|  |  |
| --- | --- |
| 基于GO+QML的跨平台桌面直播助手  总体设计V1.0 | 摘要  运用GO+QML实现跨平台的桌面直播助手。  作者：Walnut开发小组 |

**“‘Walnut’跨平台直播助手计划书”执行摘要**

“Walnut” 跨平台直播助手（以下简称“Walnut”）具有天然Go语言的跨平台特性，实现直播过程中各种信息的展示，快速呈现所需的关键内容，性能优越，很好地弥补了其他应用所缺失的功能，所不具备的特性，为广大直播用户提供了可靠的保障，市场前景广阔。

“Walnut”应用贯彻落实了“精简，高效”的开发理念，在满足用户需求的基础上，将“大道至简”完全融入到软件中去，能够让用户不需要每次设置繁杂的功能参数，也会提供给使用者更舒适便捷的用户体验。相比于现有的直播助手应用，性价比更高也更加实用，能被大多数人所接受。

本项目主要盈利是靠中后期为特定用户的具体需求提供定制，实现个性化功能。本项目所属领域是跨平台桌面应用，由于产品较为精简，亲民、高性价比，所以会被大多数人所接受，也更利于后期的扩展和发展。

“Walnut”充分利用了Go跨平台的天然优势，并发特色以及CGO的特性，将Go和Qt的优势互补，相比于现有的其他C#，Java应用，做到了精简，高效的效果。界面更加美观，也更容易被用户所接受。由于应用了Go语言协程的自带特性，使得“Walnut”生来自带处理海量数据的能力，保证在大量数据到来的情况下不漏，不慢地将所有信息准确无误的显示到用户界面上。

目录

[一、市场分析 4](#_Toc512188170)

[1.1政策方针 4](#_Toc512188171)

[1.2目标客户 4](#_Toc512188172)

[1.3发展前景 5](#_Toc512188173)

[1.4行业现状 5](#_Toc512188174)

[二、项目概述 6](#_Toc512188175)

[三、技术支持 7](#_Toc512188176)

[3.1 3D建模技术 7](#_Toc512188177)

[3.2 Unity物理引擎 8](#_Toc512188178)

[3.3 云计算服务平台 9](#_Toc512188179)

[四、功能列表 9](#_Toc512188180)

[4.1 云服务中心 9](#_Toc512188181)

[4.1.1 指纹数据库 9](#_Toc512188182)

[4.1.2 记忆化学习引擎库 10](#_Toc512188183)

[4.1.3 路径计算模块 10](#_Toc512188184)

[4.2 用户平台 11](#_Toc512188185)

[4.2.1 信号读取 11](#_Toc512188186)

[4.2.2 定点导航 11](#_Toc512188187)

[4.2.3 界面显示 11](#_Toc512188188)

[4.2.4 建筑大数据查看 13](#_Toc512188189)

[五、技术实现 14](#_Toc512188190)

[5.1 建模处理 14](#_Toc512188191)

[5.2 路径计算 14](#_Toc512188192)

[5.3 导航显示 14](#_Toc512188193)

[六、执行流程 15](#_Toc512188194)

[七、团队介绍 16](#_Toc512188195)

[7.1团队概括 16](#_Toc512188196)

[7.2团队理念 16](#_Toc512188197)

# 一、市场分析

## 1.1市场现状

经过一段时间的观察发现，大部分用户都处于“不愿使用官方自带的工具”和“没有找到真正适合自己的工具”两者之间痛苦不堪，由于今年的疫情影响，绝大多数居民居家相应国家号召，使得线上某些行业空前“繁荣”，其中就包含直播领域。大批量用户跨入直播行业，却因为“不了解”，“没人带”等各方面原因找不到一款真正适合自己的直播助手应用。

有些用户也提出，几乎所有现有的直播应用都是闭源，不可自定义修改的，这让许多用户痛苦不堪。仅有的几个功能比较全面的应用，也是界面设置参数繁杂，新人在没有指导的情况下，第一次使用根本摸不到头绪，不知从何入手。除此之外，现有工具也不支持某些用户所需的功能。

