太阳系系统

软件需求规约

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2011-07-06 | <0.1> | 建立 | 马仕青 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

目录

[1. 简介 4](#_Toc297731948)

[1.1 目的 4](#_Toc297731949)

[1.2 范围 4](#_Toc297731950)

[1.3 定义、首字母缩写词和缩略语 4](#_Toc297731951)

[1.4 参考资料 4](#_Toc297731952)

[1.5 概述 4](#_Toc297731953)

[2. 整体说明 4](#_Toc297731954)

[3. 具体需求 5](#_Toc297731955)

[3.1 功能 5](#_Toc297731956)

[3.1.1 <Use case 图> 5](#_Toc297731957)

[3.1.2 <Use case1 规约> 5](#_Toc297731958)

[3.1.3 <Use case2 规约> 5](#_Toc297731959)

[3.2 可用性 5](#_Toc297731960)

[3.2.1 <可用性需求一> 5](#_Toc297731961)

[3.3 可靠性 5](#_Toc297731962)

[3.3.1 <可靠性需求一> 6](#_Toc297731963)

[3.4 性能 6](#_Toc297731964)

[3.4.1 <性能需求一> 6](#_Toc297731965)

[3.5 可支持性 6](#_Toc297731966)

[3.5.1 <可支持性需求一> 6](#_Toc297731967)

[3.6 设计约束 6](#_Toc297731968)

[3.6.1 <设计约束一> 7](#_Toc297731969)

[3.7 联机用户文档和帮助系统需求 7](#_Toc297731970)

[3.8 接口 7](#_Toc297731971)

[3.8.1 用户界面 7](#_Toc297731972)

[3.8.2 硬件接口 7](#_Toc297731973)

[3.8.3 软件接口 7](#_Toc297731974)

[3.8.4 通信接口 7](#_Toc297731975)

[3.9 适用的标准 7](#_Toc297731976)

软件需求规约 (简化版)

# 简介

## 目的

本文档是太阳系系统SRS文档。试图从整体上给出太阳系系统的轮廓。然后从具体说明该系统的功能需求，性能需求以及相关的非功能需求进行了详细的说明，同时，对本文档中没有包含的内容都有相关的文档作为补充。其中对功能需求的描述使用了UML图的形式，给出了较为直观的用例图和较为完整的事件流包括备用流等。

本文档是需求和规格的说明，是整个系统设计的基础，同时也是测试用例编写和系统测试的主要依据。同时，该文档也是用户最终确定软件需求的依据。

## 范围

此文档使用与开发的太阳系系统软件。本软件包括一个windows下的程序，需要试下如下的功能：

1. 模拟太阳系的各大行星和卫星的运行。
2. 计算出现日全食的时间等天文现象。
3. 模拟日全食和“行星连珠”等天文学的现象。

本文档定义了太阳系系统的开发目的，数据字典以及相关的功能性和非功能性需求。同时包含了设计的用例模型等。因此，用例的实现，测试用例的选取，系统的设计等设计文档以及相关设计文档、前景文档等都需要和本文档保持一致。

## 定义、首字母缩写词和缩略语

1. 日全食：是日食的一种，即太阳被月亮全部遮住的天文现象。如果太阳、月球、地球三者正好排成或接近一条直线，月球挡住了射到地球上去的太阳光，月球身后的黑影正好落到地球上，这时发生日食现象。 在地球上月影里的人们开始看到阳光逐渐减弱，太阳面被圆的黑影遮住，天色转暗，全部遮住时，天空中 可以看到最亮的恒星和行星，几分钟后，从月球黑影边缘逐渐露出阳光，开始生光、复圆。由于月球比地球小，只有在月影中的人们才能看到日全食。
2. 行星连珠：对于“行星连珠”现象，至今并没有一个严格的定义，通常用肉眼望去，行星差不多处在一条直线上，人们就称之为“行星连珠”。
3. 太阳系：太阳、行星及其卫星与环系、小行星、彗星、流星体和行星际物质所构成的天体系统及其所占有的空间区域。
4. 行星： 行星通常指自身不发光，环绕着恒星的天体。其公转方向常与所绕恒星的自转方向相同。一般来说行星需具有一定质量，行星的质量要足够的大且近似于圆球状，自身不能像恒星那样发生核聚变反应。

