OO-第十一次作业-需求分析

2017/5/16

（一）本次作业主要包括三部分对象，分别为地图、顾客、出租车。下面就三个对象简单分析一下它们的职能。

1. 地图：

大小为80x80的网格。

地图为连通图，且只有相邻的两点才有可能有边相连，边为双向边。

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（Taxi）：

普通出租车有70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，每次将客服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。

特殊出租车也就是追踪出租车有30辆，与普通出租车主要的行为的不同在于它们可以走之前被关闭的道路，然后可以输出自己服务的详细信息。

4．红绿灯(Light):

在每一个交叉路口模拟一个红绿灯的情形，每一个红绿灯的初始的状态随机，每经过一个相同的时间其状态改变。

（二）线程设计

从上面的类的职能分析我们可以看到，出租车类和顾客类是我们这次工程的主要核心，而它们的桥梁则应该由一个调度器来分配执行。调度器能分析出每个顾客的请求，并把该请求信息送个每一个出租车对象进行分析，如果某一个出租车对象是符合条件的，那么它会把自己的“简历”投寄给这个顾客，再由顾客进行遴选。被抽中的出租车进行状态的转移。

根据SOLID原则的单一责任原则，我们应该将每一个类的责任划分到尽可能小的单元，因此我决定开这么几个类：出租车类进行模拟出租车的运动、乘客请求类进行分析每一个请求是否合法并能够提取出请求的信息、请求队列类把所有的请求放在ArrayList里以便统一分配、一个调度器类进行请求分配与安排出租车的运动、还有一个请求增加类负责读入请求并把判断出合理的请求送到请求队列里，这样每一个类各司其职，能够较好地完成任务。

程序采用多线程设计，每个出租车(Car)为一个线程，请求队列管理者(RequestAdd)为一个线程，由一个调度器(Dispatch)线程进行把请求队列中的请求分配到出租车。为了保证线程的安全，在请求队列(RequestQueue)和出租车(Car)类的部分方法加上了synchronized。

然后新增添的特殊出租车的设计继承之前写的Car类，取名为Car2类，其主要方法与Car类相同，只有部分行为不一致，满足LSP原则，具体分析在当前目录下的设计文档.docx里。