

# FreewayEquipment package 使用手册

## 1.1 更新记录

表 1 版本更新记录表

时间	版本号	创建者	备注
2020 年 7 月 13 日	0.0.5	孙铸	无
2020 年 6 月 16 日	0.0.3	孙铸	无

## 1.2 资源包简介

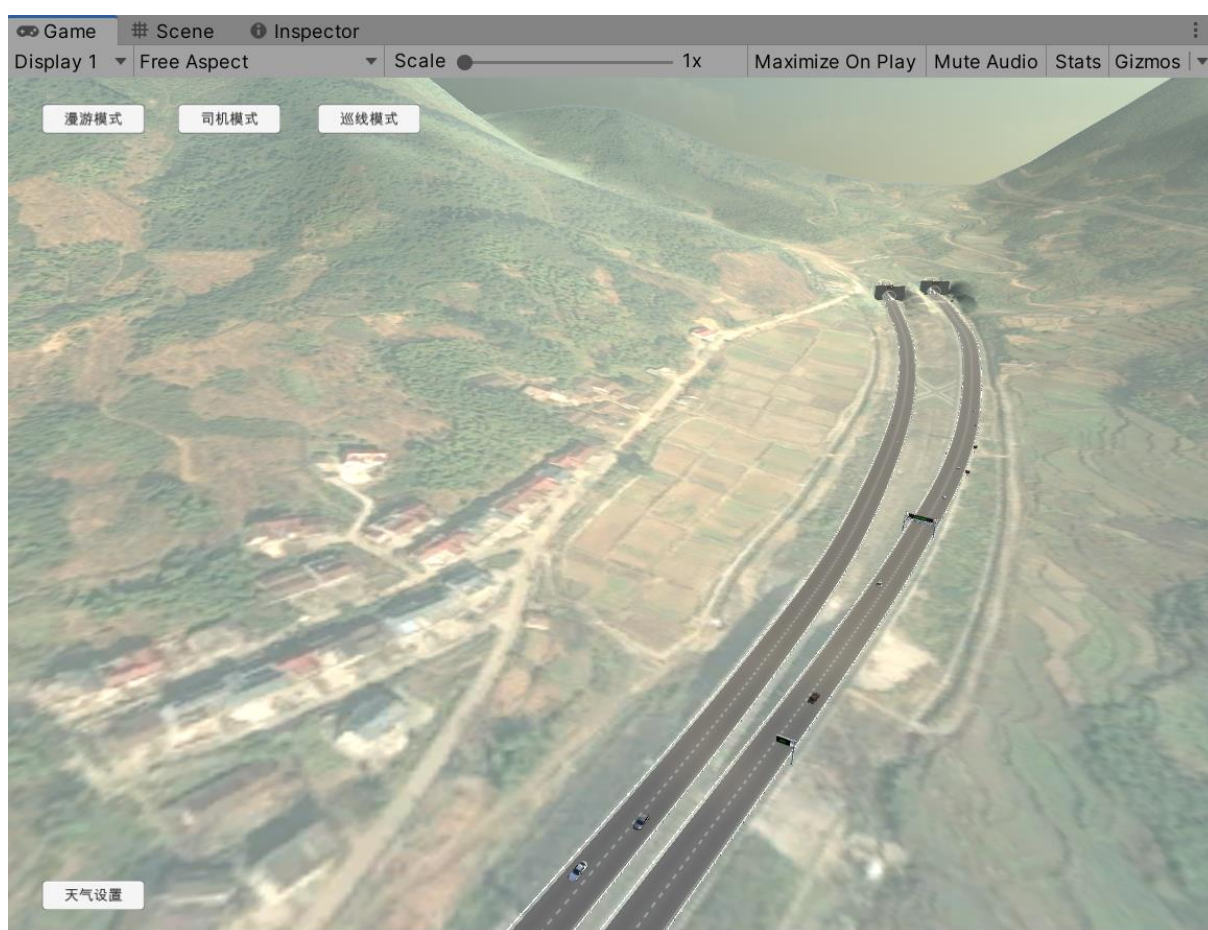


图 1 FreewayEquipment 资源包示例场景

FreewayEquipment 是一个基于 Unity 框架进行高速公路场景交通系统控制仿真开发设计的资源包。

主要特点：

- 资源包包括了交通信号灯、车道指示器、可变限速标志、可变情报板（龙门架型、F 型、T 型、悬挂型）、照明灯具、风机、防火卷帘门、声光报警装置等

多种高速公路常见设备模型；

- 大部分设备已完成功能脚本编写，可通过调用脚本接口实现设备运行控制；
- 实现了基于 TCP 协议的 Socket 通信功能，可通过其他端口或主机控制设备运行状态；
- **【New】** 可通过隧道监控系统数据库导出表，导入设备信息并实现自动布设和自动填写设备控制管理信息。
- **【New】** 以浙江衢南高速达坞隧道为样本，建立模拟场景，实现了隧道监控系统与虚拟仿真系统的联动。
- **【New】** 建立了天气效果控制功能。

## 1.3 使用方法

### 1.3.1 资源包导入

进入 Unity，点击菜单栏 **【Asset】 - 【import Package】 - 【custom package】**，选择资源包导入。

### 1.3.2 示例运行

可通过双击 Project 框中 Asset/Scenes 下的 SampleScene，打开示例场景。场景包含地形对象 **【DaWuTunnelTerrain】**，隧道模型对象 **【DaWuTunnelModel】**，道路对象 **【Road Network】**，交通流对象 **【Traffic】**，设备信息管理器 **【Controller】**，设备对象组 **【Devices】**，绑定设备信息管理器 **【Binders】**，绑定设备对象组 **【BinderDevices】**，Socket 通信服务器端 **【SocketServ】**，数据载入器 **【DataLoader】** 等对象。

