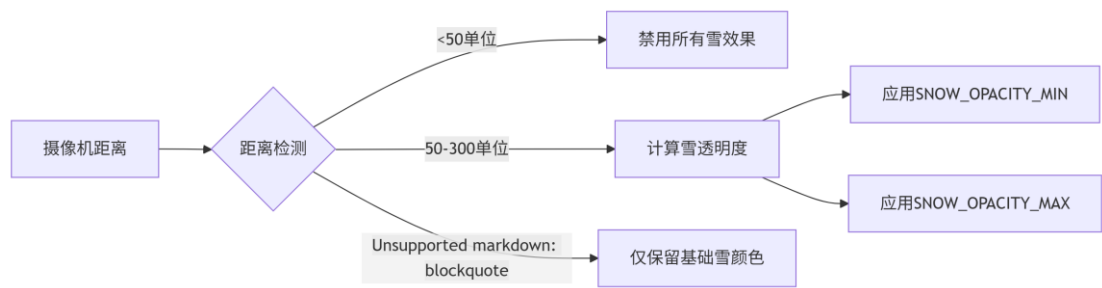
### 2.3.5 效果可见的摄像机距离

```
static const float SNOW_CAM_MIN = 50.0f; // 雪效果可见的最小摄像机距离
static const float SNOW_CAM_MAX = 300.0f; // 雪效果可见的最大摄像机距离
static const float MUD_CAM_MIN = 50.0f; // 泥泞效果可见的最小摄像机距离
static const float MUD_CAM_MAX = 300.0f; // 泥泞效果可见的最大摄像机距离
static const float ICE_CAM_MIN = 100.0f; // 冰效果可见的最小摄像机距离
static const float ICE_CAM_MAX = 350.0f; // 冰效果可见的最大摄像机距离
```

这里仅使用雪地效果来举例，摄像机距离小于此值(SNOW\_CAM\_MIN)时**完全禁用雪效果**，避免近景穿帮（如雪浮空），摄像机距离大于此值时**简化雪效果**

距离范围	渲染效果
<50 单位	完全无雪
50-150 单位	完整效果（粒子+法线）

	150-300 单位	中效（仅颜色+基础法线）
	>300 单位	极简（仅基础颜色）



效果渲染逻辑示意图

修改建议(参考)

修改目标	调整参数	推荐值范围
增强厚雪质感	↑ SNOW_OPACITY_MIN	0.97~0.99
增强薄雪透地效果	↓ SNOW_OPACITY_MAX	0.1~0.15
扩展近景雪显示范围	↓ SNOW_CAM_MIN	30~40
提升远景雪细节	↑ SNOW_CAM_MAX	400~500
性能优化（低端设备）	↓ SNOW_CAM_MAX	200~250

2.3.6 效果高级参数

2.3.6.1 雪地效果出现阈值

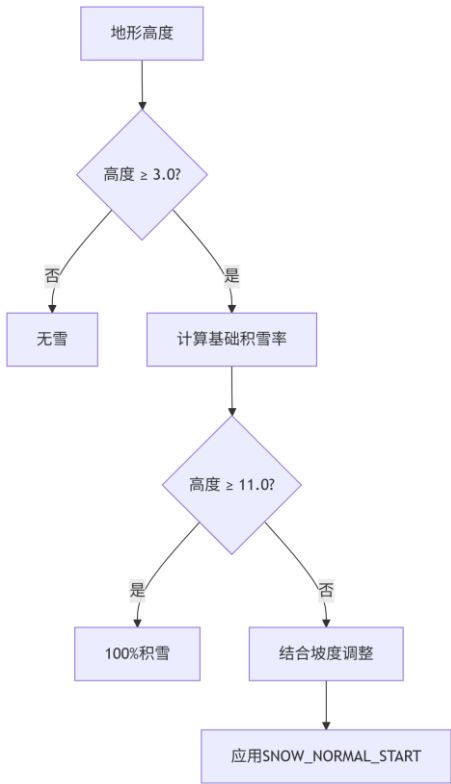
```
static const float SNOW_START_HEIGHT = 3.0f;           // 雪开始出现的高度阈值
static const float SNOW_RIDGE_START_HEIGHT = 11.0f;    // 雪在山脊处开始出现的高度阈值
static const float SNOW_NORMAL_START = 0.7f;           // 基于法线方向的雪覆盖起始值（控制坡度对积雪的影响）
```

