<基于元宇宙平台的场景搭建系统>

软件需求规约

版本 <1.0>

[注：用方括号括起来并以蓝色斜体（样式=InfoBlue）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式=Body Text）。]

[要定制 Microsoft Word 中的自动字段（选中时显示灰色背景），请选择 File>Properties，然后将 Title、Subject 和 Company 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 Edit>Select All（或 Ctrl-A）并按 F9，或只是在字段上单击并按 F9，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 Alt-F9，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 Word 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <2/11/2022> | <1.0> | <初次编写> | <花泉润，叶鹏鹏，封琪> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 定义、首字母缩写词和缩略语 4

1.3 参考资料 4

2. 整体说明 4

2.1 产品总体效果 5

2.2 产品功能 5

2.3 用户特征 5

2.4 约束 5

2.5 假设与依赖关系 5

2.6 需求子集 5

3. 功能需求 5

3.1 <Use case 图> 6

3.2 <Use case1 规约> 7

3.3 <Use case2 规约> 7

3.4 <Use case3 规约> 8

4. 非功能需求 9

4.1 易用性 9

4.1.1 各任务培训时间 9

4.2 可靠性 9

4.3 性能 10

4.3.1 事务响应速度 10

4.3.2 资源利用情况 10

4.4 可支持性 11

4.5 设计约束 11

5. 其它产品需求 11

5.1 接口需求 11

5.1.1 用户界面 11

5.1.2 软件接口 12

5.1.3 通信接口 12

5.2 适用的标准 12

软件需求规约 (简化版)

# 简介

## 目的

[阐明此 **SRS** 的目的。**SRS** 应详细地说明所确定的应用程序或子系统的外部行为。它还要说明非功能性需求、设计约束以及提供完整、综合的软件需求说明所需的其他因素。]

该文档为软件需求规约文档，给出了本项目系统的整体结构和功能结构的概貌，从总体架构上描绘了整个系统的轮廓。文档中，将对本项目的功能需求、外部接口需求和其它非功能性需求进行详细描述。其中对功能需求的描述采用了 UML 的用例模型方式，并给出了直观的用例图。

文档中详尽说明了本项目的需求和规格，这些规格说明是进行设计的基础，也是编写测试用例和进行系统测试的主要依据，可以作为组员开发时了解软件实际需求的主要参考资料，保证了后续的开发工作能够正常开展。同时，该文档也是用户确定软件功能需求的主要依据。

## 定义、首字母缩写词和缩略语

[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。建议单独撰写项目术语表，本处只需要参见该术语表即可。]

|  |  |
| --- | --- |
| **术语** | **含义** |
| 场景 | 由一系列模型文件组成，可被用户编辑、加载、导出 |
| 资源 | 用于加载进场景的文件数据，包括视频，图像，模型等 |
| 序列化 | 对场景文件进行格式转换的操作，用于支持场景数据的导入导出 |

## 参考资料

[本小节应完整列出此 **SRS** 中其他部分所引用的任何文档。每个文档应标有标题、报告号（如果适用）、日期和出版单位。列出可从中获取这些参考资料的来源。这些信息可以通过引用附录或其他文档来提供。]

软件工程：面向对象和传统的方法（原书第8版），（美）Stephen R. Schach著，邓迎春、韩松等译，机械工业出版社，2018.

软件工程（原书第10版），（英）Ian Sommerville著，彭鑫、赵文耘等译，机械工业出版社，2019.

IEEE Software Requirements Specification Template

# 整体说明

[**SRS** 的这一节应说明影响产品及其需求的一般因素。本节并不列出具体的需求，而只是提供在第 3 节中详述的各种需求的背景，以使这些需求便于理解。所包括的内容有：【产品总体效果，产品功能，用户特征，约束，假设与依赖关系，需求子集】

## 产品总体效果

本项目“基于元宇宙平台的场景搭建系统”，将在由交大DALAB开发的元宇宙平台的客户端上，增加场景搭建的独立功能，使用户能够使用本地模型文件搭建场景后上传至平台并进行摆放调整，使其能够接受其它用户的参观。