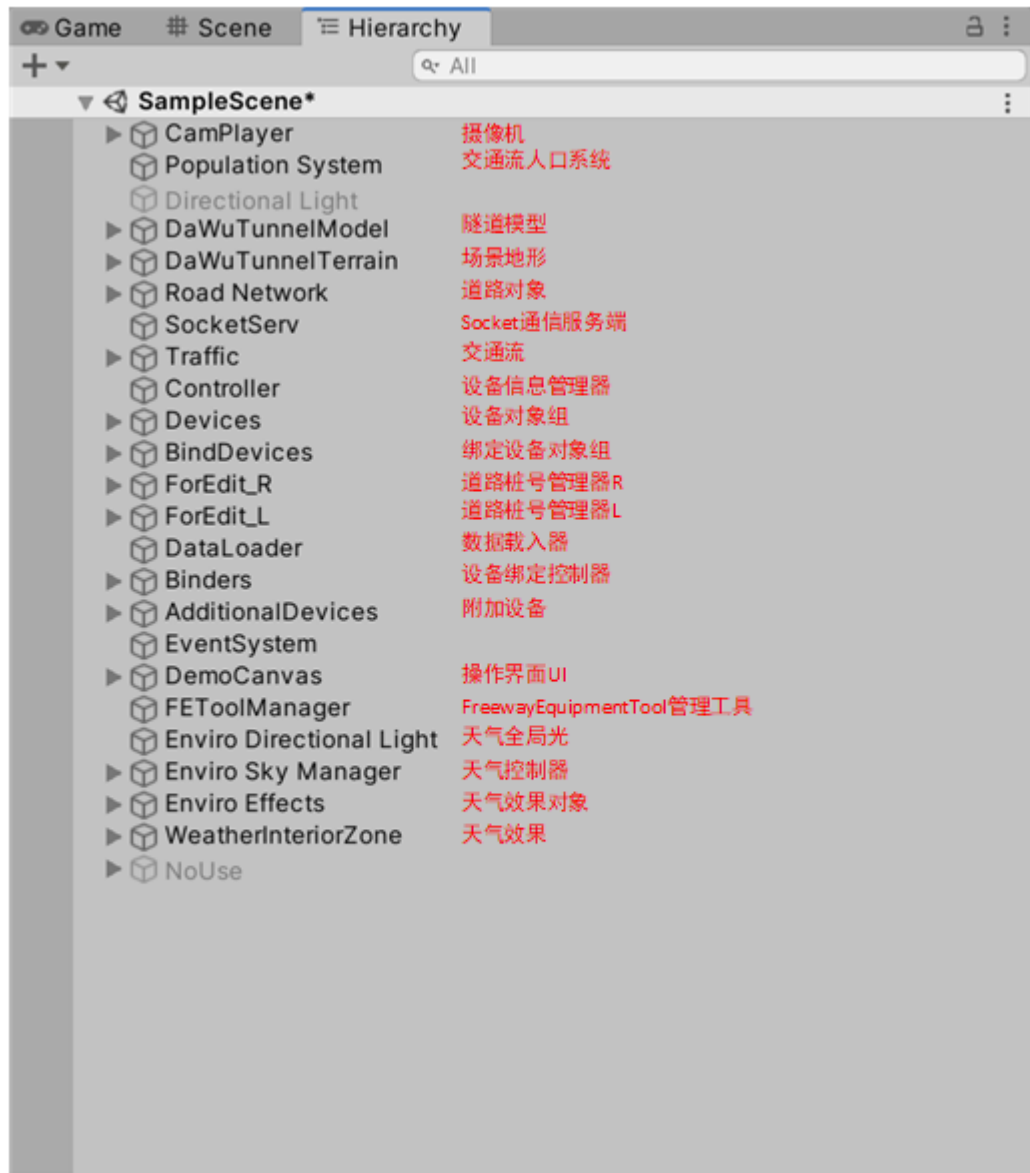


图 2 Demo 场景层级结构

场景中的设备对象在 Hierarchy 框中的 Devices 下。可通过手动调整控制各设备脚本变量实现控制效果，也可以通过 Socket 通信方式实现远程控制。各设备控制码如表 2 所示。

表 2 设备控制码

编号	名称	设备类型	控制码
1	Traffic Light_1	交通信号灯	CtrlCode: 0-绿灯, 1-黄灯, 2-红灯, 3-左转
2	LightAlarm	声光报警	CtrlCode: 0-关闭, 1-启动
3	AutoBarrier	自动栏杆机	CtrlCode: 0-落杆, 1-抬杆
4	F_VMS_1	F 型可变情报板	Display_text: 文本;

编号	名称	设备类型	控制码
			Color_code: red-红色, yellow-黄色, green-绿色, white-白色, #F00F00-十六进制色彩编码（#为十六进制标识, F00F00 为具体编码值, 可变） FontSize: 字号
5	Cam	摄像机	【未开发】
6	Fan	风机	CtrlCode: 0-停止, 1-正转, 2-反转
7	Radio	广播	CtrlCode: 0-停止, n (n>1) 播放音频源列表中的第 n 段音频
8	Fire_hydrant	消防栓	【未开发】
9	G_VMS	龙门架型情报板	Display_text: 文本; Color_code: red-红色, yellow-黄色, green-绿色, white-白色, #F00F00-十六进制色彩编码（#为十六进制标识, F00F00 为具体编码值, 可变） FontSize: 字号
10	H_VMS	悬挂式情报板	Display_text: 文本; Color_code: red-红色, yellow-黄色, green-绿色, white-白色, #F00F00-十六进制色彩编码（#为十六进制标识, F00F00 为具体编码值, 可变） FontSize: 字号
11	Lamp	照明灯具	CtrlCode: 0-关闭, 1-启动
12	SurfaceIndicator	路面指示灯	【未开发】
13	StopLine	停车线	【未开发】
14	T_VMS	T 型情报板	Display_text: 文本; Color_code: red-红色, yellow-黄色, green-绿色, white-白色, #F00F00-十六进制色彩

