操作系统进程管理项目：

十字路口交通信号灯控制系统

李万葶（1652724）

Contents

一、分析

项目目的

1、熟悉线程 / 进程同步机制

2、理解信号量机制

3、理解多线程 / 进程调度

4、体会并理解多线程概念

5、设计十字路口交通灯控制系统

项目功能要求

1、模拟出十字路口的交通控制情况。

2、考虑东、西、南、北四个方向，每条路分为两个车道，每个路口设置一盏显示灯。为简单起见, 每种灯显示时间为8秒。

3、当东西（或南北）方向红灯时，所有车辆（除了消防车、救护车、警车）均排队等待，当东西（或南北）方向绿灯时，所有车辆按序行驶（不准超车）。

题目分析

这道题目涉及多线程和线程之间信息的共享，可以通过java提供的多线程和界面绘制技术实现。要点在于红绿灯定时变换、道路占用状态和车辆的行进算法，其中单位道路的占用状态和灯光信息应当全局共享，所以单独封装在公共类中。

二、设计

类结构设计

按照之前的分析，系统的类结构分为三种：车辆、路面和红绿灯。

对于车辆，本系统设置了车辆父类，它负责实现所有车辆共有的功能，它有两个子类，即普通车辆和特种车辆，它们负责实现普通和特种车辆的行驶方式，这两个类又各自有三个子类，分别是小型车、中型车、大型车与警车、救护车、消防车，它们的单位长度分别是2、3、4、2、3、4，全面地代表了各种不同形态的车出现的情形。这些子类负责导入特定一种车型的图片和尺寸，供绘制时使用。

红绿灯类中添加单独的红绿灯面板，让红绿灯颜色按照规定周期变化，路面类中标记了单位路面的占用状态。

成员操作设计

系统总功能包括：绘制界面、初始化路面状况、绘制新车

车辆的操作包括：判断是否堵塞、判断灯光状态、判断是否到达路口处、判断是否到达拐弯点、判断路况是否允许拐弯、判断是否到达界面中道路的终点以及拐弯、直行、消失。

界面设计

绘制双向十字路口模拟地图，红绿灯作为独立面板放置界面中，红绿灯只有一个，指挥南北方向的车辆行驶，东西方向的情形与南北相反，即南北方向红灯时，东西方向车辆可以通行。

车辆每三秒增加一辆，每秒钟所有车辆进行判断和行动，车辆的位置和路面状态发生变化，同时每秒钟重新绘制所有车辆，实现“移动”。

算法设计

算法中，路面被抽象为一个12*12矩阵，其占用情况用二维数组表示。车辆具有尺寸Size、坐标xy、车头朝向Diret、出现方向froms和前往方向gos等信息属性，通过各种判断操作还能够得到jam,toturn,canturn,arrive等状态属性。这些属性共同确定一辆车在某一时刻的行动。

三、实现

红绿信号灯实现

```
public void run() {
        while (true) {
                changeColor(1, 8, 8);
        }
}（黄、绿、红颜色闪烁时间分别是1秒、8秒、8秒）
public void changeColor(int Ystop, int Bstop, int Rstop) {
        for (int i = 0; i < Ystop; i++) {
                Winkle = 0;
                colors = colorT.Yellow;
                repaint();
                pause(500);
                colors = colorT.Pause;
                repaint();
                pause(500);
        }（红色和绿色同理，参见源码）
```

车辆行进判断实现

```
public abstract class Normaltype extends Cars {
  public void action() {
          arrive();（是否到达）
          toturn();（是否到达拐弯点）
          cross();（是否到达十字路口）
          jam();（前方是否堵塞）
          if(arrive==1) {
                  delete();
             Diret=null;
             Dirtnum=0;
          }（删除已到达终点车辆）
          else if((cross==1&&Light.Winkle!=1&&(Diret==Directions.SouthII
Diret==Directions.North))Iljam==1) {
```

