

# ShaderlabVS专业版

---

## ShaderlabVS专业版

[介绍](#)

[安装](#)

[主要特点](#)

- 1. 语法突出显示
- 2. 代码完成和基本的智能感知
- 3. 悬停信息
- 4. 签名帮助
- 5. 自动格式
  - [格式文档](#)
  - [格式选择](#)
  - [在新的线路上放置开放的支撑件](#)
  - [宏对齐模式](#)
    - [带有层次结构的缩进](#)
    - [没有层次结构的缩进](#)
    - [没有缩进，但有层次结构](#)
    - [没有压进结构，也没有层次结构](#)
- 6. 转到定义
- 7. 支持SRP
- 8. 代码片段

[转录器](#)

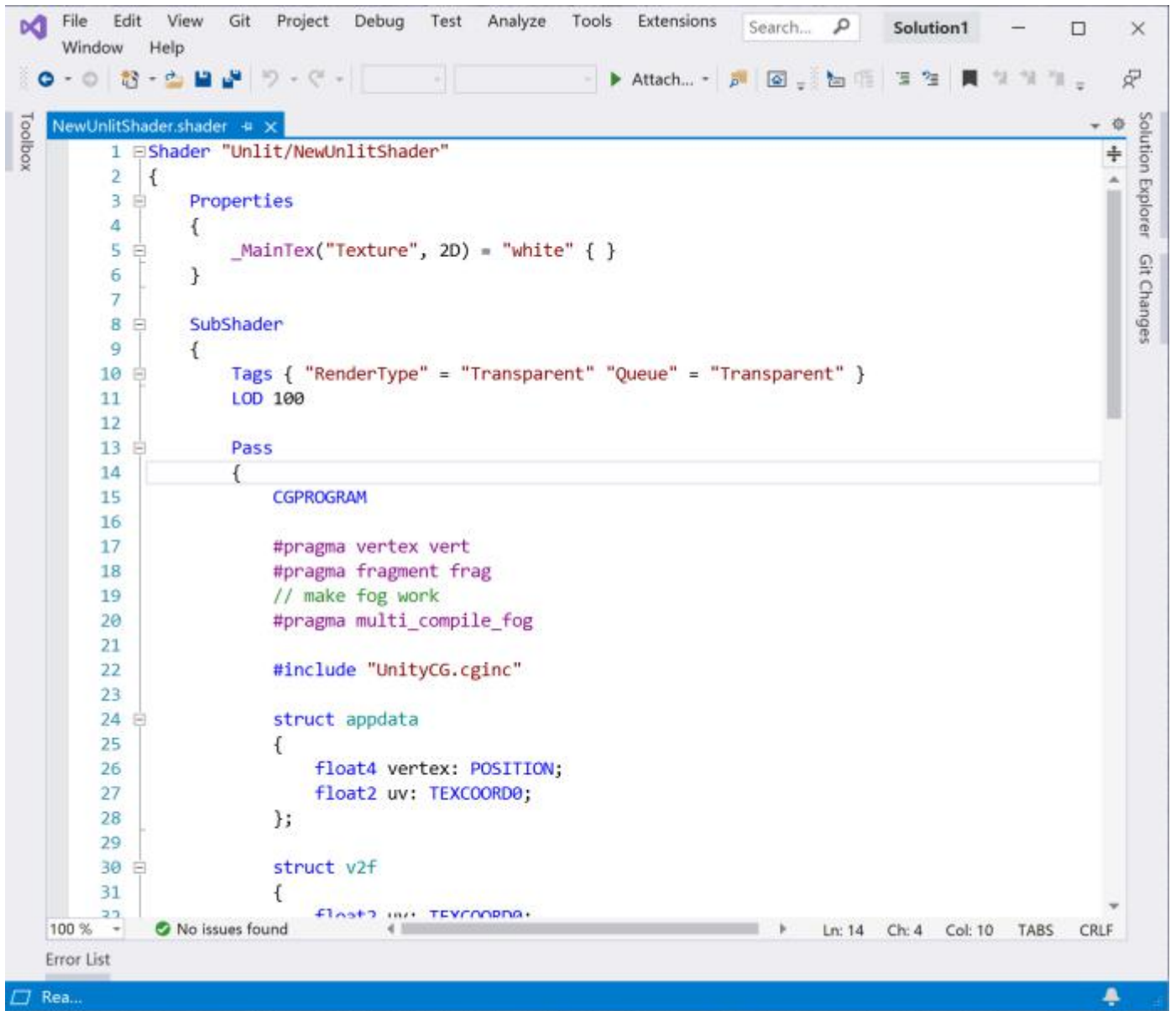
[更多信息](#)