编号	名称	设备类型	控制码
			编码（#为十六进制标识，F00F00 为具体编码值，可变） FontSize: 字号
15	SpinAlarm	旋转警报	CtrlCode: 0-关闭, 1-启动
16	Rollup Door	防火卷帘门	CtrlCode: 0-关闭, 1-启动
17	LI_new	车道指示器（单面）	CtrlCode: 0-红叉, 1-前向绿箭, 2-左向绿箭
18	LaneIndicators	车道指示器（双面）	FrontCtrlCode: 0-红叉, 1-前向绿箭, 2-左向绿箭 BackCtrlCode: 0-红叉, 1-前向绿箭, 2-左向绿箭

若使用 Socket 通信方式进行设备控制，需要按照协议设定报文结构进行控制。报文按类型分为两种：指令报文和心跳报文。指令报文即用于控制远程服务中设备运行状态的报文，心跳报文即表明客户端存在状态的报文，无控制作用。

表 3 Socket 报文格式

编号	名称	长度（字符）	说明
1	标识符	2	“ET” 为指令报文；“HT” 为心跳报文；
2	时间戳	19	格式如下：yyyy-mm-dd hh:nn:ss
3	设备类型	x	参见：设备类型对应表，如：“” TrafficSignalLamp
4	设备编号	x	如：DW_H_TS
5	指令内容	x	开关状态或者其他内容，根据设备类型变化
6	报文尾	1	“#”（23H）

若报文能够正常发送至服务端并执行，服务端会发送反馈消息，消息内容与发送内容相同，仅报文尾增加 “,ACK” 标识。若发送的指令报文存在设备类型、设备编号错误，则应答报文则会在报文尾 “#” 与 “, ACK” 之间增加错误描述信息：“找不到

设备”，“类型码错误”。

情报板指令内容较特殊，会涉及多屏显示内容轮换、显示时长、文字颜色、文字字体等信息。情报板指令信息以“@”分段，每个分段表示一个轮换分屏，每个分屏信息内，分别以[info][info]， [stayTime][stayTime]， [color] [/color]和[fontFamily] [/fontFamily]标识待显示内容的文本、停留时间、颜色和字体。

表 4 Socket 报文示例

编号	报文	解释
1	HT,2020-06-08 08:08:08,#	心跳包
2	HT,2020-06-08 08:08:08,#,ACK	服务端反馈的心跳包应答
3	ET,2020-06-08 08:08:08,Light,test1,0,	指令 Light 类，编号为 test1 的设备，其控制码变为 0；
4	ET,2020-06-08 08:08:08,Light,test1,0,#,ACK	指令正常反馈；
5	ET,2020-06-08 08:08:08,Light,test1,0,# ,找不到设备,ACK”	指令异常反馈，设备编码错误；
6	ET,2020-06-08 08:08:08,Light,test1,0,# ,类型码错误,ACK”	指令异常反馈，设备类型错误；
7	ET,2020-06-08 08:08:08,VMS,test2,[info] 谨慎驾驶\n 减速慢行 [/info][stayTime]2[/stayTime][color]red[/color][fontFamily]宋体[/fontFamily],#	指令 VMS 类编号为 test2 的设备，显示文字“谨慎驾驶\n 减速慢行” 【\n 表示换行】，显示时长 2 秒，颜色红色，字体宋体。
8	ET,2020-06-08 08:08:08,VMS,test2, [info] 谨慎驾驶\n 减速慢行 [/info][stayTime]2[/stayTime][color]red[/color][fontFamily]宋体 [/fontFamily]@[info]欢迎行驶 [/info][stayTime]2[/stayTime][color]green[/color][fontFamily]宋体[/fontFamily],#	指令 VMS 类编号为 test2 的设备，分两屏显示文字“谨慎驾驶\n 减速慢行”和“欢迎行驶”

## 1.4 场景设计

项目资源包所有设备均已打包为预制体（Prefab）。可通过菜单栏按需要创建功能对象，或从资源包中拖动预制体文件到 Hierachy 中完成设备设施实例化，从而建立一个可与隧道监控系统联动的虚拟场景。

### 1.4.1 布设准备

本资源包示例场景设计中依赖使用了以下资源包：

- EasyRoad3D，道路绘制；
- Enviro-Sky and Weather，天气效果设置；
- SampleScenes 和 StandardAssets，基础脚本/模型参考；
- WorldComposer，真实卫星地图地形生成；
- UTS\_PRO，交通流生成；
- RoadArchitech，道路附属设施模型；
- RockVR，视频推流；
- Logitech SDK，罗技方向盘摇杆 SDK；
- TextMesh Pro，文本编辑强化；

道路设备设施在虚拟场景布设，其位置与其桩号信息有关。为准确定位道路桩号位置，本资源包基于 UTS Pro 中构建的交通流路径点集，通过点集位置转换设备桩号为空间坐标。因此，使用 FreewayEquipmentTools 进行设备自动化布设前，需要提前绘制交通流路径点集【注：为准确布设设备，最好将路径点集位置绘制在道路中线上】

### 1.4.2 创建功能对象管理工具

资源成功载入 Unity 后，会在菜单栏中增加“FETools”菜单。通过点击 FETools—ToolManager—Demo，可创建功能对象管理工具，用于按照步骤设置可以与监控系统交互的场景功能部件对象。功能对象管理工具创建后，会在 Hierachy 中添加一个名为“FEToolManager”的对象，并挂载<CreateTools>脚本。