基于以上几点，我们决定开发“Walnut”。最终目的就是开发一款简洁，高效，可靠，美观的跨平台直播助手应用。

## 1.2目标客户

“Walnut”目标用户是所有直播平台的主播，目前实现了BiliBili平台的所有直播需求，后续会扩展到各大平台。最终是为所有无论是新人还是经验丰富的直播者提供帮助。

## 1.3发展前景

由于“Walnut”是一款自研，开源的项目，开发过程中会不断接收用户提出的问题以及想法，并会在扩展平台的同时，不断增加已有功能的稳定性，并且会不断获取用户反馈，综合分析后增加新的功能与特性。逐渐发展至能满足绝大多数用户的直播需求，真正实现“一站式的解决方案”。

## 1.4行业现状

今天，物联网行业早已成为万众公认的未来大势，其火热程度无与伦比。作为物联网体系中很重要的一环，室内导航定位技术正处于蓬勃发展的时期，并且已经逐步成为各行各业的关注焦点，目前主流技术有UWB（超宽带）脉冲信号、基于RFID定位、ZigBee室内结点定位技术、超声波定位技术等诸多方法。这些方法都各有其优缺点，要么斥资巨大（32线Lidar），要么难以普及（TDOA超声波测距）。

而要想实现精确的室内导航定位技术，难点在于室内定位的精确度和对信号的处理。基于交汇的算法对时间，角度很敏感，手机的传感器不能达到要求，定位精度不高；基于WiFi的信号受楼层，遮挡物等影响太大，电磁波在不同介质之间的衰减速率差异太大无法计算；很多技术需要布设信号网，测定信号源位置，导致成本开销十分巨大，在此情况下，研发成本较低，精度较高，人人可用的楼宇内导航定位技术已经成为当前社会急迫的需求和导向。

# 项目概述

“启明”系统基于手机对路由器蜂窝信号的读取做出定位判断。我们将提前对建筑进行建模，采集大量定位点的路由器信号形成指纹数据库。在定位时，通过读取楼宇内所安装的路由器信号（不需要接入局域网）比对多组信号强度并带入提前建立好的建筑模型计算出手机当前空间位置作为起点，以用户所输入的地点作为终点，在模型中根据起点和终点设计寻址算法找出最小路径。本系统可以对建筑模型进行输出标注路径和点，输出时会有左右两个画面，一个是建筑完整外观（较少内部细节以使之不至于太过杂乱而看不清）和起点，终点位置（以亮点标注）；一个是当前楼层平面剖析路线图（在当前楼层平面图中标注出在当前楼层所需行进的路线）。为用户提供简单，便捷，精确的服务。

# 三、技术支持

图3.1

如图3.1所示，“Walnut”应用的技术支持分为CGO， QML和Qt三方面。详细介绍如下。

## 3.1 CGO

CGO是Go语言的特性之一，可以支持创建调用C代码的Go包。

需要导入一个“伪包C”，能够写入正常的Go代码来调用CGO。接下来，Go的代码就可以引用一些类型，例如C.size\_t，一些变量例如C.stdout，或者是一些函数，比如C.putchar。如果在导入包“C”前紧跟注释，这个注释会被称为“前导”，在编译包的C部分时，会作为文件头部。例如：

1. // #include <stdio.h>
2. // #include <errno.h>
3. import “C”

由于Go语言的CGO特性，我们就可以利用绑定的原理，调用Qt框架，再通过信号与插槽，使得Go与QML进行双向通信，还可以使用CFLAGS, CPPFLAGS, CXXFLAGS, FFLAGS 和 LDFLAGS 变量来控制C/C++编译器的行为。

