# 软件需求规格说明书

目录

[软件需求规格说明书 1](#_Toc514252327)

[引言 1](#_Toc514252328)

[1.1编写说明 1](#_Toc514252329)

[1.2背景 1](#_Toc514252330)

[1.3参考资料 1](#_Toc514252331)

[2.任务概述 2](#_Toc514252332)

[2.1目标 2](#_Toc514252333)

[2.2用户的特点 2](#_Toc514252334)

[3.需求规定 2](#_Toc514252335)

[3.1对功能的规定 2](#_Toc514252336)

[4.运行环境规定 3](#_Toc514252338)

[4.1运行模块组合 3](#_Toc514252339)

[6.总结与展望 4](#_Toc514252340)

## 引言

### 1.1编写说明

本次设计的内容是太阳系概貌模拟系统。主要功能包括近日、远日行星的显示，放大、缩小、左、右、上、下移动功能，多视角观察太阳系概貌，行星运行、静止以及初始状态复原。在这个系统中有二种视角来观看太阳系分别为俯视和侧视。每种视角下都可以实现放大、缩小、平移、静止和运行功能。 实现此系统的一个难点就在于对行星运动的处理，想让行星在轨道上动起来就要知道这个行星下个时刻的坐标是什么，在放大、缩小和上、下、左、右移动轨道后运行轨道的参数相应变化了，要想使行星随着轨道的变动还能准确的在轨道上运行的话，就得对行星下个位置的坐标的计算公式做变型，也就是加上或减去一些偏差值。

本系统比较好的完成了所要求的任务，综合考虑了用户使用的方便，尽量做到了界面的美观和功能的实用。

### 1.2背景

随着计算机3D图形学和虚拟现实技术的飞速发展，除了可以精确描述宇宙运转机理的物理学公式外，计算机仿真也是辅助人们对宇宙世界进行探索的重要工具。太阳系，作为人类研究过的第一个也是最透彻的星系，更是计算机虚拟现实技术热切关注的话题。不仅如此，精确的太阳系仿真模型还可辅助于相关日常教学。

### 1.3参考资料

[1]美 赫恩等著，蔡士杰，宋继强，蔡敏译.计算机图形学.第三版，出版社.电子

工业出版.2005-6-1.

[2] (美)冈萨雷斯.数字图像处理(第2版英文版).出版社.电子工业出版社.2007-09-01.

[3] 美.坎宁安著.计算机图形学英文版.出版社.机械工业出版社.2008-5-1.

[4] 孙正兴.周良.郑宏源编著.计算机图形学基础教程.出版社:清华大学出版2004-7-1.

## 2.任务概述

### 2.1目标

这次我们所要做的太阳系概貌的模拟系统就是用来模拟太阳系运行是最主要最基本的特征，实现太阳系各行星运动概貌的模拟。

### 2.2用户的特点

（1）对软件的实用性要求较高；

（2）对界面设计大方直观有较高要求；

（3）要求软件操作简单；

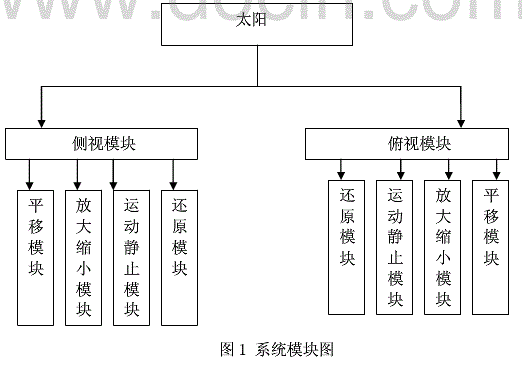
（4）要求软件功能较为全面。

## 3.需求规定

### 3.1对功能的规定

第一：系统是单用户的，这里的单用户是指不需要用户名和密码，用户进入系统可以直接对系统进行操作。第二：系统有多视角。第三：太阳系是可以放大、缩小的，并且保证放大、缩小后行星照样能正确地运行。第五：有专门的初始化工具。第六：有对应的工具按钮。