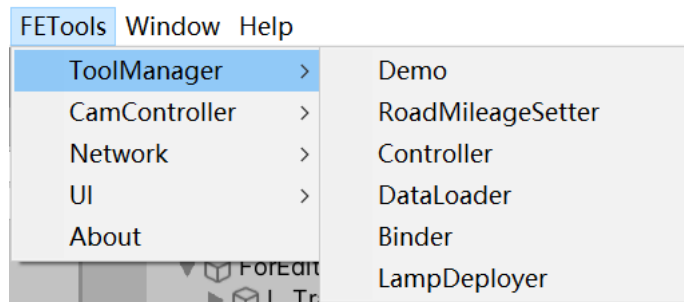


图 3 功能对象管理工具的创建

<CreateTools>脚本可编辑内容如图 4 所示。填入需要引用的参数，脚本可按照步骤生成以下功能对象：

- **桩号生成管理器：**用于设置路径桩号参数，根据给定路径，路径参考点，参考点参考桩号，可计算整个路径所有点的桩号。所有设备设施依桩号布设的功能对象，均依赖此功能对象。桩号生成管理器创建后，会在 Hierachy 中添加名为“1-RoadUpwardMileage”和“1-RoadDownwardMileage”的功能对象，分别管理道路上行方向路径桩号、设备设施布设，和道路下行方向的桩号、设备设施布设。
- **Socket 通信服务端：**用于生成与局域网其他主机交互的 Socket 通信服务端，与其他系统交互时（例：示例场景与隧道监控系统交互）依赖此功能对象。Socket 通信服务端生成后，会在 Hierachy 中添加一个名为“2-SocketServer”的功能对象。
- **设备事件管理器：**用于管理生成生成设备与生成设备的控制信息（其他系统可通过设备类型，设备编号匹配设备进行控制）。创建后会在 Hierachy 中分别添加名为“3-Devices”和“3-Controller”的功能对象，其中，“3-Devices”主要用于管理生成的设备设施对象实例（层次结构方便查看）；“3-Controller”主要用于管理生成设备设施的控制信息，如各个设备的名称、设备类型、设备编号以及对应对象实例。
- **数据读取器：**可通过数据库导出数据文件（目前仅支持\*.xlsx 格式）读入设备信息。设备设施的自动布设功能依赖此功能对象提供的设备信息。数据读取器创建后会在 Hierachy 中添加一个名为“4-Dataloader”的功能对象。
- **自定义摄像机：**为方便场景设计完成后的浏览功能所定制的摄像机。摄像机对象提供了三种摄像机浏览模式：漫游模式，驾驶模式，巡线模式。创建后在



Hierarchy 中添加一个名为“5-CamPlayer”的功能对象。

- **示例界面 UI：**即配合摄像机浏览模式选择的示例界面。创建后会在 Hierarchy 中添加一个名为“6-DemoCanvas”的功能对象。



图 4 CreateTools 脚本可编辑内容

为一定程度减少功能对象生成后的配置工作量，使用<CreateTools>脚本创建功能对象前，尽量先配置好脚本预设的参考引用变量（Roadtraffic，RoadUpward，RoadDownward），变量配置说明如下：

- **Roadtraffic：**用于创建自定义摄像机时，其驾驶模式需要引用一个交通流对象以提供车辆行驶视角。
- **RoadUpward，RoadDownward：**用于创建桩号生成管理器。路径参数可选择引用交通流功能对象，也可以使用自定义路径对象，但路径对象结构需保持 Path-Points-point 的三层结构。RoadUpward 为上行方向路径，RoadDownward 为下行方向路径（上行方向为桩号增大的方向，下行方向为桩号减小的方向，设置时需根据项目实际道路桩号增减方向设置）。

- **数据文件地址：**用于读取设备设施布设信息。

### 1.4.3 桩号生成管理器

桩号生成管理器可由功能对象管理工具（FEToolManager）的 STEP1 创建，也可以通过菜单栏 FETools—ToolManager—RoadMileageSetter 创建。桩号生成管理器包含两个脚本<RoadMileageSetting>和<SingleDeviceDeploy>。其中<RoadMileageSetting>即路径桩号生成功能的实现脚本，<SingleDeviceDeploy>即在生成路径桩号的基础上，按照桩号自动布设单个设备的功能脚本。若要实现批量设备自动布设，此功能对象中的<RoadMileageSetting>和<SingleDeviceDeploy>脚本都是必需的。