SNOW\_START\_HEIGHT = 3.0f 这是世界坐标系中的 Y 轴高度值，当地形高度大于等于此值时，雪效果开始加载，雪效果由多个参数共同控制，见下表格

参数	值	作用	与起始高度的关系
SNOW_START_HEIGHT	3.0f	雪出现的最低高度	基准线



SNOW_RIDGE_START_HEIGHT	11.0f	山脊完全积雪的高度	当高度 $\geq 11.0$ 时，积雪覆盖率为100%
SNOW_NORMAL_START	0.7f	坡度对积雪的影响	陡坡（法线 Y<0.7）减少积雪



雪效果加载逻辑图

地形高度	坡度	积雪覆盖率	视觉效果
2.5f	平缓	0%	无雪，裸露地表
5.0f	平缓	~25%	斑驳积雪
5.0f	陡峭	~10%	少量雪迹
8.0f	平缓	~62.5%	大部分覆盖
12.0f	陡峭	100%	完全积雪

2.3.6.2 雪地颜色

```
static const float3 SNOW_COLOR = float3(0.46, 0.48, 0.69); // 雪的基础颜色（冷色调）
static const float3 SNOW_WATER_COLOR = float3(0.3, 0.6, 1.0); // 雪与水面交界处的颜色
```

雪(SNOW\_COLOR)的基础颜色，冷色调（蓝灰色）模拟真实积雪，雪与水(SNOW\_WATER\_COLOR)交界处的颜色，偏蓝模拟融雪效果

### 2.3.6.3 雪的堆积强度

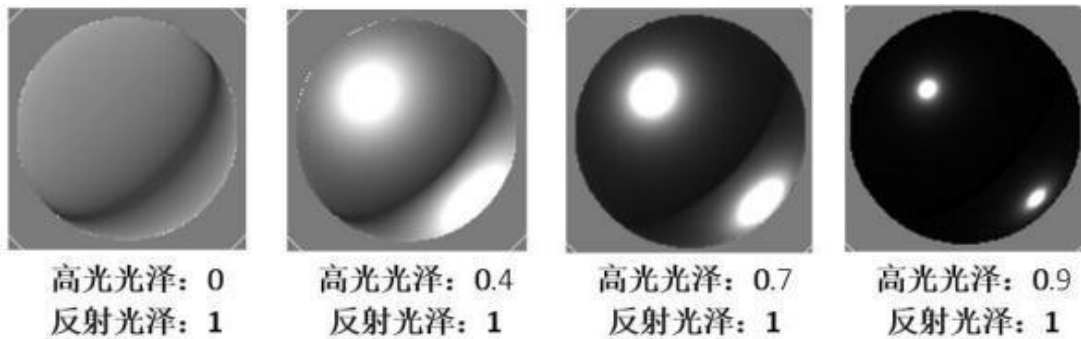
```
static const float SNOW_CLIFFS = 5.0f; // 雪在悬崖处的覆盖强
```