系统为模拟系统，外观要逼真才比较好，所以我们需要从网上下载关于行星的图片，这样看起来比较逼真。真实的太阳系是三维空间的，不同的视角下观看的结果是不一样的，所以在系统要实现从两种视角来观看太阳系，分别为俯视和侧视。这两种视角看到太阳系的感觉是完全不一样的。选定了一种视角后就可以在这种视角下观看太阳系各行星的运行，当然也可以放大，缩小，左（右，上，下）移动整个太阳系，在太阳系运行过程中，当然也可以暂停行星的运行，使太阳系处于静止状态，再次点击运行，行星会接着动起来。如果想回到最初的状态，也可以点一下复原按钮，整个太阳系又回到了此视角下的最初始的状态。如果想换个视角来看太阳系，直接点击你想要的视角的按钮，这两种视角下可以行使的功能是一样的。根据功能的划分，我们将系统划分为以下9个功能模块和两大界面，以及在工具栏上设置工具按钮。



## 4.运行环境规定

## 4.1运行模块组合

我们将行星的运动放在辅线程中，若需让行星运行的时候创建辅线程，若想让运动的行星暂停下来则只需撤销辅线程。若要保证行星在运行的时候准确地在轨道上，那么就要不停的计算下个时刻行星位置的坐标，轨道是个椭圆，我们可以有现成的数学公式用来计算，但是放在或缩小或平移后椭圆参数会发生变化，此时我们就需要让各参数加上或减去一些值，保证在轨道发生改变后那些行星依然找得到自己的路。

5.系统数据结构设计

每个图形都有自己的位置坐标，在程序中的不同功能模块中会被不断改变，必须通过变量来标识这些坐标。图片的加载涉及到图片的加载位置和图片大小的问题，而只要知道了图片的左上角和右下角的坐标，那么图片的大小自然也就可以计算出来了。于是采用如下的数据结构：

int left0,top0;//太阳外接矩形的左上角坐标

int left01,top01;//第1个行星的左上角坐标

int left02,top02;//第2个行星的左上角坐标

int left03,top03;//第3个行星的左上角坐标

int left04,top04;//第4个行星的左上角坐标

int left05,top05;//第5个行星的左上角坐标

int left06,top06;//第6个行星的左上角坐标

int left07,top07;//第7个行星的左上角坐标

int left08,top08;//第8个行星的左上角坐标

int left1,top1,right1,bottom1;//最里面轨道的外接矩形的左上角和右下角的坐标

int left2,top2,right2,bottom2;//第2条轨道外接矩形的左上角和右下角的坐标

int left3,top3,right3,bottom3;//第3条轨道外接矩形的左上角和右下角的坐标

int left4,top4,right4,bottom4;//第4条轨道外接矩形的左上角和右下角的坐标

int left5,top5,right5,bottom5;//第5条轨道外接矩形的左上角和右下角的坐标

int left6,top6,right6,bottom6;//第6条轨道外接矩形的左上角和右下角的坐标

int left7,top7,right7,bottom7;//第7条轨道外接矩形的左上角和右下角的坐标

int left8,top8,right8,bottom8;//第8条轨道外接矩形的左上角和右下角的坐标

int upoffset;//向上或下移动距离

int leftoffset;//向左或右移动的距离

int wideoffset;//放大缩小时宽度增加的大小

int highoffset;//放大时上下增加的大小

int highoffset1;//第一上轨道放大缩小时高度更改的大小

int fushitoceshi//标记此时界面是侧视还是俯视为0表示侧视为1表示俯视

## 5.总结与展望

这个系统基本实现了任务书上的所有功能，但是它还是有很多不足的。首先，它模拟的不是特别逼真，不像专业的太阳系模拟器那样立体那样美观。其次，由于我们的编程经验不是太多，写出来的代码没有优化非常占用CPU资源，以后这方面我们要注意。还有在行星运行过程中放大缩小轨道，行星就会出现找不着轨道，这是我们没有对全局变量加锁，如果将全局变量设为排斥区就可以解决这个问题，以后还得多加学习这方面的知识。我们的程序运行的时候占用了百分之六十的CPU资源，说明我们的程序对内存的利用非常不合理，以后我们需要在这方面多下功夫，这样写的程序才会比较好。