使用 RenderDoc 调试 Vulkan 程序

2022-09-09 19:25:57

前言

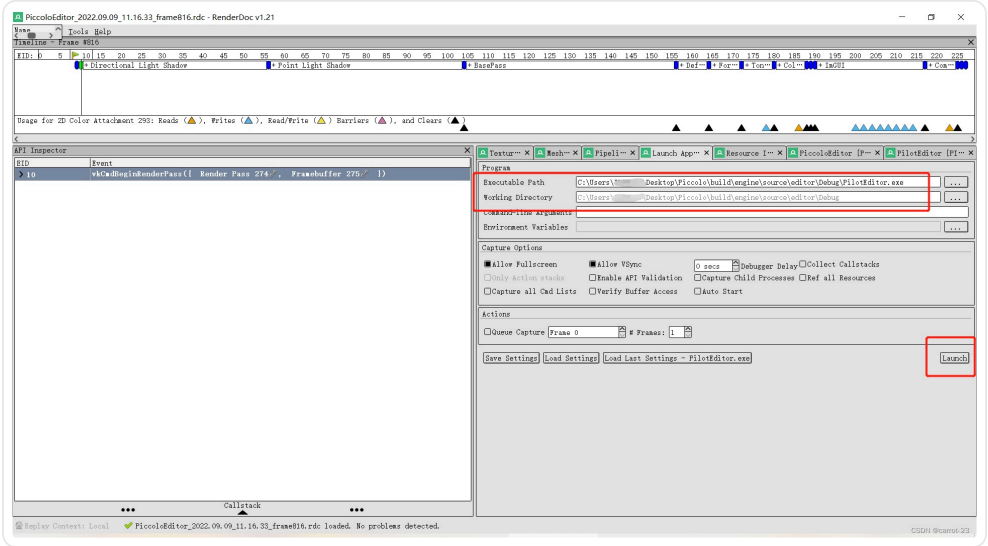
在 RenderDoc 出现之前，我们很难有一个好的方法来调试 Shader 程序，传统的做法是将一些调试信息以纹理的形式渲染到屏幕空间来供我们检查，例如将将坐标信息映射成 RGB。

但是这种做法往往效率不高，而 RenderDoc 的出现极大地加速了我们调试的效率。它能让我们像 CPU 程序一样打断点一行一行地看，并且 RenderDoc 还能够收集一帧（或者多帧）中每个渲染 Pass 的缓冲区输入输出、调用堆栈、缓冲区格式等，使得 Debug 更加方便。