悬崖处雪的堆积强度，值越大积雪越厚。

### 2.3.6.4 雪的高光光泽

```
static const float SNOW_SPEC_GLOSS_MULT = 0.2f; // 雪的高光光泽度乘数（值越小越粗糙）
```

雪的高光光泽度乘数，0.2 表示表面较粗糙



高光光泽示意图

### 2.3.6.4 雪纹理的平铺频率

```
static const float SNOW_TILING = 0.05f; // 雪纹理的平铺频率
```

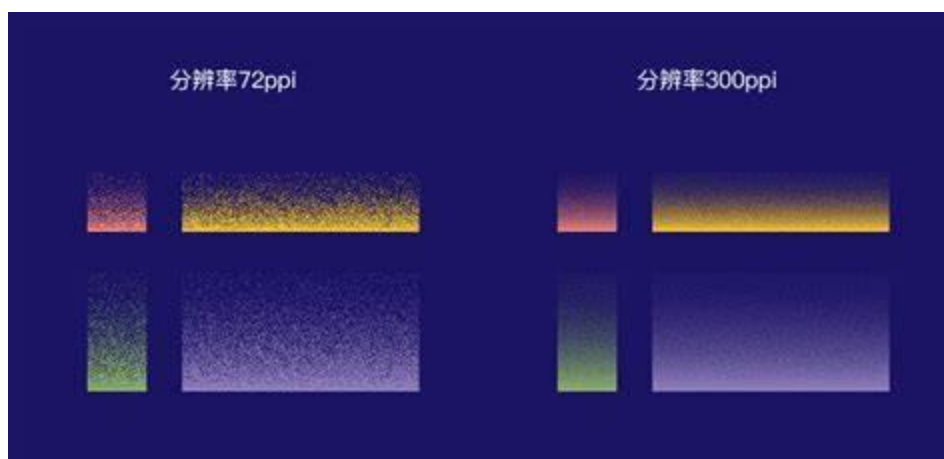
雪纹理的平铺频率，值越小纹理重复越少

### 2.3.6.4 噪点纹理的平铺频率

```
static const float SNOW_NOISE_TILING = 0.06f; // 雪噪点纹理的平铺频率（用于细节增强）
```

```
static const float SNOW_ICE_NOISE_TILING = 0.0625f; // 冰面噪点纹理的平铺频率
```

噪点纹理的平铺频率，控制细节密度



噪点纹理频率示意图

### 2.3.6.5 霜冻效果最小强度

```
static const float SNOW_FROST_MIN_EFFECT = 0.4f; // 霜冻效果的最小强度
```

霜冻效果的最小强度，低于此值不显示霜冻。



交界处即为霜冻

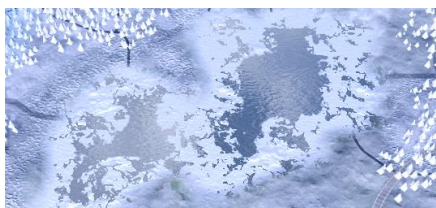
### 2.3.6.6 冰面参数

```
static const float3 ICE_COLOR = float3(0.5f, 0.6f, 0.9f); // 冰的基础颜色
```

```
static const float ICE_NOISE_TILING = 0.1f; // 冰噪点纹理的平铺频率
```

ICE\_COLOR 定义冰的基础颜色，偏蓝绿色模拟透明冰层

ICE\_NOISE\_TILING 定义冰噪点纹理的平铺频率，较稀疏的细节

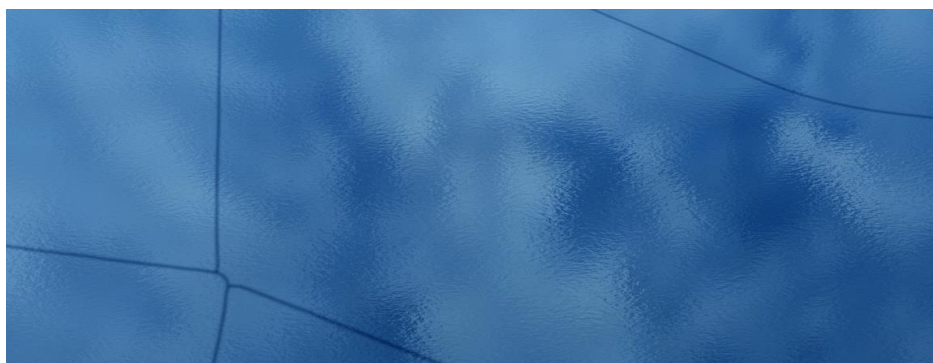


图为冰面(左)和水面(右)

### 2.3.6.7 水面参数

```
static const float WATER_COLOR_LIGHTNESS = 0.5;           // 水面的亮度 (0~1  
范围)  
static const float WATER_RIPPLE_EFFECT = 0.0025;         // 水面涟漪效果的强  
度
```

水面的整体亮度, 0.5 为中等亮度, 0.0025 表示微小涟漪



图为钢 4 中的水面

### 2.3.6.8 颜色叠加参数

```
static const float COLORMAP_OVERLAY_STRENGTH = 0.75f;     // 主颜色贴图  
的叠加强度  
static const float COLORMAP_MUD_OVERLAY_STRENGTH = 0.5f; // 泥泞颜色  
贴图的叠加强度
```

