前景（简化版）

# 简介

[此文档的目的是收集、分析和定义<<系统名>>的高层次需求和特性。它侧重于涉众和目标用户所需的功能以及这些需要存在的**原因**。<<系统名>>如何满足这些需要的详细情况记录在用例和补充规约中。]

[**前景**文档的简介应提供整个文档的概述。它应包括此**前景**文档的目的、范围、定义、首字母缩写词、缩略语、参考资料和概述。]

## 目的

[阐明此**前景**文档的目的。]

## 范围

[简要说明此**前景**文档的范围：它的相关项目，以及受到此文档影响的任何其他事物。]

## 定义、首字母缩写词和缩略语

[本小节应提供正确解释此**前景**文档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

[本小节应完整列出此**前景**文档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。]

## 概述

[此小节应说明**前景**文档中其他部分所包含的内容，并解释此文档的组织方式。]

# 定位

## 商机

[简要说明此项目面临的商机。]

## 问题说明

[提供一段说明，总结此项目正在解决的问题。可以采用以下格式：]

|  |  |
| --- | --- |
| 问题是 | [对问题进行说明] |
| 影响 | [问题影响的涉众] |
| 问题的后果 | [该问题会导致什么后果] |
| 成功的解决方案 | [应列出成功解决方案的一些主要优点] |

## 产品定位说明

[提供一段总体说明，高度概括产品将要在市场上占据的独特位置。可以采用以下格式：]

|  |  |
| --- | --- |
| 针对于 | [目标客户] |
| 谁 | [说明需要或机会] |
| 该（产品名） | 属于 [产品类别] |
| 功能 | [陈述主要优点，即促使人们购买的原因] |
| 不同于 | [主要的竞争产品] |
| 我们的产品 | [陈述主要的区别] |

[产品定位说明用于向所有相关人员传达应用程序的目的和项目的重要性。]

# 涉众和用户说明

[为有效地提供可满足涉众及用户实际需要的产品和服务，有必要在需求建模流程中确定并包括所有涉众。您还必须确定系统的用户，确保涉众群体能够充分代表这些用户。本节提供参与项目的涉众和用户的简介，以及他们希望通过所提议的解决方案来解决的关键问题。这里并不说明他们的具体请求或需求，因为这些内容将单独在涉众请求工件中记录。此处只提供之所以存在这些需求的背景和原因。]

## 市场统计

[总结促使您作出产品决策的关键消费者统计数据。说明并定位目标细分市场。估计市场的大小和增长率，估计的依据可以是潜在用户的数量，也可以是您的客户为满足您的产品或改进将要满足的需求所用资金的多少。了解行业大势和主流技术。回答以下战略性问题：

• 您的组织在这些市场的声誉如何？

• 您想获得什么样的声誉？

• 该产品或服务将如何支持您实现这些目标？]

## 涉众概要

[提供所有已确定涉众的一览表。]

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **角色** |
| 指明涉众类型。 | ［简要说明。］ | [简要说明他们在开发中的作用。  例如，确保某项工作顺利完成。] |

## 用户概要

[提供所有已确定用户的一览表。]

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **涉众** |
| 指明用户类型 | [简要说明他们在系统中的主要职责。] | [明确用户由哪个涉众来代表。  例如，由涉众 1.1 来代表］ |

## 用户环境

[详细说明目标用户的工作环境。以下是几项建议：

该任务由多少人来完成？是否总在变化？

一个任务周期需要多长时间？执行每项活动要用多长时间？是否总在变化？

是否有特殊的环境约束：移动、户外、乘机旅行等？

目前使用的是哪些系统平台？以后会使用哪些平台？

还在使用哪些应用程序？您的应用程序是否需要和这些应用程序集成？

在此处可以从业务模型中摘录一些内容来概述所涉及的任务和角色等等。]

## 关键的涉众/用户需要

[列出涉众认为现有解决方案存在的关键问题。对于列出的每个问题，需澄清以下要点：

• 为什么会出现这一问题？

• 目前如何解决该问题？

• 涉众需要什么样的解决方案？]

[务必要了解涉众或用户对解决各个问题的**相对**重视程度。分级和累积投票方法表明，**必须**解决的问题与涉众或用户希望解决的问题大有不同。]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **需要** | **优先级** | **关注的要点** | **目前的解决方案** | **提议的解决方案** | |
|  |  |  |  | |  |

## 备选方案和竞争

[确定涉众认为可以使用的备选方案。其中可能包括购买竞争对手的产品、自行设计解决方案，或者仅维持现状。列出已经存在或潜在的竞争产品。列出涉众认为各种竞争对手具有的主要优缺点。]

