初识OpenGL ES 2.0

1. 开发环境及资料文档

安装开发环境

学习OpenGL ES 的开发，可以在Windows搭建模拟环境来进行练习与编程，此环境可以简单的通过Resources 目录下的 PowerVRSDKSetup-3.0r2.exe 客户端方便的完成。

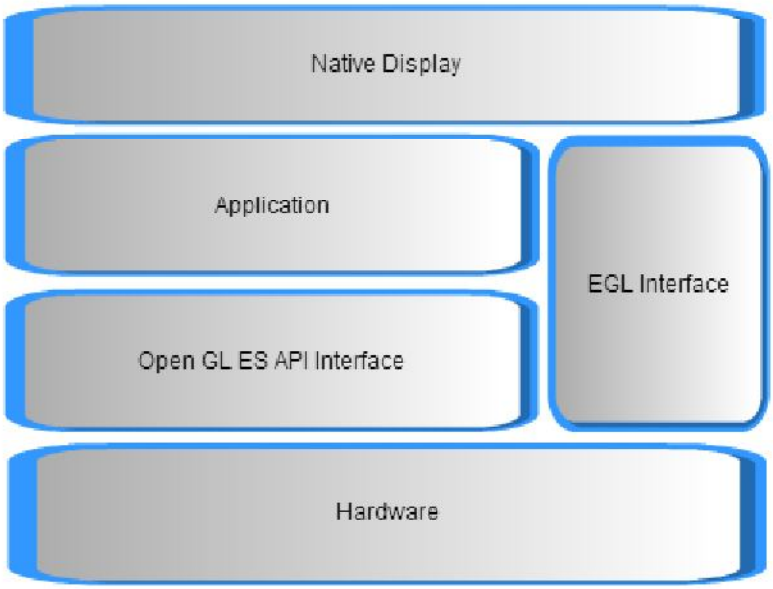
PowerVRSDKSetup客户端运行后，可以配置安装哪些平台的运行库，和需要支持的OpenGL ES 的版本，配置完成后会通过网络下载需要的资源，只支持Windows系统和OpenGL ES 2.0 版本大概会下载500多M的数据，其中包括安装时配置的平台的运行库和头文件、相关文档资源（比如示例代码）和许多有用的工具（比如PVRShaderEditer）。

有用的资料文档

1. 可以从官方网站获取所有文档 ，2.0的文档已放在共享的Resources目录中，其中：
   - OpenGL-ES-2_0-Reference-card.pdf 很有用，可以在编程的时候随时察看
   - es_full_spec_2.0.25.withchanges.pdf 为OpenGL ES 2.0 的官方文档
   - GLSL_ES_Specification_1.0.17.pdf 为 OpenGL ES 2.0 使用的 GLSL 的官方文档
   - OpenGL_ES _ 2.0_API官方在线文档
2. 示例代码可以参考PowerVR SDK 安装目录下的示例源码。
   - 默认在C:\Imagination\PowerVR\GraphicsSDK\SDK_3.0\Examples
3. 入门书籍中文没发现有什么好的，提供一本英文版的，说的比较清楚。
   - Addison.Wesley.OpenGL.ES.2.0.Programming.Guide.Aug.2008.pdf

