# 嵌入式软件工程课程项目总结报告

组长：李贞子

组员：张璐 李天宇 张佳琳 王润泽

2019.6.9

# 项目名称

MOSS你这是在杀人：ROS机器人

# 项目概述

在自动化生产高度重要的当今时代，我们意识到机器人开发是一个重要的研究领域，因此我们的重点在于开发一个具有三大功能的机器人。三个功能分别是随机避障行走，导航到目标地点，识别目标物体并抓取。在非功能需求方面，我们要求机器人运动速度不能过快；要求机器人不会发生撞击损伤；系统的可维护性和容错性强等。

最终我们成功完成三个功能的设计实现，并且在此基础上开发了前端操作界面，方便真实场景下用户的使用并能提高他们的使用体验。随机避障行走能达到的功能有，设定机器人行走速度、时间让机器人在设定的速度下行走固定的时间，并且对于空间有一定的覆盖性。导航到目标地点的功能有，根据实际地形建立新地图；建立新的目标地点；选定目标地点并导航到目标地点。抓取预设物体的功能有：机器人跟随领航员标定预设位置、机器人识别并抓取目标物体。

由于达到了上述的效果，我们的机器人可以初步应用于家庭扫地装置；仓库物体管理系统等。假若在现有功能上进行改进，我们的机器人应当还能适应更多特种用途，例如火场紧急取物等。

# 需求分析阶段总结

本项目的需求主要分为三类，分别是避障、导航和物体抓取。需求分析过程中采取的主要策略是根据机器人开发手册对机器人的应用场景进行分析，再经过小组讨论带入用户角色，并且结合机器人的软硬件条件等得需求。功能需求和非功能需求由小组进行讨论之后再把具体的任务分配给不同成员进行文档书写。文档书写过程中遇到的主要困难是对业务需求和功能需求的界限认识不是很清楚，在和老师进行沟通并查阅相关资料后对这一过程有了较为清晰的认识。除此之外，由于一开始没有真正接触到机器人，缺乏实地操作，功能需求有一部分与后期的具体实现有所脱离，比如避障模式中一开始需要对姿态进行设定，而这一设定其实是等同于速度的设定。再者，由于小组成员对机器人的硬件设置不了解，比如双目摄像头和雷达激光的具体参数和数据传输格式，这就导致了数据需求的书写不够规范。嵌入式评审过程中遇到的主要问题是非功能需求衡量标准比较模糊，比如机器人速度不能“过快”，这就需要一个具体化的数字进行衡量。之后小组成员进行了讨论，对功能需求、数据需求、非功能需求等进行了补充，比如对结合具体的实现结果修改用例模型，删除掉一些不合理的需求，解决了与最终实现的不一致以及前后叙述的冲突，对于较为模糊的描述给出了具体的数字化评判标准，还对整体的风格进行了统一，最终完善了需求分析文档。

# 设计阶段总结

1.避障模块

**架构：**

由雷达，控制，行走节点构成。雷达节点发送雷达信息，控制节点接受并处理雷达信息，判断机器人前方是否有障碍物并且在需要的情况下为机器人选择一个前方没有障碍物的方向并将相应的前进或转向信息发送给行走节点。行走节点接受消息，控制机器人行动。

**追踪关系：**

整个模块对应了机器人避障行走的需求

**设计评审中出现的问题与设计活动中的问题总结如下：**

1）设计阶段没有仔细思考雷达的消息，事实上雷达消息很复杂，由很多属性构成，在实际实现的时候要实现一些之前没有想到的算法和过程

2）设计寻找新路径时没有考虑到机器人的实际行走请况，机器人在转向时设置的角度分辨率（每次转的最小单位角度）太小，导致动作十分卡顿。

2.导航模块

**架构：**

导航部分由于需要进行导航，根据ROS手册的相应部分，采用构建地图-设定航点-导航实现的实现方式。因此导航模块总体分为三大模块：

1) 构建地图模块，调用相应ROS指令进行建图和保存，并且将新建的地图移动至规定的位置方便调用。

2) 航点标定模块，调用相应ROS指令进行航点标定，由于ROS航点标定一定需要使用Rviz界面，因此直接调出该界面。标定的航点按照一定顺序储存在预定的文件内。

3) 导航实现模块，通过读入地图文件、航点文件，并且将航点文件中保存的航点显示在下拉菜单中供用户选择，随后根据用户选择的航点标号，得到导航路径，并且根据路径到达目标航点，并且在导航到目标地点的过程中，避免与动态障碍物进行碰撞。

**追踪关系**：

根据我们导航部分的需求：建立地图，控制机器人规划路径，我们设计了对应的模块：构地建图模块针对地图需求，航点标定模块和导航实现模块针对路径规划需求。

3.抓取模块

**架构**：

由物体识别，相机、行走、导航、控制、语音、抓取节点构成: 相机节点广播kinect相机拍摄的画面，物体识别节点接收这个画面，利用机器学习的算法将画面上物体的种类和位置发送给控制节点，控制节点主要负责机器人跟随用户、标定航点和并将物体识别节点发来的信息转化为要抓取物体的航点信息。将这个信息发送给导航节点，机器人走到目标位置启动抓取节点。抓取节点负责检测面前的物体并进行抓取。

**设计评审中出现的问题与设计活动中的问题总结如下：**

1）在设计阶段没有完全搞清楚ROS机器人的运行规律，导致最后发现有些设计完全不符合实际情况，比如开始的设计需要启动机器人的usb\_cam节点，但是实验用的机器人完全没有这个外设，只能使用kinect2.