### <一个竞争对手>

### <另一个竞争对手>

# 产品概述

[此节高度概括产品的功能、与其他应用程序的接口以及系统配置。此节通常要包括以下三个小节：

• 产品总体效果

• 产品功能

• 假设与依赖关系]

## 产品总体效果

MY PC LOGO为独立的代码绘图软件，具有基于命令行与命令文件进行绘图的单机功能，以及双人单双海龟绘图、共享库函数与示例的联机功能。除此之外，还有丰富详尽的语法文档与程序样例帮助用户理解熟悉LOGO语言的语法与功能。

[**前景**文档的这一小节应将该产品放在其他相关产品环境和用户环境中进行介绍。如果该产品自成一体，应在此处说明。如果该产品是较大系统的构件，此小节则应说明这些系统如何进行交互，并确定系统之间的相关接口。要显示较大系统的主要构件、互连情况和外部接口，一种简单的方法就是通过框图来表示。]

## 功能摘要

[总结该产品将提供的主要优点和特性。例如，一个客户支持系统的**前景**文档可能会利用此部分来讨论存在问题的记录、消息传递和状态报告，而不必涉及每个功能的细节。

对功能加以组织，使客户或初次阅读该文档的其他人能够理解此功能列表。下面的简表列出了主要优点及支持的特性，该示例应足以说明问题。例如：]

**MY PC LOGO**

|  |  |
| --- | --- |
| **客户利益** | **支持特性** |
| 绘图能够分步完成而无需一次性完成 | 基于命令行的绘图 |
| 客户可以记录绘图命令，方便执行 | 基于命令文件的绘图 |
| 客户能够更加快捷地写出代码 | 智能的语法提示 |
| 代码结构更加清晰，方便客户回顾与调试代码 | 代码高亮 |
| 客户能够在协作下创作一幅图像 | 双人单双海龟绘图 |
| 方便客户进行模块化编程 | 库函数 |
| 方便用户发现与修改代码存在的BUG | 调试功能 |
| 客户能够分享与使用互联网上的库，有利于减少开发成本 | 共享库管理工具 |
| 智能地将自然语言翻译为代码，方便代码能力不足的客户（如未接触过编程的少年儿童）体验产品 | 利用自然语言通过语音方式进行绘画 |
| 给客户提供详细的语法标准，方便查阅 | 丰富准确的语法文档 |
| 方便客户学习语法，了解常用绘图方式 | 丰富的程序样例 |

## 假设与依赖关系

1. 利用自然语言通过语音方式进行绘画需要有机器学习的硬件平台支持，语音转文字的云服务支持。
2. 双人双海龟绘图、共享库管理工具需要有云服务器的支持。

[列出会影响**前景**文档中所述特性的所有因素。列出其变更将引起**前景**文档随之变化的假设。例如，有这样一项假设：将为该软件产品指定的硬件提供特定的操作系统。但如果没有提供该操作系统，就将需要更改**前景**文档。]

## 成本与定价

成本：

1. 机器学习硬件平台算力
2. 语音转文字云服务
3. 租用云服务器

定价：

免费

[对于向外部客户发售的产品和许多内部的应用程序，成本和定价问题会直接影响到应用程序的定义和施。在此节中，应记录任何相关的成本或定价约束。例如，分销成本（软盘的数量、光盘的数量、CD 制作）或其他商品销售的成本约束（手册、包装）可能对于项目的成功非常重要，也可能无关紧要，这取决于应用程序的性质。]

## 许可与安装

许可：

采用BSD的开源协议。本产品主要服务于想要进行代码入门（如青少年），以及想要通过代码绘制常规基于UI的绘图工具难以绘制图像的人群，不以盈利为目的，又由于小组人数与精力有限无法长期维护该产品，故采用在GitHub上开源的方式。又考虑到该产品商业化后或许能够为用户提供更好的服务，故采用商业友好的BSD协议，基于本产品的代码可以闭源。

安装：

客户端可以直接从互联网下载。

服务器端需要搭建好数据库与K8S集群后进行容器化部署。

[许可和安装问题也可能直接影响到开发工作。例如，如果需要支持串行化、口令安全或网络许可，则会增加在开发工作中必须予以考虑的系统需求。

安装需求还可能会影响到编码，或需要单独安装的软件。]

# 产品特性

[列出并简述产品的特性。特性是为让用户获益而必须具备的高级系统功能。每一项特性都是外部所需的服务，它通常需要一系列输入来实现预期的结果。例如，问题跟踪系统的特性是能够提供趋势报告。当用例模型成型后，更新这里的说明以指代用例。

