# 如何实现建图和导航

# 前言

软银机器人自NAOqi 2.9.1开始提供了建图工具和导航框架辅助用户进行室内导航的开发。

# 使用建图工具创建地图和点

## 安装建图工具

Pepper建图工具是基于QiSDK API构建的辅助开发者高效完成室内地图创建的应用程序。支持在NAOqi 2.9.1及以上版本使用。

建图工具目前处于内测中,请通过热线 400-639-1680、软银机器人微信公众号、售后邮箱support@softbankrobotics.com.cn,联系 技术支持提交测试申请,后续会作为中国大陆地区机器人的默认安装应用。

## 前期准备

- 由于激光传感器扫描识别范围约为3-5米,所以建图场地不应过于空旷,3米内需要有明显的标识物
- 选择人员走动最少的时间,将临时的障碍物清除干净,对于长期放置的物品应收拾整齐



## 创建地图

将Pepper放在任意建图起始点(建议将实际行走时的起始点作为建图起始点),点击上图中的新建地图按钮或者轻拍Pepper头部 开始创建地图。

确保Pepper周围3米内没有人,在等待Pepper原地定位结束后,按照语音提示,用户可以从后面接近Pepper。

推动Pepper在整个建图场所内行走(在推动过程中,步伐应尽量远离Pepper以减少创建地图中的噪音点)

点击完成建图按钮或者轻拍Pepper头部结束,然后按照提示输入地图名称来完成建图

## 删除地图



按住想要删除的地图向左划,点击删除按钮来删除地图

## 保存标记点

选择想要添加标记点的地图,在建图标记点完成定位后,将Pepper推至欲标记的位置

点击添加标记点按钮, 输入标记点名称来保存标记点

注意:在实际行走过程中,相邻两个标记点之间的距离不要超过6米,如果距离比较大,可以在中间多建一些辅助点(Pepper到达该点可以不执行任何行为)

## 修改标记点

Р 🛨 🖞 🔒 🛎		🔊 ≭ 👽 8:17
Pepper建图工具		
地图列表 123 Labby office meeting room	<section-header><section-header><section-header></section-header></section-header></section-header>	市         1         2         3         5
$\triangleleft$	0	

按住想要删除的标记点向左划,点击删除按钮来删除标记点

## 使用Navigation framework实现导航逻辑

Navigation framework 是基于软银Pepper机器人在NAOqi 2.9.1及以上版本开始支持的导航框架。

Navigation framework 提供了面向于业务需求的简单导航接口,封装了使用QiSDK API 实现导航时的复杂逻辑,让用户更专注于 业务开发。建议初级开发者可以优先使用Navigation framework来实现导航逻辑,如果有更多进阶功能定制的需求也可以直接使用 QiSDK API中Navigation相关的API。

### 功能

- 获取建图工具创建的地图列表
- 获取相对应地图的标记点列表
- 定义标记点顺序
- 自动导航
- 导航过程中的事件回调处理

## 引入依赖包

首先,在Android工程中,添加maven仓库:

```
maven {
    url 'http://maven.softbankrobotics.com.cn/releases'
}
```

在Android工程的 build.gradle 文件中添加以下依赖:

implementation 'cn.softbankrobotics.transam:navigation:1.0.6'

### 在应用中实现导航逻辑

#### 实现Pepper导航逻辑的大致步骤

- 创建Pepper机器人应用,详见链接
- 获取需要导航的地图
- 加载预先设置的标记点
- 根据业务需要对标记点进行排序
- 根据设置好的标记点顺序进行自动导航

#### Navigation Framework API介绍

• 创建 RobotMapService 服务, 这里需要注意, qiContext 需要从 onRobotFocusGained(QiContext qiContext) 函数中获得

```java @Override public void onRobotFocusGained(QiContext qiContext) { RobotMapService robotMapService = new RobotMapServiceBuild().setContext(this, qiContext).build(); }

\* 获取当前所有的地图名称

java List maps = robotMapService.listRobotMapName();

```
* 加载选定的地图数据, 并获取对应地图的标记点
```

java // 获取指定的地图 String mapName = <建图工具保存的地图名称>; RobotMap robotMap = robotMapService.getLocalRobotMap(mapName); // 获取地图中的标记点, Key为建图时输入的标记点名称, 建议在此步骤做标记点 是否存在的检查 Map positions = robotMap.getPositions();

\* 创建 NavigationService 服务

java NavigationService navigationService = new NavigationServiceBuild().setRobotMap(robotMap).setContext(qiContext).build();

\* 根据业务需要对标记点排序为名称数组并转换为Position对象

java String[] positionNames = ...; List positions = navigationService.getPositions(positionNames);

\* 添加 NavigationService 服务的监听事件

java navigationService.setListener(new NavigationListener() { /\*\* \* @param cuurentPosition 当前的标记点 \* @param nextPosition 下一 个标记点 \* @param errorMessage 错误信息 \* @param errorType 错误类型 \*/ void onError(Position currentPosition, Position nextPosition, String errorMessage, Enum errorType) { //在导航中,发生任何的错误,都会通过此函数返回。 }

```
/**
* @param position 当前的标记点
* @return 是否开始从此标记点开始进行导航
*/
boolean leaveFromPosition(Position position) {
 //离开标记点
}
/**
* @param isArrived 是否到达此标记点
* @param position 到达的标记点
* @param errorType 没有到达此标记点,返回错误类型
* @return 是否继续进行下一个标记点
*/
boolean arrivedAtPostion(boolean isArrived, Position position, Enum errorType) {
 //到达标记点
}
```

#### });

\* 根据Position 列表开启导航, 依次进行点的移动

java navigationService.startByPosition(positions);

\* 停止导航

java navigationService.stop();

### 注意事项

```
由于当前Pepper的平板是Android 6.0 (API 23) 平台,用户需要在应用运行时向其授权,而不是在安装程序时授权,让用户可以对应用的功能
```

在使用导航框架时, 需要在 AndroidManifest.xml 文件中添加:

#### xml

同时在应用中添加运行时授权(推荐使用 easypermissions 第三方库):

groovy implementation 'pub.devrel:easypermissions:1.0.1'```