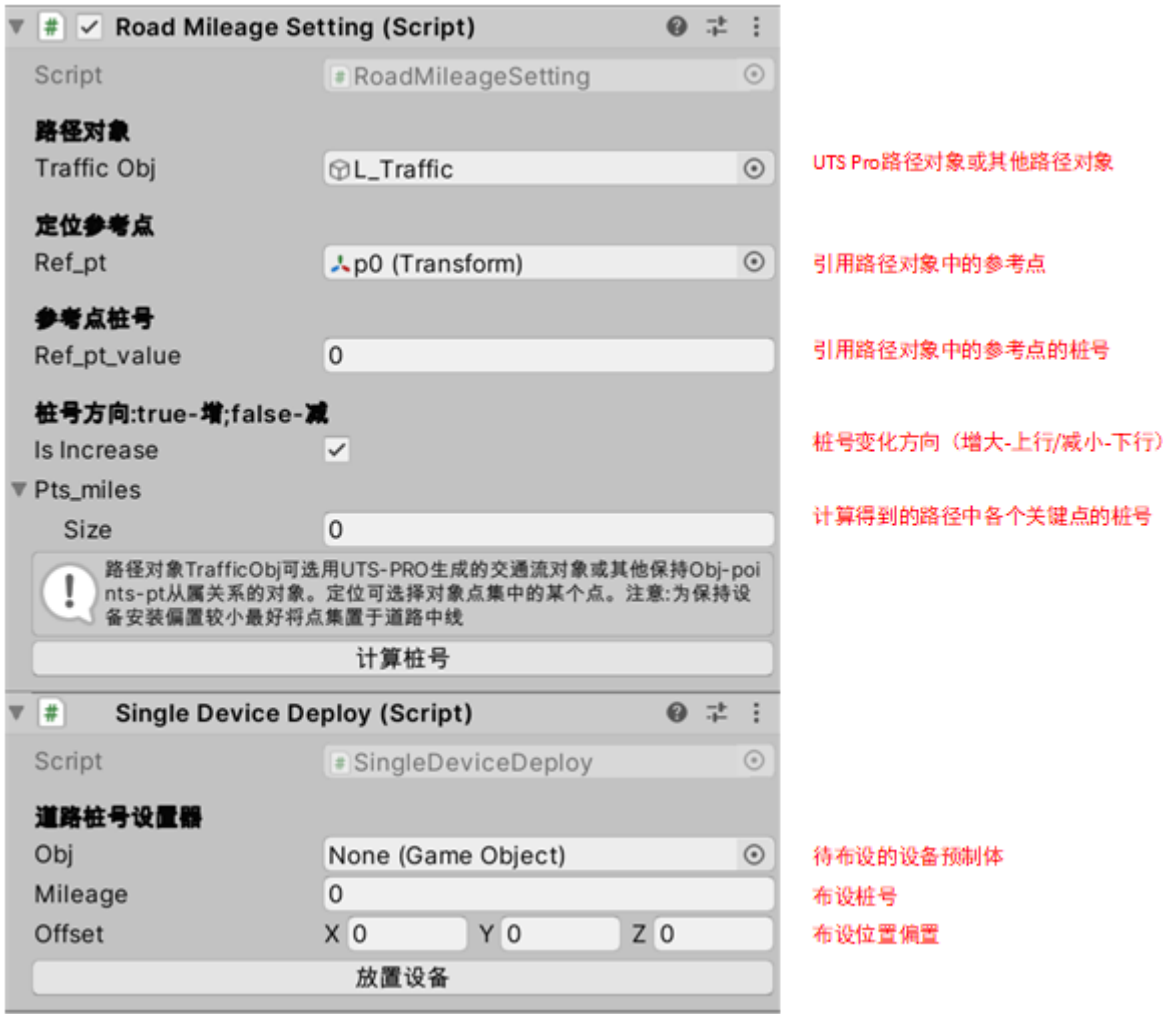


图 5 桩号生成管理器可编辑内容

图 5 展示了桩号生成管理器所包含两个脚本的可编辑内容。可编辑内容的配置说明如下：

- **RoadMileageSetting**

通过功能对象管理工具（FEToolManager）中 CreateTools 脚本创建的桩号生成管

理器中，TrafficObj, Ref\_pt, Ref\_pt\_value, IsIncrease 等变量均已设置默认值。通过菜单栏 FETools—ToolManager—RoadMileageSetter 创建的桩号生成管理器，TrafficObj, IsIncrease 等变量仍需要配置。

**TrafficObj:** 可使用 UTS Pro 资源包绘制的交通流路径或其他符合“Path-Points-Point”三层结构的路径对象（如图 6 所示路径对象点集示例）。注：为了便于设备设施自动布设，路径对象最好沿着道路路面的中心线。

