# 自动驾驶交通流仿真系统

重点功能说明和模拟流程

# 一、软件概述

## 1.1设计目标

提供一个交通流模拟软件框架，用户将需要进行仿真的交通流模型经过一定的规范写成代码之后，交由该软件动态编译，并可视化对该交通流进行演示。

## 1.2 软件主体部分

软件主体部分采用Unity引擎制作，车辆，道路，路口等部件利用碰撞系统进行逻辑交互。软件主体包含各种基本的操作功能，提供了探测器面板供用户读取交通流模型计算中的数据。

## 1.3 二次开发接口

创建一个类库framework项目，链接提供TrafficDllPacket1.dll文件并且重写需要修改的模型方法。点击重新生成，可在项目文件夹\bin\Debug中找到对应的.dll文件。在软件中通过加载自定义模型打开此文件即可加载自定义模型方法。

# 二、系统环境

## 2.1 软件环境

操作系统: Windows 10

.NET框架：.NET Framework 4.7.1

## 2.2 硬件环境

CPU：Inter i7 8750H

内存：16G

硬盘：512G NVME SSD

（注：该环境为团队软件测试环境，实际使用可降低配置）

# 三、引言

## 3.1目的

本说明书是我方团队制作的自动驾驶交通流仿真软件的说明书，该说明书面向软件使用者，以便快速上手该软件。

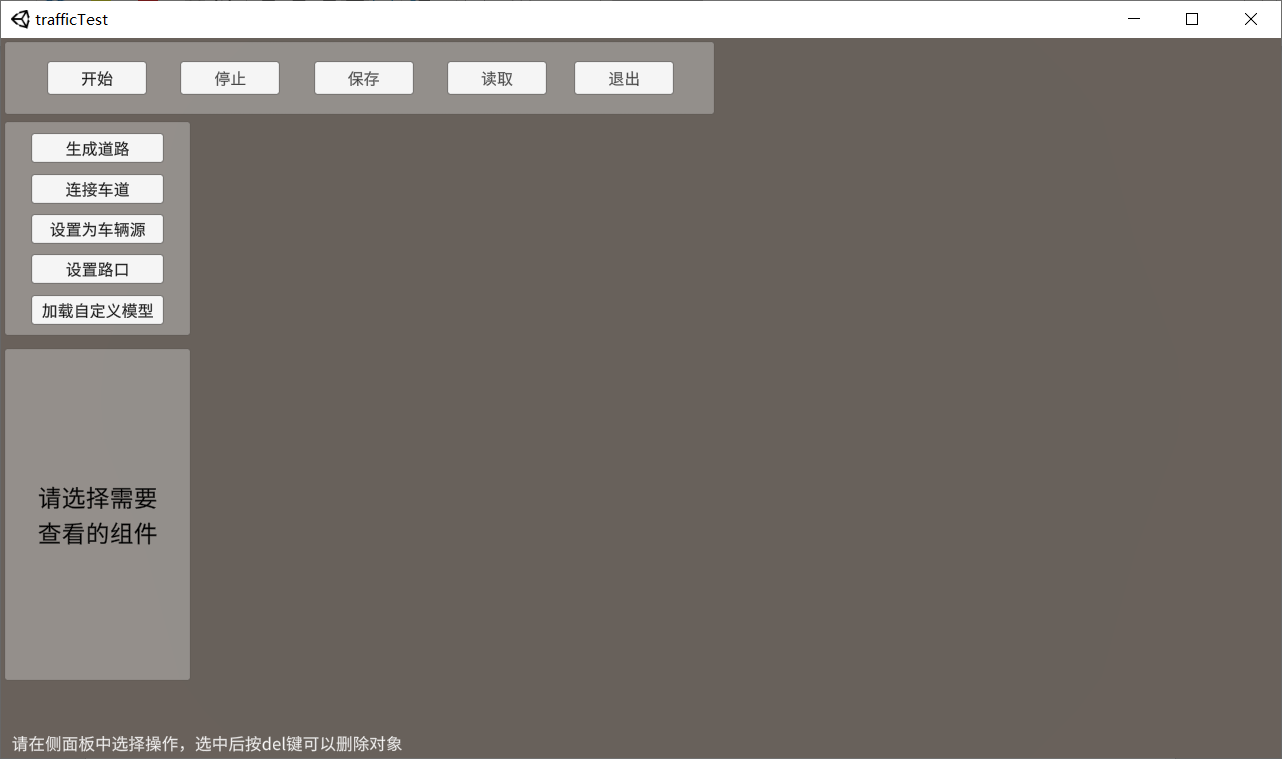
## 3.2范围

本说明书包括两个范围：基本操作和自定义模型导入。基本操作包括道路场景的设置，仿真演示的操作，以及对仿真内部数据的访问，自定义模型道路着重说明如何导入自己定义的模型，以及编写相关代码的规范。

