**ShaderForge插件介绍详解**

**https://blog.csdn.net/wwlcsdn000/article/details/78847745**

ShaderForge简介

ShaderForge是适用于Unity，基于节点的着色器编辑器 。它可让您更轻松地创建自己的着色器 ，而不需要使用代码。开始以之前我想你已经下载好了插件，如果没有的话可以从下方的链接获取。

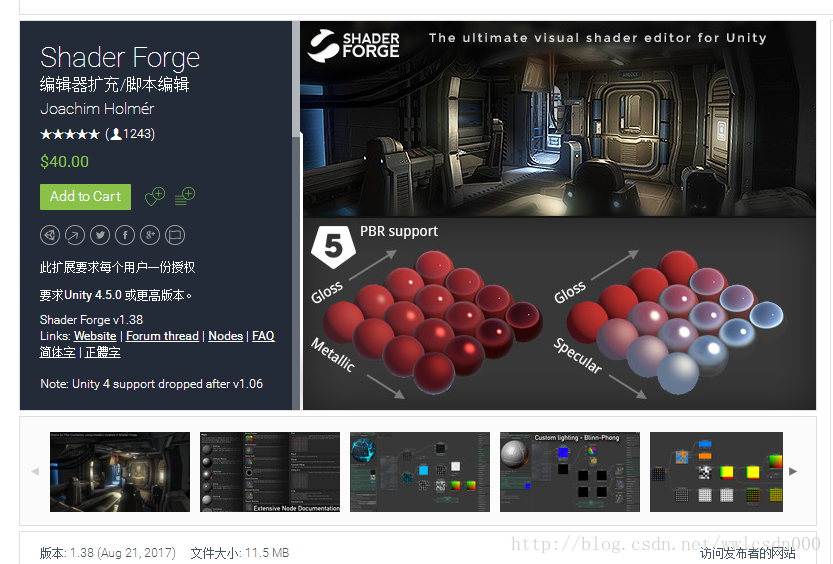
前言

1.官网介绍

官网:http://www.acegikmo.com/shaderforge/（包括文档、教程、论坛线程，链接和更多！）

商店链接 ：https://www.assetstore.unity3d.com/cn/#!/content/14147

---------------------



2.Shader

开始使用插件之前，我们首先来看下Shader是什么东西。Shader翻译过来是着色器。

着色器用于绘制游戏中的物理资源。它们用于确定当前渲染物体的表面颜色，通过使用纹理、光源、光色、网格数据等信息计算该处的颜色。

Unity内置了几个着色器，有些专用于粒子和效果，有些则更通用，例如Unity5的标准PBR着色器。

为什么选择ShaderForge?

Unity中的默认着色器固然不错，但如果您要在游戏中设计出更

加与众不同的外观，或者想达到一些非常特别效果。如果您有这种想法，那么您就需要编写自E的着色器。但是，编写自己的着

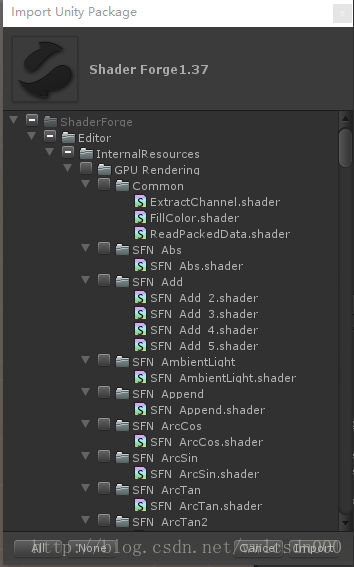
色器很困难，而且学起来也很难，对于那些不是程序员出身的人来说更是如此。这就是ShaderForge发挥作用的地方。ShaderForge不需要您知道如何编写代码就能让您创建着色器！