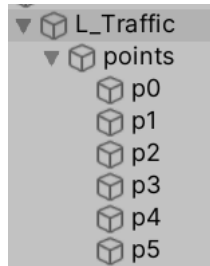


图 6 路径对象点集三层结构示例

**Ref\_pt:** 定位参考点，可选择路径对象点集内具有标志性的位置作为参考点（即可匹配桩号的位置点），如隧道洞口、道路分岔口等（可通过双击点对象在场景 Scene 中查看），方便根据参考点设置桩号。

**Ref\_pt\_value:** 参考点桩号数值，单位以米计量，例如桩号“K1531+231”或“K1531.231”需要设置桩号数值为 1531231。

**IsIncrease:** 桩号增/减（上行/下行）方向设置。IsIncrease 设置勾选时，沿行车方向桩号逐渐增大；IsIncrease 不勾选时，沿行车方向桩号逐渐减小。

**Pts\_miles:** 根据设置变量生成的点集桩号对应表。以上变量内容均设置完成后，点击“计算桩号”，计算结果即存放在 Pts\_miles 中（如图 7）。

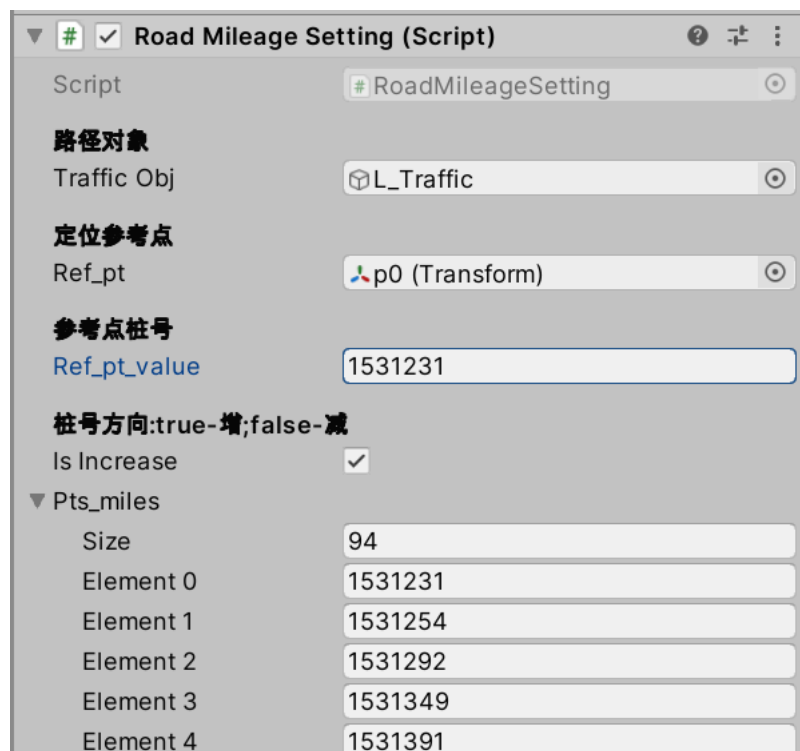


图 7 路径对象点集生成桩号列表

#### ● SingleDeviceDeploy

SingleDeviceDeploy 脚本可用于将选定预制体设备实例化在指定桩号位置。

**Obj:** 选择资源包内的预制体设备（资源路径：

Assets/FreewayEquipment/Prefabs/...，预制体文件后缀：.prefab）；

**Mileage:** 设备设施布设桩号数值，单位以米计量。

**Offset:** 设备设施布设偏置，由设定 Mileage 数值计算得到的点在选择引用的路径对象上，对于路侧或者路面上方的设置，可通过 Offset 调整最终设备设施布设的位置，偏置位置由三维向量表示，其中，X 表示前向偏置，以行车方向为正；Y 表示高度偏置，以上为正；Z 表示左右偏置，以右为正。



待实例化预制体设备

布设桩号

布设位置偏置，x: 前向偏置距离（行车方向为正）；  
y: 高度偏置（向上为正）；z: 左右偏置（向右为正）；

图 8 SingleDeviceDeploy 脚本可编辑内容

#### 1.4.4 Socket 通信服务端



图 9 SocketServ 脚本接口变量

Socket 通信服务端 SocketServer 功能对象挂载了<ServSocket>脚本，其中，ServSocket 默认 ip 为本机环回地址，端口 10010。如果没有特殊需要，一般可以不用修改。若在编辑中修改了地址和端口号，需要向有联动需求的应用提供此配置信息。一个模拟场景中，仅需配置一个 Socket 通信服务端。

#### 1.4.5 设备事件管理器

本资源包将设备管理和事件管理分为两个功能对象，即 Devices 和 Controller。其中，Devices 的主要功能即将实例化的设备作为子对象存放，方便管理。Controller 的主要功能即管理设备信息，并通过 Socket 通信匹配待控制设备，触发设备动作。Controller 中挂载了<EventTrigger>脚本（注：此 EventTrigger 脚本为命名空间 sz.network 中的事件触发器，而非 UnityEngine 中的事件触发器）。功能脚本变量内容如图 10 所示。

**Socket\_Serv:** 用于事件触发控制的 Socket 通信服务端对象。

**GameEvent:** 用于事件触发控制的设备信息。脚本事件触发功能执行时，通过设备类型、设备编号匹配待控制设备的实例化对象。此信息可通过 DataLoader 功能对象从数据库导出数据读入，亦可自定义设备信息。自定义设备信息时，通过修改 GameEvent 变量的 size 增加设备信息记录（增加一条记录时，设置 size 为原有 size 数值加 1）。填写设备信息时，设备 Title 可随意填写，其内容与设备匹配无关，但最好与 Devices 中实例化设备的名称一致，方便管理。Devicetype（设备类型）与 id（设备编号）应与联动监控系统中对应该待控制设备的类型与编号一致，Controller 应设置场景中实例化的设备对象。

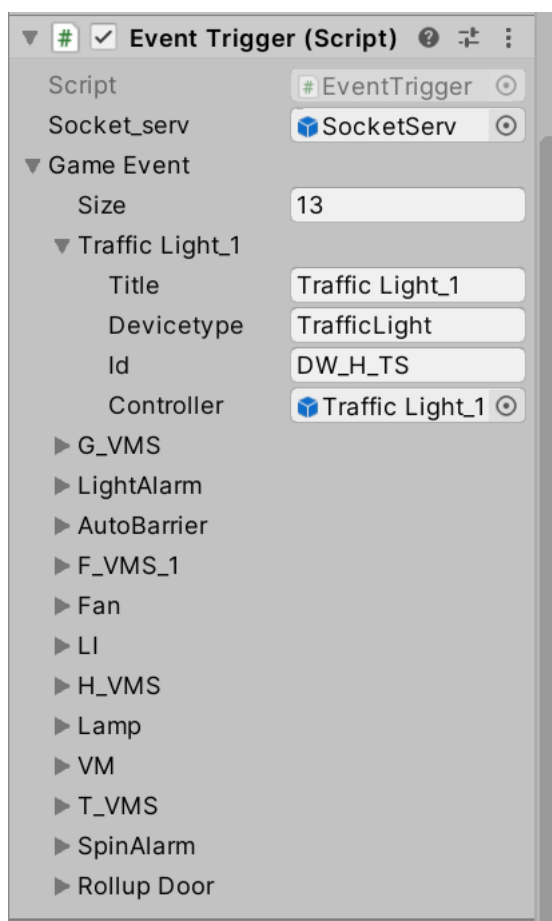


图 10 Controller 预制体脚本变量接口

#### 1.4.6 数据载入器

数据载入器挂载了<DeviceInfoLoaderXLS>和<MultiDeviceDeploy>两个功能脚本。其中，<DeviceInfoLoaderXLS>脚本主要用于从\*.xls 或\*.xlsx 文件中读取设备信息。<MultiDeviceDeploy>脚本主要用于根据<DeviceInfoLoaderXLS>读入的设备信息，实例化设备预制体并布设在设备信息中保存的位置，同时将设备信息添加到 EventTrigger 中。

