# Virtual Bicycle 人工智能方案(草稿)

## 通用组件:

### 概述:

通用组件的作用是把一个城市的街道,人行道,障碍物都表示出来

下面的方法不一定是最终版,需要很多的修改

**需要综合考虑的地方是**

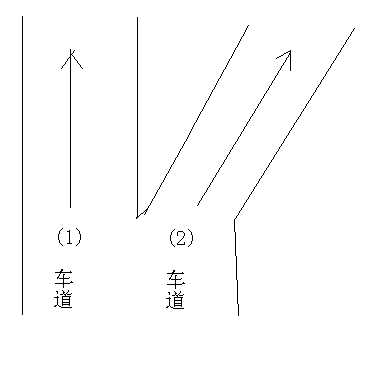
1. 普通车辆必须能够在指定的车道上行动
2. 比赛车辆可以在赛道的任何一个地方穿行
3. 行人必须在人行道上行走
4. 行人在红绿灯的时候可以穿过马路(斑马线)

综上所述,必须将所有的内容联动考虑,也就是说,公路,路口,人行道,斑马线,交通灯就是一个整体,如果是按照分离的方式来建立,反而有诸多的不便

### 公路:

#### 定义及表示方法

定义很简单,就是在城市里面可供车辆通行的道路

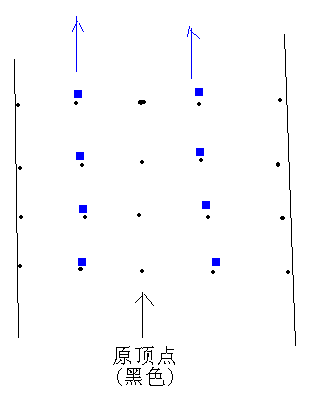


(图1)一条道路的表示:由车道和方向来组成

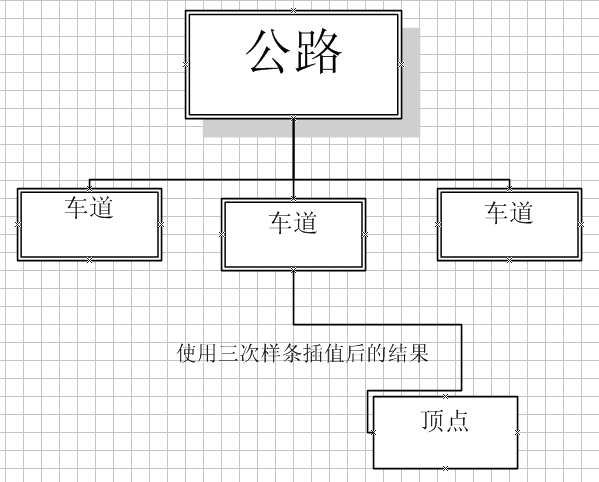
**车道的定义是**:在公路上能允许一辆汽车通行的空间,就为道路

车道可以通过在我们道路生成方案的元数据中得到.比如我们可以在原有的顶点表面上标记来得到,下面是我提供的一些标记的建议:

1. 车道数目
2. 车道的中点序列
3. 车道的方向



(图2)标记车道的方法



(图3)公路,车道,顶点的关系

**车道和车道间的转换关系**

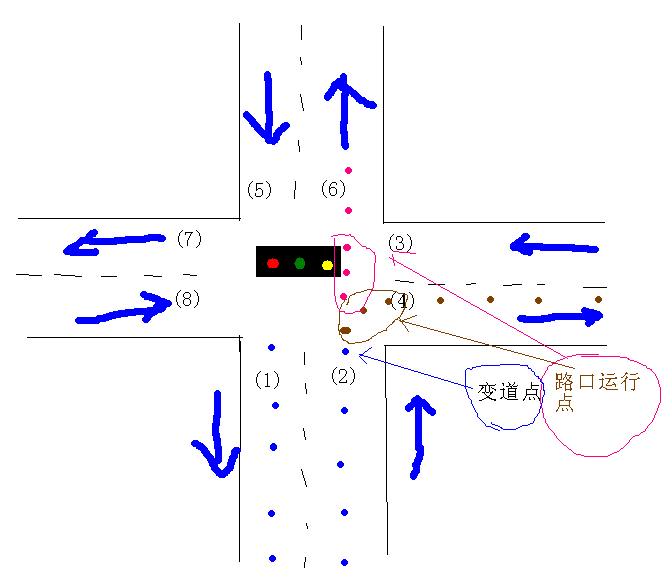
主要有两种:

1. 平行的多车道,车辆可以交换到其他任何一个道之中(变道)

过程:

1. 统计当前车道以及邻居车道中,所包含的车辆的数目
2. 有较大差距 -> 变道
3. 差距不太大 -> 概率决策
4. 当前车辆位于当前车道所有长度的1/2~1/3处(随机变道,不过如果变过一次了,就不再变道了,免得出现开过来又开过去的神经质行为)
5. 任何变道都需要满足,另外一条车道可以使得当前车过去(没有其他车挡着)
6. 多车道间可以设置车辆数目比例,以使得看起来更加真实
7. 前后关系的多车道,车辆可以经过一条曲线转换到另一个车道中.(转换曲线所处的位置也就是平时说的路口)

这个我还是先画一张图吧.



(图4)路口形式的变道情况

**过程:**

1. 一条车道可以有一个或者多个路径通往下面的车道

比如车道(2)可以通往(4)或者(6),通过时候的路径是在建立道路的时候就计算出来的(不过实时计算也是可行的,只是要麻烦一点)

1. 通过路口的时候有两种方案
2. 无条件,你可以随意通向下一条道路.比如单行道间的自然连接
3. 有条件,比如说红绿灯或者路障,如果是红绿灯就可以指出当前允许通向的下一条道路是哪一条

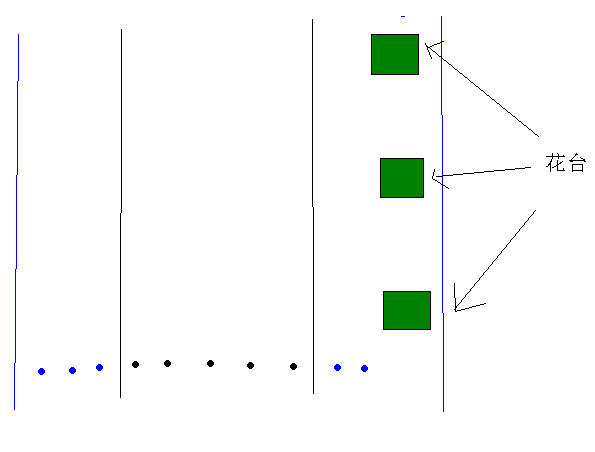
比如车道(2)在有些时候可以通向(4).有些时候可以通向(6)

1. 车辆在碰到条件选择下一条道路的时候可以随机的决策(也就是说可能会出现等待红绿灯的情况

### 人行道:

#### 定义

人行道的定义就不说了,可以跟车道类似,在原有的公路基础上直接定义,人行道在公路两边的位置,宽度,和道路的数目直接建立



(图5)人行道用蓝色的点标记,每个蓝色的点表示一条人行道

从图中也可以看出,花台是禁止通行的区域

**人行道的一些特点**

1. 和车辆差不多,如果两个人在一条道上拥堵了,那么一个人将会等待,另一个人变道行走
2. 人行道没有方向
3. 行人只会在一条道路上走来走去,不用通过路口也不用通过斑马线(免去一些不太容易被注意到的细节)
4. 如果行人碰到了道路的边界,那么就是简单的折返回来
5. 行人也会随机的变道

## 比赛对手:

1. 采用和自己完全一样的操作方式(加速减速,转弯/角度,刹车)
2. 速度/体力决策

当前条件允许加速时,当体力允许下才可以加速

1. 角度决策

当需要超车/转向时,规划路线,并且算出目的位置

1. 决策的过程是不停的走向下一个点,下一个点是通过枚举四周可以到达的点,计算他们的分数来得到的