接下来让我们开始进入ShaderForge的世界。

ShaderForge基本介绍

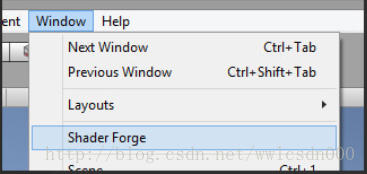
1.导入到Unity项目中

直接拖入到Unity的项目视图面板或者通过菜单导入，不做过多解释。

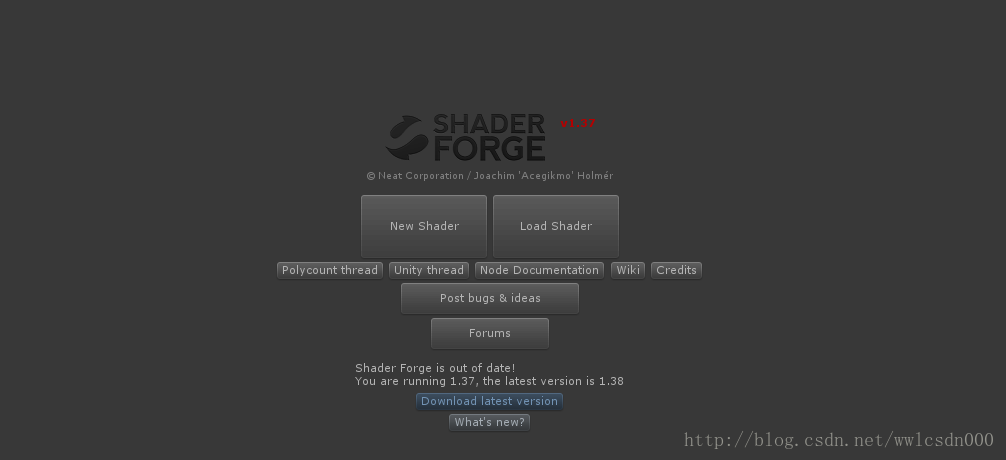


### 2.界面基本介绍

打开Shader Forge操作面板



打开之后的面板



New Shader：新建一个Shader文件

Load Shader：导入已有的Shader文件

PolyCount thread：PolyCount页面

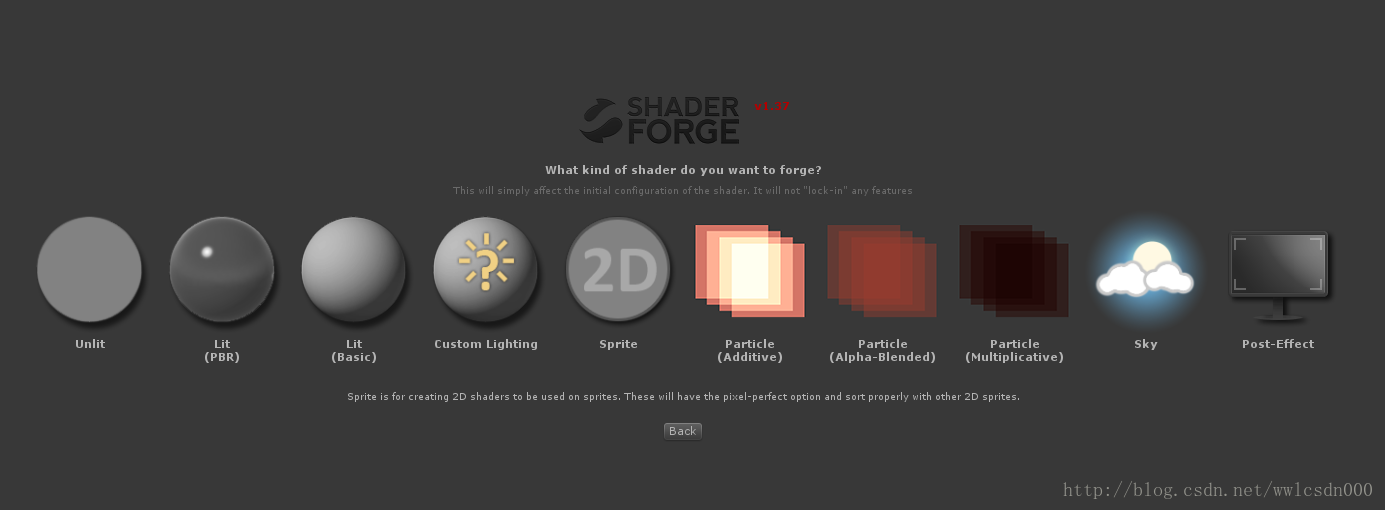
Unity：Unity Forum的页面，有问题可以去里面提问。

Node Documentation：Shader Forge中的节点说明文档（重要）

Wiki：Shader Forge的维基文档

---------------------