2）在设计时没有将抓取模块跟导航模块结合，因为抓取时应该实现导航，但由于刚开始人员分工不够合理，不同设计人员缺乏沟通，导致没有调用接口而是自己重写，浪费了时间。

3）设计时将功能分的太细碎，把整个抓取模块分成了跟随、拍摄，转化图片，识别、导航和抓取五个小模块，事实上如果把中间三个模块割裂开来的话就要耗费大量的时间进行消息的传递，应该把他们放在一个模块里以便传递变量。

**追踪关系：**

1）程序根据kinect相机得到的照片，能够识别物体的种类和其位置，这对应了识别物体的需求。

2）程序在通过一定算法得出航点后，机器人需要走向那个航点以完成抓取，这个对应了控制机器人规划路径的需求。

3）到达指定航点后，机器人可以抓取物体，这对应了抓取物体的需求。

4. UI模块

**架构：**

UI模块基于PyQt5开发，分为界面和响应事件两部分，界面主要搭载了图片、文字、文本框、按钮、下拉菜单等并且使其布局美观且人性，响应事件包含下拉菜单、文本框和按钮等控件的功能，具体表现为通过用户通过下拉菜单和文本框设置参数，通过按钮来将输出重定向到命令行，启动launch文件来完成启动机器人的相应功能。

**追踪关系：**

UI模块对应用户界面需求

**设计评审中出现的问题与设计活动中的问题：**

无

# 迭代1阶段总结

## 任务分配

第一迭代阶段我们主要实现了避障功能模块和导航功能模块，成员被分为两个小组：李贞子和王润泽负责避障功能和UI框架的搭建；张璐、张佳琳、李天宇负责导航功能；每小组成员也分别需要编写对应的UI。第一阶段两小组成员都如期完成了各自的功能模块和对应UI，测试效果也十分理想。尽管组员很好地完成了第一阶段的任务，但是任务分配停留在口头层面，因此我们在之后利用了github的看板进行管理，这一阶段的具体任务分配和看板对应如下表所示：

**表1 第一次迭代任务分配**

|  |  |  |  |
| --- | --- | --- | --- |
| **人员** | **任务模块** | **具体内容** | **看板对应** |
| 李贞子 | 避障 | 设计避障功能算法: | <https://github.com/sebuaa2019/Team201/projects/3#card-22018811> |
| 完成main和callback函数编写，实现基本避障架构和核心函数 | <https://github.com/sebuaa2019/Team201/projects/3#card-22018833> |
| 实现UI主体界面 | <https://github.com/sebuaa2019/Team201/projects/3#card-22053287> |
| 实现避障界面UI设计 | <https://github.com/sebuaa2019/Team201/projects/3#card-22018845> |
| 对避障功能进行测试 | <https://github.com/sebuaa2019/Team201/projects/3#card-22018861> |
| 根据测试结果修复bug | <https://github.com/sebuaa2019/Team201/projects/3#card-22033641> |
| 王润泽 | 避障 | 编写demo测试机器人行走功能 | <https://github.com/sebuaa2019/Team201/projects/3#card-22019406> |
| 测试雷达功能，查看雷达发布的话题和消息的组成部分 | <https://github.com/sebuaa2019/Team201/projects/3#card-22019431> |
| 实现功能接口函数（检查前方是否有障碍物，找到一个没有障碍物的方向） | <https://github.com/sebuaa2019/Team201/projects/3#card-22019443> |
| 编写UI窗口缩放功能，使其更加美观易用 | <https://github.com/sebuaa2019/Team201/projects/3#card-22052068> |
| 编写UI中按钮的响应事件 | <https://github.com/sebuaa2019/Team201/projects/3#card-22019468> |
| 张佳琳 | 导航 | path\_plan.cpp文件对point.txt进行访问，以获得目标航点名称 | <https://github.com/sebuaa2019/Team201/projects/4#card-22031367> |
| 导航UI下拉菜单设计 | <https://github.com/sebuaa2019/Team201/projects/4#card-22031308> |
| 导航UI按钮设计 | <https://github.com/sebuaa2019/Team201/projects/4#card-22031189> |
| 张璐 | 导航 | 导航部分按钮响应事件，前后端对接 | <https://github.com/sebuaa2019/Team201/projects/4#card-22030546> |
| 搭建导航模块代码框架 | <https://github.com/sebuaa2019/Team201/projects/4#card-22030441> |
| 修改path\_plan文件，解决运行时异常问题 | <https://github.com/sebuaa2019/Team201/projects/4#card-22029979> |
| 完成机器人导航的底层算法设计 | <https://github.com/sebuaa2019/Team201/projects/4#card-22029970> |
| 李天宇 | 导航 | UI变更与美化，统一着色风格和布局 | <https://github.com/sebuaa2019/Team201/projects/4#card-22528487> |

## 代码提交

第一次迭代的技术层面一开始十分艰难，由于小组成员对ROS系统了解甚少，上手也比较困难。在参考手册、ROS wiki等资料和大量实地测试之后，大家都对具体的设计有了一定的想法。这一阶段的代码提交没有较好地使用github，停留在线下交流代码、“面对面”编程的阶段，因此在这之后我们分别对自己的代码模块进行了github提交，利用github进行代码管理，具体的代码提交记录和内容对应如下表所示：