图3.1 建模过程

图3.2 初步模型

## 3.2 QML

Unity3D是由Unity Technologies开发的一个综合型游戏开发工具，是一个全面整合的专业[游戏引擎](https://baike.baidu.com/item/%E6%B8%B8%E6%88%8F%E5%BC%95%E6%93%8E)。Unity类似于Director,Blender game engine, Virtools 或 Torque Game Builder等利用交互的图型化[开发环境](https://baike.baidu.com/item/%E5%BC%80%E5%8F%91%E7%8E%AF%E5%A2%83" \t "https://baike.baidu.com/item/Unity3D/_blank)为首要方式的软件。其编辑器运行在Windows 和Mac OS X下，可发布游戏至[Windows](https://baike.baidu.com/item/Windows" \t "https://baike.baidu.com/item/Unity3D/_blank)、[Mac](https://baike.baidu.com/item/Mac/173" \t "https://baike.baidu.com/item/Unity3D/_blank)、[Wii](https://baike.baidu.com/item/Wii" \t "https://baike.baidu.com/item/Unity3D/_blank)、[iPhone](https://baike.baidu.com/item/iPhone" \t "https://baike.baidu.com/item/Unity3D/_blank)、[WebGL](https://baike.baidu.com/item/WebGL" \t "https://baike.baidu.com/item/Unity3D/_blank)（需要[HTML5](https://baike.baidu.com/item/HTML5" \t "https://baike.baidu.com/item/Unity3D/_blank)）、Windows phone 8和Android平台。也可以利用Unity web player[插件](https://baike.baidu.com/item/%E6%8F%92%E4%BB%B6)发布网页游戏，支持Mac和Windows的网页浏览。它的[网页播放器](https://baike.baidu.com/item/%E7%BD%91%E9%A1%B5%E6%92%AD%E6%94%BE%E5%99%A8" \t "https://baike.baidu.com/item/Unity3D/_blank)也被Mac 所支持。

## 3.3 Qt

采用H3Cloud云计算操作系统软件，将多台h3C R390机架服务器组建HA集群，在虚拟机上部署“启明”云端计算服务，并配合HA和动态负载均衡等高级功能，实现业务的连续性，提高资源利用率，实现强大的运算能力。

# 四、功能列表

图4.1

如图4.1所示，“Walnut”系统大致分为直播信息显示和用户点歌两部分。直播信息显示部分可以显示用户的弹幕，礼物，VIP进场等关键信息，并且分不同区域，以不同颜色的字体显示其重要程度；而能与用户互动的部分就是用户点歌功能，用户通过输入点歌关键字，如“BGM”，用户就可以通过“BGM 关键词”的方式搜索相应歌曲进行自动播放。详细介绍如下。

## 4.1 直播信息显示

### 4.1.1 用户弹幕显示

将提前测量的定位点的信号数据存放于物理引擎中，形成指纹数据库，在使用时，将用户于当前地点采集到的路由器信号强度数据输入，通过与数据库中的数据进行比对，得到与当前数据差异最小的一组定位点数据，再通过空间测距定位计算出用户与该定位点的距离误差，最后得到用户在空间中的实际位置。

### 4.1.2 用户礼物显示

引擎库中存储了之前已计算出的路径数据以及机器学习得到的用户行为模式分析，从指纹数据库得到起点终点数据后，运用动态规划和分块原理，将每一次计算都拆分成多个子路径（一般以楼层为单位），若引擎库中之前存有子路径的计算结果则直接调用，不再需要重复运算，若不存在，则将需要计算的子路径输入到路径计算模块中，并且接收计算结果，不断更新维护引擎库数据，始终保证系统能够在最短的时间内计算出最精确的导航路径。同时采用机器学习模型分析用户行为模式（如不同年龄段用户在楼层较少的情况下对楼梯和电梯的选择），做出更加精确和个性化的路径导航。

### 4.1.3 VIP进场显示

通过实时接收记忆化学习引擎库中输入的子路径信息，使用Dijkstra、contraction hierarchies 和 highway hierarchies算法互相去除误差后得到子最短路径并重新输出至引擎库。