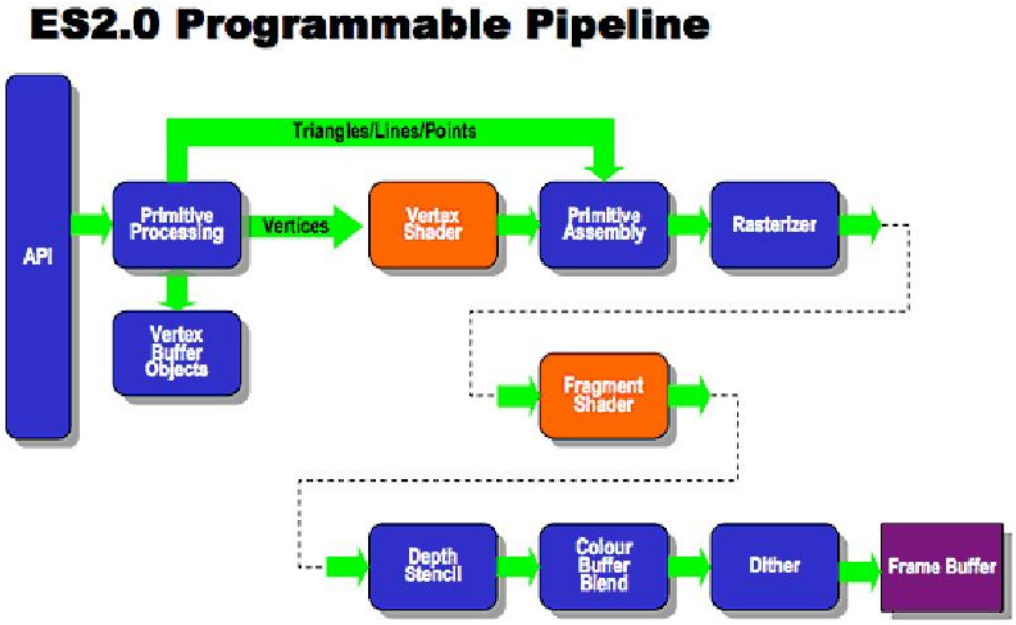
2 OpenGL 图形渲染系统结构

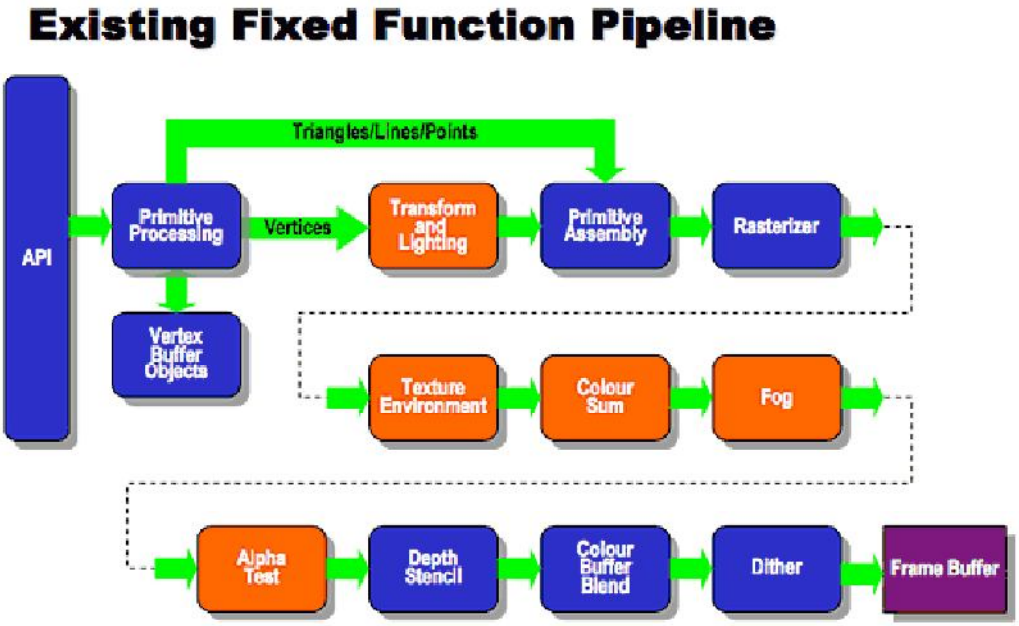
OpenGL ES 系统结构



OpenGL ES 渲染流程

3D渲染过程为流水线操作，首先需要准备好需要处理的输入数据，包括顶点数据和颜色数据，然后设置好控制参数，然后调用绘制函数，一次渲染就开始了。





3 EGL简介及初始化

EGL的功能

EGL是OpenGLES模块和外部模块沟通和协作的桥梁，它主要负责以下几大功能：

1. 沟通OpenGLES和系统本地窗口
2. 查询和配置绘制表面
3. 建立绘制表面
4. 同步OpenGLES渲染过程和其它渲染API（比如OpenVG）的渲染过程
5. 管理渲染资源，比如纹理等

EGL的初始化过程

```
EGLBoolean initializeWindow(EGLNativeWindow nativeWindow)
{
   // 指定需要的渲染表面需要具备的一些特性
   const EGLint  configAttribs[] =
   {
      EGL_RENDER_TYPE, EGL_WINDOW_BIT,
      EGL_RED_SIZE, 8,
      EGL_GREEN_SIZE, 8,
      EGL_BLUE_SIZE, 8,
      EGL_DEPTH_SIZE, 24,
      EGL_NONE
   };
   // 指定需要为OpenGL ES 2.0进行初始化
   const EGLint  contextAttribs[] =
   {
      EGL_CONTEXT_CLIENT_VERSION, 2,
      EGL_NONE
   };
   // 1. 首先获取本地窗口
   EGLDisplay dpy;
   dpy = eglGetNativeDispay(EGL_DEFAULT_DISPLAY);
   if(dpy == EGL_NO_DISPLAY)
   {
      return EGL_FALSE;
   }
   // 2. 对本地窗口初始化EGL
   EGLint major, minor;
   if(!eglInitialize(dpy, &major, &minor))
   {
      return EGL_FALSE;
   }
   // 3. 查询想要建立的渲染表面的一些特性是不是本地窗口和OpenGL ES 都支持
```

```
    EGLConfig  config;
    EGLint  numConfigs;
    if(!eglChooseConfig(dpy, configAttribs, &config, 1,
       &numConfigs)) {
          return EGL_FALSE;
    }
    // 4. 根据配置建立渲染表面
    EGLSurface window;
    window = eglCreateWindowSurface(dpy, config, nativeWindow, NULL);
    if(window == EGL_NO_SURFACE)
    {
       return EGL_FALSE;
    }
    // 5. 根据配置建立一个上下文,此对象OpenGL ES用来维护渲染状态
    EGLContext context;
    context = eglCreateContext(dpy, config, EGL_NO_CONTEXT,
                               contextAttribs);
    if(context == EGL_NO_CONTEXT)
    {
       return EGL_FALSE;
    }
    // 6. 启用当前的本地窗口、渲染表面和渲染上下文, EGL初始化完成。
    if(!eglMakeCurrent(dpy, window, window, context))
    {
       return EGL_FALSE;
    }
    return EGL_TRUE;
}
```