**表2 第一次迭代代码提交**

|  |  |  |
| --- | --- | --- |
| 人员 | 代码内容 | 提交记录 |
| 李贞子 | main, autonomous\_behave函数 | <https://github.com/sebuaa2019/Team201/commit/fc55668a10ff48812aa8f8bf437b452e476144b8> |
| LeftTableWidget文件，  实现UI主体界面， | <https://github.com/sebuaa2019/Team201/commit/113802a01dfbda49ae6590ea4a3c07247faa685c#diff-fce90544871426c432b1c2e5a3eb0253> |
| 实现避障功能UI | <https://github.com/sebuaa2019/Team201/commit/113802a01dfbda49ae6590ea4a3c07247faa685c#diff-fce90544871426c432b1c2e5a3eb0253> |
| 王润泽 | check\_front\_obstacles  find\_path函数 | <https://github.com/sebuaa2019/Team201/commit/4afd65a5a1d1c640868ad89d0fb0cfa5dc2dbc56> |
| UI窗口缩放移动以及最上层窗口的编写 | <https://github.com/sebuaa2019/Team201/commit/5ec70ccf98cb018d81cfe27847193a65e95316ec> |
| 编写my\_vel.launch | <https://github.com/sebuaa2019/Team201/commit/8a84ae2d56338a3841506bc0e6e6f52f178a0b15> |
| 编写package.xml文件 | <https://github.com/sebuaa2019/Team201/commit/9d20fa83417d09d25b744523e9ade167107395e5> |
| 编写CMakeList.txt文件 | <https://github.com/sebuaa2019/Team201/commit/fecdedf04bde3e98c9e40b6bcf33c58bb37677e2> |
| 编写抓取部分按钮响应事件函数 | <https://github.com/sebuaa2019/Team201/commit/4abc8c5bc5baeb85034fb2e28798855f649af5fb> |
| 张璐 | 增加path\_plan.cpp文件 | <https://github.com/sebuaa2019/Team201/commit/6e3f058805979b00f419ef41c774f7bf3edcd999> |
| 更新path\_plan.cpp文件，基本完成导航部分设计 | <https://github.com/sebuaa2019/Team201/commit/2c5dfd75c3b88b1b440f48c892960345c0b48fb9> |
| 完成后端接口设计，实现前后端对接 | <https://github.com/sebuaa2019/Team201/commit/4b33c3393ac4355691b886efd149f4a657c88491> |
| 张佳琳 | 增加航点最大个数 | <https://github.com/sebuaa2019/Team201/commit/77bb8a4ccf9b1ccee510329ac37da5298ab2db09> |
| 更新path\_plan.cpp文件，删除EntranceCB | <https://github.com/sebuaa2019/Team201/commit/b0bbf2575172b1d0caa76ca5ac0b780cc605cf32> |
| 设计导航部分初始UI布局 | <https://github.com/sebuaa2019/Team201/commit/378c743eaa17a8ade7b6f54ff83eff5017550883> |
| 建立click响应函数 | <https://github.com/sebuaa2019/Team201/commit/880bdc0ad144b26edba8a5cd5e3604044199ed07> |
| 李天宇 | 导航功能UI美化 | <https://github.com/sebuaa2019/Team201/commit/9903b9b5974bd2306267067dd408c57a9daca52a> |

## 测试

第一次迭代的测试主要是针对功能的测试，即优先和主要考虑功能的完善度和可靠程度，基于这样的想法，避障模块采用不同的障碍物摆放方式和速度、时间的随机设置进行了功能的测试，还加入了测试员的随机遮挡，测试结果比较理想。导航模块则按照不同的组成模块进行划分，测试了每一个小模块的正常功能，最终进行了整体的测试，也取得了不错的效果。尽管机器人可以较好地完成功能，但是也暴露了很多问题，在5.4节中我们进行了具体的阐述。

**表3 第一次迭代测试**

|  |  |  |
| --- | --- | --- |
| 功能模块 | 测试内容 | 测试结果 |
| 避障 | 测试避雷达、速度控制是否正常运行 | 雷达、速度模块均可正常运行，可以获取雷达信息，机器人正常行走； |
| 在机器人周围摆放不同障碍物，进行人为随机设障，设置多组速度和时间，查看机器人是否可以及时正确避障 | 机器人正确避障，按照预期的速度、时间行走；没有碰到任何障碍物，可以及时避障；一开始运行比较卡顿，解决bug后运行流畅 |
| 导航 | 测试构建与存储地图功能是否能按预定方式将构建的地图存储到对应的位置 | 可以根据周围环境构建完整、准确的地图，并保存在指定的路径下 |
| 设定并保存航点，测试是否能设定新航点并将新航点信息正确保存 | 可以在地图上标定新的航点信息，航点内容可以保存为在waypoint.xml文件下 |
| 测试导航过程中是否可以避免与动态障碍物的撞击 | 机器人可以在导航过程中遇到障碍物停止，不会撞击障碍物；障碍物去除可以继续行走至目标地点 |
| 测试是否可以按照选定航点进行路径规划和导航 | 可以根据选定的航点进行路径规划，并且行走至目标地点 |

## 问题

第一次迭代在测试的过程中发现了很多问题，主要来源于测试过程中与预期的不符合和开发过程中的设计问题。github的issue记录了我们的问题和解决方案，以及小组成员的讨论、交互过程，下表是具体的问题和issue对应：

**表4 第一次迭代问题及解决**

|  |  |  |
| --- | --- | --- |
| 问题 | 解决 | issue对应 |
| 辨别雷达的转向、range数组元素对应的角度以及角度分辨率 | 编写demo，输出range数组大小，分别遮挡特定角度雷达，查看距离的变化，以获取所有的数据 | <https://github.com/sebuaa2019/Team201/issues/3> |
| 避障转动卡顿 | 调大分辨率 | <https://github.com/sebuaa2019/Team201/issues/5> |
| 时间到但机器人未停 | 改变消息发布和订阅缓冲队列长度，重复发送消息次数 | <https://github.com/sebuaa2019/Team201/issues/6> |
| 实现UI与速度、时间信息的交互 | 通过文件读取交互 | <https://github.com/sebuaa2019/Team201/issues/7> |
| 关闭避障结点之后其他所有结点如何关闭 | 修改避障结点的required属性 | <https://github.com/sebuaa2019/Team201/issues/8> |
| 代码无法编译 | 修改Makefile文件，将源代码放在与Team201平行的文件夹中，可以成功编译并运行 | <https://github.com/sebuaa2019/Team201/issues/23> |

# 迭代2阶段总结

## 任务分配

第二阶段的任务主要分为两部分：抓取功能、导航动态避障以及bug修复。这一阶段还是将小组成员分为了两部分：王润泽和李贞子完成抓取功能，张璐、李天宇、张佳琳完成导航模块。本阶段大家依然如期完成了任务，抓取部分由于机械臂损坏没有测试，但物体识别和到达指定航点功能已经进行了测试，结果符合预期。导航模块也完成了动态避障的功能，即对于导航过程中出现的障碍物可以停止，移开后又可以进行导航，同时小组成员还修复了导航功能出现的bug。具体的任务分配和看板管理如下表所示：