由于**前景**文档将由各种各样的相关人员来复审，所以不应太过详细，应让所有人对此都有大致的了解。但是，应该向团队提供他们创建用例模型所需的必要详细信息。

要有效地管理应用程序的复杂性，对于任何新系统或对现有系统的增量部分，我们建议将功能提炼到较高的程度，这样 25 到 99 项特性较为合理。这些特性为产品定义、规模管理和项目管理提供了基础。每项特性的详细程度都将在用例模型中得到较深入的扩展。

贯穿此节的始终，都应能让用户、操作人员或其他外部系统从外部觉察到每项特性。这些特性应包括功能性的说明以及必须考虑的任何相关的可用性问题。以下原则将会适用：

• 避免设计。使特性说明保持一定的概括程度。侧重于说明所需的功能以及为什么要（而不是如何） 实现这些功能。

• 如果您使用的是 Requisite 工具包，应将需求类型选择为“所有”，以便于引用和跟踪。]

## 基于命令行的绘图

代码会以命令的形式分步执行。

## 基于命令文件的绘图

代码可以被记录在一个文件中，一次性执行。

## 智能语法提示

根据用户输入在下拉列表中动态给出相似的词汇，以及根据语法结构自动推断接下来的关键字，如if-else等。

## 5.4 代码高亮

通过构建代码的中间生成树，对关键字、变量、常量等不同代码元素进行色彩上的区分，有利于代码结构的清晰化。

## 5.5 双人单双海龟绘图

两名用户能通过命令行在互联网环境下共同绘图。

## 5.6 库函数

用户可以将代码打包成一个库方便调用。

## 5.7 调试

支持断点、单步执行、单步跳出、跨越执行、执行至断点等调试功能。

## 5.8 共享库管理工具

类似Python的pip工具，用户可以分享自己编写的库函数与使用他人共享的函数。

## 5.9 自然语言语音绘图

智能地将用户语言转化为绘图代码，实现用“嘴”绘图。

## 5.10 语法文档

对LOGO语言的语法进行全面细致的说明。

## 5.9 程序样例

使用LOGO语言绘制简单而优美的图像示例，目的在于帮助用户理解语法以及发现LOGO语言强大功能。

# 约束

[记录所有设计约束、外部约束或其他依赖关系。]

# 质量范围

[定义性能、强壮性、容错、可用性以及特性集内没有记录的类似特征的质量范围。]

# 优先级

[定义不同系统特性的优先级。]

# 其他产品需求

[在较高层次上列出适用的标准、硬件或平台需求、性能需求以及环境需求。]

## 适用的标准

[列出产品必须符合的所有标准。其中可能包括法律和法规（FDA、UCC）标准、通讯标准（TCP/IP、ISDN）、平台一致性标准（Windows、Unix 等）以及质量和安全标准（UL、ISO、CMM）。]

## 系统需求

[确定支持该应用程序所必需的任何系统需求。其中可能包括所支持的主机操作系统及网络平台、配置、内存、外围设备和配套软件。]

## 性能需求

[本节用于详细说明性能需求。性能问题可能包括在各种负载条件下的用户负载因素、带宽或通信容量、吞吐量、精确度以及可靠性或响应时间。]

## 环境需求

[根据需要详细说明环境需求。对于基于硬件的系统，环境因素可以包括温度、振荡、湿度、辐射等。对于软件应用系统，环境因素可以包括使用条件、用户环境、资源可用性、维护问题、错误处理和恢复。]

# 文档需求

[此节说明为支持成功部署应用程序而必须制作的文档。]

## 用户手册

[说明用户手册的目的和内容。讨论预期长度、详细程度，是否需要索引、词汇表、教程与参考手册策略等。还应确定格式和打印约束条件。]

## 联机帮助

[许多应用程序提供了联机帮助系统来协助用户。这些系统的性质对于应用程序开发来说独特的，因为它们综合了编程（如超链接）和技术写作（组织、演示）的各个方面。许多人发现联机帮助系统的开发本身就是一个受益于先期规模管理和计划活动的项目。]

## 安装指南、配置文件、自述文件

[在提供全套的解决方案时，提供包括安装说明和配置指南的文档是非常重要的。此外，自述文件通常也要作为一个标准构件包括在内。自述文件可以包括一个“本发布版中的新特性”部分，并讨论与以前发布版的兼容性问题。多数用户也希望在自述文件中列出任何已知的错误和变通方法。]

## 标签与包装

[目前最先进的应用程序从产品包装开始就提供了一致的外观，这种一致还体现在安装菜单、启动屏幕、帮助系统、GUI 对话框等等。此节定义对标签的需求和和标签类型，以便合并到代码中。举例来说，标签和包装涉及版权和专利声明、公司徽标、标准化的图标以及其他图形元素。]