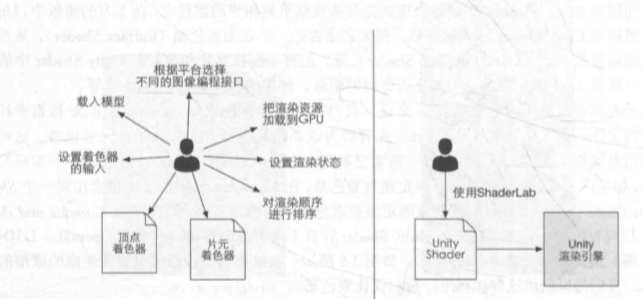
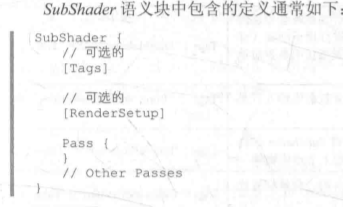
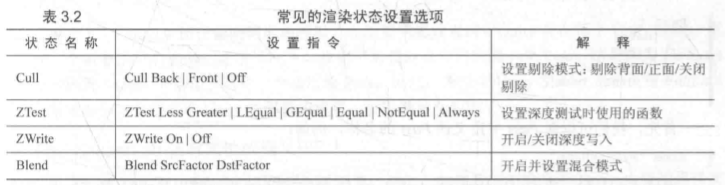
# Unity Shader基础

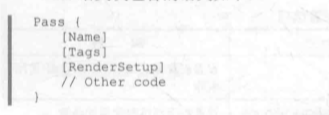
* Unity shader 概述
  + Unity shader与 材质

使用unity shader的常见流程

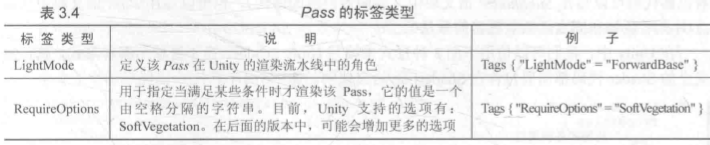
1. 创建一个材质
2. 创建一个shader，并赋给创建的材质
3. 把材质赋给需要渲染的对象
4. 调整shader的属性

* ShaderLab
  + Unity提供的专门为unity shader服务的语言，让开发者轻松地控制渲染
  + 
  + 基础结构
  + Shader "Shader Name"
  + {
  + Properties
  + {
  + }
  + SubShader
  + {
  + }
  + FallBack "Diffuse"
  + }
* Unity Shader的结构
  + Properties
    - Properties
    - {
    - // unmbers and sliders
    - \_Int ("My\_Int", Int) = 2
    - \_Float ("My\_Float", Float) = 1.5
    - \_Range ("My\_Range", Range(0.0, 5.0)) = 3.0
    - // colors and vectors
    - \_Color ("My\_Color", Color) = (1, 1, 1, 1)
    - \_Vector ("My\_Vector", Vector) = (2, 3, 4, 5)
    - // Textures
    - \_2D ("My\_2D", 2D) = " " {}
    - \_Cube ("My\_Cube", Cube) = "white" {}
    - \_3D ("My\_3D", 3D) = "Black" {}
    - }
  + SubShader
    - 一个shader文件能够包含多个SubShader语句块，unity会选择一个能够在目标平台上使用的SubShader进行渲染。
    - 设置状态
    - 标签 Tags 

上述Tags只能够在SubShader中声明，不能在Pass中声明。（在pass中声明的tags与上述的不同）

* + - Pass语句块

定义Pass名字，通过UsePass和Pass的名字可以使用其他Unity Shader中的Pass。（使用UsePass时要全部大写，uinty内部会将Pass名字全部转换为大写）

* + - Pass中的标签
    - 特殊的Pass：UsePass，GrabPass(抓取屏幕并将结果存储在一张纹理中，用于后续Pass处理)
  + FallBack
    - 如果上述的所有SubShader无法使用，则使用FallBack之后的Shader
* Unity Shader形式
  + 表面着色器
  + 顶点/片元着色器
  + 固定函数着色器