# 软件使用手册

## 4.1 软件主界面

运行程序直接进入软件的主界面，软件左上角为工具栏，左侧从上到下依次为菜单栏和探测器面板，左下角为状态栏。其余部分为场景，仿真场景布置时都在场景进行。



图片 1 软件主界面

## 4.2 基本操作功能

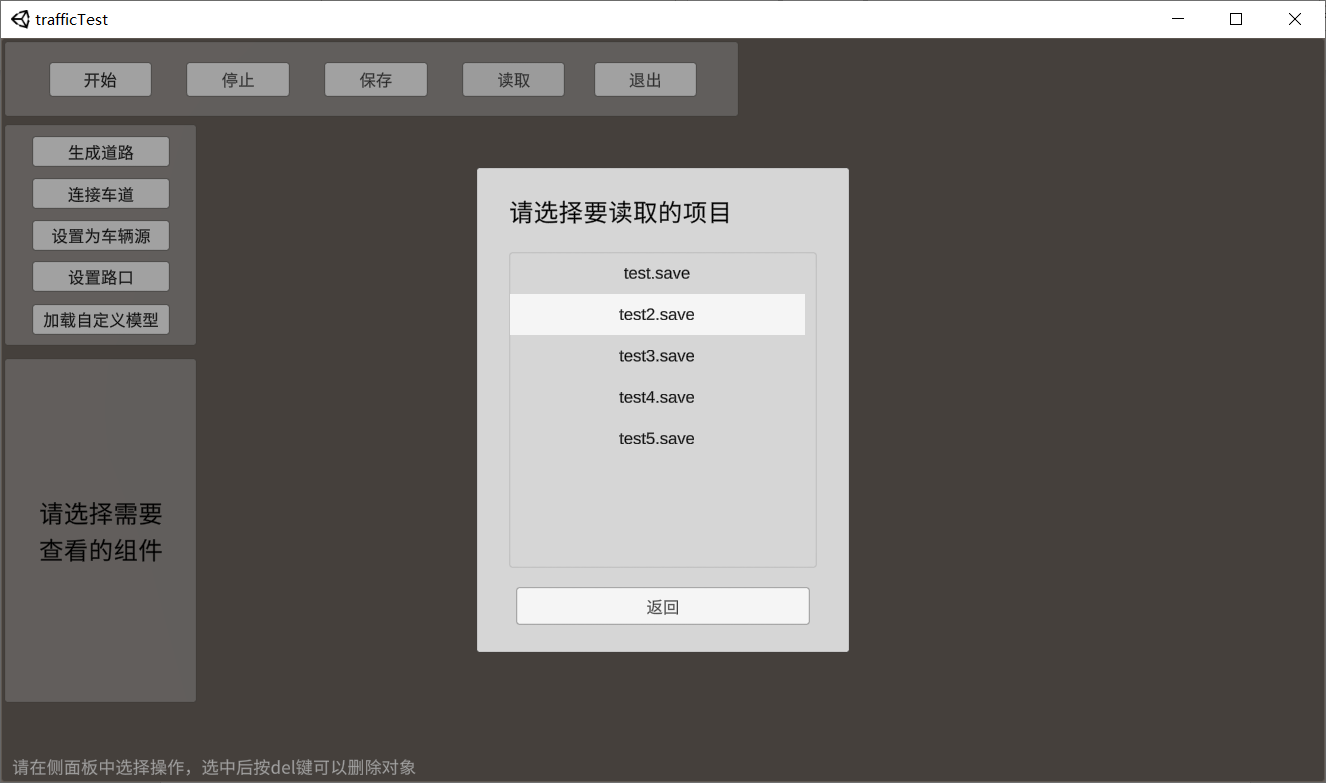
软件主要操作方式为鼠标。鼠标左键可以点击选中场景上的组件。按住左键拖动可以同时选中多个组件。软件允许一定程度上的调整视角，使用鼠标滚轮可以进行缩放，按住鼠标中键并拖动可以移动显示范围。

左上角的菜单栏包含开始/暂停、停止，保存和读取四个按键。点击保存按钮后会跳出对话框，用于输入存档的文件名。存档保存在软件同一目录下的trafficTest\_Data\saves文件夹中。



