OO\_7说明文档

1. 程序功能功能说明：

程序模拟出租车的乘客呼叫与应答系统，采用多线程的设计，并结合了相应的GUI即时显示结果。

1. 程序运行环境与运行指令规范

程序运行环境为Java(TM) SE Runtime Environment (build 1.8.0\_73-b02)，运行IDE为Eclipse Neon.2 Release (4.6.2)或者Intellij Idea 2017.1。如果用命令行编译并运行，编译指令为javac -encoding utf-8 TaxiSys.java，运行指令为java TaxiSys。最后输出结果保存在一系列logi.txt文件中。其中logi.txt记录了用户i的请求过程。

1. 程序注意事项

由于官方GUI的性能问题，在同时输入过多请求的时候可能产生程序运行的迟缓现象，因此如果想避免卡顿，可以将InputRequest类的第56、57行注释掉，取消显示用户请求范围的效果。

程序提供了一些Taxi类中的接口函数来查询出租车当前状态，它们是：getX():返回出租车当前所在行，getY()返回出租车当前所在列（注意这里的行列都是以1开始，以80结束的数字），getState()获取出租车当前运行状态，queryTaxi(int \_state):查询状态为\_state的所有出租车，返回一个ArrayList，至于出租车当前时刻，则调用getCurTime()即可返回当前毫秒数（这里的毫秒数是相对于出租车刚刚产生的时间，而不是相对于1970年那个时间）。

1. 程序输入说明

程序可以接受 [CR,(srcX,srcY),(dstX,dstY)]形式的输入，输入之间的空格都会被替换因此接受含有空格的输入。对于其中坐标值(srcX,srcY,dstX,dstY)是以1开始，以80结束的形式表示，这与以后的路径显示和车辆位置记录中所采用的是同一种表示形式。此外，程序还接受一行不超过10个请求输入，例如[CR,(srcX,srcY),(dstX,dstY)]; [CR,(srcX,srcY),(dstX,dstY)]...其中要求请求之间用分号隔开，而且注意程序采取的分隔方式是严格按照分号分隔，两个分号之间、第一个分号与输入行首位之间、最后一个分号与输入行末尾之间都被认为是一个请求。此外，程序接受输入”end”作为整个程序结束的标志。

输入的地图文件中允许有空行，地图的连通性需要由测试者保证。

1. 程序输出说明

程序对于格式错误的输入会报错，并且对于和以前的请求重复或者起始点重复的输入也会报错。程序会调用本次官方提供的GUI显示车辆与乘客状态。程序还会为每个用户产生一个记录文件，记录文件中记录了用户刚刚发出请求时，周围车辆的状态，信用(Customer#i First Logging: )。以及抢单时间窗口内所有抢过单的车辆(Taxis tried to grap Customer#i's order:)，最后被选中的车辆(Taxi chosen by Customer#i : Taxi#j)以及车辆将乘客从起点送到终点的路径(Path found for Customer#i)。注意乘客假如乘不到车那么这些记录就是不完全的。建议用Sublime Text等编辑器打开输出的log文件，那样显示效果会更好。

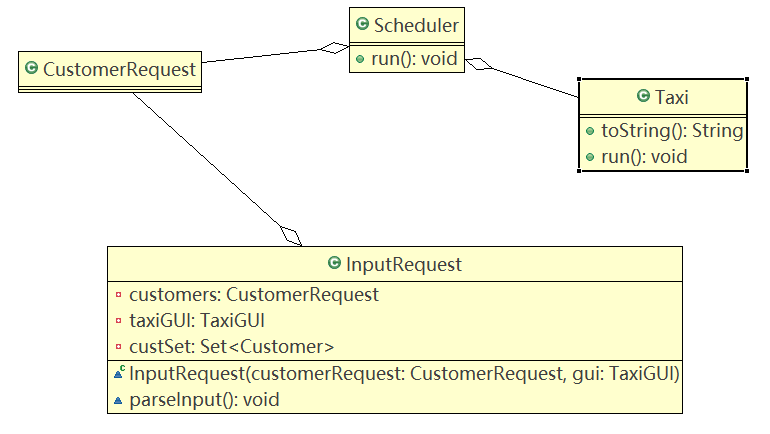
1. 需求分析

本次作业要求设计的是一个模拟出租车与乘客交互的系统，根据实际，我们可以自然地抽象出这样几个类：Taxi,Customer,Map，分别表示出租车，乘客，地图。此外还需要一些辅助的类来调度整个系统，如请求队列类，输入类，调度器类

1. 地图能处理给定的地图文件，得到地图文件的邻接表表示形式。便于后面处理最短路。
2. 出租车作为一个可以run的对象，能在没有任务时根据相应规则到处游走，在有任务时及时响应并完成任务。出租车的状态转换规则指导书已经说明，此处不再赘述。
3. 乘客可以抽象为一个请求，包含起始位置与终止位置。
4. 请求队列负责装载用户的请求。
5. 输入类提供将控制台输入转换为用户请求的能力，并与请求队列协同。
6. 调度器类负责处理请求队列中的用户请求。

为此，程序采用SOLID原则中的单一职责原则，每个类都只管理自己的事，例如出租车只负责导航与运行，乘客只负责上车等。

根据生活经验与本次指导书要求，我们可以设计出如下应答方式：乘客在发出请求的瞬间产生一个呼叫区域，这个区域持续3s，3s内任何到达该4x4区域(包括边界)的车辆都会给该乘客发请求(投简历)，乘客在3s后作出决定，从简历箱中选出此刻能响应的最优出租车，然后呼叫，出租车便立即去往乘客所在地，将其接走，送到目的地后重新回到等待服务状态。具体的关系如下：



总而言之，是调度器类从用户请求队列中不断取出到时间的请求，将其放入合适的出租车，出租车线程会根据输入的请求自动运行，而输入线程在另一边可以将输入的请求放入该请求队列，实现了测试的需求。