**F4+2 团队项目软件设计方案**

目录

1引言

1.1编写目的

1.2背景

1.3定义

1.4参考资料

2总体设计

2.1需求规定

2.2运行环境

2.3基本设计概念和处理流程

2.4功能描述

2.5功能器求与程序的关系

2.6人工处理过程

2.7性能

2.8尚未问决的问题

3接口设计

3.1用户接口

3.2外部接口

3.3内部接口

4运行设计

4.1运行模块组合

4.2运行控制

4.3运行时间

5系统数据结构设计

5.1逻辑结构设计要点

5.2物理结构设计要点

5.3数据结构与程序的关系

6测试计划

6.1测试目的

6.2测试启动与结束准则

7系统出错处理设计

7.1出错信息

7.2补救措施

7.3系统维护设计

**1引言**

**1.1编写目的**

本阶段完成系统的大致设计并明确系统的数据结构与软件结构。本概要设计说明书的目的就是进一步细化软件设计阶段得出的软件概貌，把它加工成在程序细节上非常接近与源程序开发的软件表示。

预期读者：软件测试员、程序开发员、软件分析员

**1.2背景**

说明：

a.       待开发软件系统的名称；动态的太阳系模型

b.       列出此项目的任务提出者、开发者、用户以及将运行该软件的计算站（中心）。

            项目任务提出者：马仲山

项目任务开发者：马仲山，马绍辉，马世芳，张俊逸，马婧，马婧  
 用户：初一学生

实现该软件的计算机网络：校园网

**1.3定义**

逻辑结构：简称数据结构，用来实现用户观点的系统结构的表示。

 管理员：系统的最高级权限的用户

**1.4参考资料**

《软件工程导论》 张海潘 牟永敏 编著 清华大学出版社

**2总体设计**

**2.1需求规定**

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。第四：有专门的初始化工具。第五：有对应的工具按钮。

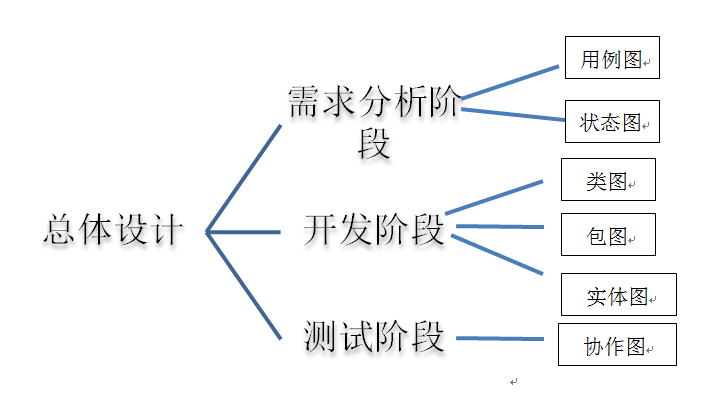
系统为模拟系统，外观要逼真才比较好，所以我们需要从网上下载关于行星的图片，这样看起来比较逼真。真实的太阳系是三维空间的，不同的视角下观看的结果是不一样的，所以在系统要实现从两种视角来观看太阳系，分别为俯视和侧视。这两种视角看到太阳系的感觉是完全不一样的。选定了一种视角后就可以在这种视角下观看太阳系各行星的运行，当然也可以放大，缩小，左（右，上，下）移动整个太阳系，在太阳系运行过程中，当然也可以暂停行星的运行，使太阳系处于静止状态，再次点击运行，行星会接着动起来。如果想回到最初的状态，也可以点一下复原按钮，整个太阳系又回到了此视角下的最初始的状态。如果想换个视角来看太阳系，直接点击你想要的视角的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

