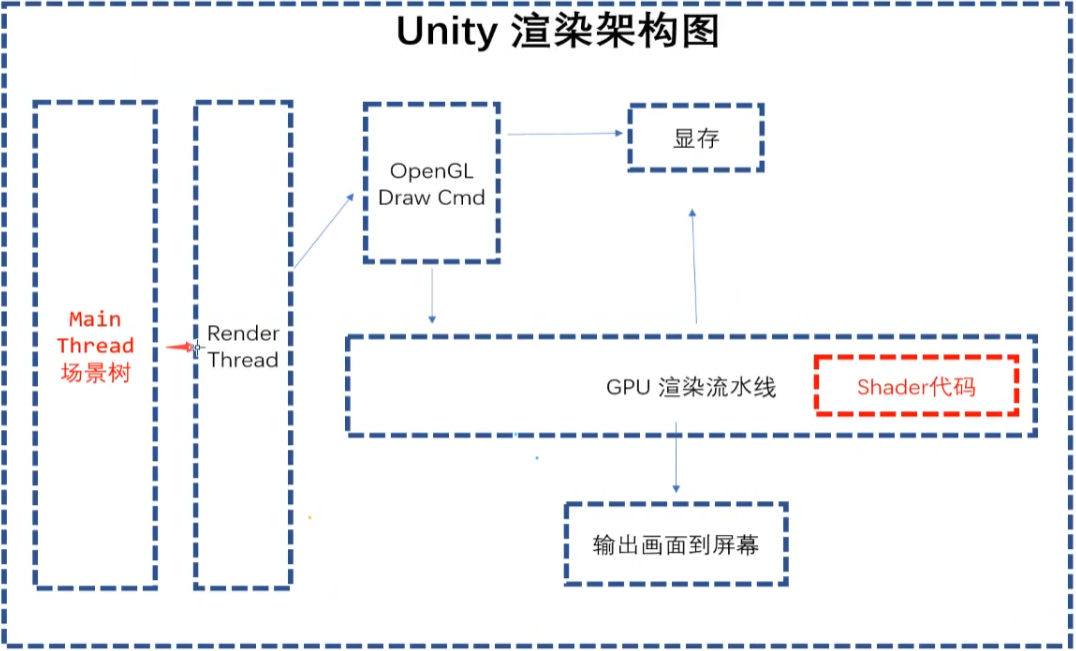
Unity渲染架构图



Unity多线程，写代码经常用main线程处理，包括调用mono相关接口

1. Main Thread : 遍历场景，把需要绘制的物体提交给Renderer Thread
2. Renderer Thread：调用OpenGL/Dx 等接口，操作GPU和显存
   1. 配置渲染管线(绘制物体的标准流程)
      1. 装载代码，装载设置管道渲染流水线的参数
      2. Shader代码装载进渲染管线
      3. Set Pass Call：绘制完整个游戏场景，配置了多少次GPU渲染管线。非常消耗性能。一个游戏不要用太多不同的shader，尽量用同一种shader来绘制一批物体。
      4. 把Shader提前加载到内存，提升配置渲染管线的速度。
   2. GPU的渲染管线，虽然是标准流程，但是流程中的环节，会开发出来给用户编程控制。
      1. 给GPU写程序可以控制渲染管线绘制出来的效果。
      2. 普通的开发人员能够绘制3D物体，是因为unity已经写好了一些自带的渲染管道的程序--🡪就是Shader，给GPU执行的代码指令。
3. 把mesh，纹理 提交到显存里，给Renderer Thread 给GPU下达绘制命令
4. GPU绘制到屏幕中

Draw call：Render thread 给GPU下达一次绘制命令，就是一次Draw call

Set pass call : 配置渲染管线，装载参数，装载shader

假设绘制2个物体，render tread 向GPU下达2次命令，draw call = 2

如果2个物体使用同一个渲染管线，set pass call = 1，如果使用不同渲染管线，那么中间会切换x次渲染管线，那么Set pass call = x。

优化方案

1. 要减少渲染管线的切换
2. 注意场景，一类物体使用同一渲染管线
3. 常用的Shader-----🡪Cache。