图片 2 文件保存对话框

读取存档时，软件会跳出对话框让用户选择存档，用户只需要在列表中点击对应的存档就可以直接加载。保存和加载结束后都会有对话框提示。



图片 3 读取对话框以及文件列表



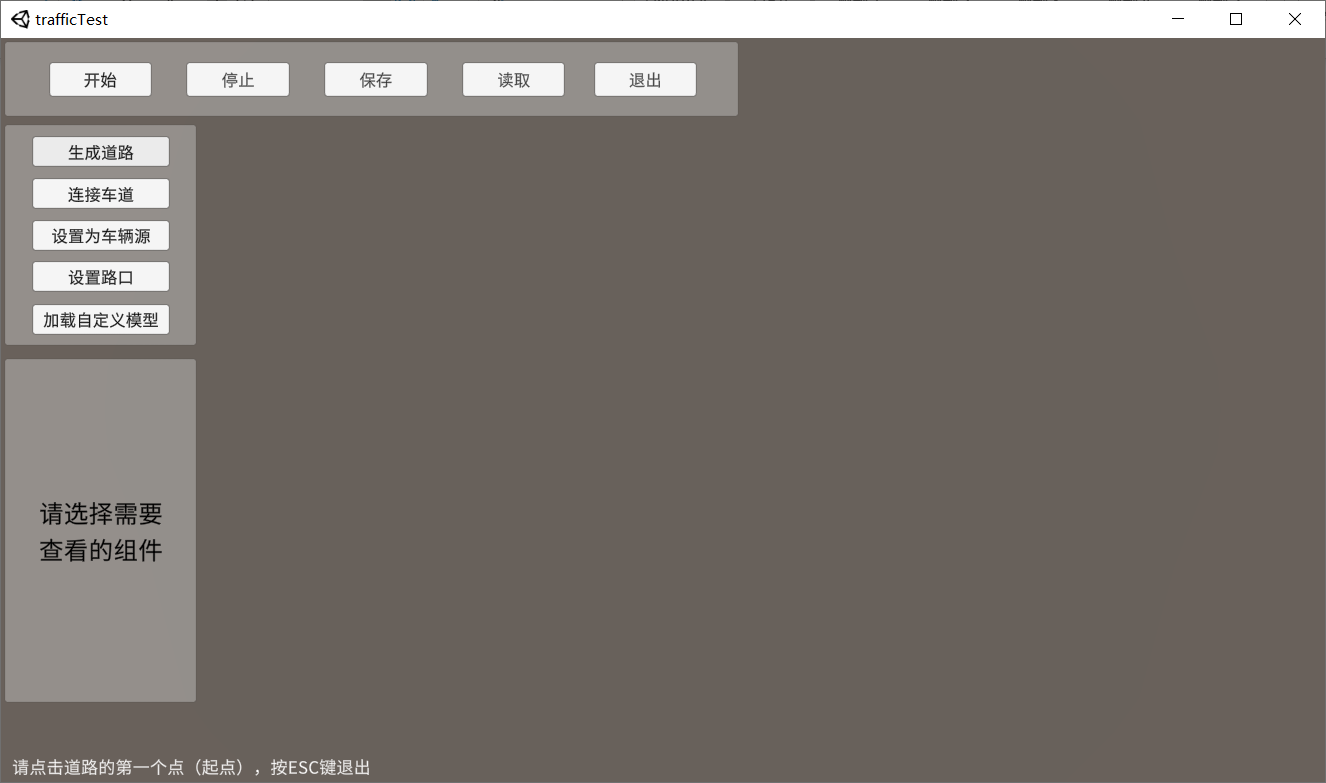
图片 4 读取成功的提示

## 4.3 仿真场景的布置

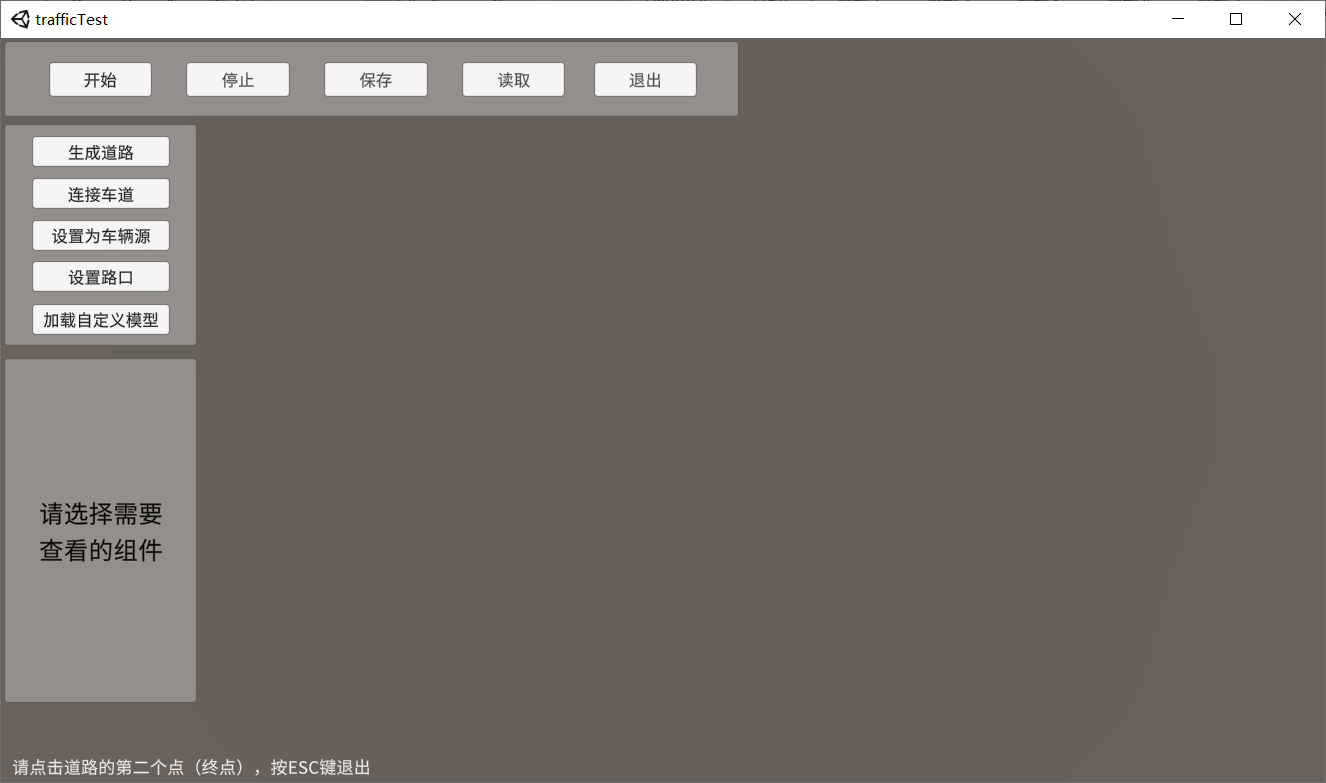