**表5 第二次迭代任务分配**

|  |  |  |  |
| --- | --- | --- | --- |
| 人员 | 任务模块 | 具体内容 | 看板对应 |
| 李贞子 | 抓取 | darknet\_ros包下载及安装，用以识别物体 | <https://github.com/sebuaa2019/Team201/projects/6#card-22018666> |
| 测试机器人标定航点和导航效果,进行测试和改进 | <https://github.com/sebuaa2019/Team201/projects/6#card-22018682> |
| 完成结合物体识别和标定航点、导航的代码 | <https://github.com/sebuaa2019/Team201/projects/6#card-22018703> |
| 增加darknet传递的消息类型，获取识别结果数组长度 | <https://github.com/sebuaa2019/Team201/projects/6#card-22018713> |
| 录制视频 | <https://github.com/sebuaa2019/Team201/projects/6#card-22019916> |
| 王润泽 | 抓取 | 编写相机驱动文件 | <https://github.com/sebuaa2019/Team201/projects/6#card-22019177> |
| 编写darknet\_ros节点订阅/image\_color以识别物体 | <https://github.com/sebuaa2019/Team201/projects/6#card-22019177> |
| 获取识别到的物体种类和在图中的位置 | <https://github.com/sebuaa2019/Team201/projects/6#card-22019196> |
| 设计将识别信息（种类的位置）转化为航点信息的算法 | <https://github.com/sebuaa2019/Team201/projects/6#card-22019212> |
| 张璐 | 导航 | 解决前端与后端roslaunch命令调用问题 | <https://github.com/sebuaa2019/Team201/projects/5#card-22030640> |
| 修复前端bug，对前端Waypoints文件不存在情况进行异常处理 | <https://github.com/sebuaa2019/Team201/projects/5#card-22030646> |
| 实现机器人动态避障 | <https://github.com/sebuaa2019/Team201/projects/5#card-22030653> |
| 张佳琳 | 导航 | 修改【保存航点】按钮对应的事件。 | <https://github.com/sebuaa2019/Team201/projects/5#card-22031457> |
| 李天宇 | 导航 | 测试功能测试项 | <https://github.com/sebuaa2019/Team201/projects/5#card-22529308> |

## 代码提交

第二次迭代由于任务难度的增加设计难度与之前相比更为复杂，这一次的设计在小组成员的协作配合之下也完成地不错，代码管理方面依然采用了github工具，具体的代码内容和提交记录如下表所示：

**表6 的第二次迭代代码提交**

|  |  |  |
| --- | --- | --- |
| 人员 | 代码内容 | 提交记录 |
| 李贞子 | 完成抓取主体架构，语音标定航点、导航功能，完成结合物体识别和标定航点、导航的代码 | <https://github.com/sebuaa2019/Team201/commit/c66001e94961ebba1275654d7c6d07c104ad5fc9> |
| 增加darknet消息传递类型 | <https://github.com/sebuaa2019/Team201/commit/a292cba5f34d8f693fe9a8e8d11cf29b89a3a148> |
| 编写launch文件 | <https://github.com/sebuaa2019/Team201/commit/b53ce3a4e53fb0e6f75c05bc861f7bd16fec6197> |
| 编写package.xml文件 | <https://github.com/sebuaa2019/Team201/commit/12586324996315543f93348bfa47fceb510b433e#diff-42deb6e568603885d3d8da059571f97a> |
| 王润泽 | 编写接收到darknet\_ros提供的物体识别信息后使用的回调函数，用于确定航点 | [https://github.com/sebuaa2019/Team201/commit/04ed6e69fbe21f579cbdcbf2ad740215afc005a5#diff-0e9fc0642e789aada570f8d5368a76c3](#diff-0e9fc0642e789aada570f8d5368a76c3) |
| 编写订阅darknet\_ros的物体识别信息节点 | [https://github.com/sebuaa2019/Team201/commit/d9ce6cd81e2e3452462bd361cf4a50611f4f0692#diff-0e9fc0642e789aada570f8d5368a76c3](#diff-0e9fc0642e789aada570f8d5368a76c3) |
| 张璐 | 更新path\_plan.cpp文件，增加对读入文件point.txt是否存在的判断 | <https://github.com/sebuaa2019/Team201/commit/0300e616fb3edad786bb7028ee0d839d56761d4b> |
| 修复读入文件时Waypoints文件不存在的bug | <https://github.com/sebuaa2019/Team201/commit/61eff716746c97fdde3a8e90bff9db4dbf257bd2> |
| 张佳琳 | 更新UI关于保存航点和选择航点的操作 | <https://github.com/sebuaa2019/Team201/commit/7b22949b9fa213f5352af117becae7f7f36b6053> |
| 修复waypoints.xml中航点名称命名顺序bug | <https://github.com/sebuaa2019/Team201/commit/faa541315aca177b155d02d4fba4a28f53e5afbc> |

## 测试

第二次迭代对抓取功能的语音航点标定和物体识别进行了测试，我们采用了多种物体，但由于杂物影响、光线等因素，最终采取了bottle和person作为识别的目标。除此之外，导航模块也对动态避障功能和之前的bug进行了测试，具体的测试内容如下表所示：

**表7 第二次迭代测试**

|  |  |  |
| --- | --- | --- |
| 测试模块 | 测试内容 | 测试结果 |
| 导航 | 驱动相机，识别不同物体，测试机器人识别效果 | 可以对不同的具有明显特征的物体进行识别 |
| 语音标定航点，测试机器人是否可以根据识别结果走到目标物体所在的航点 | 可以正确走到目标物体所在的航点 |
| 导航 | 导航过程中动态避障，测试员出现在导航过程中的机器人之前，测试机器人是否可以及时停止 | 机器人可以及时停止，而没有撞击，如果测试员走开，机器人可以继续进行导航 |

