

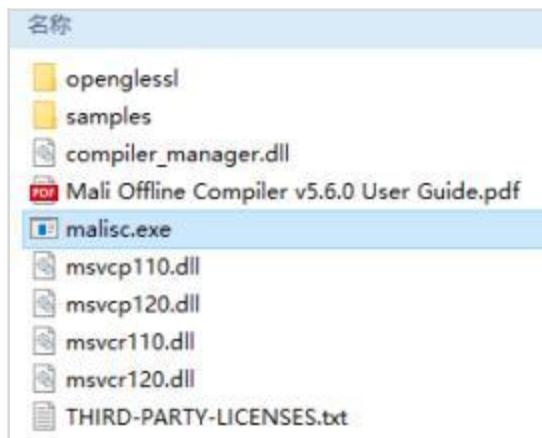
Mali Offline Compiler 使用方式

1. 下载

UWA 目前使用的是5.6 版本:

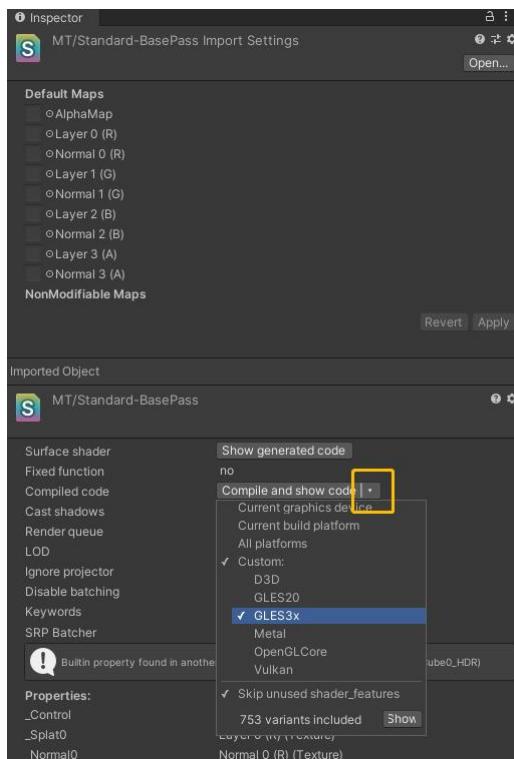
https://armkeil.blob.core.windows.net/developer/Files/downloads/opengl-es-open-cl-offline-compiler/5.6/Mali_Offline_Compiler_v5.6.0.af0c20_Windows_x64.zip

解压后，文件夹中的malisc.exe 即所需的命令行工具



2. Unity Shader 的处理

选中待分析的Shader，在Inspector 中点击黄框按钮，将弹出面板设置为下图的状态（只勾选了GLES3x）



关闭面板后，点击“Compile and show code”按钮，Unity 会开始编译该Shader 的各个变体的“预编译版本”，编译完成后会自动弹出预编译版本的.shader 文件，其中每个变体都有对应的vertex 和fragment 代码。变体信息：

```
-- Hardware tier variant: Tier 1
-- Fragment shader for "gles3":
Shader Disassembly:
// All GLSL source is contained within the vertex program
...
///////////////////////////////
Global Keywords: DIRECTIONAL LIGHTPROBE_SH SHADOWS_SHADOWMASK
Local Keywords: <none>
-- Hardware tier variant: Tier 1
-- Vertex shader for "gles3":
```

对于fragment 代码，都是被宏定义FRAGMENT 包含，开头部分如下：

```
#ifdef FRAGMENT
#version 300 es
#ifndef GL_EXT_shader_texture_lod
#extension GL_EXT_shader_texture_lod : enable
#endif
...
```

找到希望分析的变体，找到对应的fragment 代码，将 #ifdef FRAGMENT 和 #endif 之间的代码拷贝到新的文件中，比如target.frag 文件

3. 获取复杂度

通过命令行 malisc.exe --fragment target.frag 即可获取复杂度结果，如下：

```
ARM Mali Offline Compiler v5.6.0
(C) Copyright 2007-2017 ARM Limited.
All rights reserved.

No driver specified, using "Mali-T600_r13p0-00rel0" as default.

No core specified, using "Mali-T830" as default.

No core revision specified, using "r2p0" as default.

8 work registers used, 16 uniform registers used, spilling used.

      A      L/S      T      Bound
Instructions Emitted: 110      20      12      A
Shortest Path Cycles:  26      18      11      A
Longest Path Cycles:  37      20      12      A

A = Arithmetic, L/S = Load/Store, T = Texture
Note: The cycles counts do not include possible stalls due to cache misses.
Note: Shaders with loops may return "N/A" for cycle counts if the number of cycles cannot be statically determined.

Compilation succeeded.
```

其中，建议参考的是 Longest Path Cycles 的数值。

PS:

如果使用过程中遇到如下报错：

```
'binding' qualifier is not allowed in language version 300 es
```

将Shader首行的 #version 300 es 改为 #version 320 es 即可。

```
1 #version 300 es
2
3 precision highp float;
4 precision highp int;
5 #define HLSLCC_ENABLE_UNIFORM_BUFFER_SUPPORT
6 #if HLSLCC_ENABLE_UNIFORM_BUFFERS
7 #define UNITY_UNIFORM
8 #else
```