选择新建Shader



Unlit：无光照的Shader模板，选择此模板，使用该Shader的材质将不受光照影响。

Lit(PBR)：符合Unity PBR的Shader模板，采用基于物理的光照模型。

Lit(Basic)：使用基础光照模型的Shader模板。

Custom Lighting：自定义光照Shader，在此Shader模板中实现自己的光照模型。

Sprite：2D精灵Shader模板，基于2D。

Particle(Additive)：主要应用在发光效果的粒子Shader。

Particle(Alpha-Blended)：主要应用在细碎效果的粒子Shader，如风沙。

Particle(Multiplicative)：主要应用在昏暗效果的粒子Shader，如黑烟。

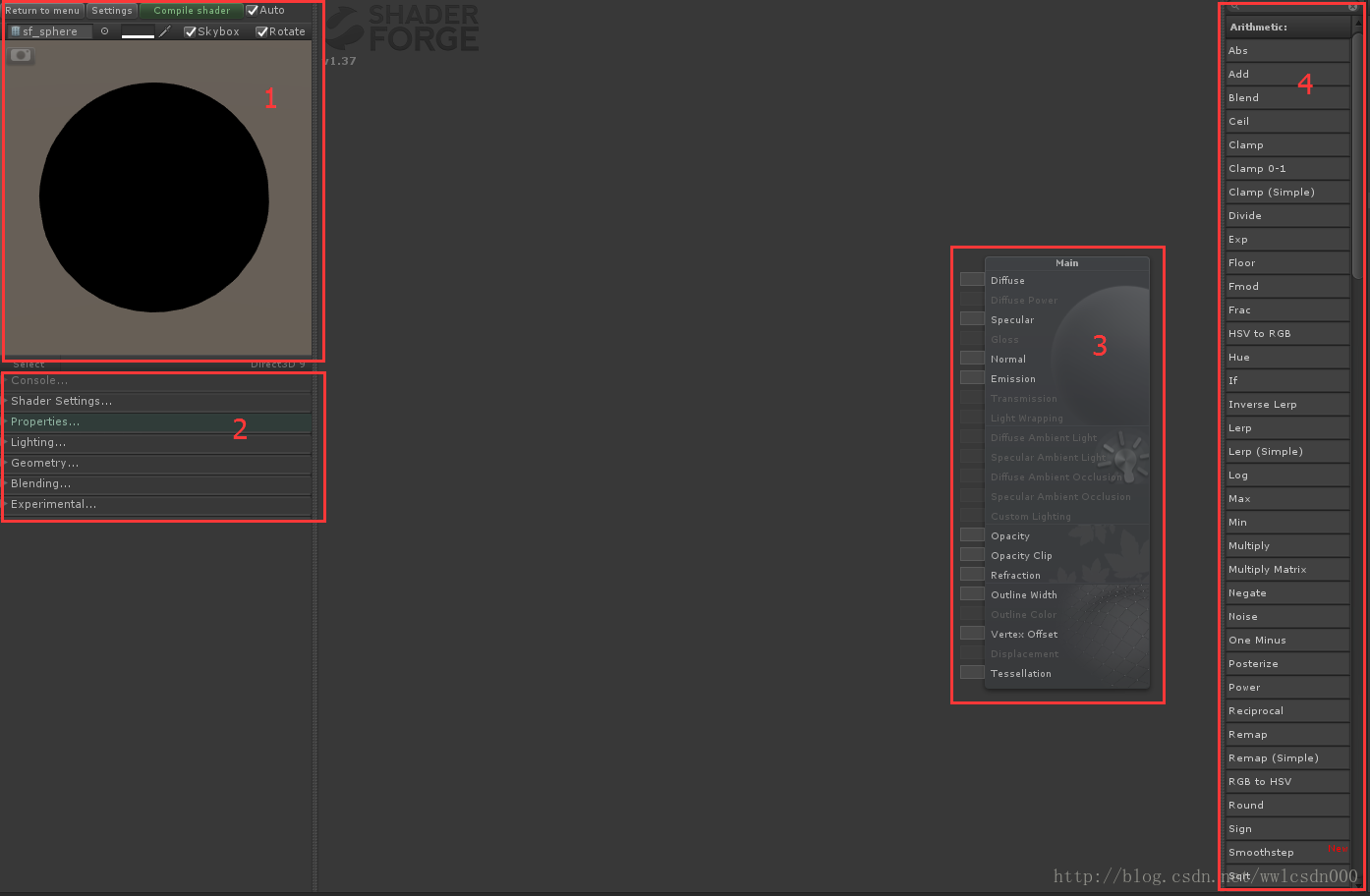
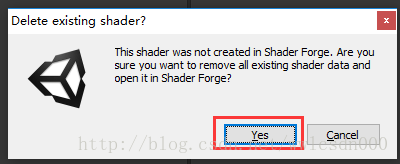
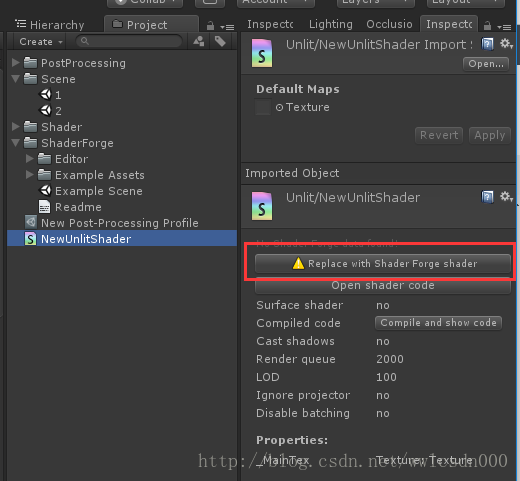
Sky：天空盒。

