

需求分析

1、 系统简述

1) 系统功能描述

本次作业模拟了出租车的乘客呼叫与应答系统。

乘客可以发出乘车请求，然后系统会给出 3 秒的时间用来让指定范围内的出租车进行抢单，然后系统会根据一定的调度规则来选择一辆符合要求的出租车，最后被派单的出租车会按照最短路径到达乘客乘车点，最后载上乘客按照最短路径到达目的地。

平时未接单的出租车会自动在等待与停止两种状态之间切换。

2) 系统性能要求

java 环境：

jdk1.8.0_162 或 jdk-10

运行的 IDE：

Eclipse IDE for Java Developers

Version: Oxygen.2 Release (4.7.2)

Build id: 20171218-0600

在 eclipse 中导入 project 后直接运行 Main.java 文件即可。

系统运行过程中对性能的要求，取决于输入乘客请求的数量，输入请求越多，对系统的性能要求越高。

出租车每行驶一个格子需要 200ms，每 20s 空闲时间后需要 1s 的时间进入停止状态。在等待抢单的 3s 时间过后系统对参与抢单的出租车进行分配的时间取决于参与抢单的出租车数量，由几 ms 到几十 ms 不等，参与抢单的出租车数量越多，分配需要花费的时间越长。

3) 约束条件

程序通过控制台输入，每行只能输入一条请求，请求的格式类似于[CR,(10,10),(5,5)]，不支持多余的空格，前导 0，正号和其他符号，也没有结束符。

程序的输出结果输出到文件，每一条合法请求对应一个文件，文件位置在程序所在的主目录下。每次测试前请删除上次测试留下的文件，否则可能对新的测试结果形成干

扰。

对于无效请求的提示信息直接在控制台输出，测试接口对出租车的查询信息也在控制台输出。

2、 交互分析

1) 与系统有交互关系的对象

本次作业与系统有交互关系的对象主要包括三部分，地图、顾客、出租车。下面就三个对象分析它们的属性。

1. 地图：

大小为 80x80 的网格。

地图为连通图，任意两点都有路径可以到达。

出租车、顾客分布在点上。其中出租车的起始坐标通过随机产生，其后按照一定的运行规则运动。

2. 顾客

每个请求包含起始点坐标、目的地坐标，每次有 3 秒的时间留给出租车们来进行抢单。在这 3 秒钟，符合条件的出租车会像该顾客发送抢单请求，3 秒后选择一个最优的顾客来进行行驶到目的地。如果不存在一辆可以搭载顾客的出租车，则该请求者将不能坐车。

3. 出租车

出租车有 100 辆，最初随机出现在地图上。

出租车具有信用，初始值为 0，每抢单一次成功信用值加一，每次将顾客送到目的地信用值加 3。

出租车具有不同的状态，对应不同的行为。状态分为停止运行、等待服务、接单状态、服务状态

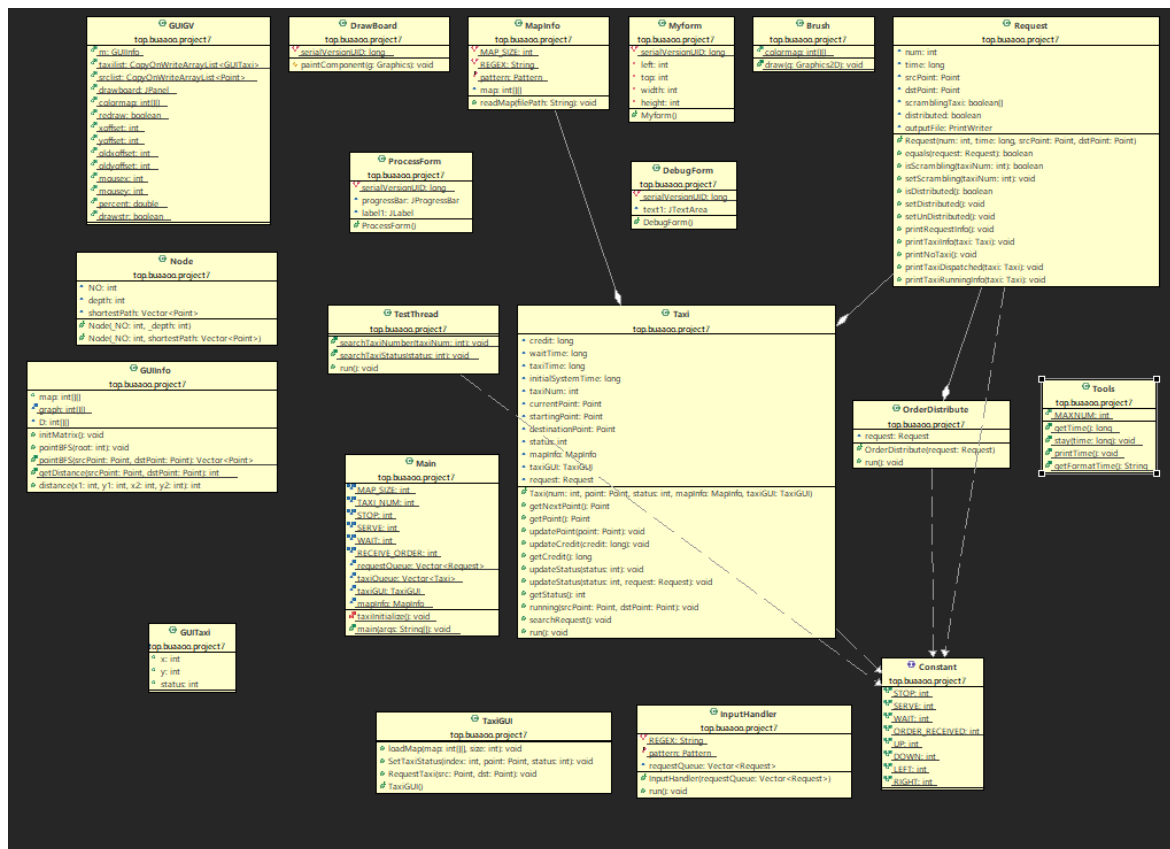
- 停止服务：在这 1 秒内停止；其后变为服务状态或者等待状态。
- 等待状态：在地图上随机行驶，只要在用户的请求窗口内便可以抢单。如果被顾客选中，那么装填变为接单状态。每等待状态形成 20 秒变进入一下停止服务状态。
- 接单状态：从出租车的位置以最短路径开到顾客的起始位置。随后进入运行状态。

•运行状态：以最短路径从顾客的起始位置行驶到终止位置。随后进入等待状态。

2) 待开发系统与上述识别对象之间的交互

交互发生在乘客请求与出租车之间，乘客发出请求后会向附近的出租车发出派单信号，出租车收到信号后会进行抢单，乘客发出请求与出租车运行及抢单是两个独立的过程，同时发生。

3) 对象识别与构造



4) 并发分析

对于 100 辆出租车每辆各有一个独立的线程，请求输入处理及队列管理者为一个线程，加入队列后会新建一个分配线程，负责抢单窗口的新建，以及最后订单的分配。

为了保证线程的安全，分配线程和出租车线程的部分方法加上了 `synchronized`，保证对共享资源的访问不会出错。