**准备工作**

* 拷贝exercise\_ec\_ws到自己建立的ROS工作目录的/src文件夹下面，然后运行catkin\_make编译。(需要实验人员熟悉ROS的基本操作)
* **Exercise 1： 路径规划**

**问题描述**

* 简介：给定二维栅格地图，栅格地图根据激光数据标记栅格障碍物占用信息。实验人员需要实现一种路径规划方法，输入栅格地图中机器人当前位置、目标位置以及当前栅格地图障碍物占用情况，输出能够到达目标位置的安全路径。
* 具体工作：编写路径规划函数pathPlan。该函数位于src/exercise\_1/src/car\_move.cpp文件中。
* 函数接口如下

std::vector<std::string> pathPlan(const std::vector<std::vector<GridCell> >& grid\_map, std::string start\_point, std::string target\_point);

**1、输入：**

grid\_map: 环境栅格地图，以二维数组形式表示，每一个元素是一个结构体变量，包含该栅格的名称，数值和坐标三个信息。说明：（a） 名称以Ax\_y表示，每个栅格对应一个唯一的名称。Ax\_y代表二维栅格地图中第x行，第y列。（b）数值表示该栅格占用情况，0表示无障碍物， 1表示有障碍物。（c）坐标表示该栅格在全局地图中的数值位置。

Start\_point: 路径规划起始点，以地图上栅格名称表示

Target\_point: 路径规划终点，以地图上栅格名称表示

**2、返回：**

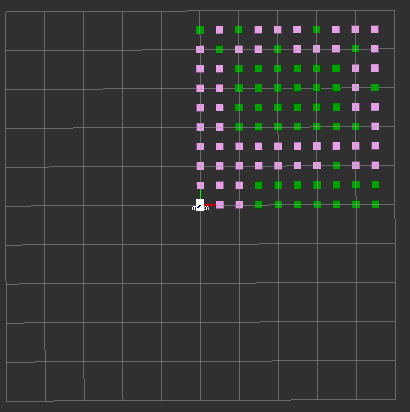
返回值类型为std::vector<std::string>: 表示由当前位置到目标位置需要经过的所有栅格名称序列。

**仿真环境启动步骤**

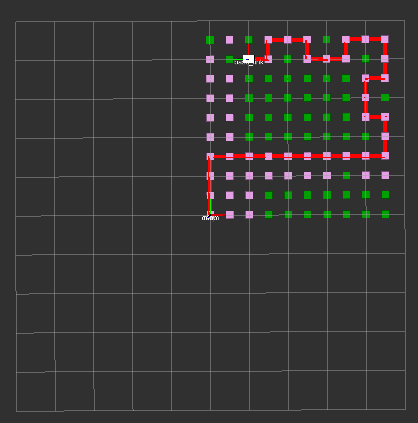
* 加载launch文件运行程序

roslaunch exercise\_1 exercise\_1.launch

可视化界面中会显示机器人当前位置和栅格地图，白色长方体代表机器人当前位置，粉色方块代表地图中非障碍点，绿色方块代表地图中障碍点。如下：



输入‘s’开始运行程序观察效果，红色为通过算法Dijkstra算法搜索出来的路径。



* 编写路径规划函数pathPlan。
* 修改完后，运行catkin\_make编译通过后，运行launch观测结果是否正确：

roslaunch exercise\_1 exercise\_1.launch

注：实验人员可通过修改scripts下面的地图数据(mapdata.json, mapData.txt)修改地图大小(不建议)，也可通过修改障碍物数据(obsdata.json, obsData.txt)修改障碍物信息增加测试用例来验证算法。

程序58行 std::string file\_path\_name = "/home/handsomexian/tutorials\_ws/src/exercise\_ec\_ws/src/exercise\_1/scripts/obsData.txt"; 改成自己电脑本地的路径。

**Exercise 2： 轨迹规划**

**问题描述**

* 简介：给定一系列路径点，机器人当前位姿与速度，实验人员需要计算一系列轨迹运动控制指令，使机器人能够实现经过指定路径点运动，同时确保运动平滑且尽可能减少原地转向。
* 具体工作：编写路径规划函数trajectoryPlan。该函数位于src/exercise\_2/src/car\_move.cpp文件中。
* 函数接口如下

vector<RobotSpeed> trajectoryPlan(const vector<Point>& waypoints, const RobotPose& cur\_pose, const RobotSpeed& cur\_speed, double period, double command\_length)

**1、输入：**

waypoints: 路径点坐标

current\_pose: 机器人当前位姿

current\_speed: 机器人当前速度

period: 控制周期，即两次控制指令发送间隔

command\_length: 一次计算的控制指令条数

**2、返回：**

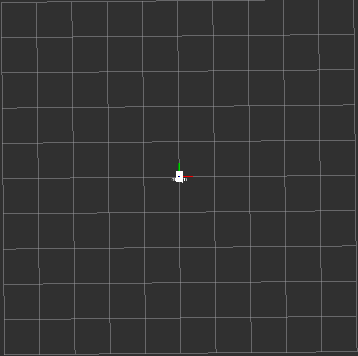
返回值类型vector< RobotSpeed>: 表示轨迹规划出来的一系列机器人速度控制指令。

**仿真环境启动步骤**

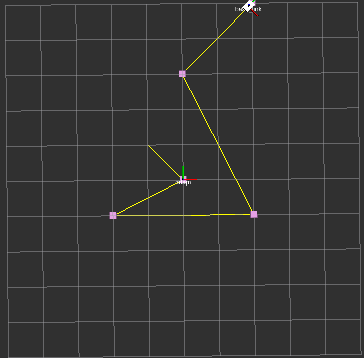
* 启动可视化界面

roslaunch exercise\_2 exercise\_2.launch

可视化界面中会显示机器人当前位置，以白色长方体代表:



输入‘s’开始运行程序观察效果，黄色为轨迹规划跟踪的路径。



* 编写路径规划函数trajectoryPlan。
* 修改完，运行catkin\_make编译通过后，运行launch检查结果是否正确

roslaunch exercise\_2 exercise\_2.launch

注：该仿真中机器人没有平滑经过所有路径点，而是在每个路径点上面原地旋转，没有任何平滑处理，只是起到演示效果，请实验人员自己编写轨迹规划代码。