## 介绍

---

ShaderlabVSPro是一个用于Unity阴影实验室编程的可视化SisuilStudio插件。

[文档](#) | [论坛](#) | [电子邮件](#) | [资产 存储](#)

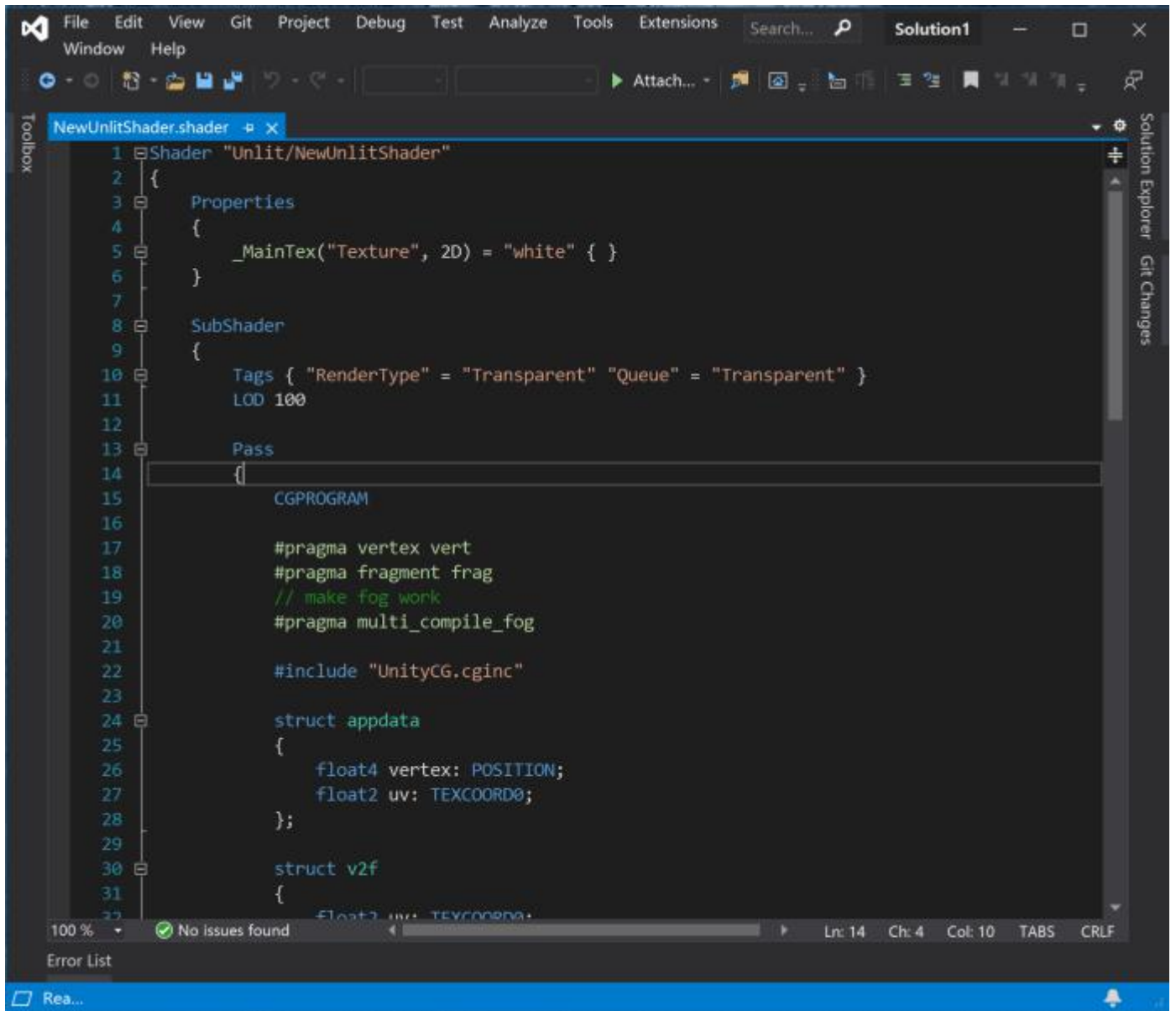


## 安装

1. 得到。路径下的vsxi文件 资产/ShaderlabVS/VSPlugin
2. 双击。vsxi文件，一个VisualStudio扩展安装窗口将弹出
3. 在安装窗口时，单击“安装”按钮
4. 安装成功后，重新启动VisualStudio