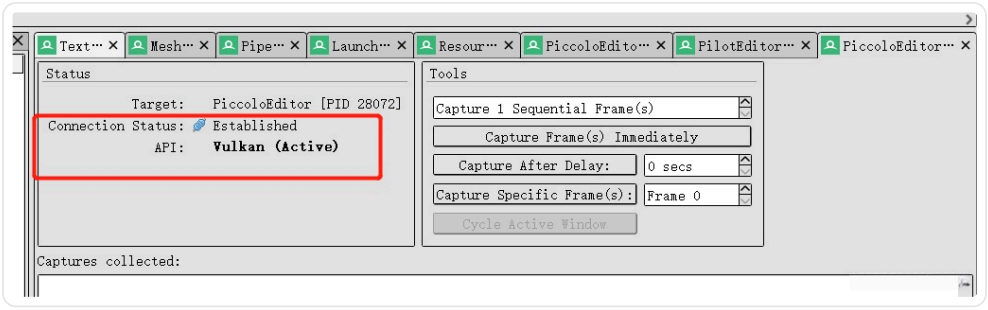
官网链接：https://renderdoc.org/

快速上手

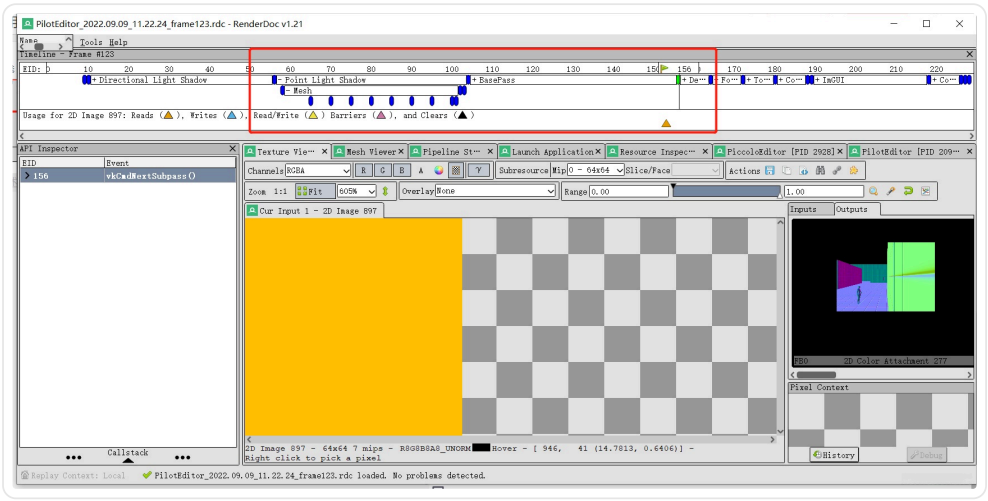
安装完成后，打开 RenderDoc，点击 Launch Application，选择要调试的 Vulkan 应用程序。这里需要注意的是我们在编译 Shader 的时候需要开启编译调试符号信息，否则我们没办法捕获 Shader 的运行状态。



程序启动后，我们就能够看到 RenderDoc 和 Vulkan API 建立了连接，右边 Capture Frame 可以捕获当前帧的运行信息。



捕获完成后，我们就能够在上面的 Timeline 窗口中看到渲染一帧的全流程了。



事件浏览器

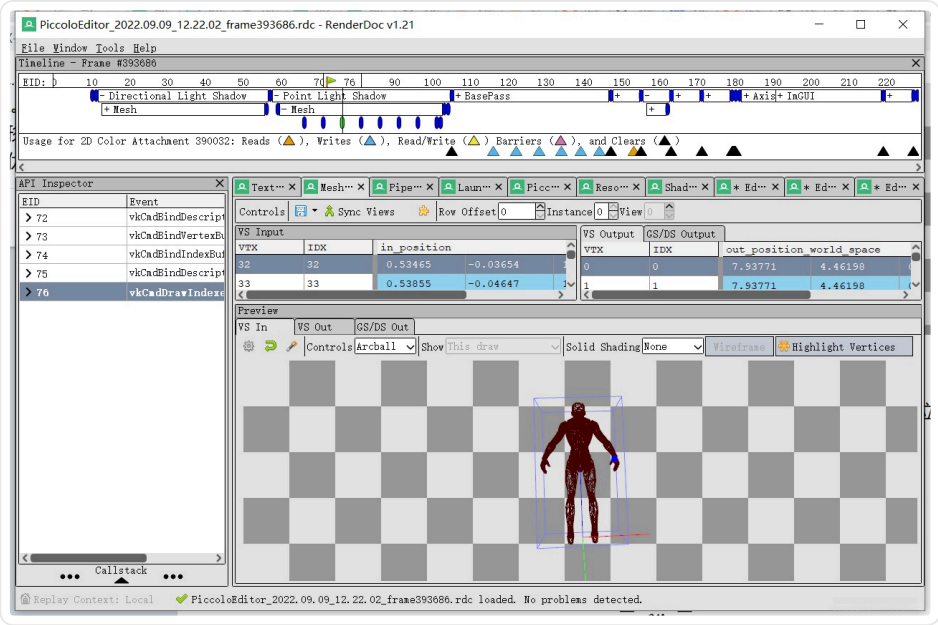
绘制事件

点击 Timeline 中的蓝色端点，可以查看一次 DrawCall 的详细过程：

Texture Viewer： 查看该次事件所用到的输入、输出缓冲区，注意缓冲区可以有多个

Pipeline State： 当前 DrawCall 的 Pipeline 包含哪些步骤，例如 VS、GS、FS 等

Mesh Viewer：当前 DrawCall 的 Mesh 信息，可以查看每个点的输入和输出



DrawCall 统计

在 Even Browser 中我们可以有选择地过滤一些事件，例如我们想统计以下 Draw Call 的数量，我们可以在事件过滤栏里输入 `$action() Draw` ，有兴趣的朋友可以数一下，其实和 Statistics Windows 下得到的结果是一样的。