主颜色贴图的叠加透明度, 0.75 表示较强覆盖, 泥泞颜色贴图的叠加透明度, 0.5 中等强度

### 2.3.6.9 假立方体贴图颜色

假立方体就是利用单一颜色常量替代真实环境数据，其具有零采样开销的优势，但代价是牺牲真实性换取性能，以下表格举例说明假立方体贴图的优劣势

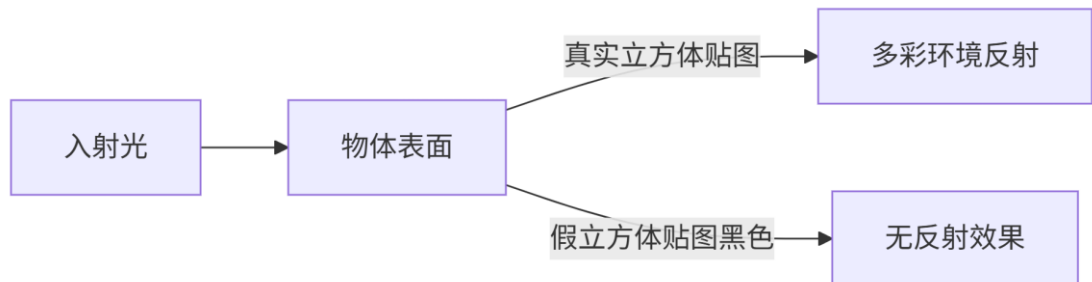
特性	真实立方体贴图	假立方体贴图
反射质量	高动态范围，精确环境反射	平坦的单色反射
性能开销	高（多次采样+插值）	极低（仅常量读取）
内存占用	高（6张 512x512 纹理≈8MB）	0（仅 12 字节常量）
典型应用场景	角色金属盔甲、水面反射	远景小物体、移动设备、风格化渲染

```
static const float3 FAKE_CUBEMAP_COLOR = float3(0.0f, 0.0f, 0.0f); //
```

用于模拟环境反射的假颜色

代码中 FAKE\_CUBEMAP\_COLOR = 全黑，约等于未启用，在后续的模组中可以尝试启用

效果说明：



设计意图：

完全禁用环境反射效果，最大程度节省性能开销

适用于：低优先级物体（如远景树木），非金属材质（木材、岩石），风格化渲染（卡通/像素风）

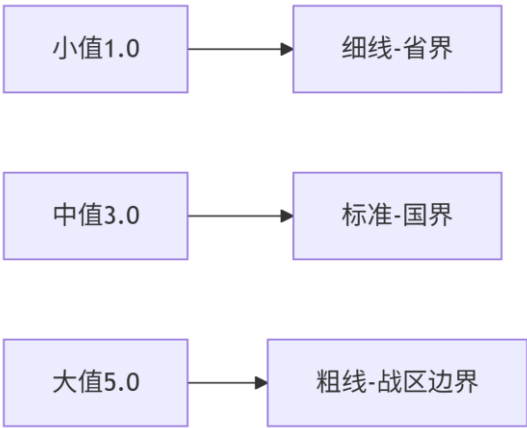
### 2.3.6.10 冬季参数(未定义)(未说明)

```
// 冬季参数（实际值在 defines.lua 中定义）：  
// MILD_WINTER_VALUE = ###  
// NORMAL_WINTER_VALUE = ##  
// SEVERE_WINTER_VALUE = ###
```

2.3.6.11 边框参数

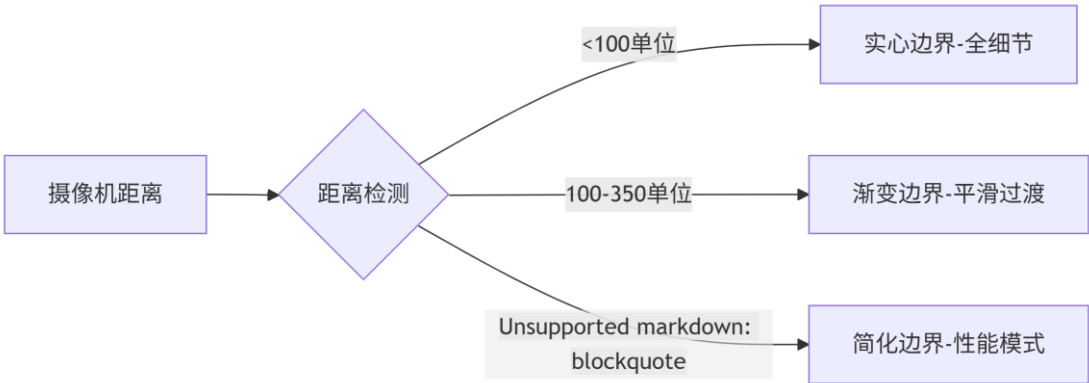
```
static const float BORDER_TILE = 0.4f; // 边框瓦片的平铺频率
// BORDER_WIDTH = ### (实际值在 defines.lua 中定义)
```

