Three.js是一个基于WebGL编写的轻量级的渲染引擎。可以很轻松的创建三维应用程序，并且将渲染结果显示到一个Canvas节点上嵌入到DOM的结构内。

Three.js的核心代码行数28997行，包括javascript代码和导出的typescript代码，不过typescript代码只有类的定义没有具体的实现。源码中包含的包数量有16个，但是具体核心的包有14个（除去extras和helpers如图1）。经过挑选核心类来分析，主要包括的类有XXX个，在附件的three.js.mdj文件中有缩略的类图。

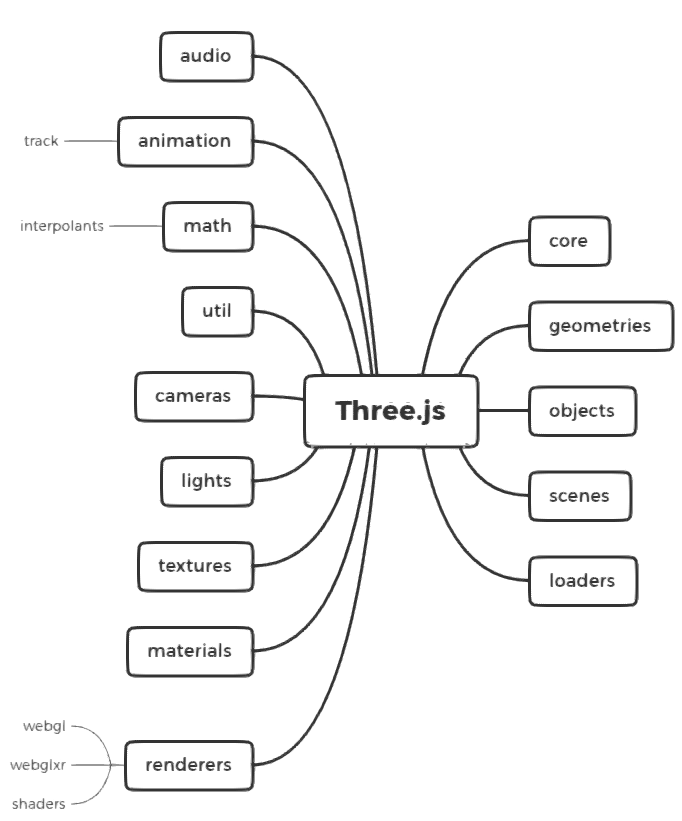


图1核心包结构

经过我们分析后，我们主要挑选最核心的部分包来分析整个项目，我们认为最核心的部分包括core, geometries, objects,scenes, loaders, audio, animation, cameras, lights, textures, materials, renderers, 对于math, util和heplers三个包我们认为主要是作为工具类或者是方便开发者调试的辅助类，并不会对体系结构造成影响，因此我们忽略掉他们。

我们首先从core包开始分析包内的类继承情况并分析core的在项目中的地位，进而分析其他包与core包的关系得出整个项目是采用的分层体系结构来构建的，并画出了结构图如图2。

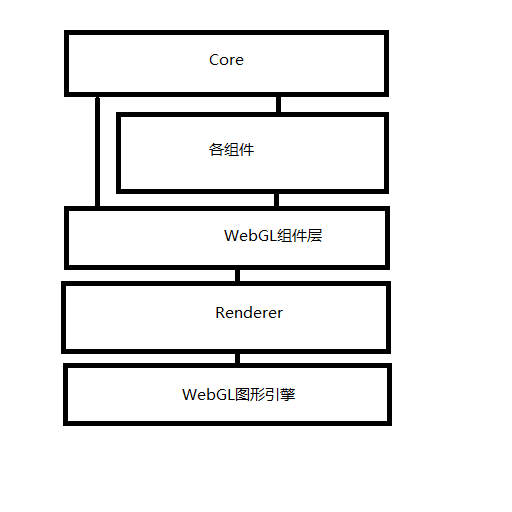


图2 three.js分层结构示意图

需求分析

Three.js作为一个基于WebGL的渲染引擎，渲染管线已经由WebGL图形库实现，Three.js就需要在此基础上驱动渲染管线完成渲染工作。渲染的过程主要是把几何数据的逻辑描述与渲染参数结合，计算出屏幕上每个像素的颜色强度并现实到屏幕上，而WebGL已经完成了最后一步操作，Three.js只需要管理渲染需要的参数并在渲染过程将参数传入到驱动WebGL工作的渲染器即可。因此，经源码分析，总结Three.js主要功能包括:

* 描述几何数据
  + 加载几何文件
    - 加载三维模型文件
    - 加载JSON文件
  + 保存
    - 导出Three.js定义的JSON文件
* 构造渲染参数
  + 摄像机参数
    - 正交投影
    - 透视投影
    - 双透视投影
    - 多视图透视投影
  + 光照参数
    - 点光源
    - 方向光
    - 聚光
    - 半球光
  + 着色器参数
    - 着色器代码
    - 贴图
    - 法线
    - ……
  + 渲染

不过Three.js在支持渲染的基础上，还增加了动画基础库，让开发者可以解析模型带有的动画数据并控制动画的播放，另外还增加了对声音的支持，让场景更加生动。为了方便开发者可以更容易地调试，为一些组件开发支持工具，可以完成一些可视化的操作，例如SkeletonHelper可以呈现模型的骨骼分布，PointLightHelper可以显示点光源的位置，CameraHelper可以显示摄像机的视锥体和位置。

除了核心源代码，Three.js还需要方便开发者扩展。因此项目设计就需要有很好的扩展性，在整个项目源码中，有其他开发者基于核心源代码开发的其他工具包，包括物理模块，后期处理模块，模型导出模块等，不过没有被整合到核心代码内，只是作为样例程序保存在项目examples文件夹下，供其他开发者参考借鉴，根据自身需求开发自己需要的工具。