Post-Effect：屏幕空间后期处理Shader

选择无光照进入，中间需要保存Shader文件路径自行选择即可。

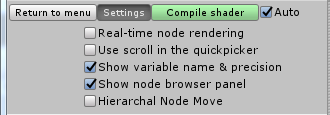
也可以自己手动创建一个Shader文件，然后使用ShaderForge打开。

---------------------



1.3D预览，用于查看着色器运行时的外观

* Return to menu：返回界面1。
* Settings：Shader Forge界面的一些设置。



Real-time node rendering：实时节点渲染，是否开启看需求。

Using scroll in the quickpicker：在快速拾取中使用滚动。

Show variable name & precision：显示变量名和精度，看需求开启。

Show node browser panel：显示节点浏览面板，是否开启看个人习惯。

Hierarchal Node Move：层次节点移动。

Compile shader：编译为.shader文件，编译成功按钮为绿色，编译不成功or改动后按钮为红色。

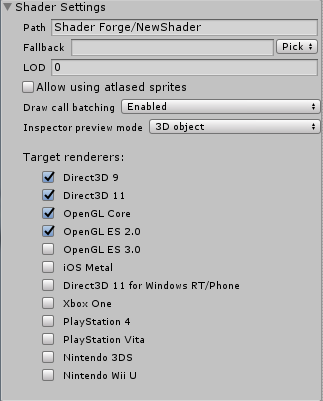
Auto：是否自动编译shader，建议选中。 Skybox：选择天空盒，在不受天空影响的模板中被冻结。

Rotate：选中后模型自动旋转。 左上角摄像机按钮：保存当前的连接节点图。

预览面板也支持缩放和旋转视图。

2.着色器设置，用于配置着色器的渲染方式

Shader Settings标签



Path：文件路径，可修改。

Fallback：在运行平台硬件不支持该shader的情况下，选择替代shader。

Allow using atlased sprites：2D开发使用，3D默认即可。

Draw call batching：批处理绘制调用，默认Enable。

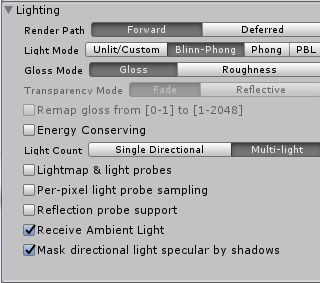
Inspector preview mode：3D/2D/sky，制作的什么类型选什么。

Target renderers：目标渲染器，如无特殊需求，默认。

Properties标签



可以拖动改变显示位置，同时在Inspector面板更新。



Render Path：渲染路径，正向渲染或延迟渲染，移动平台选择正向渲染即可。

Light Mode：光照模型，对应不同的Shader模板。分别是无光照（UnLit模板）、Blinn-Phong光照模> > 型，Phong光照模型，PBL（基于物理光照，Lit（PBR）模板）。

Gloss Mode：使用光泽度还是粗糙度模型，根据自己需要。

Transparent Mode：透明度模式，淡出or反射。

Remap gloss from [0,1] to [1,2048]：重映射光泽度从[0,1]到[1,2048]。

Enery Conversing：能量守恒，PBR需要勾选。

Lighting Count：光源数量，单一定向光（太阳光）or多重光源

Lightmap & light probes：灯光贴图&灯光探针。

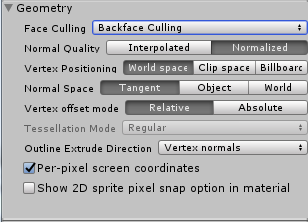
Per-pixel light probe sampling：逐像素光照探针采样。