右侧的菜单栏面板有“生成道路”、“连接车道”、“设置为车辆源”、“设置路口”、“加载自定义模型”五个按钮，在点下对应按钮时即进入场景的编辑模式。此时左下角的状态栏会显示步骤提示，指示使用者如何进行以及停止操作。

### 4.3.1 生成道路

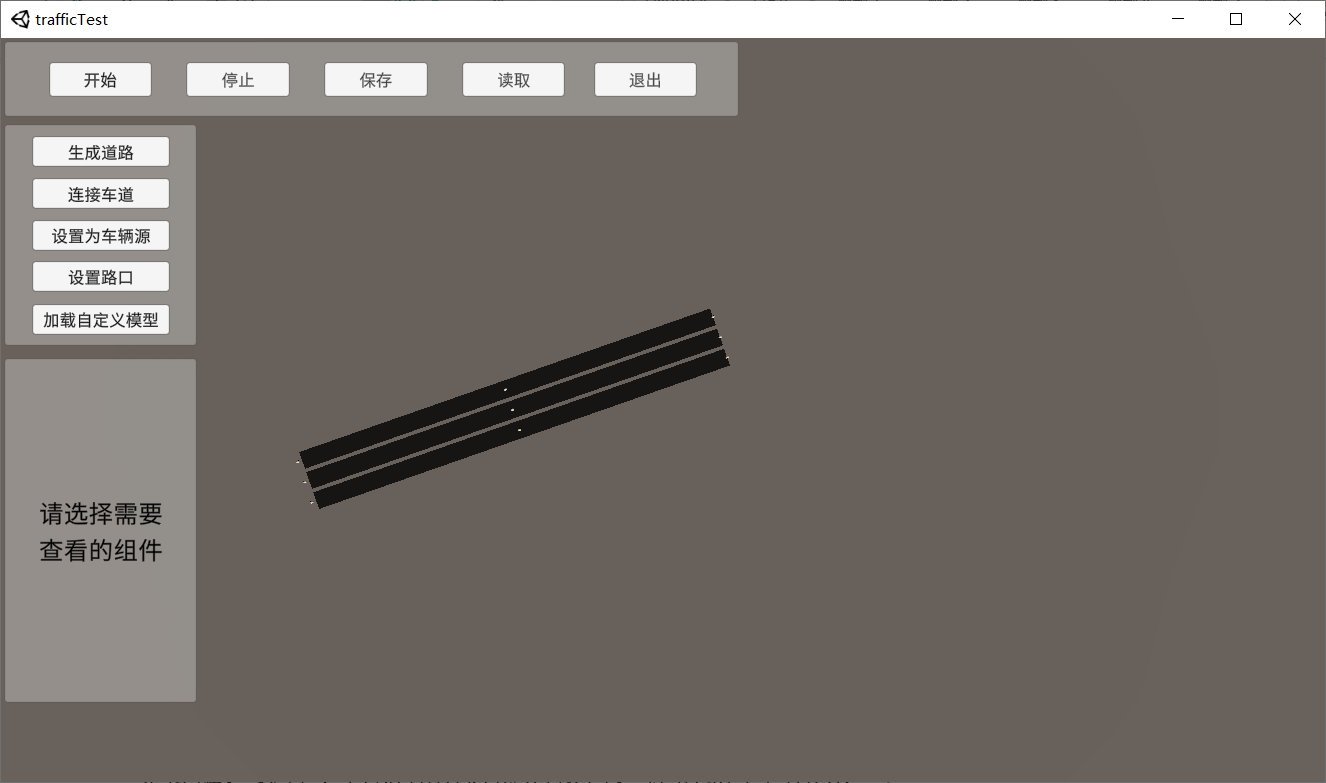
点击“生成道路”按钮后，软件进入道路生成模式。用户通过在场景中依次点击两个点来确定道路的位置。每一个步骤左下角均有操作提示，如下图：