- DeviceInfoLoaderXLS

**Filepath:** 数据库导出文件的路径。填入 Filepath 后点击“读入数据”，如果文件读取正常，设备信息会读入并保存至 DeviceInfoList 中。

- MultiDeviceDeploy

**IsEditable:** 是否允许将设备信息填入添加到 Controller 功能对象的 EventTrigger 中。默认为可以添加，可根据实际需要配置，有些不需要控制的设备可不用添加到 EventTrigger 中。

**Area1, Area2, Area3:** 设备布设区域, Area1 为道路上行方向, Area2 为道路下行方向, Area3 可根据需要设置, 默认与 Area1 的区域相同, 自动布设后可根据需要手动调整设备设施的位置。变量引用对象应为创建的桩号生成管理器 (即挂载有 <RoadMileageSetting> 和 <SingleDeviceDeploy> 脚本的功能对象)。

**EventTrigger:** 事件管理器 (Controller, 即挂载有 <sz.network.EventTrigger> 脚本的功能脚本)。

**ObjParent:** 设备管理器 (Devices)。实例化后的设备会从属于设置对象, 此用法主要方便设备管理。

各类设备预制体: 各类待实例化设备类型所使用的预制体设备模型。

各类设备安装偏置: 各类待实例化设备类型在实例化时的布设偏置。

以上配置完成后, 点击“设备部署”, 实例化的设备会添加到 ObjParent 设置的功能对象下; 实例化的设备信息存在 EventTrigger 中的 GameEvent 中。

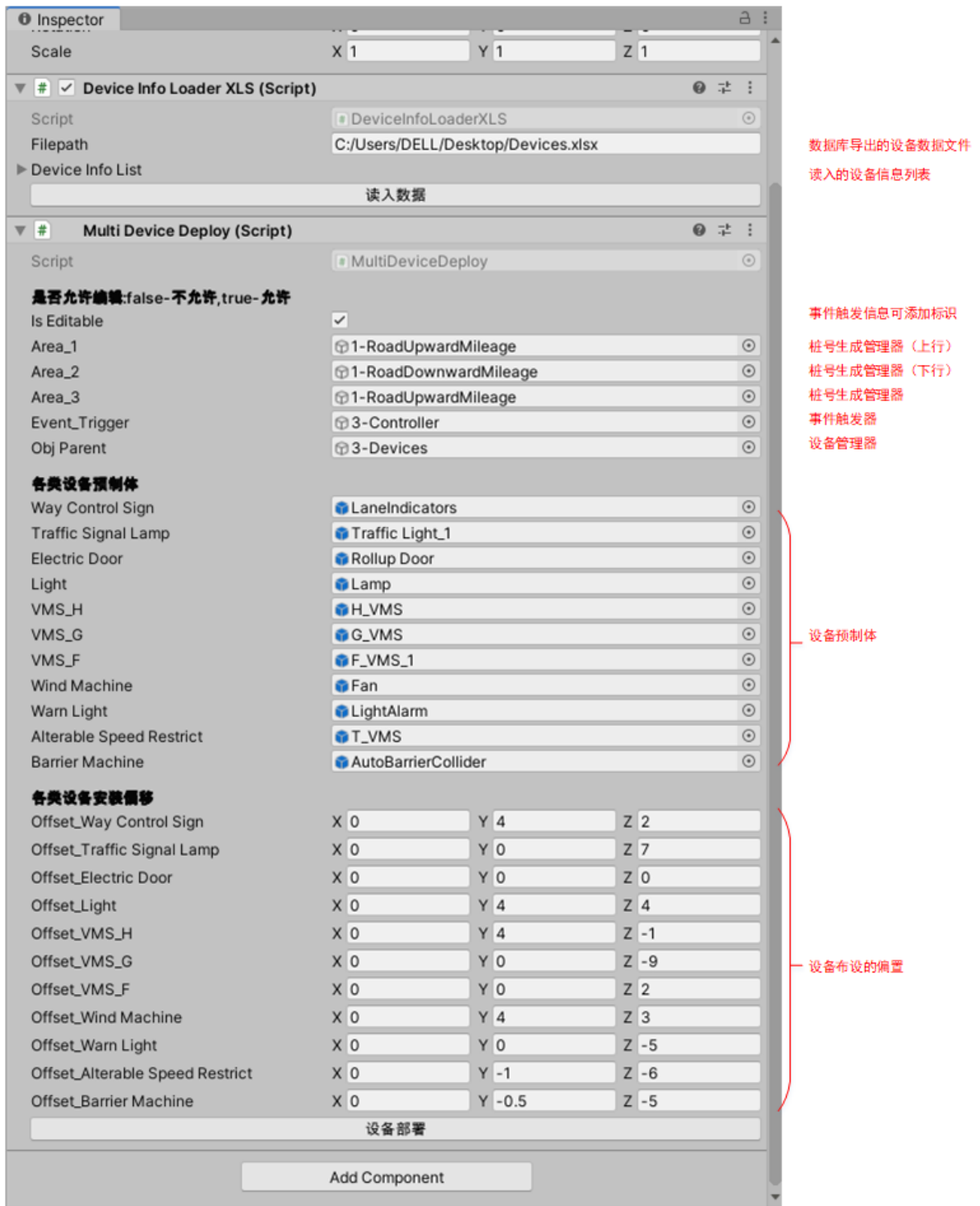


图 11 数据读入器可编辑内容

图 12 为数据库导出的数据样表。数据表包含多个属性，本资源包主要关注属性 ID（设备编号）、SPECIFICATION\_ID（设备类型）、POSITION（桩号）、AREA（安布设方向）、NAME（名称）。数据读入后，程序会根据 SPECIFICATION\_ID 匹配设备预制体进行实例化，因此，数据读入前，首先需要查看所有记录的 SPECIFICATION\_ID 是否满足需要（即查看 SPECIFICATION\_ID 是否都能够对应<MultiDeviceDeploy>中枚