Reflection probe support：支持反射探针。

Receive Ambient Light：是否接受环境光。

Mask directional light specular by shadows：定向光高光是否被阴影遮挡。

Geometry标签



Face Culling：面剔除，默认是剔除背面，可选剔除正面，也选择双面渲染。

Normal Quality：法线质量，插值或归一化，默认使用归一化。

Vertex Positioning：顶点位置，世界空间or裁剪空间or广告牌（法线面向摄像机）。

Normal Space：法线空间选择，切线or建模or世界，默认切线空间。

Vertex offset mode：顶点偏移模式，相对偏移or绝对偏移。基于模型空间的顶点偏移叫做相对偏移，世界空间的顶点偏移叫做绝对偏移。

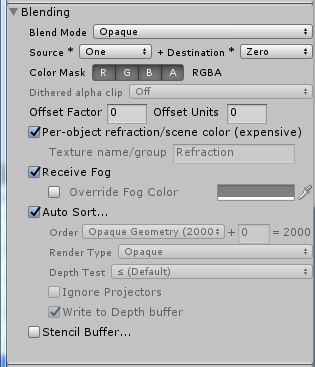
Tessellation Mode：曲面细分模式，一般不需要用到曲面细分。

Outline Extrude Direction：描边挤出方向，一般都是沿着顶点法线方向。

Per-pixel screen coodinates：逐像素屏幕坐标。

Show 2D sprite pixel snap option in material：在材质中显示2D精灵像素单元设置，3D开发忽略。

Blending标签



Blend Mode:混合模式，如果是不透明的就选择Opaque，其余根据需要选择。

Source\*\_+Destination\*\_：源\*\_+目标\*\_，决定了源图像与目标图像以哪种方式混合。

Color Mask：颜色遮罩，需要过滤的通道取消选择。

Dithered alpha clip：抖动透明裁剪，开or关。

Offset Factor：偏移因子。

Offset Units：偏移单位。

Per-object refraction/scene color (expensive):