在下面的例子中使用的延迟渲染，在每个 Pass 中会处理多个 Mesh，因此可以看到在每个 Pass 中都需要调用许多次 DrawCall。

常用的有以下一些命令：

Draw

Clear

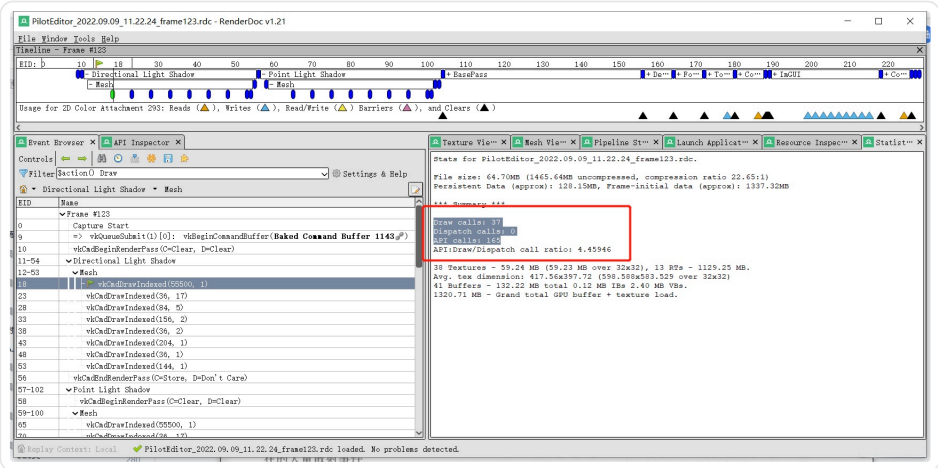
Copy

Depth

Indexed

Instanced

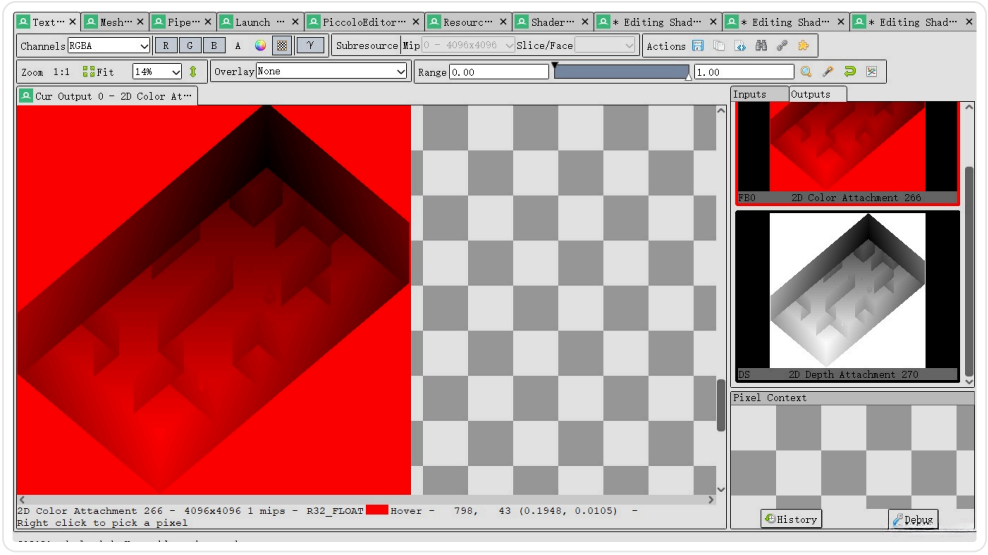
Depth



Texture Viewer

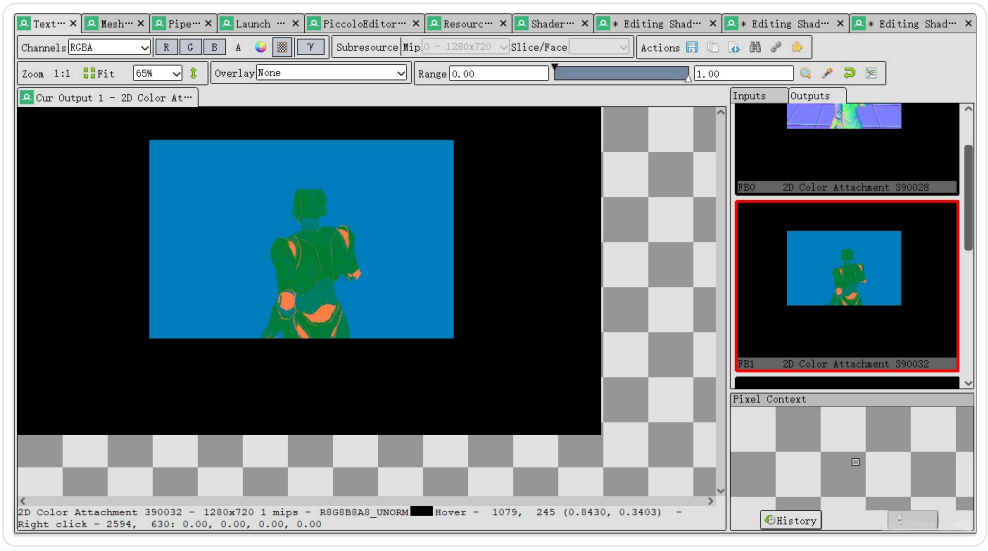
观察深度缓存

在 Texture Viewer 中我们可以观察到一个事件所使用的所有 FrameBuffer，下面例子中我们在光源视角生成了一张深度图，这张深度图也就是所谓的 Shadow Map



观察 G-Buffer

在生成 G-Buffer 的阶段，我们可以看到在 FrameBuffer 中一共生成了 4 种信息，分别是法线、漫反射颜色、粗糙度、深度。这些信息生成后就处于屏幕空间坐标系，在后面的光照 Pass 会使用到这些信息进行着色。



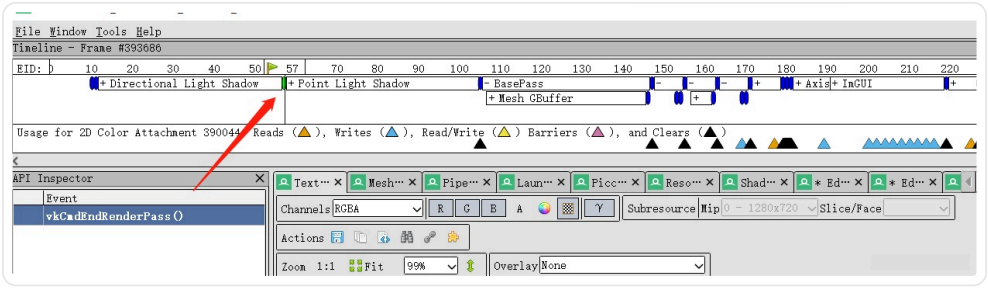
观察 RenderPass 和 SubPass