边框参数在钢 4 中的应用甚广，详细见下表格



边框应用示意图

在 P 社游戏中，这些参数常通过 defines.lua 动态调整，配合 DLC 或 MOD 实现不同历史时期的边界表现



距离渐变效果逻辑

2.3.6.12 树木季节过渡参数

```
static const float TREE_SEASON_MIN = 0.5f; // 树木季节变化的最小插值阈值
static const float TREE_SEASON_FADE_TWEAK = 2.5f; // 季节过渡的平滑度调整
```

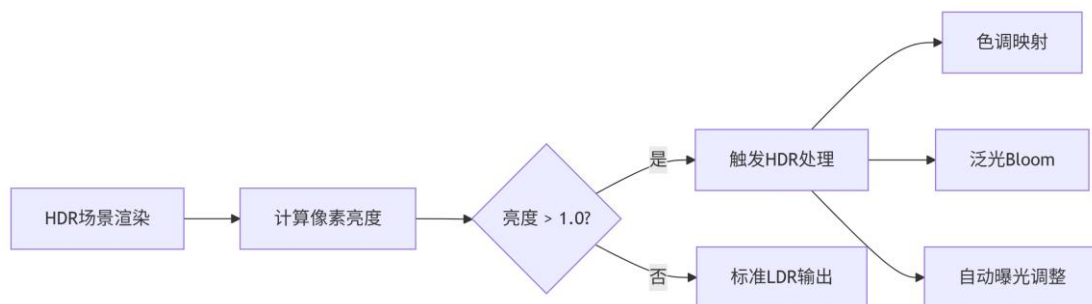
## 2.4 HDR 参数

```
static const float3 LUMINANCE_VECTOR = float3(0.2125f, 0.7154f, 0.0721f); // HDR 亮度计算权重 (RGB 通道的贡献比例)
```

超越传统 0-1 范围的亮度表示能力，支持真实世界的高对比度场景

公式：亮度  $L = R \times 0.2125 + G \times 0.7154 + B \times 0.0721$

通道	权重	人眼敏感度	物理依据
红(R)	0.2125	中等	锥细胞对红光响应较弱
绿(G)	0.7154	最高	人眼 55-60%锥细胞感应绿色
蓝(B)	0.0721	最低	短波光线在大气中散射最强



HDR 使用功能示意图

常见 HDR 使用标准：

标准名称	红	绿	蓝	应用场景
CIE 1931	0.2127	0.7152	0.0722	色彩科学基础
BT.709 (HDTV)	0.2126	0.7152	0.0722	高清视频
BT.2020 (UHD)	0.2627	0.678	0.0593	超高清显示
游戏渲染	0.2125	0.7154	0.0721	游戏渲染优化版

依据天气可对 HDR 参数做出一下调整：

天气	亮度修正	视觉表现
晴天	$lum *= 1.5$	增强高光，锐利阴影
阴天	$lum *= 0.7$	降低对比度，柔和画面
暴风雨	$lum *= 0.4 + lightning$	动态闪电高光冲击

## 2.5 树木参数

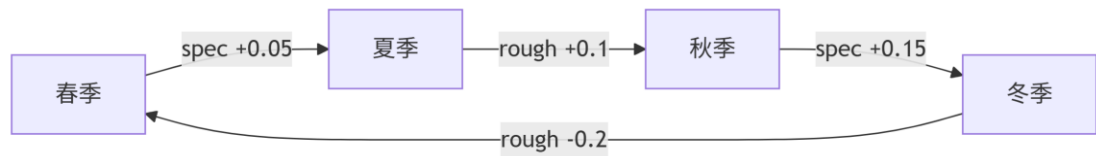
```
static const float TREE_SPECULAR = 0.1f;    // 树木材质的高光强度
static const float TREE_ROUGHNESS = 0.6f;    // 树木材质的粗糙度（值越大表面越粗糙）
```

参数说明：

参数	值	物理意义	视觉效果
TREE_SPECULAR	0.1	高光反射强度	微弱的湿润反光
TREE_ROUGHNESS	0.6	表面粗糙度	漫反射主导的磨砂质感

季节调整建议：

季节	参数修改建议值	视觉效果
夏季	specular=0.1 roughness=0.6	树叶油亮反光 树皮粗糙质感
秋季	specular=0.15 roughness=0.7	增强金叶反光 干燥树皮纹理
冬季	specular=0.3 roughness=0.4	雪霜强反光 结冰平滑表面
雨季	specular=0.25 roughness=0.3	湿润高光 水流光滑效果