## 参考资料

Roger S.Pressman 著 《软件工程——实践者的研究方法》（原书第6版）郑人

杰、马素霞、白晓颖 等译

中华人民共和国国家标准——计算机软件文档编制规范》 GT/T 8567—2006

百度百科：baike.baidu.com

## 概述

本文档首先从整体上给出整个太阳系系统的轮廓。然后从具体说明该系统的功能需求，性能需求以及相关的非功能需求进行了详细的说明，同时，对本文档中没有包含的内容都有相关的文档作为补充。其中对功能需求的描述使用了UML图的形式，给出了较为直观的用例图和较为完整的事件流包括备用流等。

本文档根据用户需求的实际情况，进行动态的更新和变更。

# 整体说明

1. 本太阳系系统的总体效果：
2. 本太阳系系统功能主要有：模拟太阳系系统的运行；计算日全食等天文学现象；计算和模拟日全食和“行星连珠”等天文学现象。详细的在下面的需求中会有说明。
3. 本太阳系系统的用户以及他们的用户特征为：
4. 本系统开发的约束：
5. 本系统的假设和依赖：
6. 本系统的需求子集：

# 具体需求

## 功能

本程序主要是实现对太阳系系统运行状态的模拟和各种太阳系系统现象的计算。主要实现的是正常的状态和两种特殊的现象：日全食和行星连珠。Use Case图如下所示：

### <Use case1 规约>：模拟正常情况下运动的行星

简介：模拟正常情况下运动的各个太阳系系统状态。

正常事件流：

1. 用户打开本软件界面。主窗口的界面嵌套了小窗口，其中就是各大行星按照一定的规律运动。
2. 用户可以选择在其中进行放大、缩小，调整视角，调整观察的视野，暂停和还是等操作。这一步也可以跳过。
3. 用户可以选择变换时间，改变当前显示的行星运动的状态。随后跳到2。
4. 用户停止相关操作。

异常事件：

1. 异常步骤：

处理办法：

1. 异常步骤

处理办法：

1. 异常步骤：

处理办法：

### <Use case2 规约>： 模拟日全食

简介：模拟发生日全食的情况。

正常事件流：

1. 用户打开本软件，可看到正常状态下的太阳系的运动状态。
2. 用户选择了观看日全食的时间。
3. 系统开始模拟发生日全食的过程。
4. 用户选择观看发生日全食的视角和场景等。

异常事件流：

1. 异常步骤：

处理办法：

1. 异常步骤

处理办法：

1. 异常步骤：

处理办法：

### <Use case3 规约>： 模拟日全食

简介：模拟发生日全食的情况。

正常事件流：

1. 用户打开本软件，可看到正常状态下的太阳系的运动状态。
2. 用户选择了观看日全食的时间。
3. 系统开始模拟发生日全食的过程。
4. 用户选择观看发生日全食的视角和场景等。

异常事件流：

1. 异常步骤：

处理办法：

1. 异常步骤

处理办法：

1. 异常步骤：

处理办法：

## 可用性

本系统的设计完全符合最基本的要求，同时最大程度的降低用户的操作复杂度。典型情况，一般的用户和专家用户无需经过培训即可使用。操作较NASA官网系统相似。

### <可用性需求一>

/\*\*

[在此给出需求说明。]