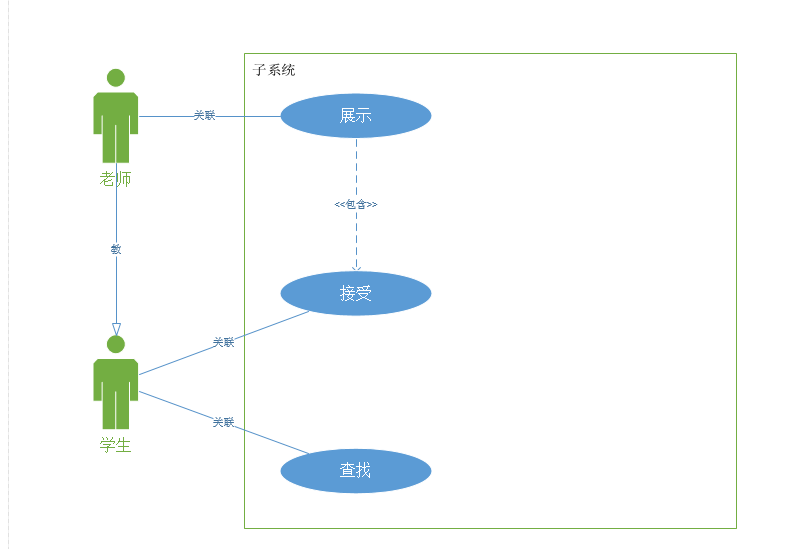
**2.2运行环境**

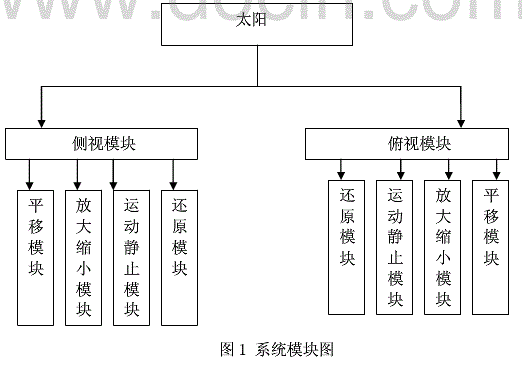
操作系统在windowXP以上

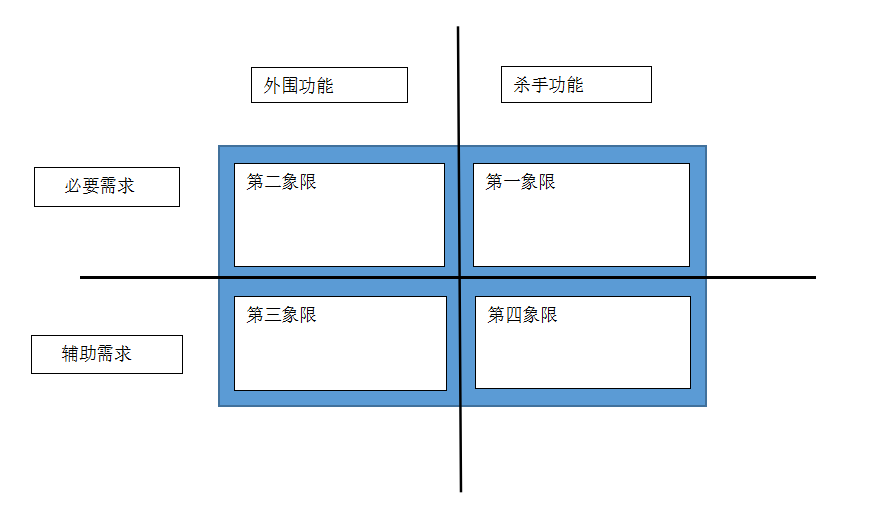
客户端要配备JDK和SQL等应用开发软件

**2.3基本设计概念和处理流程**

****







## ****2.4****功能描述

1.系统有多视角。

2.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。

模拟系统界面逼真且美观，功能实用。

3.用户进入系统可以直接对系统进行操作。

4.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。

**2.5功能器求与程序的关系**

本条用一张如下的矩阵图说明各项功能需求的实现同各块程序的分配关系：

|  |  |  |
| --- | --- | --- |
|  | 教师模块 | 学生模块 |
| 单用户 | √ | √ |
| 多视角 | √ | √ |
| 任意放大缩小 |  | √ |
| 初始化工具 |  | √ |

**2.6人工处理过程**

需要时刻的备份数据，以上不可抗拒因素导致数据的丢失。

**2.7性能**

性能：单用户，多视角，任意放大缩小

1.单用户：

用户进入系统可以直接对系统进行操作。

2.多视角：系统有多视角。可多视角观察太阳系概貌，行星运行、静止以及初始状态复原。

3.任意放大缩小： 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。模拟系统界面逼真且美观，功能实用。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。