## 主要特点

### 1. 语法突出显示



## 2. 代码完成和基本的智能感知

```
float2 uv: TEXCOORD0;
UNITY_FOG_COORDS(1)
float4 vertex: SV_POSITION;
};

sampler2D _MainTex;
float4 _MainTex_ST;

v2f vert(appdata v)
{
    v2f o;

    o.vertex = UnityObjectToClipPos(v.v);
    o.uv = TRANSFORM_TEX(v.uv, _MainTex);
    UNITY_TRANSFER_FOG(o, o.vertex);
    return o;
}

fixed4 frag(v2f i): SV_TARGET
{
    // sample the texture
    fixed4 col = tex2D(_MainTex, i.uv);
}
```

### 3. 悬停信息

```
texture", 2D) = "white" { }

derType" = "Transparent" "Queue" = "Background" }

AM

vertex vert
fragment frag
fog_work
multi_compile_fog

e "UnityCG.cginc"

appdata

float4 vertex: POSITION;
```

Determine in which order your objects are drawn using the Queue tag.

### 4. 签名帮助

```
44 |         o.vertex = UnityObjectToClipPos(v.vertex);
45 |         o.uv = TRANSFORM_TEX(v.uv, _MainTex);
46 |         UNITY_TRANSFER_FOG(o, o.vertex);
47 |         return o;
48 |     }
49 |
50 |     fixed4 frag(v2f i): SV_TARGET
51 |     {
52 |         // sample the texture
53 |         fixed4 col = tex2D(
54 |             ▲ 1 of 16 ▼ float4 tex2D(sampler2D samp, float2 s)
55 |             // app performs a texture lookup in a given 2D sampler and, in some cases, a sha
56 |             UNITY_TRANSFER_FOG(o, o.vertex);
57 |         return col;
58 |     }
59 |     ENDCG
60 |
61 | }
62 | }
```

## 5. 自动格式

### 格式文档

通过菜单编辑器->高级->格式文档

### 格式选择

### 在新的线路上放置开放的支撑件

我们可以在工具->选项->文本编辑器->阴影实验室->格式化页面中设置大括号格式化样式。如果“在新线上放置打开支撑”为真，格式化后，打开支撑将在新线中，如下所示

```
如果（真）
{

}
```

如果 在新的线路上放置支架 是，格式化后，打开括号不会在新行。

```
如果（真）{

}
```

## 宏对齐模式

在设置中，阴影标签vscode部分中有一个名为“格式化：样式”的项，请将宏对齐模式更改为为宏设置格式化模式。

### 带有层次结构的缩进

```
下色器
{
    通过
    {
        cg程序
        空白宏测试()
        {
            浮动c;
            #if 0
                c = 0;
            #if 1
                c = 1;
            #endif
            #endif
        }
        恩德克
    }
}
```

### 没有层次结构的缩进

```
下色器
{
    通过
    {
        cg程序
        空白宏测试()
        {
            浮动c;
            #if 0
                c = 0;
            #if 1
                c = 1;
            #endif
            #endif
        }
        恩德克
    }
}
```