\*/

## 可靠性

/\*\*

[对系统可靠性的需求应在此处说明。以下是一些建议：

• 可用性—指出可用时间百分比 ( xx.xx%)、使用小时数、维护访问权、降级模式操作等。

• 平均故障间隔时间 (MTBF) – 通常表示为小时数，但也可表示为天数、月数或年数。

• 平均修复时间 (MTTR) — 系统在发生故障后可以暂停运行的时间。

• 精确度 — 指出系统输出要求具备的精密度（分辨率）和精确度（按照某一已知的标准）。

• 最高错误或缺陷率—通常表示为每千行代码的错误数目 (bugs/KLOC) 或每个功能点的错误数目 (bugs/function-point)。

• 错误或缺陷率—按照小错误、大错误和严重错误来分类。需求中必须对“严重”错误进行界定，例如：数据完全丢失或完全不能使用系统的某部分功能。]

\*

本系统可用时间在24小时左右。MTBF大于24小时，MTTR不超过半个小时。程序代码中达到一般的标准以上，Bug数目小于25个/KLOC。每个功能点的错误数数目小于5个。

对错误的分类：

1. 严重错误：数据丢失、错误，功能不完整，或是错误。不满足需求等错误。
2. 大错误：程序出现宕机，没有反应等现象。
3. 小错误：系统反应时间长，操作设计不合理，界面设计不合理。

### <可靠性需求一>

/\*\*

[需求说明。]

\*/

## 性能

/\*\*

[此节应概述系统的性能特征。其中需包括具体的响应时间。如果可行，按名称引用相关用例。

• 对事务的响应时间（平均、最长）

• 吞吐量，例如每秒处理的事务数

• 容量，例如系统可以容纳的客户或事务数

• 降级模式（当系统以某种形式降级时可接受的运行模式）

• 资源利用情况，如内存、磁盘、通信等

\*/

系统的最长响应时间不超过一分钟。平均时间在10秒之内。

系统占用资源，磁盘不超过100M，内存不超过0.5G。

### <性能需求一>

/\*

[在此给出需求说明。]

\*/

## 可支持性

/\*\*

[此节应列出将提高所构建系统的可支持性或可维护性的所有需求，其中包括编码标准、命名约定、类库、维护访问权和维护实用程序。]

\*/

系统的设计应当以易于维护为目标。代码风格遵循一般的DMR-C风格。使用匈牙利命名法对变量进行命名。

### <可支持性需求一>

/\*\*

[在此给出需求说明。]

\*/

## 设计约束

/\*\*

[此节应列出所构建系统的所有设计约束。设计约束代表已经批准并必须遵循的设计决定。其中包括软件语言、软件流程需求、开发工具的指定用途、构架及设计约束、购买的构件、类库等。]

\*/

系统使用visual studio 2010开发，使用openGL。包括了openGL的辅助库。但都是开源，可以从网上直接下载的。无需购买。遵循一般的软件工程的工作开发流程。

### <设计约束一>

/\*\*

[在此给出需求说明。]

\*/

## 联机用户文档和帮助系统需求

/\*\*

[如果存在对联机用户文档、帮助系统、关于声明的帮助等的需求，请在此说明。]

\*/

## 接口

/\*\*

[此节规定应用程序必须支持的接口/界面。它应非常具体，包含协议、端口和逻辑地址等，以便于按照接口/界面需求开发并检验软件。]

\*/

### 用户界面

/\*\*

[说明软件将实现的用户界面。]

\*/

### 硬件接口

/\*\*

[此节指出软件所支持的所有硬件接口，其中包括逻辑结构、物理地址、预期行为等。]

\*/

### 软件接口

/\*\*

[此节说明软件系统中与其他构件之间的软件接口。这些构件可以是购入的构件、取自其他应用程序重新利用的构件，也可以是为此 **SRS** 范围之外的子系统开发，但该软件应用程序必须与之交互的构件。]

\*/

### 通信接口

/\*\*

[说明与其他系统或设备（如局域网、远程串行设备等）的所有通信接口。]

\*/

## 适用的标准

/\*\*

[通过引用，此节说明了所有适用的标准以及适用于所述系统的相应标准的具体部分。例如，其中可以包括法律、质量及法规标准；业界在可用性、互操作性、国际化、操作系统相容性等方面的标准。]

\*/