Receive Fog：应用雾。

Auto Sort：自动排序，如需手动改变排序，在Order中改变数值。数值越小，在渲染队列越靠前。

Depth Test：深度测试。

Ignore Projectors：忽略投影器。

Write to Depth buffer：是否写入深度缓存。

Stencil Buffer：是否开启模板缓存。

3.输出通道

Diffuse：漫反射

Diffuse Power：漫反射力道，相当耗能

Metallic：金属质感，值越大时漫反射越小

Gloss：光泽，值越高光泽度越高，1为极值，物极必反

Emisson：自发光

Transmission：透射，决定有多少光线可以透过材质

Light Wrapping：环绕光，让光线或某个通道的值环绕对象，可做出皮肤的效果

Diffuse Ambient Light：漫反射环境光，添加光线或环境光或立体贴图，不用自发光或光源，也能显亮

Specular Ambient Light：镜面反射环境光，可映出环境的贴图？？？

Diffuse Ambient Occlusion：漫反射环境光遮罩，抑制间接漫反射光线

Specular Ambient Occlusion：镜面反射环境光遮罩，抑制间接镜面反射光线

Custom Lighting：自定义照明，被光照射时，自定义照射效果

Opacity:不透明度，控制最终像素的透明度

Opacity Clip：不透明度剪切，根据阿尔法通道剪切不透明区域

Refraction：反射，用于折射背景像素的屏幕空间 UV 偏移。

请确保在使用前将 Opacity（不透明度）设置为低于 1 的值，以便折射效果可见

Outline Width：描边宽度

Vertex Offset：顶点偏移，插入每个顶点所需偏移量的 XYZ 坐标

DX11 Displacement：DX11 位移，此选项和 Vertex Offset（顶点偏移）的作用非常类似，但是与