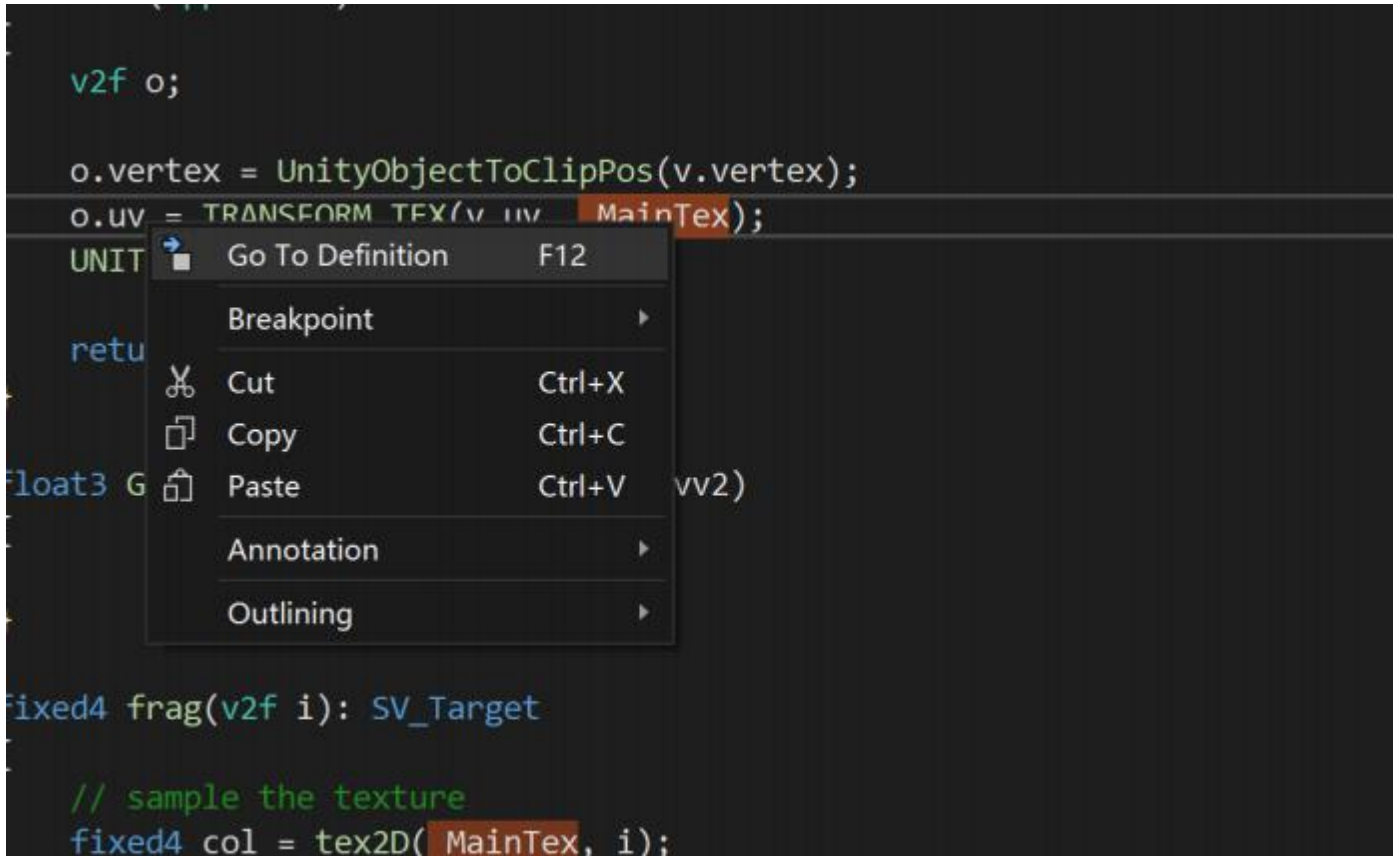
没有缩进，但有层次结构

```
下色器
{
    通过
    {
        cg程序
        空白宏测试()
        {
            浮动c;
        }
    }
    #if 0
        c = 0;
    #if 1
        c = 1;
    #endif
    #endif
    恩德克
}
```

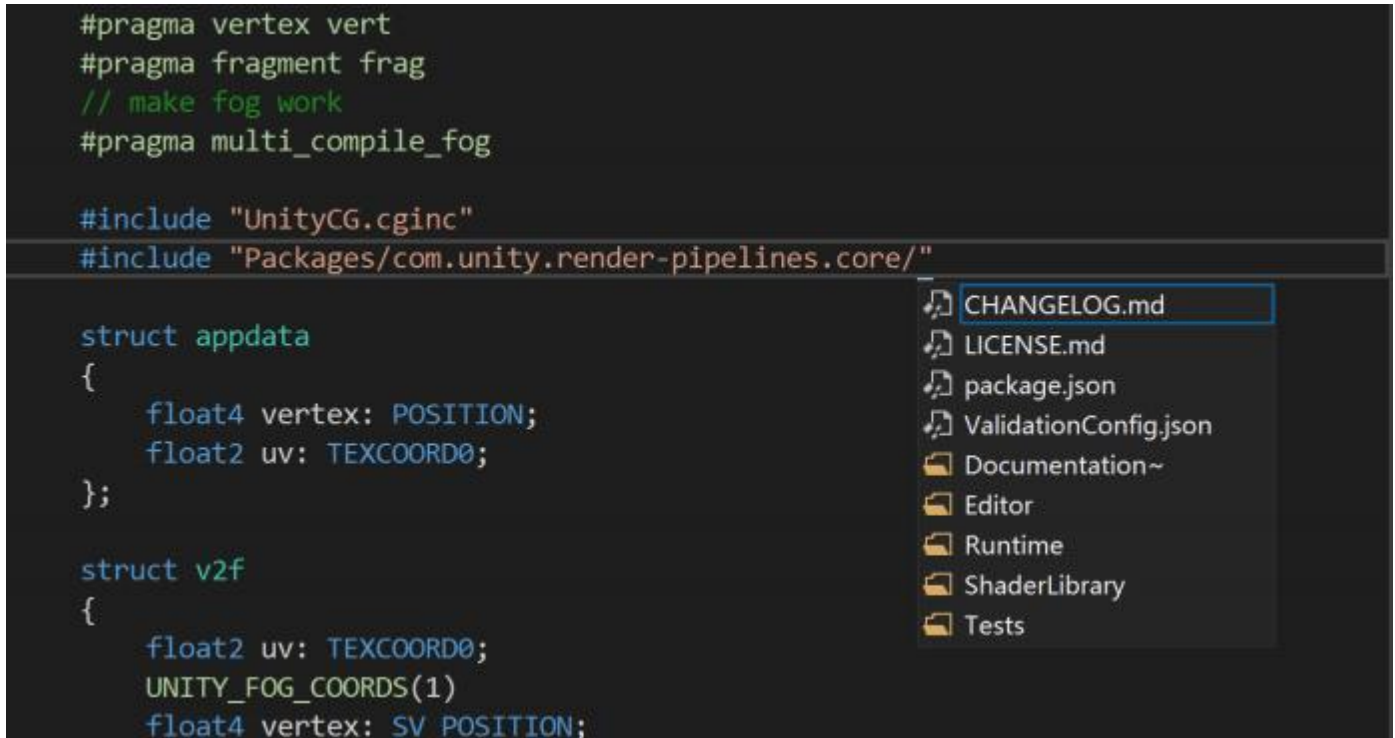
没有压进结构，也没有层次结构

```
下色器
{
    通过
    {
        cg程序
        空白宏测试()
        {
            浮动c;
        }
    }
    #if 0
        c = 0;
    #if 1
        c = 1;
    #endif
    #endif
    恩德克
}
```

