|  |  |
| --- | --- |
| 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：Walnuts开发小组 |

**“‘Walnuts’跨平台直播助手计划书”执行摘要**

“Walnuts” 跨平台直播助手（以下简称“Walnuts”）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。

“Walnuts”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。

“Walnuts”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语言协程的特性，使得“Walnuts”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完整，快速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](#_Toc38544894)

[1.1市场现状 4](#_Toc38544895)

[1.2目标客户 4](#_Toc38544896)

[1.3发展前景 4](#_Toc38544897)

[1.4行业现状 4](#_Toc38544898)

[二、 项目概述 5](#_Toc38544899)

[三、技术支持 6](#_Toc38544900)

[3.1 CGO 6](#_Toc38544901)

[3.2 QML 7](#_Toc38544902)

[3.3 Qt 7](#_Toc38544903)

[四、功能列表 8](#_Toc38544904)

[4.1 直播信息显示 8](#_Toc38544905)

[4.1.1 用户弹幕显示 8](#_Toc38544906)

[4.1.2 用户礼物显示 9](#_Toc38544907)

[4.1.3 VIP进场显示 9](#_Toc38544908)

[4.2 用户点歌 10](#_Toc38544909)

[4.2.1 信号读取 10](#_Toc38544910)

[4.2.2 定点导航 10](#_Toc38544911)

[4.2.3 界面显示 10](#_Toc38544912)

[4.2.4 建筑大数据查看 12](#_Toc38544913)

[五、 技术实现 13](#_Toc38544914)

[5.1 建模处理 13](#_Toc38544915)

[5.2 路径计算 13](#_Toc38544916)

[5.3 导航显示 13](#_Toc38544917)

[六、 执行流程 14](#_Toc38544918)

[七、 团队介绍 15](#_Toc38544919)

[7.1团队概括 15](#_Toc38544920)

[7.2团队理念 15](#_Toc38544921)

# 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”两者之间的状态痛苦不堪，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。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，想要提出建议也大概率不被采纳。

基于以上几点，我们决定开发“Walnuts”。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

“Walnuts”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“Walnuts”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”。并且由于其精简，高效，容易上手的特点，也很容易被广大用户所接收，从而增加用户量，市场占有率。

## 1.4行业现状

今天，直播行业已然成为万众公认的未来娱乐平台大势，其火热程度无与伦比。作为用户直播体系中很重要的一环，一款功能完备，界面美观的直播助手应用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，目前主流应用有“小葫芦”，“弹幕姬”以及官方推出的种种版本，大部分基于C#开发，都各有优缺点，要么功能繁杂，不利于新人使用，要么占用资源巨大，对配置有较高要求，难以真正普及。

而要想实现完美的直播助手应用，难点在于功能的完备程度和对资源占用的处理。由于实现的功能逐渐变多，很容易造成界面设计繁杂，上手难，占用资源过大的情况，很多用户由于种种原因面对各种直播工具望而却步。想要实现结合界面，功能，资源占用多方面与一体，难度极大。

# 项目概述

“Walnuts”是基于Go + QML实现跨平台级别的界面美观，精简高效的直播助手。我们利用Go语言的CGO特性，建立与Qt的绑定，结合了Go + QML，即Go负责后端地处理数据，QML负责前端展示页面，两者实现了前后端分离，通过信号与插槽机制进行通信。即Go模拟客户端连接直播间对应的弹幕服务器建立连接，发送所需的握手包，并且每30秒发送一次心跳包维护连接过程，期间不断接收从服务器上传回来的各种直播数据，例如弹幕，礼物，进场等关键信息，再结合抓包获取的官方API展示一些页面上的常见信息，例如总榜，直播间标题，主播昵称等，即可实现完美匹配真实客户端的效果。综合考量项目需求和业务场景，最终决定使用qamel这款小巧精致的Qt绑定，只链接了不多的Qt模块，使得整个应用具备了变得小巧，编译快的特点。

项目的目标是实现全平台直播助手，目前仅实现了BiliBili平台的直播相关需求，后续会实现更多的平台，以“选择平台”的方式进行选择接入。

# 三、技术支持

图3.1

如图3.1所示，“Walnuts”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提供了Go代码与C代码相互调用的机制，是Go语言的特性之一。

首先需要设置GO ENV CGOENABLED=1，来开启CGO，随后通过导入一个“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。