图片 5 道路生成步骤1



图片 6 道路生成步骤2

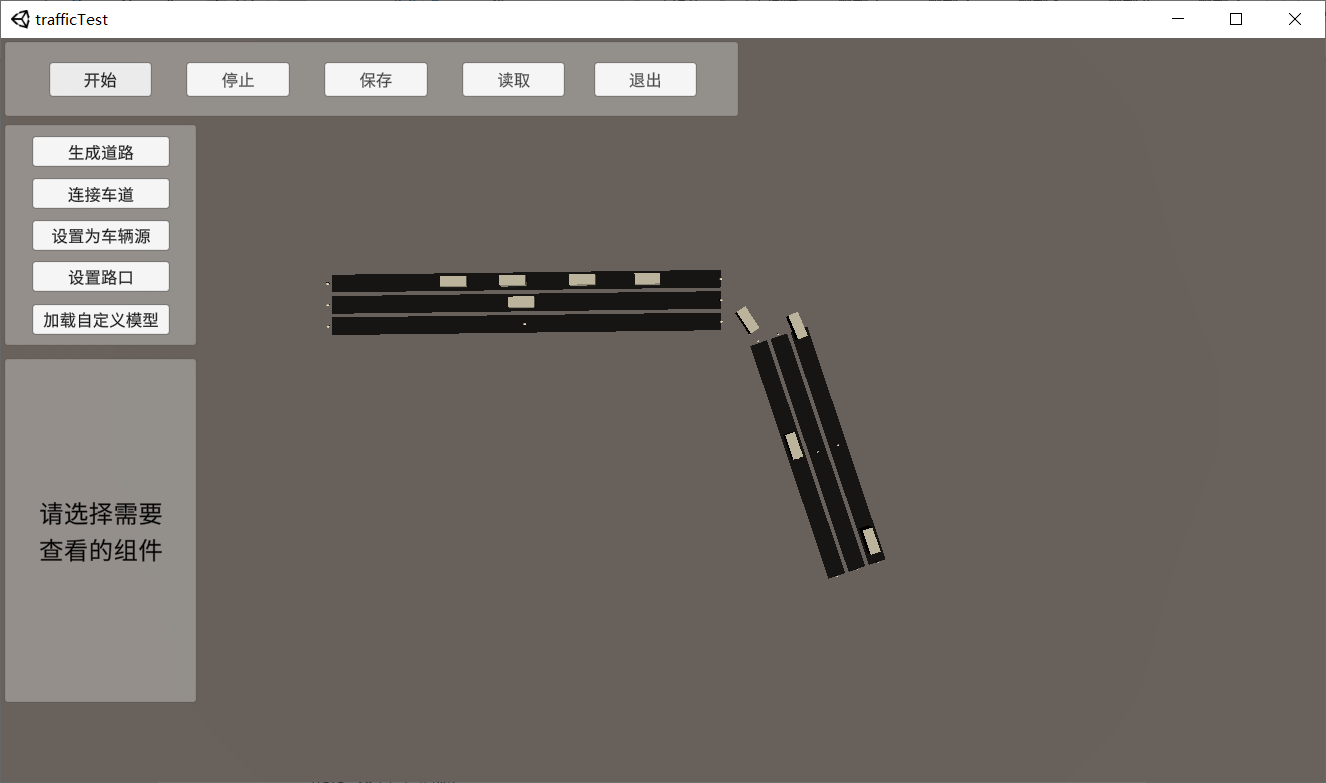


图片 7 道路生成完成

需要注意：1. 在道路生成完成后，默认不会生成车辆。若要使道路生成车辆需要点击“设置为车辆源”按钮。该功能将在后面描述。2. 道路生成时具有顺序，道路的行驶方向是从鼠标点击的第一个点到第二个点。