## 问题

这一阶段出现的问题较多，抓取部分更多地与物体识别有关，相机驱动和检测包的结合需要进行多方面的调整；再者，由于抓取功能的实现需要不同功能包之间的协作，编译、包、消息文件都需要有条理地进行编写，因此遇到了不少困难。具体的问题和issue对应如下：

**表8 第二次迭代问题及解决**

|  |  |  |
| --- | --- | --- |
| 问题 | 解决 | issue对应 |
| 如何驱动相机 | 可以仿照已有代码的相机驱动打开 | <https://github.com/sebuaa2019/Team201/issues/9> |
| 编译无法通过 | 修改CMake, package.xml, msg文件 | <https://github.com/sebuaa2019/Team201/issues/10> |
| 语音关键词无法识别 | 挑选有明显特征的语音关键词 | <https://github.com/sebuaa2019/Team201/issues/12> |
| bouding\_boxes数组计算出错 | 改动darknet\_ros发布的消息，增加数组大小信息 | <https://github.com/sebuaa2019/Team201/issues/13> |
| 物体检测不稳定 | 设置标志位重复检测，直达识别到物体为止 | <https://github.com/sebuaa2019/Team201/issues/14> |
| Waypoints文件不存在，程序运行报错 | 生成指定格式的Waypoints文件，使得程序运行更加稳定。 | <https://github.com/sebuaa2019/Team201/issues/20#issue-448891988> |
| 再次选择航点时，命名不应从1开始 | 采用正则表达式替换。 | <https://github.com/sebuaa2019/Team201/issues/22#issue-448894024> |
| 动态避障问题 | 实现遇到障碍物停止，障碍物移开继续前进，已经能够达到避开动态障碍物的要求。 | <https://github.com/sebuaa2019/Team201/issues/24#issue-448897575> |

# 迭代3阶段总结

## 任务分配

第三次迭代的任务主要分为抓取机械臂测试，各功能单元测试、UI完善三个部分。人员分为两小组：王润泽和李贞子负责机械臂测试、避障抓取单元测试和对应UI的完成；张璐、张佳琳、李天宇负责导航模块的单元测试和对应UI的完成。具体的任务分配与看板管理如下所示：

**表9 第三次迭代任务分配**

|  |  |  |  |
| --- | --- | --- | --- |
| 人员 | 任务模块 | 具体内容 | 看板对应 |
| 李贞子 | 抓取 | 完成UI"使用方法"部分 | <https://github.com/sebuaa2019/Team201/projects/7#card-22018515> |
| 完成抓取部分UI界面设计 | <https://github.com/sebuaa2019/Team201/projects/7#card-22018558> |
| 测试机械臂，调节各项参数，实现抓取功能 | <https://github.com/sebuaa2019/Team201/projects/7#card-22014815> |
| 完成check\_front\_obstacle函数单元测试 | <https://github.com/sebuaa2019/Team201/projects/9#card-22313335> |
| 完成find\_path函数单元测试 | <https://github.com/sebuaa2019/Team201/projects/9#card-22313321> |
| 王润泽 | 抓取 | 完成抓取部分UI按钮对应的响应事件 | <https://github.com/sebuaa2019/Team201/projects/7#card-22019037> |
| 完成UI “联系我们“部分 | <https://github.com/sebuaa2019/Team201/projects/7#card-22019054> |
| 录制演示视频 | <https://github.com/sebuaa2019/Team201/projects/7#card-22019080> |
| 完成FindObject函数单元测试 | <https://github.com/sebuaa2019/Team201/projects/9#card-22312439> |
| 完成Speak函数单元测试 | <https://github.com/sebuaa2019/Team201/projects/9#card-22312391> |
| 完成Findkeyword函数单元测试 | <https://github.com/sebuaa2019/Team201/projects/9#card-22312381> |
| 张佳琳 | 导航 | 完成UI"硬件设置"部分 | <https://github.com/sebuaa2019/Team201/projects/8#card-22032864> |
| 完成Init\_waypoints方法的测试 | <https://github.com/sebuaa2019/Team201/projects/8#card-22032959> |
| 张璐 | 导航 | 完成UI"注意事项"部分  整体UI的补充美化 | <https://github.com/sebuaa2019/Team201/projects/8#card-22032914> |
| path\_plan.cpp中Speak方法的测试 | <https://github.com/sebuaa2019/Team201/projects/8#card-22032940> |
| 李天宇 | 导航 | 完成UI"遇到问题"部分 | <https://github.com/sebuaa2019/Team201/projects/8#card-22529270> |