## 6. 转到定义



7. 支持SRP



8. 代码片段

以下是片段：

片段	描述



blend1-dc_1	混合OneMinusDst颜色1
blend1_1-sa	混合一个OneMinusSrcAlpha
blend1_1	混在一起
blenddc_0	混合DstColor零
blenddc_sc	混合DstColorSrc颜色
blendsa_1-sa	混合SrcAlphaOneMinusSrcAlpha
cgp	cg程序...
为了	为了 环
forr	反向 为了 环
glp	... 格拉斯普林斯格尔
hlp	...hlsl程序修正ls1
如果	如果 {}...
如果其他	如果 {} 其他 {}.....
inc	#include ""
incpkg	#包括“包”
incucg	#include "UnityCG.cginc"
prop2d	2D 类型属性
propc	颜色 类型属性
支柱立方体	立方 类型属性
propf	浮点 类型属性
支柱范围	范围 型质
小道具	属性 {}...
propv	矢量 类型属性
地区	...//region //endregion
region2	...//#region //#endregion
着色器	着色器 {}...
结构	结构 {。..};
subshader	SubShader { }...

标记	标签{。..}
tagtt	同时具有渲染类型和队列的标记是透明的

# 转录器

## v1.1.2

- 在“程序包”文件夹中添加对着色器的支持
- 添加对VisualStudio2020预览的支持
- 修复当结构包含宏时，在代码完成中丢失的一些错误●

## v1.1.1

- 改进了自动格式化中的宏格式化
- 修复异常时，放置支撑在新行是假的自动格式
- 修复了包含方法成员的结构代码完成不正确的错误

## v1.1.0

- 添加格式选择功能
- 更好的HLSL支持添加更多的关键字和内置方法
- 更好地概述支持区域和项目块
- 更好的宏格式。我们可以在宏对齐模式中更改不同的样式，设置更好的类型和方法的语法突出显示●
- 修复了结构化缓冲区在悬停信息中显示为缓冲区类型的错误
- 修复一些破坏着色器编译

## v1.0.5

- 更好的代码完成匹配
- 添加版本更新通知
- 修复一些小错误

## v1.0.4

- 改进所有事件的高亮显示将忽略注释行
- 提高了对大文件的性能
- 修复了对最后一行注释不工作的错误

## V1.0.3

- 在新的行格式化样式上添加放置开放的大括号
- 在悬停信息中添加类型信息
- 修复了在转到定义中找不到方法中的变量的错误
- 在某些情况下，某些变量/成员的修复类型名称是错误的

## V1.0.2 Beta3

- 为方法/变量/类型添加“要转到定义”的功能
- 为选定单词的所有出现情况添加突出显示
- 为由宏定义的方法添加签名帮助支持
- 改善支架匹配标记的背景颜色
- 修复了由宏定义的方法在完成过程中缺失的错误
- 修复了在“代码已完成”中重复完成项的错误
- 修复了内置包含的着色器库缺失的错误

## V1.0.1 beta2

- 改进语法突出显示
- 添加代码片段支持

## v1.0 beta1

初始化承诺

# 更多信息

---

参观地点<http://www.amlovey.com/shaderlabvs>