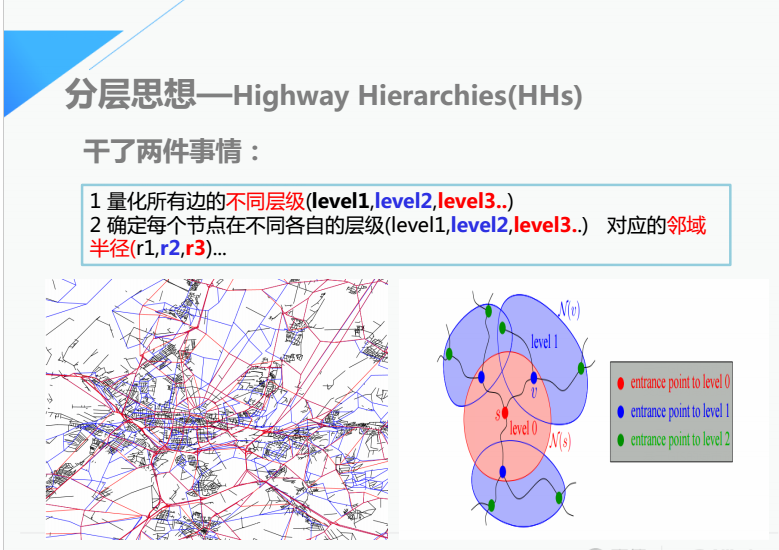


图4.2 highway hierarchies算法图解

## 4.2 用户点歌

### 4.2.1 信号读取



图4.3

通过手机自带的WiFi信号读取功能获取附近路由器蜂窝信号强度（图4.3为tensor log APP读取的信号），选择强度最大的3组信号获取其对应路由器mac协议地址和强度数据一起发送至云服务平台。

### 4.2.2 定点导航

用户输入目标地点后会弹出数个目标选项，用户选择其一设置为终点，系统自动通过信号读取功能匹配指纹数据库获取用户当前所在位置为起点。将起点终点一起输入云服务中心的导航路径计算模块，通过导航计算模块与记忆化学习引擎库模块一起计算出最短路径后开始导航。

### 4.2.3 界面显示

图4.4

如图4.4所示， 用户端的显示主体分为两个界面，各占一半。上半部分是该建筑完整外观透视图，下半部分是用户当前所在楼层平面剖析图。

4.2.3.1 三维模型图

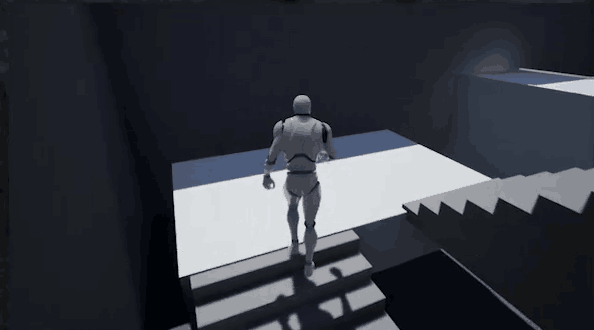


图4.5

如图4.5所示，三维模型图显示用户所在位置的三维建筑模型，包括用户所在楼层大部分内部结构。通过不断检测用户当前位置信息更新数据，用模拟人物实时显示用户当前位置，模拟用户行走路径。可通过拖动屏幕更改视角、更改大小、图像平移等。