**2.8尚未解决的问题**

本软件项目还有很多功能没有实现，有些操作过于复杂，希望以后可以简化。

**3接口设计**

**3.1用户接口**

应用程序接口是一组定义、程序及协议集合，通过API 接口实现计算机软件之间的相互通信。目前主流API 接口的有SGI 开发的OpenGL和微软开发的DirectX。两者分别有各自的优势，OpenGL可跨平台使用, 具有高度移植性并且是最为广泛使用的2D/3D 图形 API。DirectX 仅局限于Windows平台，常应用于游戏领域，除了具有图形API，另还有声音、输入、网络等API 接口。本系统中仅用到图形API 接口，同时需要考虑仿真系统跨平台后的适用性，因此本系统最终选择OpenGL 图形库接口。

**3.2外部接口**

暂无

**3.3内部接口**

程序内的各个模块之间采用函数调用、参数传递、返回值得方式进行信息传递。

**4运行设计**

**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。

**4.2运行控制**

**运动控制作为系统基础功能，通过模型变换控制天体位置、天体自转及公转速度以实现天体运动的效果。为增加系统的生动性及真实感，在每个场景中均表现出天体运动场景。**

**动画演示模块**

1. 公转自转动画模块：点击按钮，观看公转自转教学演示动画。
2. 行星介绍动画模块：点击按钮，观看行星介绍教学演示动画。
3. 配音及背景音乐模块：配音辅助学生理解记忆。

**4.3运行时间**

24小时都可运行

**5系统数据结构设计**

**5.1逻辑结构设计要点**

给出本系统内所使用的每个数据结构的名称、标识符以及它们之中每个数据项、记录、文卷和系的标识、定义、长度及它们之间的层次的或表格的相互关系。

**5.2物理结构设计要点**

（1）八大行星围绕太阳的公转。

（2）八大行星和太阳的自转。

（3）个性化的人工界面。

**5.3数据结构与程序的关系**

通过在数据库内建立主外键来关联这些数据。

**6测试计划**

**6.1测试目的**

测试该软件能否正常运行，需要的功能是否已经全部实现。

**6.2测试启动与结束准则**6.2.1  测试启动准则  
  系统测试的启动准则(应同时满足)  
  上一轮系统测试所有发现的缺陷均被处理;  
  上一轮系统测试所发现的应被修正的缺陷已被修正完成;  
  测试版本已经过开发人员的冒烟测试;  
  测试版本内容符合版本规划;  
6.2.2  测试结束准则  
  实行了该阶段所有的测试策略并达到结束标准。  
  所有本轮的测试用例都被执行过。

6.2.3 测试暂停/中止标准

发现有重大设计或编码问题，其导致流程性阻塞;

发现有重大设计或编码问题，其导致20%以上的测试用例无法执行或执行无意义，测试相关特性已经无意义;

发现有重大设计或编码问题，其修改会导致30%以上的函数接口,功能等的数量级的变化测试相关特性已经无意义;

项目再启动时，测试进度重新安排或顺延。

如有新的项目需求，则在原测试计划下做相应的调整。

若项目中止，则对已完成的测试工作做测试活动总结。

**7系统出错处理设计**

**7.1出错信息**

|  |  |
| --- | --- |
| 错误类型 | 错误提示 |
| 输入信息不合法 | 输入信息不正确，请再次输入 |
| 程序运行错误 | 给予相应的出错提示 |

**7.2补救措施**

 a.当出现断电或者电脑出现故障（死机、蓝屏）的话，可以通过以前备份的形式对数据进行回复。

 b.当用户需要在此增加功能的时候，可以通过在原系统的基础上进行扩充。

**7.3系统维护设计**

    1）在数据库设计的时候，学生信息与学生上机、学生充值、学生退卡记录都不在一个表中，防止信息删除或更新的时候，信息的不一致重叠罗嗦。

    2）在程序内部通过建立模块，不同窗体调用一个模块的形式进行设计，减少了代码的操作量，提高了运行的速度。