## 产品功能

本项目支持用户在场景布置的操作界面上，进行模型上传、摆放的功能，实现所需场景的搭建。以及支持场景分享，让其他用户能够进行参观。通过该虚拟会议平台，用户能自由地进行学术探讨、论文分享等活动。

## 用户特征

本项目用户即为使用虚拟会议平台进行场景搭建的使用者，主要包括高校学生和教师，拥有相当的软件操作技能，应能够在本系统的界面指引下完成场景搭建的操作。

## 约束

在本项目的模型上传模块中会使用到unity的导入接口，能够支持读取的文件类型以Unity 2019.4.31所支持的文件格式为准。

## 假设与依赖关系

本项目将在DALAB的前期产品基础上进行开发。前期项目已完成虚拟形象的控制与场景内容的交互逻辑等部分，本项目要在不影响原有系统的基础上在平台客户端用户界面添加场景搭建的额外功能。

## 需求子集

本项目分为下述三个模块

1. 资源管理模块

该模块管理用于场景搭建使用的任何模型、图片、视频等媒体资源。通过该功能用户能够更直观地浏览资源列表，以及进行相关操作。

1. 场景编辑模块

该模块用于可视化场景当前状态，并提供视角操控、物体选中、属性更改等基本编辑功能，是本项目的核心需求。

1. 场景序列化模块

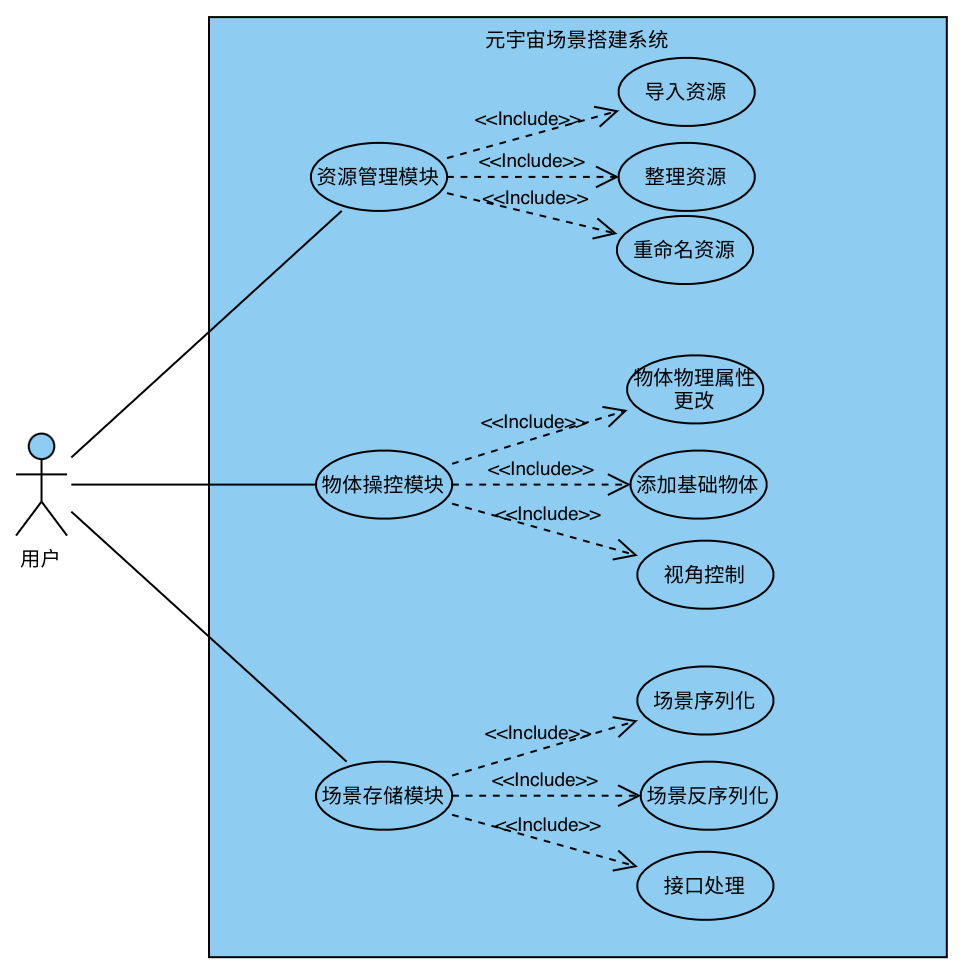
支持用户对制作的场景进行导入导出操作，增加了系统的可扩展性。

# 功能需求

[此节为以UseCase模型和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ＵseCase或特性来组织，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和安全性。]

