数据库存储：

数据库表：

道路表、站点表、路口表、坐标表

1. 道路信息：T\_RoadInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Id | ID | Int | 自增序列 |
| 道路名称 | RoadName | Varchar[256] |  |
| 行驶方向 | Direction | Int | 0:上行 1:下行 |
| 起始点卡号 | StartCard | Int |  |
| 起始点路口编号 | StartCrossingCode | Int |  |
| 起始点路口方向 | StartCrossingDirect | Int | 0:东 1:西 2:南 3:北 |
| 结束点卡号 | EndCard | Int |  |
| 结束点路口编号 | EndCrossingCode | Int |  |
| 结束点路口方向 | EndCrossingDirect | Int | 0:东 1:西 2:南 3:北 |

道路名称+行驶方向组成唯一标识。

CREATE TABLE `T\_RoadInfo` (

`ID` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`RoadName` TEXT,

`Direction` INTEGER,

`StartCard` INTEGER,

`StartCrossingCode` INTEGER,

`StartCrossingDirect` INTEGER,

`EndCard` INTEGER,

`EndCrossingCode` INTEGER,

`EndCrossingDirect` INTEGER

);

1. 站点信息：T\_Station

|  |  |  |  |
| --- | --- | --- | --- |
| Id | ID | Int | 自增序列 |
| 站点名称 | StationName | Varchar[256] | 唯一标识 |
| 对应道路Id | RoadID | Int |  |
| 对应道路行驶方向 | RoadDirect | Int |  |
| 站点卡号 | StationCard | Int |  |
| 站点分组 | StationGroup | Int | 每一组内的站点可以到达，每组的站点互相不能到达 |

CREATE TABLE `T\_Station` (

`ID` INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,

`StationName` TEXT,

`RoadID` INTEGER,

`RoadDirect` INTEGER,

`StationCard` INTEGER,

`StationGroup` INTEGER

);

1. 路口信息：T\_Crossing

|  |  |  |  |
| --- | --- | --- | --- |
| Id | ID | Int | 自增序列 |
| 路口名称 | CrossingName | Varchar[256] |  |
| 路口编号 | CrossingID | Int |  |
| 路口方向 | Direction | Int | 0:东 1:西 2:南 3:北 |
| 直行编号 | StraightLine | Int | -1:不存在 其他:存在且唯一(本路口) |
| 直行对应方向 | StraightDirect | Int | 0:东 1:西 2:南 3:北 |
| 右转编号 | DextroversionLine | Int | -1:不存在 其他: 存在且唯一(本路口) |
| 右转对应方向 | DextroversionDirect | Int | 0:东 1:西 2:南 3:北 |

路口编号+路口方向组成唯一标识

CREATE TABLE `T\_Crossing` (

`ID` INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,

`CrossingID` INTEGER,

`Direction` INTEGER,

`StraightLine` INTEGER,

`StraightDirect` INTEGER,

`DextroversionLine` INTEGER,

`DextroversionDirect` INTEGER

);

1. 卡号坐标信息：T\_Card

|  |  |  |  |
| --- | --- | --- | --- |
| Id | ID | Int | 自增序列 |
| 卡号 | Card | Int | 唯一标识 |
| X轴 | Xpos | Int |  |
| Y轴 | Ypos | Int |  |

CREATE TABLE `T\_Card` (

`ID` INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,

`Card` INTEGER,

`Xpos` INTEGER,

`Ypos` INTEGER

);