一个 RenderPass 可以包含多个 SubPass，这其实也指明了他们之间的关系。RenderPass 通常会伴随 FrameBuffer 的切换，而 SubPass 是发生在一个 RenderPass

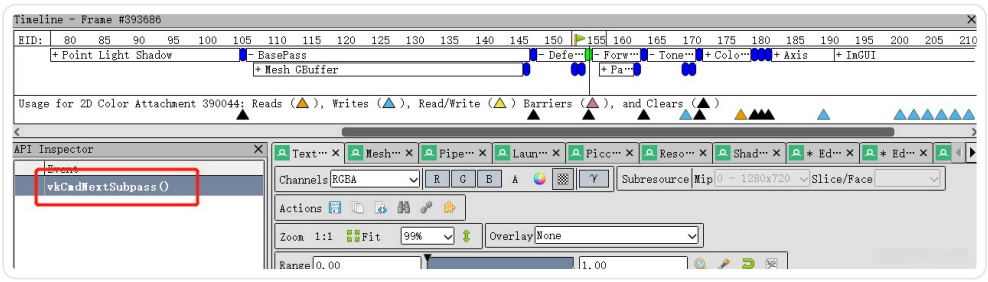
里面，可以利用前一个 Pass 的输出作为输入。

例如我们生成 ShadowMap 和 G-Buffer 的过程就分布在两个不同的 RenderPass，而生成 G-Buffer 和着色的过程就能发生在同一个 RenderPass 里面。

例如下图在生成太阳光的 ShadowMap 以后调用了 `EndRenderPass()`



而延迟渲染的后面几个步骤是调用 `NextSubpass()`，说明他们都是在同一个 RenderPass 中



全部评论 (0)

0041950719

发表一个友善的评论吧...

发送

还没有任何评论哟~

0041950719

这个人很神秘，什么都没有写

0 文章　**0** 总访问量

分类

java　c++　数据库　编程语言　前端　python　人工智能　移动开发

后端　c#　数据结构与算法　音视频　开源　小程序　运维　服务器

标签

web　springboot　vue2　ssm　数据库　c++　express+mongoDB　深度学习

图像处理　OpenCV　静态网页　vue3　神经网络　element　mybatis　mvc

机器学习　react

文章目录