## <Use case 图>

[画出Use case 图，并对每个actor或usecase有简要说明。如果系统比较大，则可按用户或子系统进行组织]



用户进行场景后可以进行一下三个模块的操作：

1. 资源管理模块。在资源管理模块中，用户可以进行导入资源、整理资源、重命名资源等操作。
2. 物体操控模块。在物体操控模块中，用户可以进行物体移动、旋转、缩放，添加基础物体以及视角控制操作。
3. 场景存储模块。在场景存储模块，用户可以进行场景的序列化与反序列化对制作好的场景进行导入导出。

## <Use case1 规约>

[对每个usecase 要有详细规约，说明其事件流等信息。]

**用例名称**

资源管理模块

**用例说明**

用户进行本地资源的上传、整理

**参与者**

所有用户

**基本流**

1. 用户上传本地资源
2. 系统判断资源格式是否支持
3. 系统加载上传的资源
4. 用户对资源进行修改、重命名等操作

**备选流**

1. 上传资源格式不支持

* 系统弹出错误信息，用户上传正确地资源格式

1. 用户修改违规

* 系统弹出错误信息，用户重新进行修改

**前置条件**

用户已经进入元宇宙搭建系统

**后置条件**

系统能正确地接受用户的操作信息

## <Use case2 规约>

**用例名称**

物体操控模块

**用例说明**

用户对物体进行移动、编辑等操作

**参与者**

所有用户

**基本流**

1、用户选择操控物体

2、用户对物体进行移动、旋转、缩放操作

3、系统接受信号进行物体属性处理

4、用户利用鼠标进行视角操控

5、系统接受信号进行视角处理

**备选流**

1、修改属性超出限制

* 系统弹出错误信息，用户重新进行修改

**前置条件**

用户已经进入元宇宙搭建系统

**后置条件**

系统能正确地接受用户的操作信息

## <Use case3 规约>

**用例名称**

场景存储模块

**用例说明**

用户对制作的场景进行导入导出

**参与者**

所有用户

**基本流**

1、用户制作场景

2、用户对场景进行导出

3、系统对场景进行序列化存储

4、用户导入场景

5、系统根据场景模块与系统平台的接口进行反序列化

6、加载导入的场景

**备选流**

1. 系统序列化导出失败

* 系统跳转到错误信息页面，用户重新修改

1. 系统反序列化导入失败

* 系统弹出错误信息，用户输入正确地反序列化信息

**前置条件**

用户已经进入元宇宙搭建系统

**后置条件**

系统能正确地接受用户的操作信息

# 非功能需求

## 易用性

### 各任务培训时间

本系统的大部分功能参考Unity的模式，而Unity的场景搭建功能非常易于上手：操作直观、界面控件易懂、辅助功能丰富、信息可视化程度高、记忆要求少，并且对多种功能提供敏捷方法（快捷键等）同时满足高级用户的快速操作需求。

#### 场景搭建任务

考虑普通用户的易用性，Unity编辑器对一系列场景搭建功能有提供纯鼠标操作，包括视角操控、物体拖拽等。如果用户完全未接触过立体场景搭建相关应用，需要一定时间熟悉视角的四种基本操作（普通平移、Pan平移、旋转、缩放）与物体的三种基本操作（平移，旋转，缩放），只需懂得其基本规则，可通过界面上的带图标按钮快速得知使用方法。高级用户则可从Unity、Unreal等三维物理引擎或Blender等三维建模软件的使用经验中快速识别界面控件的功能，并本系统将提供与Unity基本一致的场景控制相关快捷键。

#### 资源导入任务

本系统同样参考Unity的模式，提供四种方法：

1. 在窗口菜单栏选择”编辑”->”导入外部资源”
2. 在资源视图右键，菜单列表选择”导入外部资源”
3. 打开资源管理器，拖拽资源文件至本系统应用窗口内
4. 打开资源管理器，在本系统的目标资源目录里直接放入资源文件，后切回本系统应用窗口