### 4.3.2 连接车道

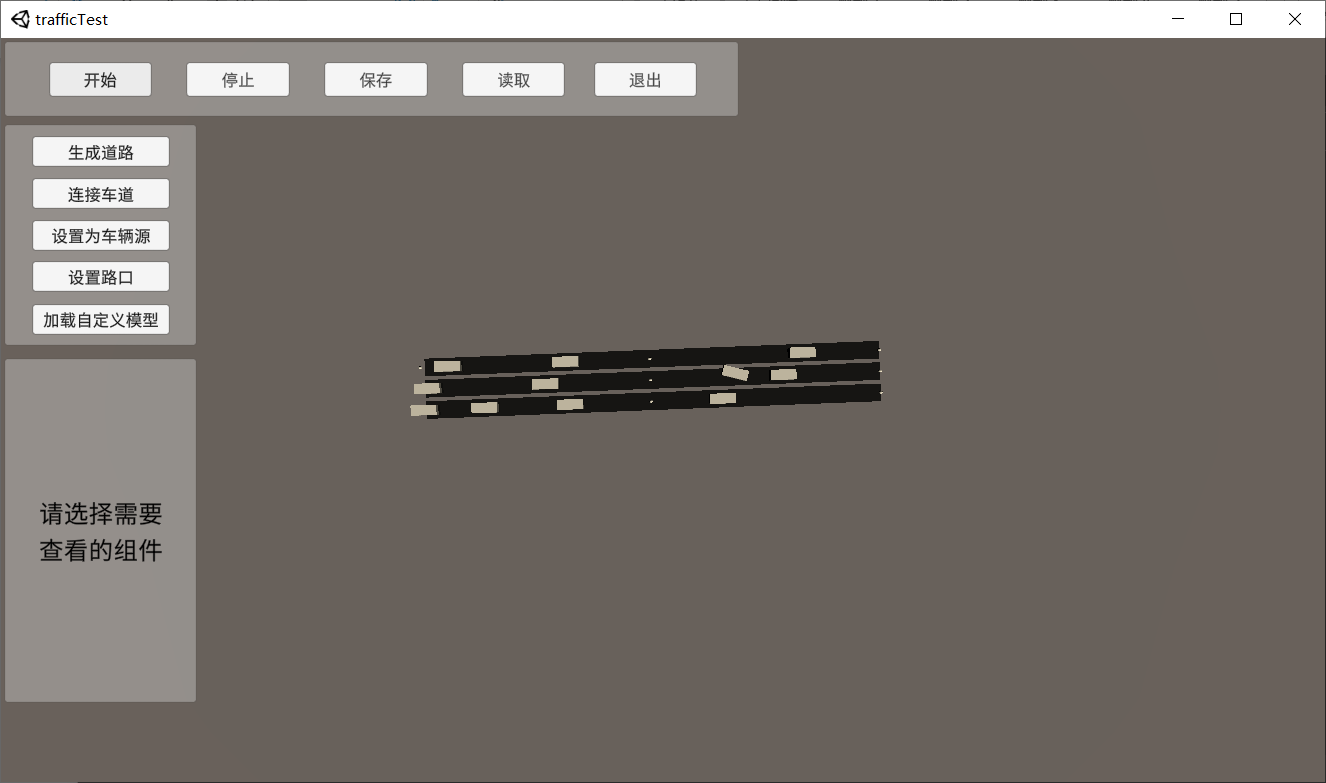
连接车道按钮同样位于菜单栏中。点击后即进入连接车道模式。左下角同样会有操作步骤提示。在连接时，点击的第一个对象应为车辆驶入交点的车道，第二个对象是车辆驶出的道路。该功能需要和“设置路口“配合使用。需要在两道路交汇的地方摆放路口部件。