季节变化调整

## 2.6 水面参数

制作此类 DLC 请认真阅读，如动态海平面变化（气候变化 DLC）

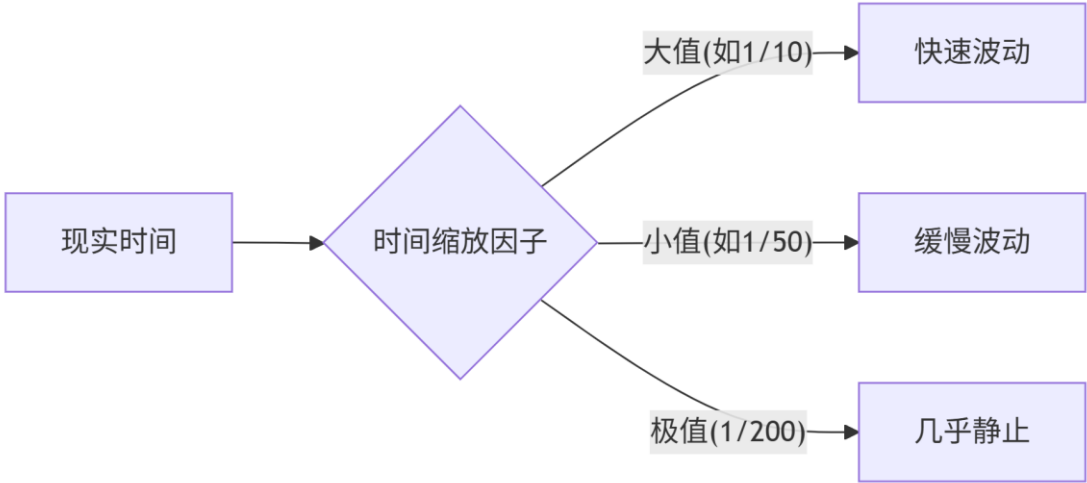
### 2.6.1 水面波动的时间缩放因子

```
static const float WATER_TIME_SCALE = 1.0f / 50.0f;    // 水面波动的
时间缩放因子（值越小波动越慢）
```

控制水面波动动画速度的调节系数,  $1/50 = 0.02$ （即现实 1 秒对应游戏中的 0.02 秒波动进度）



动画速度 = 真实时间 × WATER\_TIME\_SCALE



时间因子原理

数值所对应的视觉效果：

缩放因子值	波动频率	适用场景	性能影响
1/20	高频快速	暴风雨海面	GPU 负载高 ↑↑
1/50 (默认)	中频缓波	平静湖泊/河流	平衡负载 ↔
1/100	低频慢波	大型水库	GPU 负载低 ↓
0	完全静止	冰面效果	无波动计算

2.6.2 水面高级参数

```
static const float WATER_HEIGHT = 9.5f; // 水面的基准高度
static const float WATER_HEIGHT_RECP = 1.0f / WATER_HEIGHT; // 水面高度的倒数（用于优化计算）
static const float WATER_HEIGHT_RECP_SQUARED = WATER_HEIGHT_RECP * WATER_HEIGHT_RECP; // 倒数平方（进一步优化）
```

该部分的主要作用是减少水面对 GPU 的负载消耗

水面基准高度（WATER\_HEIGHT）：核心作用是给水面一个基础的高度，以便水面可以使用游戏内的高度参数

水面高度倒数（WATER\_HEIGHT\_RECIP）：核心用途是避免重复除法运算的优化手段

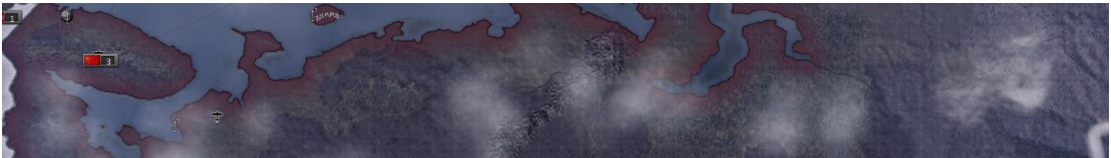
水面高度倒数平方（WATER\_HEIGHT\_RECIP\_SQUARED）：进一步优化计算，核心是优化水里面和水底的光效果计算，使其减小对 GPU 的占用量

