# 软渲染器需求文档

|  |  |  |
| --- | --- | --- |
| 修改时间 | 编写人 | 备注 |
| 2018 8 5 | 刘想 | 开始编写 |
| 2018 9 2 | 刘想 | 对类图进行设计 |
| 2018 9 5 | 刘想 | 对对象图进行修改 |
| 2018 9 8 | 刘想 | 对测试进行相关扩充 |
| 2018 9 10 | 刘想 | 对文档进行最后总体审核 |
|  |  |  |

目录

[软渲染器需求文档 1](#_Toc524346807)

[1. 序 3](#_Toc524346808)

[1.1编写目的 3](#_Toc524346809)

[1.2编写背景 3](#_Toc524346810)

[1.3定义 3](#_Toc524346811)

[1.4参考资料 3](#_Toc524346812)

[2. 软件产品的一般性描述 3](#_Toc524346813)

[2.1运行环境与资源 3](#_Toc524346814)

[2.2产品的功能描述 3](#_Toc524346815)

[3功能行为需求 4](#_Toc524346816)

[3.1引言 4](#_Toc524346817)

[3.2业务需求功能模型 4](#_Toc524346818)

[3.3活动图 7](#_Toc524346819)

[3.4类图 9](#_Toc524346820)

[4.性能需求 10](#_Toc524346821)

[4.1数据精度 10](#_Toc524346822)

[4.2时间特性 10](#_Toc524346823)

[5.附录 10](#_Toc524346824)

## 序

### 1.1编写目的

本文档对软件的基本功能实现进行简单描述，以及对基本设计进行描述，主要面向于开发人员对于开发时代码的编写。

### 1.2编写背景

### 1.3定义

### 1.4参考资料

《计算机图形学 第四版》《 computer graphics with Opengl》

## 软件产品的一般性描述

### 2.1运行环境与资源

1.采用基本io库在不配置软件环境下可以在linux，windows环境下运行

2.不使用GPU资源。对于能正常运行的基本计算机硬件系统即可

### 2.2产品的功能描述

产品主要实现简单的光线跟踪，仅仅使用cpu进行软渲染，不涉及硬件，拥有较好的跨平台，使用标准io库。

## 3功能行为需求

### 3.1引言

### 3.2业务需求功能模型

#### 3.2.1顶层用例模型



1.“光线跟踪用例

作用：对所要显示的图进行计算

2.交互显示输出

”作用：用户和计算机的接口，主要对渲染的图形进行显示

#### 3.2.2第一层用例模型



1. 光线与几何体求交

目的：求交运算

1. 求反射光线

目的：反射光线

1. 生成初始视线

目的：生成眼睛到，视平面某个像素的视线向量

1. 计算反射光强

目的：通过光的数据，和几何体的数据生成几何体的的phong模型的光强



### 3.3活动图

#### 3.3.1总体活动图



#### 3.3.2光线跟踪活动图

#### 

### 3.4类图

#### 3.4.1数学相关mymath



#### 3.4.2几何体相关myGeometry3d



#### 3.4.3光照



#### 3.4.4设备类mydevice



### 3.5时序图

#### 3.5.1初始化过程



#### 3.5.2渲染过程



## 4.性能需求

### 4.1数据精度

### 4.2时间特性

渲染帧数2帧

运行截图：

## 附录

### 5.1测试相关

相关数据解释

1. myVec向量用4值表示，(x,y,z,r)：r表示是点还是向量，点r=1,向量r=0
2. 颜色也用4值表示，（x，y，z，r）：分别别表示（r，g，b，a）；

#### 5.1.1输入数据

场景生成数据：

myDevice<float> dev;

//设置场景球体红色

mySphere<float> phere(myVec4<float>(0,200, 0, 1),200);

//box是绿色

myCube<float> cubeDown(myVec4<float>(-400, -20, -400, 1), myVec4<float>(400, 0, 400, 1));

//设置颜色

phere.color = myVec4<float>(1, 1, 1, 0);

cubeDown.color = myVec4<float>(0, 1, 0, 0);

myVec4<float> tempDir(0, -1, 0, 0);

tempDir = tempDir.getNormal();

scence.Light = myLightRay<float>(myVec4<float>(0, 600, 0, 1), tempDir, myVec4<float>());

注释：

球体：坐标为（0，200，0），半径200，球体是白色（1，1，1，0）

长方体：左下角坐标（-400，-20，-400）右上角坐标（400，0，400），绿色（0，1，0，0）

光照：矢量光从上往下照射（0，-1，0，0）

相机数据：

myCamera<float> ca(myVec4<float>(0,300,-800,1), myVec4<float>(0,300,-400,1), myVec4<float>(0, 1, 0, 0),800,600);

//视平面网格

myViewMesh<float> viewMesh(800, 600, 800, 600, ca);

相机：视点（0，300，-800，1）视平面点中点（0，300，-400，1）

视平面：视平面大小，分辨率都是800X600

结果：