某些第三方库可能只有 C/C++ 的实现，完全用纯 Go 的实现可能工程浩大，这时候 CGO 就派上用场了。可以通 CGO 在 Go在调用 C 的接口，C++ 的接口可以用 C 包装一下提供给 Go调用。被调用的 C 代码可以直接以源代码形式提供或者打包静态库或动态库在编译时链接。推荐使用静态库的方式，这样方便代码隔离，编译的二进制也没有动态库依赖方便发布也符合 Go的哲学。

## 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)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)、[Mac](https://baike.baidu.com/item/Mac/173)、[Wii](https://baike.baidu.com/item/Wii)、[iPhone](https://baike.baidu.com/item/iPhone)、[WebGL](https://baike.baidu.com/item/WebGL)（需要[HTML5](https://baike.baidu.com/item/HTML5)）、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)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，“Walnuts”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 用户信息显示

连接弹幕服务器成功后，循环接收服务器发来的消息，针对每一个消息包可能包含多个类别，多条消息的情况，进行了循环判断的处理，保证不漏掉一条消息。获取到相关信息后，根据服务端的“CMD”字段对进行分类，送入相对应的channel中，每个channel单独处理各自的消息。这有利于前端对消息的分类显示。每个channel在接收到原始数据后，根据自身消息的类型，提取其中必要的信息，序列化成字符串后发送给QML进行显示。

### 4.1.2公共信息显示

由于某些信息是一段时间固定不变的，例如直播间标题，主播昵称，等级等内容，因此官方采用了建立RESTful风格的API来进行消息获取的方法，通过抓包的方式获取到浏览器的各种请求后，调整好相关参数，在每次打开主界面时请求相关的API，获取返回的信息，一次性的显示在页面上。 期间持续监听服务器传回的消息，如果需要更新相关内容，即时捕捉，传回到前端即可。此类信息较为繁杂，软件的功能根据需要仅获取了其中部分信息，例如总榜排行，直播间描述等消息可根据后续需求任何调整修改。

## 4.2 用户点歌

### 4.2.1关键字点歌



图4.2

用户可以通过右上角的绿色图标输入关键字来开启和关闭点歌功能。在输入完关键字并点击“开启点歌”后，就会自动监听直播间观众发送的弹幕信息，针对“关键字 搜索词”的模式搜索相关歌曲进行播放，并带有相关提示，如图：



图4.3

界面中用户可以选择查看播放列表，歌曲数目，选择单曲循环或列表循环，切歌，调节音量等功能。

# 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

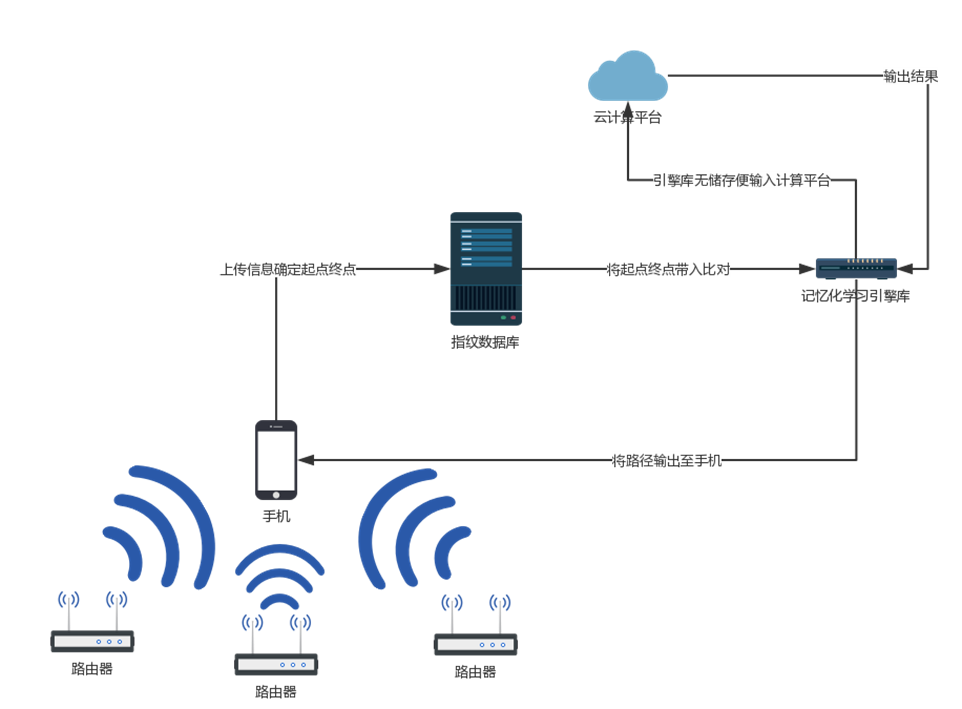


图6.1

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。