预测普通用户从1、2、3，高级用户从2、3、4进行操作。

#### 物品行为设置任务

#### 本系统为安全与普通用户的易用性考虑，仅提供内置行为组件，因此无编写代码环节。

内置行为组件将显示在物体详细信息内（与Unity的Inspector视图相似），用户在信息视图滚动能浏览附加的内置行为的所有内容与参数设置。对于任何用户，需要先浏览本系统提供的所有行为组件有哪些内容，并尝试不同设置的效果。虽然会参考Unity对组件进行图标化的方式辅助用户理解，但后续会对这部分优先写出说明文档。

如果我们能提供足够多的预设物品，也将使用户借助这些物品上的行为组件得知其存在与使用效果，是一个更理想的培训过程。

[此节应包括所有影响易用性的需求。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易用性需求

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

## 可靠性

本系统对可靠性的规定为以下：

·可用时间为99.0%

·平均故障间隔时间为14天

·平均修复时间为4小时

·最高错误为0.5 bugs/function-point

·错误类型：

·严重：序列化、反序列化失败，造成场景无法保存/读入

·中等：资源文件无法导入，或材质贴图无法按预期显示

·小：场景搭建功能不正确

[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。]

## 性能

### 事务响应速度

除了场景搭建等显然可以立即完成的零碎任务外，资源导入是主要的拥有可见耗时的任务，耗时随文件大小、格式变化。目前期待值规定，在500MB以内的FBX格式模型资源文件，能以平均1.0秒的时间导入（参考自Unity的导入速度）。

### 资源利用情况

多数场景搭建任务占少量的内存读写，而资源导入任务会占用一定的磁盘读写量，这会将目标文件拷贝自资源收纳路径，该读写规模随用户导入的资源容量增加。

资源导入时可能会添加头文件作为文件、目录标识符，而标识符文件大小通常在1KB以内（参考自Unity）。

在上传完成的场景数据时，将占用网络通信，大小随场景的内容量以及使用资源的容量而增加。

[此节应概述系统的性能特征。其中需包括具体的响应时间。

• 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]

## 可支持性

编码标准：所有代码采用UTF-8编码

命名约定：C#默认格式

·类、属性、public函数名称头字母大写

·常量全字母大写

·以上规定外的名称头字母小写

·private成员变量加”\_”前缀

类库：C#自带库与Unity自带库

其它维护规定：

·对任何不支持多对一组件的类添加[DisallowMultiComponent]字段。

·对任何需暴露在Inspector但无需公开访问权的成员使用[SerializeField]字段，非必要避免使用public获得Inspector暴露。

·对getter / setter访问等级不同的成员使用属性。

[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。]

## 设计约束

本项目采用的开发方法为基于UML的面向对象方法，并且使用Unity引擎2019.4.31版本进行开发，编程语言为C# (.Net Framework 4.x) ，编程工具使用Visual Studio 2022，UML编辑软件使用PowerDesigner，测试工具使用UnityUPR。

[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。]

# 其它产品需求

## 接口需求

[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

#### 物体列表视图

此视图的列表以其名称表示场景内存在的所有物体，支持用户通过该列表重命名物体、选中物体、或更改排序。被选中的物体名称将被高亮。

#### 场景视图

此视图是场景搭建的核心区域，显示当前的相机视角下的三维场景情况。可通过该视图改变视角方向，选中物体，改变物体位置、旋转、缩放。可接受从该资源视图拖拽进新的模型。

#### 资源视图

此视图显示本系统资源收纳目录下的所有资源文件，可拖拽至场景视图内使用在搭建工作里。

#### 物体信息视图

此时图显示物体的名称、位置、旋转、缩放、与绑定的行为组件，各信息可在同一视图内进行更改。

[说明软件将实现的用户界面。]

### 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

本系统唯一通信行为是场景数据的上传，目前本系统所依赖项目的服务器模块正在调查，因此详细接口还在规定中。

[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

本系统有关易用性、互操作性、操作系统相容性等方面标准遵循于Unity引擎。

[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；业界在易用性、互操作性、国际化、操作系统相容性等方面的标准。]