## 代码提交

这一阶段主要提价了UI和单元测试的代码，具体的代码提交内容和github对应如下所示：

**表10 第三次迭代代码提交**

|  |  |  |
| --- | --- | --- |
| 人员 | 代码内容 | 提交记录 |
| 李贞子 | 实现抓取功能UI | <https://github.com/sebuaa2019/Team201/commit/9f98f10c09a4b8fc44cea7be78d30b63edbcc7d0> |
| 实现“使用方法”UI | <https://github.com/sebuaa2019/Team201/commit/9f98f10c09a4b8fc44cea7be78d30b63edbcc7d0> |
| check\_front\_obstacle函数单元测试 | <https://github.com/sebuaa2019/Team201/commit/dc5446b2af58d059c24b11c2bd473e7f3e7e0ccd#diff-2e9d464597a13197e93f77725169f114> |
| find\_path函数单元测试 | <https://github.com/sebuaa2019/Team201/commit/be90de1b76be50f473753ca24ced7d8711f42d56#diff-c983d7533563f6c1c75f6d639125beda> |
| 王润泽 | 完成“联系我们”部分 | <https://github.com/sebuaa2019/Team201/commit/9ffeefdd825f3e338e2c9dccc7c8da915717982d> |
| 完成抓取部分UI按钮对应的响应 | [https://github.com/sebuaa2019/Team201/commit/4390eab005e7507702d133723727f32b58963ce1#diff-fce90544871426c432b1c2e5a3eb0253](#diff-fce90544871426c432b1c2e5a3eb0253) |
| findObject函数单元测试 | <https://github.com/sebuaa2019/Team201/commit/44ba77f6b50e85b9b69b2567f8d8658500107bad#diff-a3f22cb0361e259404597b9fea872f4c> |
| speak&talk函数单元测试 | <https://github.com/sebuaa2019/Team201/commit/602d4d2cb922edf92b823ad3b424596ddbbb290f#diff-af996862dccfdc66d898dc3d3f4c51ec> |
| findkeyword函数单元测试 | <https://github.com/sebuaa2019/Team201/commit/2a7562f51941a95941da01c4864073f7b3dc94cb#diff-4e006cc19a0e785a4c0ac1b755ad4dc3> |
| 张璐 | 完成注意事项部分UI | <https://github.com/sebuaa2019/Team201/commit/1ad7974e69644e8486fc8d732bd3211b72390b15> |
| 单元测试test\_Speak.cpp | <https://github.com/sebuaa2019/Team201/blob/master/ROS_CODE/TEST/test_Speak.cpp> |
| 完成“功能选择”,“关于MOSS”，“联系与帮助”部分的UI及美化 | <https://github.com/sebuaa2019/Team201/commit/6af7178ae6fc53e80293af4bd6689bad50a885d3> |
| 张佳琳 | 更新LeftTabWidget.py，完成“”硬件设置“”部分ui | <https://github.com/sebuaa2019/Team201/commit/27fa19dcab83ebf7877659e550d1da7179a774d8> |
| 单元测试test\_Init\_waypoints.cpp文件 | <https://github.com/sebuaa2019/Team201/blob/master/ROS_CODE/TEST/test_Init_waypoints.cpp> |
| ui整体界面美化 | <https://github.com/sebuaa2019/Team201/commit/99736f7e8df4d1517693847651d636e78da62a2d> |
| 李天宇 | 更新LeftTabWidget.py，完成“遇到问题”部分ui | <https://github.com/sebuaa2019/Team201/commit/9903b9b5974bd2306267067dd408c57a9daca52a> |

## 测试

**表11 第三次迭代测试**

|  |  |  |
| --- | --- | --- |
| 功能模块 | 测试内容 | 测试结果 |
| 导航 | InitWayPoints单元测试，测试是否能把航点标号按要求加入数组 | 符合预期 |
| Speak单元测试，测试机器人是否能正确发出相应语音 | 符合预期 |
| 导航部分集成测试 | 符合预期 |
| 抓取 | 抓取集成测试：语音标定航点，根据识别结果，测试机器人是否可以对指定物体进行抓取 | 可以正确根据识别结果并进行抓取 |
|  |  |
|  |  |
|  |  |
| 避障 | find\_path单元测试，测试函数正常功能，并进行分支覆盖 | 断言正常，函数功能正确 |
| check\_front\_obstacle单元测试，测试函数正常功能，并进行分支覆盖 | 断言正常，函数功能正确 |

## 问题

这一部分的问题主要和抓取有关，抓取很大程度上与参数设定、距离等因素有关，具体的问题和issue对应如下所示：

**表12 第三次迭代问题及解决**

|  |  |  |
| --- | --- | --- |
| 问题 | 解决 | issue对应 |
| 抓取参数设定 | 在wpb\_home.yaml里面修改grab\_lift\_offset和grab\_forward\_offset参数 | <https://github.com/sebuaa2019/Team201/issues/15> |
| 抓取距离过近，物体向前走 | 严格设定抓取距离,即物体据人50cm，人据机器人50cm | <https://github.com/sebuaa2019/Team201/issues/16> |

# 测试总结

## 测试策略

对于需求的覆盖，我们采用了全覆盖测试的策略。具体来说，TC1、TC2、TC3是三个集成测试，应用于具体的测试情境，覆盖了避障、导航、抓取所有的功能需求。TC11-TC20是针对单个功能需求具体的测试，根据每一个功能的不同组成模块进行拆解，并且设置多组数据进行测试，涵盖边界数据、复杂情况、和容错测试等，覆盖了所有的功能模块。除此之外还测试了接口的正确性和规范性，前后端之间的正确响应。非功能需求因为无法单独设计测试样例，因此覆盖方式采用“边测边观察”，测试表明系统满足安全性、易用性、可维护性等多种非功能需求。此外，单元测试部分（TC4-TC10）是为了重要函数设计的专门测试，全部考虑到了分支覆盖，达到了100%的函数内的分支覆盖率。

测试的设计在三种不同的层面上。有针对单个重要函数的测试（TC4-TC10），称为单元测试；有针对单个功能的测试（TC11-TC20），称为功能测试；有针对一个整体功能，即避障、导航、抓取（TC1-TC3）设计的测试，称为集成测试。测试样例基本上一次成功，但是也发现了一些问题，比如抓取的成功率不够高，主要和识别过程中的环境条件及距离有关，较好地排除周围环境的干扰并且采用合适的距离，抓取的成功率可以得到显著提升。