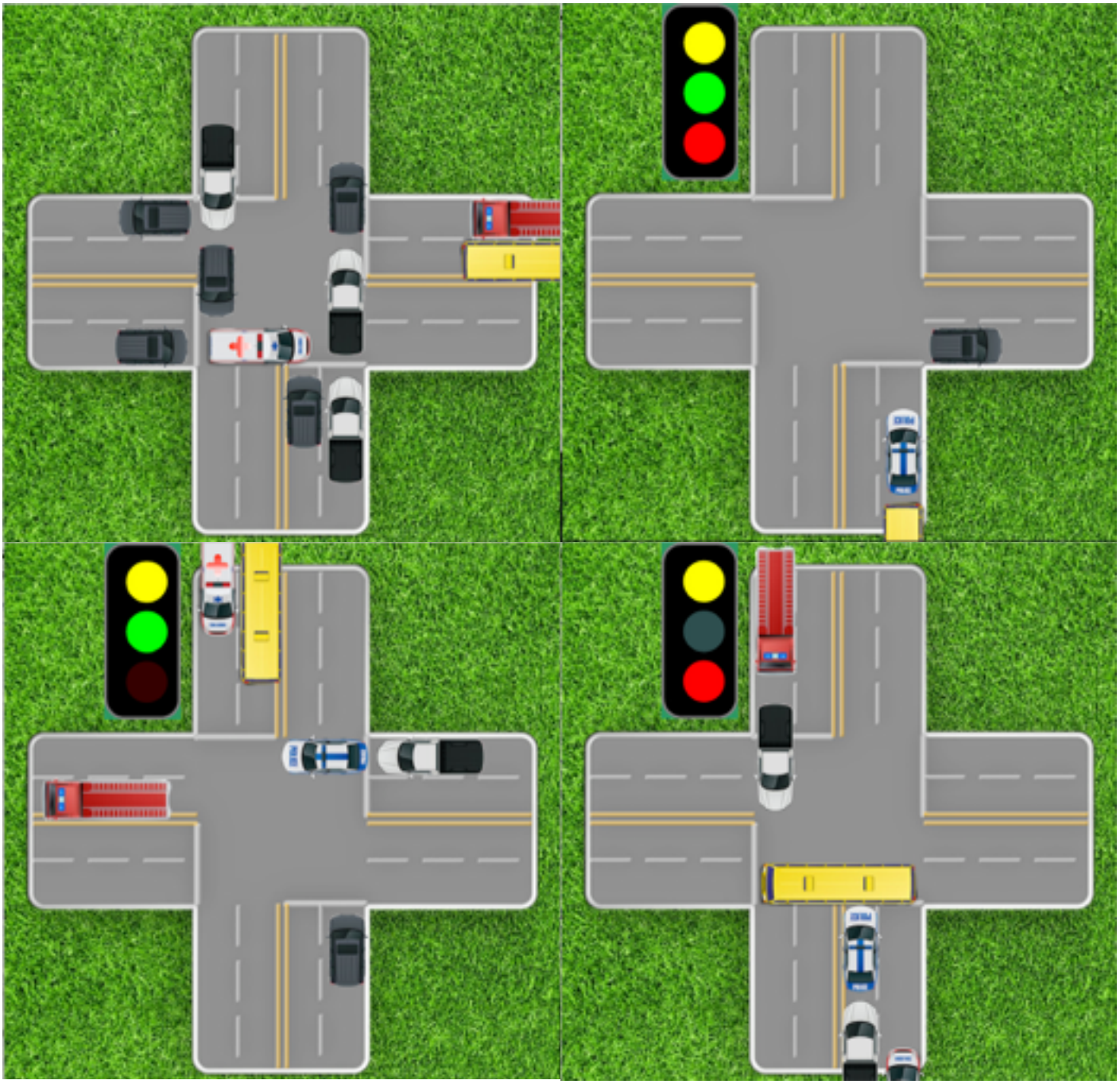
```
                }
                else if((cross==1&&Light.Winkle!=2&&(Diret==Directions.WestII
Diret==Directions.East))Iljam==1) {
                }（需要停止的情形，如果是在SpecialType中，则不需要判断十字路口和灯光
情形，只有jam==1情况下才停车，见源码）
                else{
                        if(toturn==1) {（到达拐弯点）
                  canturn();（路况允许拐弯）
                  if(canturn==1){
                                delete();
                                turn();
                                refresh();
                                straight();（拐弯，车头方向为目的地方向）
                  }
                        }
                        else straight();（直行）
                }
        }
      }
```

界面绘制实现

```
            timer.schedule(
                new TimerTask(){
                        int Timesnap=0;
                 public void run() {
                        Timesnap++;
                   if(Timesnap%5==1) nextOne();（每5秒创建新车 ）
                   for(int i = 0;i < cars.length;i++){
                        if(cars[i]!=null) {
                                if(cars[i].arrive==1) cars[i]=null;
                                else {cars[i].action();}（车辆逐一行动）
                        }else continue;
                   }  repaint();
                } },0,1000);（界面每隔1秒重绘一次  ）
```

四、测试

实现效果图



经本机测试，程序运行状况正常，正确地模拟了十字路口的车辆行进情况，符合题目要求与交通规则。如果可执行程序打开异常，请老师耐心重新运行一次，java源码也可以在eclipse中编译运行，谢谢您！