图片 连接后的两条道路，车辆从一条行驶至另一条

### 4.3.3 设置车辆源

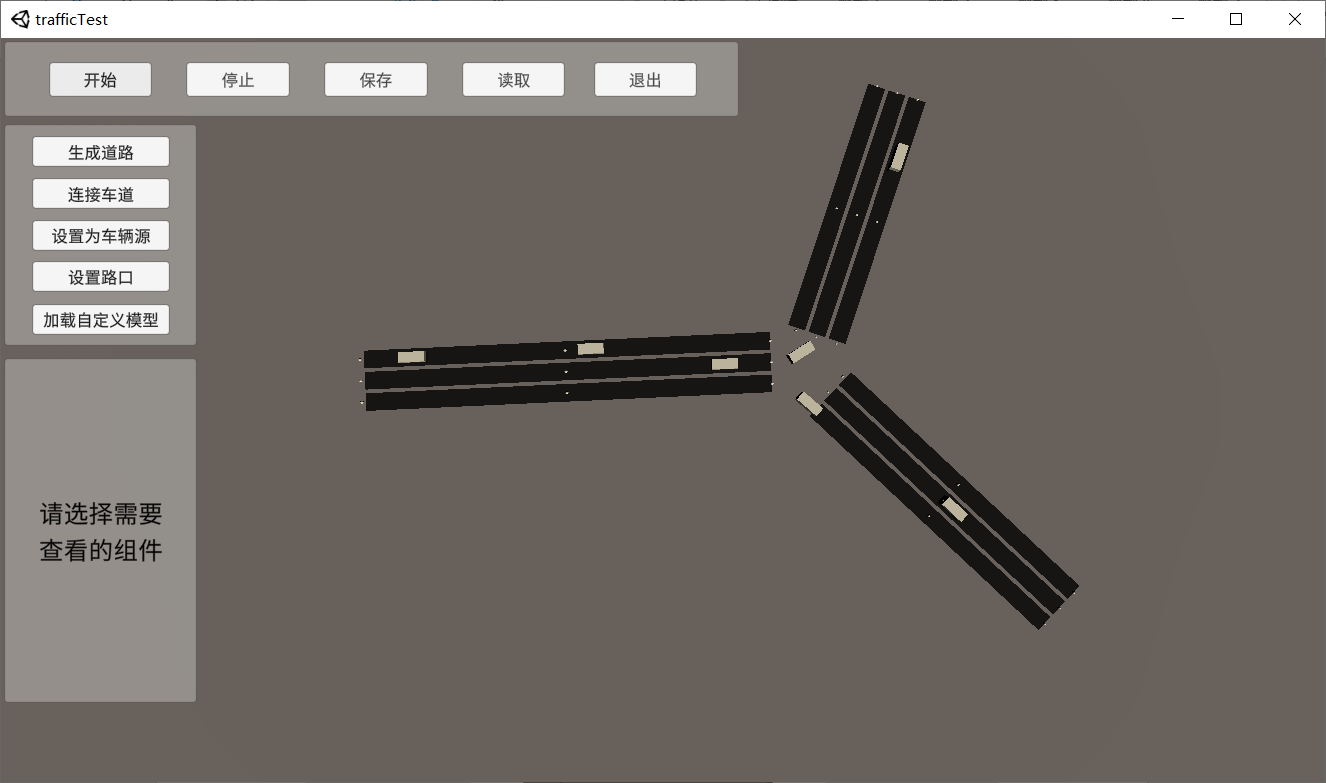
点击左侧菜单栏的“设置为车辆源”按钮将进入设置车辆源的模式。此时用户点击需要设置为车辆源的道路，或者按ESC键取消操作。被点击的道路将会设置为车辆源。需要注意，车辆是从道路的第一个点的一侧产生，并向第三个点（即上文“生成道路”中鼠标点击的第二个点）的一侧行驶。