## 测试用例概述

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例分类 | 测试用例名称 | 编号 | 测试用例描述 |
| 集成测试 | 避障用例 | TC1 | 避障需求的集成测试 |
| 导航用例 | TC2 | 导航需求的集成测试 |
| 抓取用例 | TC3 | 抓取需求的集成测试 |
| 单元测试 | FindKeyWord | TC4 | 检测字符串A中的字符串B |
| InitWayPoints | TC5 | 检测航点编号加入数组函数 |
| Speak | TC6 | 机器人语音播报信息情况 |
| CheckFrontObstacle | TC7 | 检测前方障碍物返回函数 |
| FindObject | TC8 | 从数组检出预设物体函数 |
| FindPath | TC9 | 找出前往预设物体路径函数 |
| Speak&Talk | TC10 | 机器人follow领航员并跟读函数 |
| 功能测试 | 导航地图构建 | TC11 | 导航地图构建与存储 |
| 构建航点 | TC12 | 航点新建与保存 |
| 航点异常处理 | TC13 | 航点在地图外的异常处理 |
| 地图未设定异常处理 | TC14 | 未设定地图时新建航点异常处理 |
| 避障速度测试 | TC15 | 避障模式速度设定 |
| 避障时间测试 | TC16 | 避障模式时间设定 |
| 避障随机性 | TC17 | 避障模式机器人行走随机性测试 |
| 预设目标地点 | TC18 | 抓取模式预设目标地点 |
| 识别预设物体 | TC19 | 抓取模式从预设目标地点中选出预设物体 |
| 导航并抓取 | TC20 | 导航到目标物体并抓取 |

## 测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例分类 | 测试用例名称 | 编号 | 测试结果 |
| 集成测试 | 避障用例 | TC1 | 所预期的 |
| 导航用例 | TC2 | 所预期的 |
| 抓取用例 | TC3 | 与要求有小幅偏差 |
| 单元测试 | FindKeyWord | TC4 | 所预期的 |
| InitWayPoints | TC5 | 所预期的 |
| Speak | TC6 | 所预期的 |
| CheckFrontObstacle | TC7 | 所预期的 |
| FindObject | TC8 | 所预期的 |
| FindPath | TC9 | 所预期的 |
| Speak&Talk | TC10 | 所预期的 |
| 功能测试 | 导航地图构建 | TC11 | 所预期的 |
| 构建航点 | TC12 | 所预期的 |
| 航点异常处理 | TC13 | 所预期的 |
| 地图未设定异常处理 | TC14 | 所预期的 |
| 避障速度测试 | TC15 | 所预期的 |
| 避障时间测试 | TC16 | 所预期的 |
| 避障随机性 | TC17 | 所预期的 |
| 预设目标地点 | TC18 | 所预期的 |
| 识别预设物体 | TC19 | 所预期的 |
| 导航并抓取 | TC20 | 与要求有小幅偏差 |

## 测试问题与分析

**主要发现一大问题：**

机器人抓取成功率比预想低10%左右（实测20次，抓起14次，预期16次）。

**问题分析和解决如下：**

修正这个问题需要我们使用更加精细的雷达或相机，对当前环境进行动态的数据处理，而不是使用预设的数据对机械臂进行操作。除此之外，抓取功能的实现对环境布置和识别算法有一定要求，目前采用的YOLO可以识别多个物体，但是准确率不够高，要求不能有杂物，光线也有一定影响。因此修正这个问题需要更加强大的软硬件资源和一定人力的软件实现，这会导致整个产品发布的拖慢。因此这里评估认为10%抓取成功率的低于预期是可以被接受的。故没有提出issue要求解决发现的这个问题。

# 团队协同总结

要点：从团队协同角度，概述团队成员的分工模式，并梳理各个成员实际承担和完成的任务。并对本项目团队的协同效果进行评价。

# 感受与建议

## 个人感受

**李贞子：**

软件工程这门课的学习过程中，我收获了很多，从一开始对嵌入式开发、软件开发的不熟悉，到现在对ROS系统有更加深入的了解和认识，以及作为组长的角色对小组的管理更加成熟，自己的确学习到了很多，也收获了很多。

首先，自己对ROS系统和嵌入式开发有了更加深刻的认识和了解，从一开始对ROS系统完全陌生的状态，到现在和小组成员一起开发出具有避障、导航、抓取功能的机器人，自己的代码能力得到了提高，同时也接触到了一个全新的领域，这个过程是辛苦的，也是十分开心的。软硬件的结合不仅需要对代码有成熟的运用，也需要对硬件的组成和运用方式有深入的了解，嵌入式开发使自己从另一个角度认识到了软件工程在多领域的应用。

其次，团队管理方面，作为组长的角色管理一个团队，的确是一项有挑战的任务，自己以前从未有过小团队项目管理的经验，但是本学期的软工课程让自己对这一角色有了更加深刻的认识和体会。无论是小组成员的管理、协调，还是任务分配和沟通，都让自己的各方面能力得到了提高，也可以从整体和宏观的角度对软件开发流程有较为清晰的认识。看板管理和github各种工具的使用也使得管理更加高效和便捷。

再者，本课程提供了很多机会进行文档书写，以前从未有过课程对文档书写有较多的需求，而文档对于软件工程来说，具有指导性的作用，项目开发计划帮助我们对整个开发流程有更加清晰的认识，需求文档则可以明确开发的每一项任务,设计文档对于代码的开发至关重要。因此，每一项文档的书写，都使得自己体会到了软件工程的内涵。

除此之外，课堂、老师和助教对我们的帮助无疑也是巨大的，在学习的过程中给了我们小组很多支持和帮助。总而言之，本学期的课程从管理、代码、文档等各种方面都使自己对软件工程的开发有了更加清晰的认知和了解。

**张璐：**

经过软件工程这门课程的学习，我在以下方面收获颇多：

代码能力方面，因参与设计并撰写了导航部分的源代码与界面ui，实现在Ubuntu系统下的软件开发与调试，现已较熟悉Ubuntu系统下的软件开发过程。本课程是嵌入式软件开发，经过课堂的教授和课下的训练，现已较熟悉嵌入式开发的流程。

文档撰写方面，因参与撰写了需求分析文档，设计文档，测试文档等，对文档撰写方式及组织形式有了一个比较清晰的认识，经过一次次文档的评审及修改，我对整个项目的认识也得到了进一步的提升。软件工程课程让我认识到了文档的重要性，在以后的开发过程中，我将按照这门课程学到的文档管理方式合理撰写文档。

团队协作方面，之前的课程都是自己独立完成作业，软件工程课程上，我首次与自己组内的同学合作撰写一部分代码，并用github记录自己的提交记录，用看板记录自己的任务完成进度，用issues发表与讨论设计中遇到的问题。在这个过程中，我们从最开始的难以沟通到最后的合作有序，这些都培养了我们的团队协作意识。