4.2.3.2 平面导航图

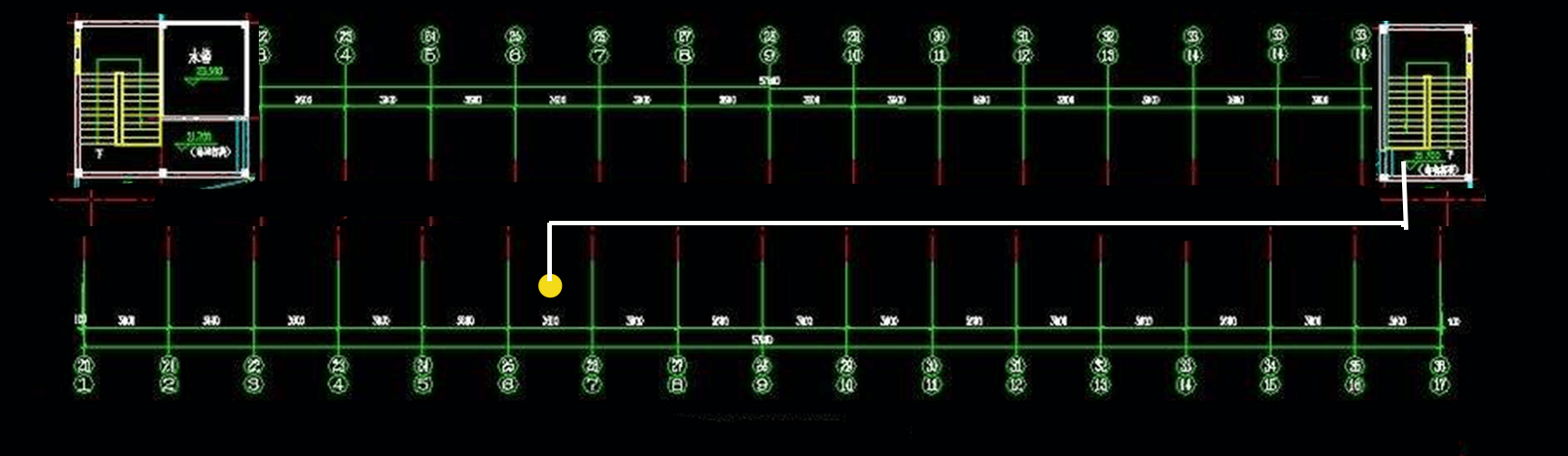


图4.6

如图4.6，平面导航图为用户当前所在楼层平面图，只显示当前楼层大致情形，并用光标标出用户所在位置，显示当前楼层导航路径和目标点位置。可通过拖动屏幕更改视角、更改大小、图像平移等。

### 4.2.4 建筑大数据查看

本系统还可提供建筑相关信息，如大型医院的值班医生表，行政办公楼的作息时间等，用户可在获取信息后对自己的后续行为作出安排和调整。

# 技术实现

## 5.1 建模处理

建筑建模采用3ds Max进行物理建模，导入Unity物理引擎，使用C++， C#对模型进行编程。带入定位点路由器mac协议地址和对应信号强度数据，对模型进行预处理，形成指纹数据库。

## 5.2 路径计算

先将自指纹数据库得到的起点和终点输入记忆化学习引擎库，使用动态规划和分块原理将路径拆分为多个子路径（一般以楼层为单位），与引擎库之前存储的路径比对，若有结果则直接调用，无结果则将子路径起点终点输入云平台路径计算模块，依次采用Dijkstra、contraction hierarchies 和 highway hierarchies算法计算子路径，互相去除误差后得到子最短路径并输出给用户，同时对记忆化学习引擎库进行更新维护，保证引擎库中储存有大量已计算的路径数据，有效减少重复计算量。

