

Object 3D

Model(读取的模型){

直接用Project里面，读取，draw

}

Element（自己构建的模型）{

类似 Model，指定顶点坐标，Draw

}

Shader

Camera

Light

Texure

所有的Entity要实现Blur，根据Z轴的深度实现模糊效果。（片段着色器里面）

ModelEntity: Model + shader + Texure + Position + Material + 运动

{

包括Model本身的信息，Model对应的shader（指针），Model的Position，Model的Texture，Model的Material，Model的运动信息。

实现的功能，接受外部的camera信息和light信息，绘制Model

注意：状态机的clean，所有绑定一定要解绑，所有的空间都要free

}

ElementEntity: Element + shader + Texture + Position + Material

elementEntity {

派生类

球；

椭球；

立方体；

圆柱；

多面棱柱；

包括Element本身的信息，Element对应的shader（指针），element的Position，element的Texture，Element的Material，Element的运动信息。

实现的功能，接受外部的camera信息和light信息，绘制Element

注意：状态机的clean，所有绑定一定要解绑，所有的空间都要free

}

Application 读取场景配置 json 文件，json文件描述：

1. 场景背景天空盒（六个面的材质的路径）
2. 场景初始Camera，位置，朝向
3. 场景初始光源，有几个，每个的类别（Direct还是Spot）位置，朝向，亮度，颜色。
4. 场景内有多少Entity
5. 对于每一个Entity，指定类别（是Model还是Element）
6. 对于Model 类型的，指定obj文件路径，指定scale，指定rotation，指定初始position，指定材质路径，指定材料性质，指定初速度，指定加速度
7. 对于element 类型，指定Element的类别，根据不同的类别，指定对应的描述参数。指定scale，指定rotation，指定position，指定材质路径，指定材料性质，指定初速度，指定加速度。

Application控制所有Entity和场景信息 {

场景漫游，控制Camera

控制灯光信息

计算每一帧每一个Entity的位置，朝向，缩放

（实现一个简单的物理引擎）

提供接口给上层使用

}

Application 直接调用背景

ElementEntity + 派生 酸奶

ModelEntity + Application（除物理引擎）

迁移到iOS + 简单UI 添砖

曲面建模（Blender）+ Application（物理引擎）

背景是一个 skybox

背景单独一个类，一个文件组织

---

长期

blender