总而言之，软件工程这门课让我学到了许多知识，多次文档撰写，三次迭代和测试虽然任务量大，但跟着老师的规划一步步执行，最终也能实现最初设定的功能。

**李天宇：**

在学习软件工程这门课的过程中，我有很多收获。

首先通过课上环节的教学，我理解了一个团队进行软件开发的过程。团队进行的软件开发首先都是规模不算小的软件系统，因此不能上来就莽代码。通常，整个过程是经过需求分析、设计，最后才是实现和测试发布。其次由于是团队进行开发，因此管理模式和协作开发的方式是十分重要的。在实际生活的软件开发中，一个软件很可能是由身处不同国家的开发者协同完成的，这就需要良好的说明文档规范和良好的代码风格来保证开发的高效性。

虽然本次课程使用了github进行代码管理，但是由于个人问题没能好好利用。实际上在这门课程的学习之后我对git有了更深的理解，也理解到代码版本管理和备份到可靠地方的重要性（参考某次我们组代码被删除临时又写一遍的情况）。

关于课下收获的部分，由于我前期在代码初构建时忙于别的事情，疏忽了和团队的交流。导致团队那边的开发都基本进行的差不多了我才回头发现这个问题，自己实际上没有做特别多的代码工作。因此在最后测试部分和一些文档上希望下更多的功夫来弥补一下工作量的问题。这也说明了整体团队的协作性和沟通交流是十分重要的——我没有主动找到和我负责同一个项目的同学进行沟通，导致我在前期开发没有什么参与度这个结果。虽然后来也参与到开发的收尾工作中，但是实际上肯定比不上前期开发的工作。

结合上述经历，我总结的收获就是，软件工程是一个专门的学问。我们通过合适的开发流程和管理方案实现一个软件项目的开发，并且通过审查手段保证整体项目的质量。

**张佳琳：**

这学期接触到了这门新的必修课，感受还是非常多的。据说这是软件工程第一次作为必修课出现，也说明了学院对我们工程能力培养的重视。而今年又新开设了嵌入式的这个方向，为同学们提供了更多的选择。这种多方向的开课其实是对同学非常好的，大家可以选择自己感兴趣的方向，最终都能达到一个比较满意的效果。总体来看，这个学期的收获如下：软件工程这门课是一个非常考验团队协作的课，与去年大二的面向对象设计与构造主要锻炼个人的能力不同，这门课没有整个团队的协作配合是完成不了的，在整个学期的分析设计、代码实现、测试等环节的工作中，大家都是必须靠着合理的分工、有序的协作来完成的；另外，在学习的过程中，我们也对软件工程中很多的步骤有了更深的了解，比如在需求设计的环节，我们了解了需求分析的过程中需要关注到的重要的内容，在测试的环节，我们了解了诸如单元测试、功能测试、集成测试等不同部分测试的重点；在嵌入式开发上，我们接触到了这样一个非常新奇的项目，学到了很多在之前没有了解过的知识，感谢老师在课程中对我们在设备上的支持。由于篇幅的限制，就不说太多了。这学期，非常感谢老师们和助教们的辛苦工作，没有老师的耐心的教导，我们就不会学到这些在开发中非常关键的各种知识和技术。希望这门课在未来的教学将会越来越好。

**王润泽：**

学完这门课后收获了很多，以前以为软件开发就等同于写代码，但事实上这门课教会我怎么系统性地去规划整个过程，首先明确需求，其次构思设计，然后再开始写代码，这样写代码的时候思路就会提前在脑子里有一个很清楚的架构，实际实现时也就不是毫无规划的去写。再者就是在github上管理项目真的十分方便，能够实时跟踪团队进度、解决开发中的问题、查看代码版本，一目了然，这对于规模化系统化的开发无疑是一个强大的辅助。

再有就是深深感到用户需求对软件的重要性。成功的软件产品是建立在成功的需求基础之上的，而高质量的需求来源于用户与开发人员之间有效的沟通与合作。当用户有一个问题可以用计算机系统来解决，而开发人员开始帮助用户解决这个问题，沟通就开始了。需求获取可能是最困难、最关键、最易出错及最需要沟通交流的活动。对需求的获取往往有错误的认识：用户知道需求是什么，我们所要做的就是和他们交谈从他们那里得到需求，只要问用户系统的目标特征，什么是要完成的，什么样的系统能适合商业需要就可以了，但是实际上需求获取并不是想象的这样简单，需要在设计过程中反复讨论需求的可行性和完整性并对其进行修改

## 建议

1. 机器人只有一个，**在高峰期使用人数太多**，在1027等一天实际使用时间也就**2-3**个小时，浪费很多时间在**无意义的排队等待**上。课程中如果**增加机器人数量**或者**拉长每次迭代的持续时间**就可以缓解扎堆现象。

2. github项目管理应该在开始的课上就作为**强制和统一**的需求提到（开始是说**最好**用某个团队工程开发工具），否则导致一些组后期才开始补上commit记录、issue和看板（之前并没有受过这方面的训练，根本没有这个意识），我们**非常理解课程组对于这方面的要求的初衷，在实际操作中也切身感受到这样做对于管理整个开发过程是非常有意义的**，所以课程应该有提前统一的规划，不应让管理沦为形式。

3. 万事开头难，绝大多数计算机学院的学生从未接触过嵌入式开发的过程，应该在课堂上涉及一些启智机器人开发的**基本知识/**多给一些**参考资料**/设计一些关于启智机器人基本模块的**实验**（移动、抓取、导航、麦克风、相机）。掌握了这些知识以后学生也可以运用这些基本模块自己去编写一些**更加复杂和系统化的**的开发，才能实实在在锻炼规模化的软件开发能力，而不是仅仅依赖和执着于修改（甚至照抄）既有例程来达到课程的要求。