## 5.3 导航显示

通过物理引擎输出建筑模型，并能做到同时输出外观透视图和平面剖析图，同时使用光标标出用户当前所在位置、目标点位置和路径。比一般的导航软件显示更加直观清晰。

# 执行流程

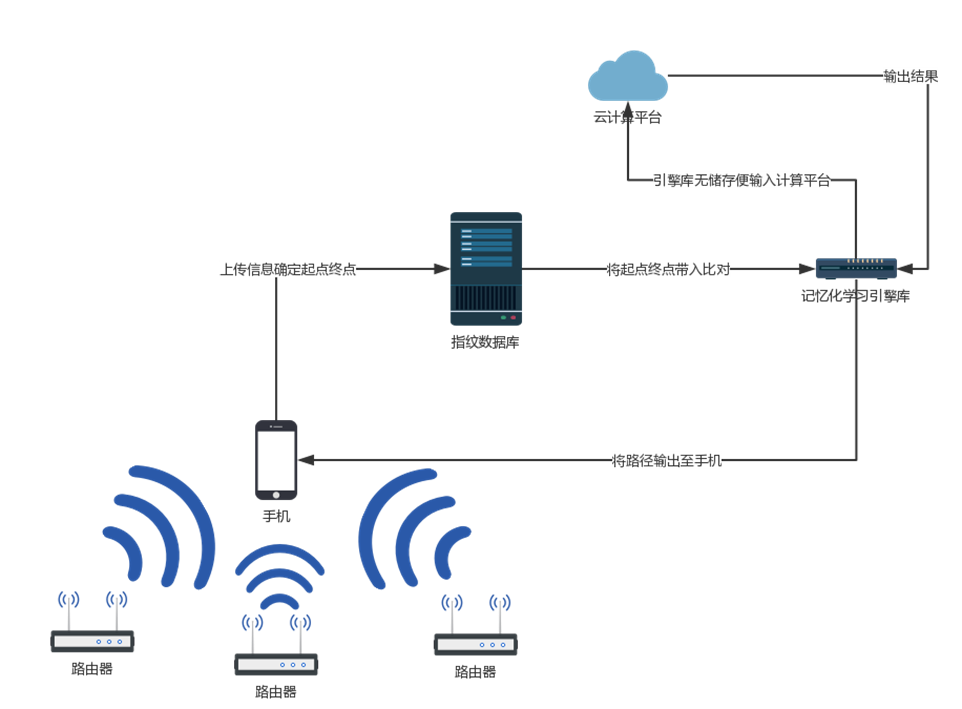


图6.1

如图6.1所示，“启明”系统将提前建立好建筑模型，采集多个定位点的路由器数据强度和对应的mac协议地址，形成指纹数据库。当用户使用“启明”系统时，系统将通过手机传感器接收附近的路由器信号强度和对应的mac协议地址，过滤嘈点后将数据传入指纹数据库，在指纹数据库中比对得出与现有信息最相近的一组，再通过空间测距定位计算出用户与该定位点的距离误差，最后得到用户在空间中的实际位置。将用户的实际位置带入物理引擎作为起点，用户输入的地点作为终点，输入记忆化学习引擎库，采用动态规划和分块原理，将每一次计算都拆分成多个子路径（一般以楼层为单位），比对之前的计算结果，若有结果则直接使用，无结果则输入云服务路径计算模块，依次采用Dijkstra、contraction hierarchies 和 highway hierarchies算法计算子路径，互相去除误差后得到子最短路径并输出给用户，同时对引擎库进行更新维护，通过不断维护引擎库保证系统始终能够在最短的时间内计算出最精确的导航路径。

# 团队介绍

## 7.1团队概括

项目组是一个五人团队，均是计算机学院本科学生，其中4人为2016级，1人为2015级。项目组拥有一流的致力于物联网行业的管理，集物联网应用、云计算、移动端开发为一身的技术型团队。专业的团队，过硬的技术力量使得我们的团队非常的有竞争力。

我们项目组熟悉物联网行业，具有共同的事业远景和创新精神，具有较强的社会实践能力，拥有部分项目经营、资本运作的理念和经验。团队成员拥有了良好的分工合作及团队协作精神，项目组内部凝聚力强，成员责任心强，确保了项目高效运转和高速发展。

## 7.2团队理念

文化决定团队的基因，技术是后盾，过程是纽带，人则是挥舞纽带的艺术大师。艺术因团队而美丽，团队因过程而精彩。以冰霜之操自励，则品日清高；以穹窿之量容人，则德日广大；以切磋之谊取友，则学问日精；以慎重之行利生，则道风日远。