DX11 曲面细分一起使用。

（请注意，DirectX 仅适用于 Windows，需要 DX11 GPU 且必须在 Unity 设置中启用）

DX11 Tessellation：DX11 曲面细分，此选项控制您要将三角形分割为多少个细分。

（请注意，DirectX 仅适用于 Windows，需要 DX11 GPU 且必须在 Unity 设置中启用）

4.节点浏览器，用于查找所需的全部节点

加减乘除：Add、Subtract、Multiply、Divide

乘方、平方根、对数：Power、Sqrt、Log

绝对值：Abs

Sign：取正负号及0

Ceil：进位取整，输出时将其输入向上舍入为最近整数

Round：四舍五入

Floor：舍位取整，输出时将其输入向下舍入为最近整数

Trunc：去零取整

Step：比较，输出0或1，用来区分出每一层的明暗，输出的是黑白纹理

If：条件

Frac：取小数部分，此节点在与 Time（时间）节点一起使用时非常有用，会随时间变化带来锯齿波

Fmod：取模，输出 [A] 除以 [B] 的余数

Clamp：限定范围

Lerp：插值，如果 [T] 为两者间的任一值，则将输入两者的线性混合

Posterize：色调分离，基于经过 [Steps] 的值对值进行四舍五入取整。

[Steps] 值为 5 时将在 0 到 1 的范围内创建 5 段

Blend：混合，有不同的混合方式，用时查看API

Remap：重映射，将值从一个范围重映射到另一个范围，用来加深或减弱

Noise：噪音，根据两分量输入（例如 UV 坐标）生成伪随机点

One Minus：1 减，输出 1 减去其输入。与颜色输入一起使用时，将反转颜色

Negate：取反，输出其主输入乘以 -1。实质上让正值变负，让负值变正