## 2.7 雾效

### 2.7.1 雾气(天气类型)

```
static const float3 FOG_COLOR = float3(0.12, 0.28, 0.6); // 雾的颜色（蓝灰色调）
static const float FOG_BEGIN = 1.0f; // 雾效起始距离（摄像机近裁剪平面）
static const float FOG_END = 150.0f; // 雾效结束距离（超过此距离完全被雾覆盖）
static const float FOG_MAX = 0.35f; // 雾的最大密度（0~1 范围）
```

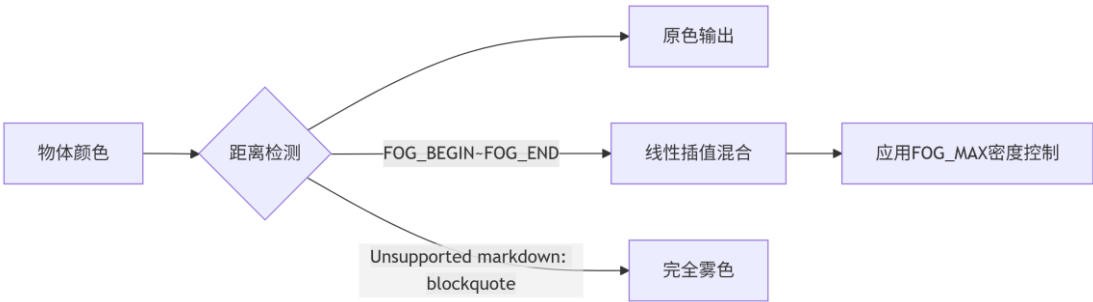
如图所示，图中的雾效属于天气类型的雾



雾效(天气类型)示意图

参数	单位	数学意义	视觉表现
FOG_BEGIN	游戏单位	雾效开始影响的最小距离	摄像机近处无雾
FOG_END	游戏单位	雾效完全覆盖的最大距离	超出此距离物体完全被雾遮挡
FOG_MAX	0-1 范围	雾的极限不透明度	值越大雾越浓

摄像机距离在 FOG\_BEGIN 和 FOG\_END 之间时，才会显示雾气

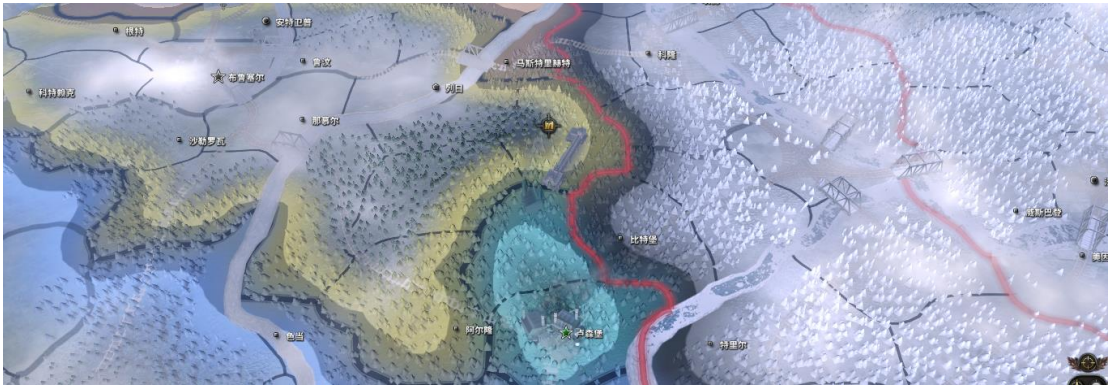


雾效渲染管线逻辑

在设计包含天气系统的 DLC 中可参考下表进行数值调整

参数	当前值	设计考量	可调范围建议
FOG_BEGIN	1	避免近处穿帮，匹配近裁剪平面	0.5-5.0
FOG_END	150	平衡视野与性能（中距离场景）	50-500
FOG_MAX	0.35	保持 35%物体可见度，避免完全遮挡	0.1-0.7

2.7.1 雾气(战争迷雾)



战争迷雾 上图为存在战争迷雾 下图为不存在战争迷雾



---

```
static const float FOW_MAX = 0.5f;           // 战争迷雾的最大不透明度
static const float FOW_CAMERA_MIN = 200;      // 战争迷雾可见的最小摄像机距离
static const float FOW_CAMERA_MAX = 500;      // 战争迷雾可见的最大摄像机距离
```