图片 设置为车辆源的单条道路

### 4.3.4 设置路口

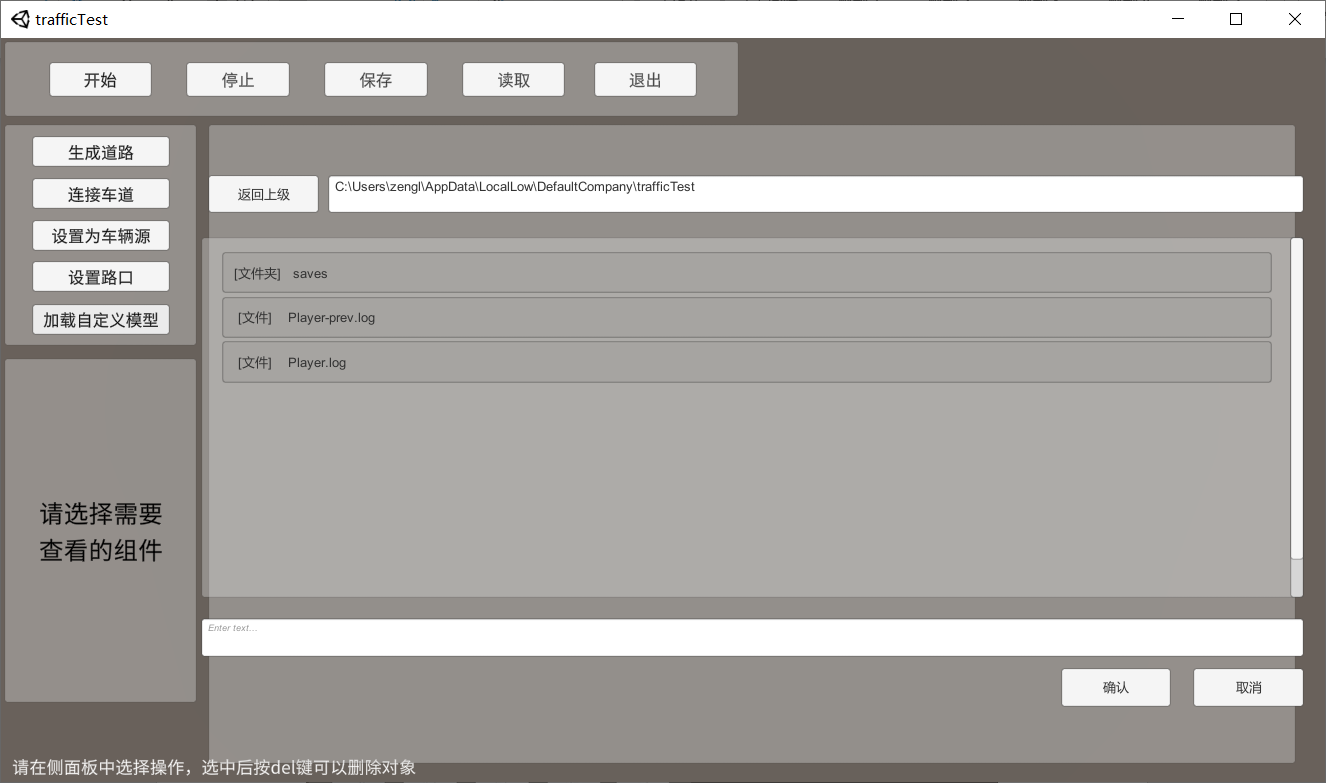
当连接两条或者多条道路时，需要在场景中设置路口组件保证仿真的正常运作。点击左侧菜单栏的“设置路口”按钮后即开始摆放路口，此时用户通过鼠标点击确定路口的位置，点击后路口即被摆放。需要注意：路口组件是全透明的，但是可以被选中删除。



图片 三岔路口的交通仿真

## 4.3.5 加载自定义模型

用户需要加载自定义的交通流模型时，需要点击左侧菜单栏的’“加载自定义模型”按钮来进行加载。点击后将显示一个文件选择对话框。用户在对话框中选择编写好的交通流模型脚本，软件将进行加载并弹出对话框返回结果。脚本编写相关内容请参见4.5 自定义交通流模型编写细则。



图片 选择自定义模型的文件选择对话框

## 4.4 探测器面板

软件提供了探测器面板以供用户对交通流模型进行检验，探测器面板利用反射将交通流模型代码中的变量和数值暴露在探测器中。用户可以用这些数据对自己导入的交通流模型进行检验。



图片 12 对选中车辆的内部数据的展示

## 4.5 自定义交通流模型编写细则

### 4.5.1 OriginCustom类：

用户自定义需要继承的类，推荐继承后类名称为Custom。

通过重写此类中方法实现对模型的修改。

1、float CustomGM(OCar m\_car, float c, float m, float l, OCar previous)

跟驰模型，返回float类型的当前加速度

m\_car 当前车辆

c

m

l

previous 前方最近车辆

2、Vector3[] CustomCP(OCar car, OLine targetLine)

换道路径生成算法，返回三维点数组，由数组中的控制点形成贝塞尔曲线作为换道路径。

car 当前车辆。

targetLine 目标道路

3、float CustomJV(OCar car, OLine line)

换道价值判断算法，返回0-1之间的数值作为是否换道的根据。

car 当前车辆

line 目标道路

4.5.2 OCar类：

车辆类，表示车辆的状态等属性。

1、float MaxVelocityNoRoad 限制的最大速度

2、State state 车辆的状态

3、float accel 车辆当前加速度

4、float maxAccel车辆的最大加速度

5、float velocity 车辆的速度

6、float s 当前路径已行驶长度

7、OLine line 当前所在道路

8、float lineT 贝塞尔曲线的控制值

9、float segment 对路径的分段段数

10、bool lineChange 修改该属性发出换道命令

11、OCar followCar 后方车辆

12、float Km2m() 当前速度以m/s表示形式

13、OCar PreCar()路径上当前车辆

14、OCar NextCar()路径上下一辆车

15、OCar CarClosest(OLine line)路径上最近的车辆

16、bool judgeLocation(OCar pointer, OCar target)

### 4.5.3 OLine类：

车辆行驶的路径

LinkedList<OCar> cars车辆形成的链表

四、ORoad类：

车辆行驶的整条道路