道路规划流程设计：

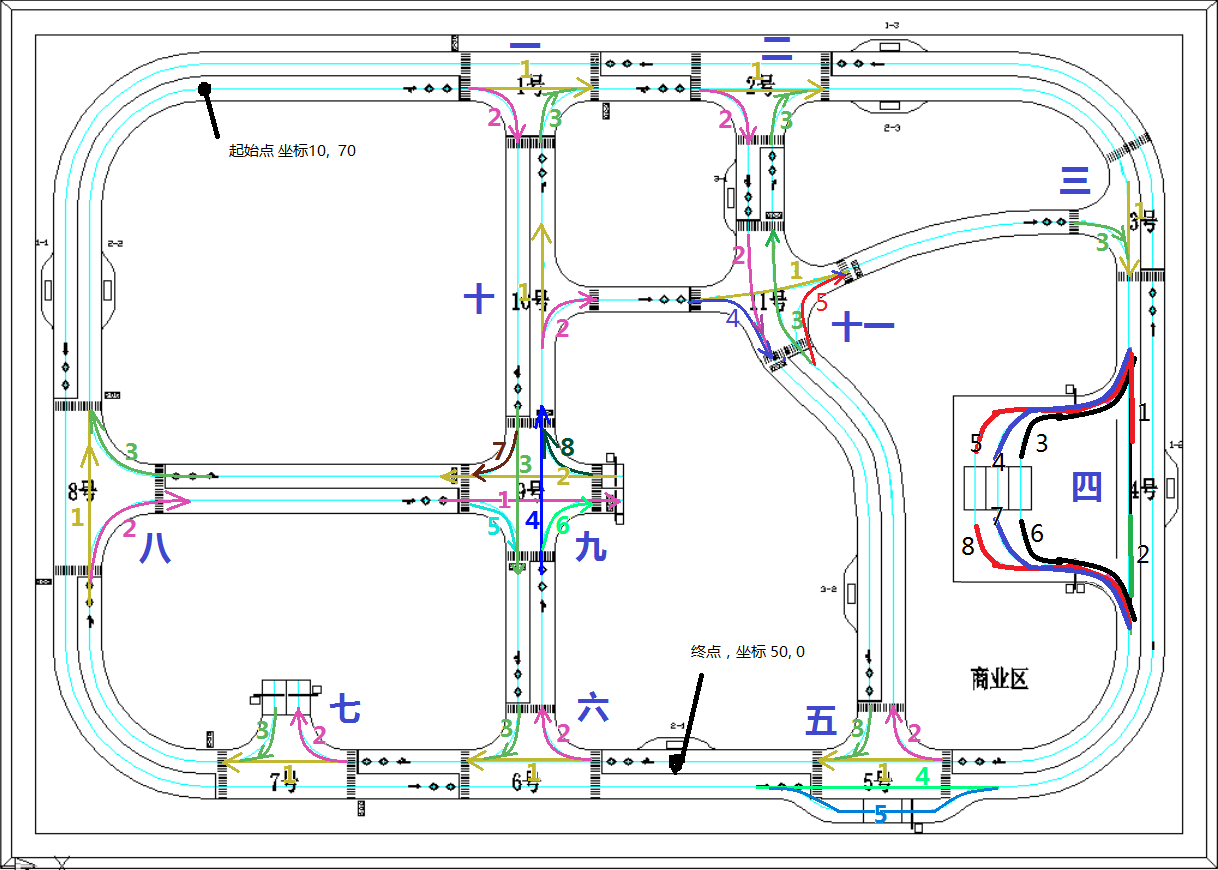
原理：

以地图的左下角为零点，纵向Y轴，横向X轴。将部分卡号（道路的起始点、结束点、站点卡号）的坐标记录到数据库。通过坐标的相对位置确定车辆的走向。由于本次系统的道路只存在直行和右转。所以本次道路规划的原则是：优先X轴方向靠拢，再Y轴方向靠拢。基于这个原则基本可以实现唯一的路径规划。

另，由于道路外环和内环不通，需要提前配置内环的站点和外环的站点连接信息。提前规避不能到达的站点。

具体流程如下：

地图坐标示例展示：



1.起始点和重点如图所示，当车辆启动后，行驶到达第一个十字路口（一号路口），坐标（35，70）。由于X轴坐标35小于终点X轴坐标，所以在十字口继续（向东）直行。

2.之后行驶到第二个十字路口（二号路口），坐标（55，70）。由于X轴坐标大于终点X轴坐标，所以在路口（向南）右转。

3.继续行驶到第十一号十字路口（十一号路口），坐标（55，55）。由于X轴坐标大于终点X轴坐标，Y轴坐标大于终点Y轴坐标所以继续（向南）直行。

4.继续行驶到第五号十字路口（五号路口），坐标（60，0）。由于X轴坐标大于终点X轴坐标，并且Y轴坐标等于终点Y轴坐标。所以在十字口（向西）右转。

5.判断该路段是否存在终点。存在终点记录终点编号。

以上就是路径规划的基本流程。依据形式过程中的相对坐标确定在每个路口的行驶方向。规则是：1.有限X轴方向靠拢，当X轴坐标小于终点坐标，向东行驶，当X轴坐标大于终点坐标，向西行驶。2.其次Y轴方向靠拢。当Y轴坐标大于终点坐标，向南行驶。当Y轴坐标小于终点坐标，向北行驶。

一些具体的设计：

1. 通过起始点查找道路名称方法：

select t1.RoadName from T\_RoadInfo t1, T\_Station t2 where t1.ID == t2.RoadID and t2.StationName = ''

1. 获取道路终点坐标方法：

select t1.Xpos, t1.Ypos from T\_Card t1 where t1.Card = (select t1.EndCard from T\_RoadInfo t1, T\_Station t2 where t1.ID == t2.RoadID and t2.StationName = '')

1. 坐标位置比较获取行驶方向方法：

如果由西向东行驶，当前X<终点X，则直行。反之右转。

如果由东向西行驶，当前X>终点X，则直行。反之右转。

如果由北向南行驶，当前Y<终点Y，则直行。反之右转。

如果由南向北行驶，当前Y>终点Y，则直行。反之右转。

行驶方向的判断方法为，判断道路结束点对应的十字路口的哪个方向。东则由东向西，西则由西向东，南则由南向北，北则由北向南。

1. 通过行驶方向获取对应路口的道路名称方法：

#define Straight 0 //直行

#define Dextroversion 1 //右转

Struct RoadInfo

{

|  |
| --- |
| ID |
| RoadName |
| Direction |
| StartCard |
| StartCrossingCode |
| StartCrossingDirect |
| EndCard |
| EndCrossingCode |
| EndCrossingDirect |

}

Long GetNextRoadInfo(const RoadInfo & CurrentInfo, const int nDirect, RoadInfo & NextInfo)

{

String str = “”;

Switch(nDirect)

{

Case Straight:

Str = “select t1.StraightLine, t1.StraightDirect from T\_Crossing t1 where t1.CrossingID = '' and t1.Direction = ''”;

Break;

Case Dextroversion:

Str = “select t1.DextroversionLine, t1.DextroversionDirect from T\_Crossing t1 where t1.CrossingID = '' and t1.Direction = ''”；

Break;

Default:

Break;

}

//查找下一个道路的信息

select \* from T\_RoadInfo t1 where t1.StartCrossingCode = '' and t1.StartCrossingCode = ''

}

1. 通过道路名称搜索终点方法：

select t1.StationName from T\_Station t1 where t1.RoadID = ''