< 移动迷宫 >

软件需求规约

用于<子系统或特性>

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <07/04/2017> | <1.0> | <初稿修订> | <组员> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 项目的目标和成功准则 4

1.4 参考资料 4

1.5 概述 4

2. 目前系统 4

3. 建议的系统 4

3.1 概述 4

3.2 功能需求 5

3.2.1 <功能性需求一> 5

3.3 非功能需求 5

3.3.1 可用性 5

3.3.2 可靠性 5

3.3.3 性能 6

3.3.4 可支持性 6

3.3.5 设计约束 6

3.3.6 接口 6

3.3.7 法律、版权及其他声明 7

3.3.8 适用的标准 7

3.4 系统模型 7

3.4.1 场景 7

3.4.2 用例模型 7

3.4.3 对象模型 7

3.4.4 动态模型 7

3.4.5 用户界面 7

软件需求规约

# 简介

**软件需求规约** **(SRS)** 的简介应提供整个 **SRS** 的概述。它应包括此 **SRS** 的目的、范围、定义、首字母缩写词、缩略语、参考资料和概述。]

[注：软件需求规约(**SRS**) 记录对系统或系统的一部分的完整软件需求。 以下是一个典型的 **SRS** 概述，用于以传统的自然语言风格表达需求而**不涉及用例建模**的项目。它在一个文档中记录了所有的需求，而适用的部分可从补充规约（此后将不再需要）中插入。对于涉及用例建模的 **SRS** 模板（由包含用例模型的用例、适用的补充规约及其他支持信息的包组成），请参见 rup\_**SRS**-uc.dot。]

[**SRS** 可能有许多不同的组织方式。有关这些方式的进一步阐述以及 **SRS** 的其他结构组织方式，请参见 [IEEE830-1998]。]

## 目的

本文档的编写目的是对移动迷宫的软件需求进行描述和规约，为后续的系统分析、设计和实现工作奠定基础。文档将详细地定义系统的功能和非功能需求，获取场景和用例。本文档也用于和客户进行沟通，明确客户需求的细节。

## 范围

本文档适用的软件为：移动迷宫。

与该软件相关的特性、子系统、模型等均符合本文档中的内容。

## 项目的目标和成功准则

[本小节应提供正确理解此 **SRS** 所需的全部术语的定义、首字母缩写词和缩略语。 这些信息可以通过引用项目词汇表来提供。]

## 参考资料

（1）《面向对象软件工程——使用UML、模式与Java》（第3版），清华大学出版社，2011.

（2）IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998

## 概述

本文档包括引言、目前系统和建议的系统三部分。目前系统部分对当前潜入类游戏市场进行分析，指出其不足并给出本系统开发的必要性；建议的系统部分列举系统的功能需求，并从不同方面规定非功能需求。该部分还描述了系统的各种场景并归纳为系统的用例，如单人游玩、多人游玩等，并给出了初步的用户界面设计。

# 目前系统

# 建议的系统

**SRS** 的这一节应包含所有的软件需求，其详细程度应使设计人员能够设计出可以满足这些需求的系统，并使测试人员能够测试该系统是否满足这些需求。 当利用用例建模时，这些需求在用例和适用的补充规约中记录。如果没有利用用例建模，则可以将补充规约的概要直接插入此节。如下所示。]

## 概述

## 功能需求

[此节为以自然语言风格表达的需求说明为此设计的系统功能性需求。对于许多应用程序，此节会成为 **SRS** 包的主体部分，所以应仔细考虑此节的组织方式。此节通常按特性来组织，但也可能会有其他适用的组织方式，例如按用户或子系统组织的方式。功能性需求可能包括特性集、性能和安全性。

当利用应用程序开发工具（如需求工具、建模工具等）来获取功能性时，此节文档将引用获取相应数据的方法，并指出用来获取数据的工具的位置和名称。]

### <功能性需求一>

[需求说明。]

## 非功能需求

### 可用性

[此节应包括所有影响可用性的需求。例如，

• 指出普通用户和高级用户要高效地执行特定操作所需的培训时间

• 指出典型任务的可评测任务次数或根据用户已知或喜欢的其他系统确定新系统的可用性需求

• 指出在符合公认的可用性标准（如 IBM 的 CUA 标准和 Microsoft 的 GUI 标准）方面的需求]

#### <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。]

#### <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

#### <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。]

#### <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。]

#### <设计约束一>

[在此给出需求说明。]

### 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。]

### 法律、版权及其他声明

[此节说明软件涉及的所有必需的法律免责声明、保证、版权声明、专利声明、字标、商标或徽标符合性问题。]

### 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。]

## 系统模型

### 场景

[如果有补充的场景，在此处添加。]

### 用例模型

[如果有补充和修改的用例，在此处添加。]

### 对象模型

[分类别列出每一个类的介绍]

表 实体类定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 实体类名称 | 属性 | 关联类 | 定义 |
|  |  |  |  |
|  |  |  |  |

表 边界类定义

|  |  |
| --- | --- |
| 边界类名称 | 定义 |
|  |  |
|  |  |

表 控制类定义

|  |  |
| --- | --- |
| 控制类名称 | 定义 |
|  |  |
|  |  |

[列表方式介绍完类后，画出一张类图。]

### 动态模型

1. 系统顺序图与操作契约

(1)用例X

* 为每一个用例画出一张系统顺序图
* 为该用例中的系统消息定义操作契约

1. 顺序图

[围绕系统消息画顺序图，为其中每一个系统消息单独画一张顺序图。]

1. 状态图

[给出某些具备复杂状态的类的状态图。]

### 用户界面

[在需求获取阶段的基础上，进一步细化界面。]