Component Mask ：分量遮罩，遮罩某个通道

Channel Blend ：通道混合

Fresnel：菲涅耳，边缘亮显的效果

更多具体的节点信息，可以查看官方的说明

节点详细信息

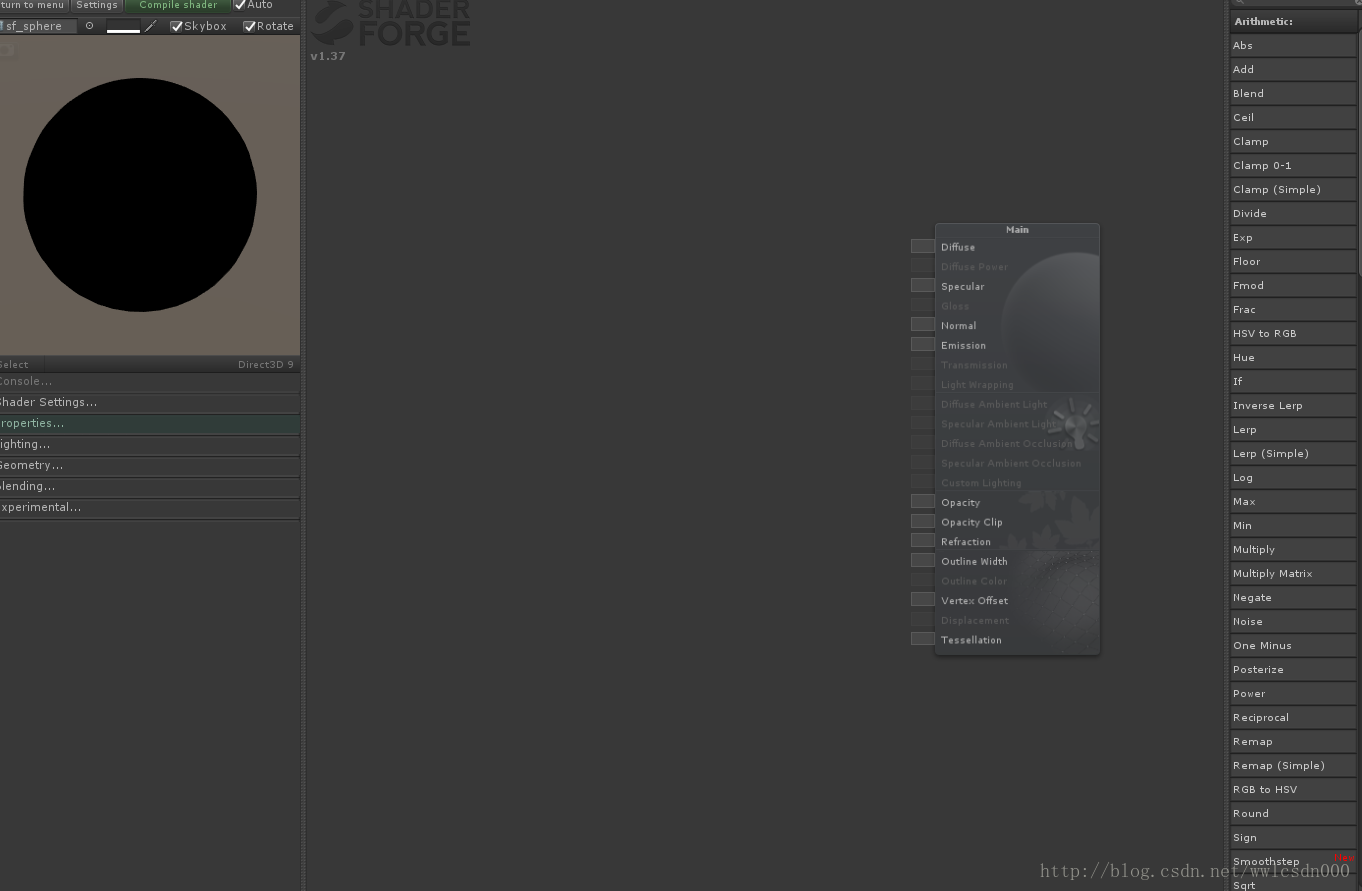
绿色的节点，代表是可以进行编辑的属性，就是可以显示到Inspector面板。

5.其他中间黑色区域为节点编辑区域

用于创建和编辑节点。下面通过操作介绍来演示。

---------------------

### 3.界面操作基本介绍



打开着色器=Windows>Shader Forge

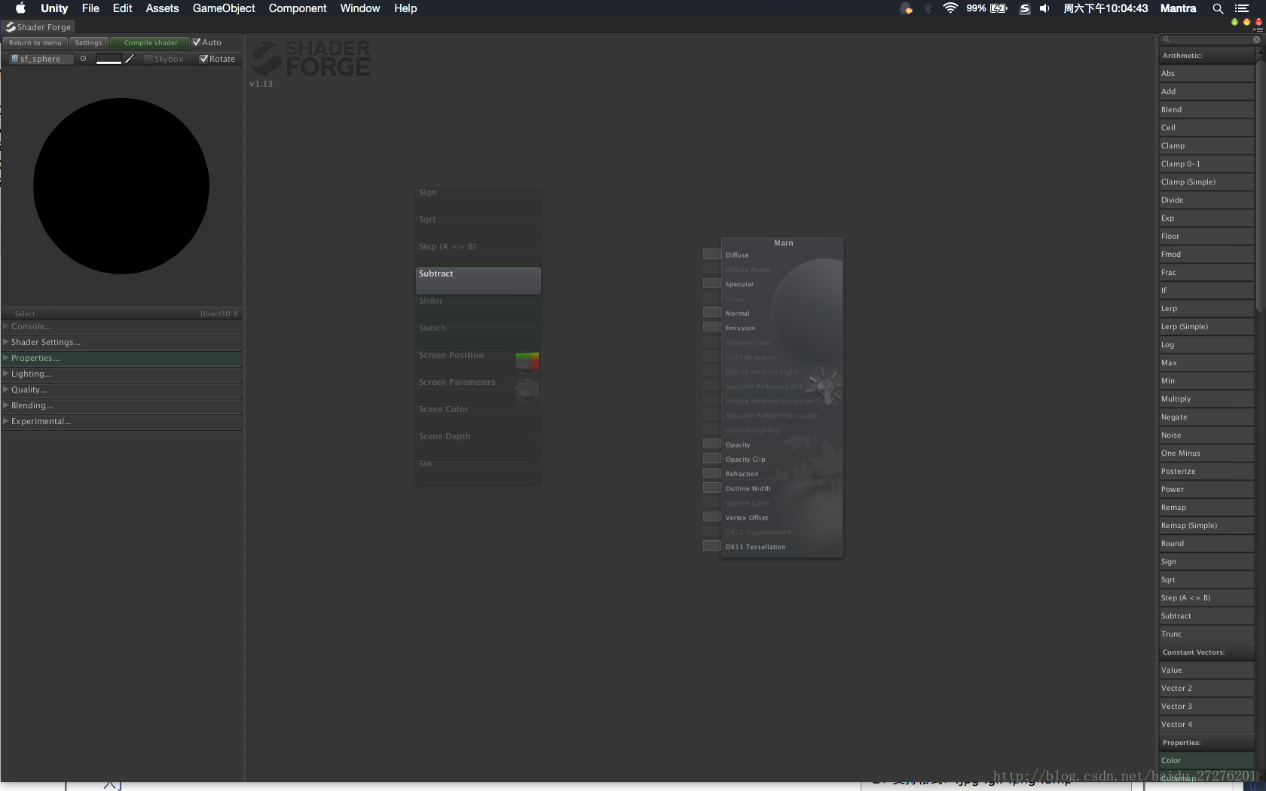
平移=单击(左键或者中键)并拖动空白空间

缩放=鼠标滚轮

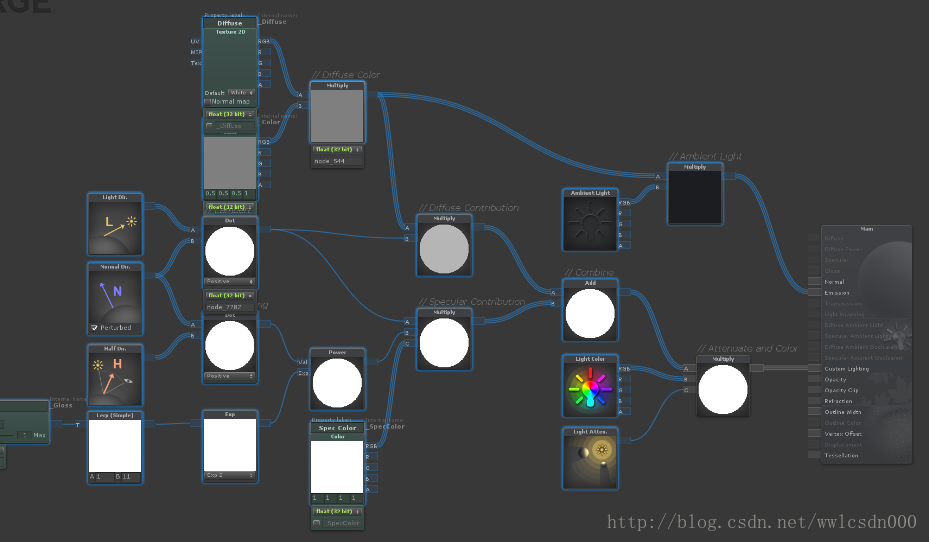
添加节点

1.右键弹出节点分类，进行选择。

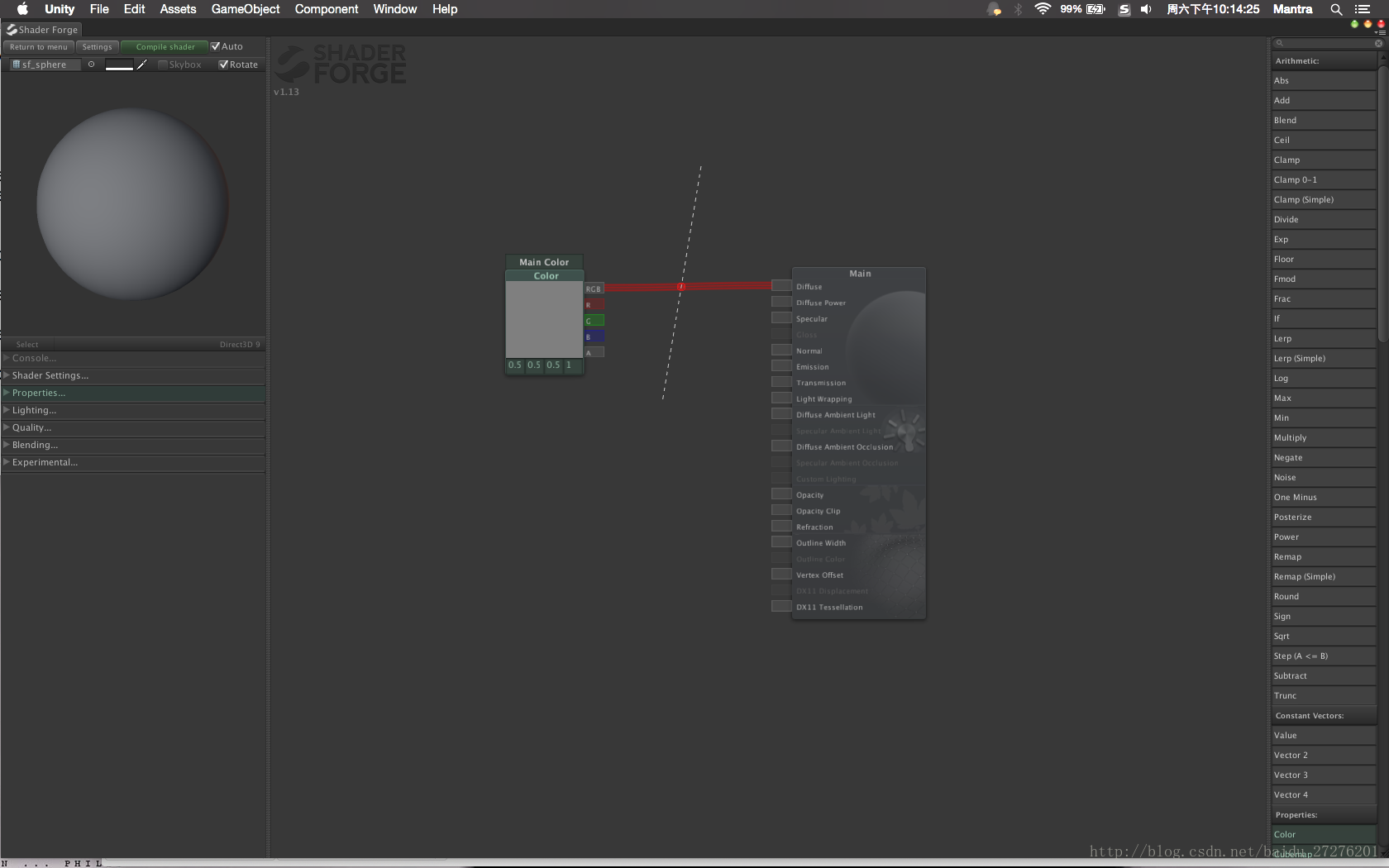
2.节点首字母 +选择节点或从右面板拖动并放置或保存节点名的第一个字母，然后滚动到所需的节点，然后单击(需要开始预览区域的相关设置)



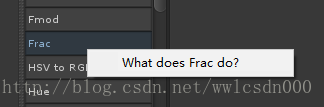
移动节点=单击和拖动   
选择节点=单击以选择、移动/控制单击以添加到选择   
框选择节点=按住Alt +拖动。移动/控制点击添加到选择



删除节点=选择并按下删除   
删除连接=按住Alt键并在连接线上右键滑动进行切割。



对于不认识的节点，右键选择what(节点树面板和节点浏览面板都支持)，出现API官网，可选简体字



连线的数量表示数据类型，一条连线表示为单个数值（float或者int），两条连线表示Vector2（如UV），三条连线表示Vector3（如Color Position等）