举的设备预制体名称，若没有则需要修改编码，例如导出数据的情报板

SPECIFICATION\_ID 均为 VMS，为了进一步区分情报板类型，导入数据前将

SPECIFICATION\_ID 根据标注名称细分为悬挂式 VMS\_H、龙门架式 VMS\_G、F 式

VMS\_F）。

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	ID	CODE	SPECIFICATION ID	ROAD ID	POSITION	AREA	REFERENCE DISTANCE	AT OUT DOOR	MEMO	NAME	IN USE	ORGAN ID	CONSTRUCTION CODE	ALTITUDE	
1	DW_N.LS35	DW_N.LS35	WayControlSign	101	1533.563	1	200	1	1	主车道车指26	1	1	HQNL.DWSD	NULL	
2	DW_N.LS36	DW_N.LS36	WayControlSign	101	1533.563	1	100	1	1	超车道车指26	1	1	HQNL.DWSD	NULL	
3	DW_N.LS27	DW_N.LS27	WayControlSign	101	1533.893	1	200	1	1	主车道车指27	1	1	HQNL.DWSD	NULL	
4	DW_N.LS28	DW_N.LS28	WayControlSign	101	1533.893	1	100	1	1	超车道车指28	1	1	HQNL.DWSD	NULL	
5	DW_N.LS29	DW_N.LS29	WayControlSign	101	1534.223	1	200	1	1	主车道车指29	1	1	HQNL.DWSD	NULL	
6	DW_N.LS30	DW_N.LS30	WayControlSign	101	1534.223	1	100	1	1	超车道车指30	1	1	HQNL.DWSD	NULL	
7	DW_N.LS31	DW_N.LS31	WayControlSign	101	1534.553	1	200	1	1	主车道车指31	1	1	HQNL.DWSD	NULL	
8	DW_N.LS32	DW_N.LS32	WayControlSign	101	1534.553	1	100	1	1	超车道车指32	1	1	HQNL.DWSD	NULL	
9	DW_N.LS33	DW_N.LS33	WayControlSign	101	1534.883	1	200	1	1	主车道车指33	1	1	HQNL.DWSD	NULL	
10	DW_N.LS34	DW_N.LS34	WayControlSign	101	1534.883	1	100	1	1	超车道车指34	1	1	HQNL.DWSD	NULL	
11	DW_N.LS35	DW_N.LS35	WayControlSign	101	1535.213	1	200	1	1	主车道车指35	1	1	HQNL.DWSD	NULL	
12	DW_N.LS36	DW_N.LS36	WayControlSign	101	1535.213	1	100	1	1	超车道车指36	1	1	HQNL.DWSD	NULL	
13	DW_N.LS37	DW_N.LS37	WayControlSign	101	1535.543	1	200	1	1	主车道车指37	1	1	HQNL.DWSD	NULL	
14	DW_N.LS38	DW_N.LS38	WayControlSign	101	1535.543	1	100	1	1	超车道车指38	1	1	HQNL.DWSD	NULL	
15	DW_N.LS39	DW_N.LS39	WayControlSign	101	1535.873	1	200	1	1	主车道车指39	1	1	HQNL.DWSD	NULL	
16	DW_N.LS40	DW_N.LS40	WayControlSign	101	1535.873	1	100	1	1	超车道车指40	1	1	HQNL.DWSD	NULL	
17	DW_N.LS41	DW_N.LS41	WayControlSign	101	1536.203	1	200	1	1	主车道车指41	1	1	HQNL.DWSD	NULL	
18	DW_N.LS42	DW_N.LS42	WayControlSign	101	1536.203	1	100	1	1	超车道车指42	1	1	HQNL.DWSD	NULL	
19	DW_N.LS43	DW_N.LS43	WayControlSign	101	1536.533	1	200	1	1	主车道车指43	1	1	HQNL.DWSD	NULL	
20	DW_N.LS44	DW_N.LS44	WayControlSign	101	1536.533	1	100	1	1	超车道车指44	1	1	HQNL.DWSD	NULL	
21	DW_N.LS45	DW_N.LS45	WayControlSign	101	1536.863	1	200	1	1	主车道车指45	1	1	HQNL.DWSD	NULL	
22	DW_N.LS46	DW_N.LS46	WayControlSign	101	1536.863	1	100	1	1	超车道车指46	1	1	HQNL.DWSD	NULL	
23	DW_N.LS47	DW_N.LS47	WayControlSign	101	1537.142	1	200	1	1	主车道车指47	1	1	HQNL.DWSD	NULL	
24	DW_N.LS48	DW_N.LS48	WayControlSign	101	1537.142	1	100	1	1	超车道车指48	1	1	HQNL.DWSD	NULL	
25	DW_H.LS01	DW_H.LS01	WayControlSign	101	1533.563	2	200	1	1	主车道车指01	1	1	HQNL.DWSD	NULL	
26	DW_H.LS02	DW_H.LS02	WayControlSign	101	1533.563	2	100	1	1	超车道车指02	1	1	HQNL.DWSD	NULL	
27	DW_H.LS03	DW_H.LS03	WayControlSign	101	1533.893	2	200	1	1	主车道车指03	1	1	HQNL.DWSD	NULL	
28	DW_H.LS04	DW_H.LS04	WayControlSign	101	1533.893	2	100	1	1	超车道车指04	1	1	HQNL.DWSD	NULL	
29	DW_H.LS05	DW_H.LS05	WayControlSign	101	1534.223	2	200	1	1	主车道车指05	1	1	HQNL.DWSD	NULL	
30	DW_H.LS06	DW_H.LS06	WayControlSign	101	1534.223	2	100	1	1	超车道车指06	1	1	HQNL.DWSD	NULL	
31	DW_H.LS07	DW_H.LS07	WayControlSign	101	1534.553	2	200	1	1	主车道车指07	1	1	HQNL.DWSD	NULL	

图 12 数据库导出表样例

### 1.4.7 自定义摄像机

为了构建场景不同展示模式效果，本资源包根据应用需要自定义了摄像机对象。摄像机对象可进行漫游模式、驾驶模式、巡线模式进行场景的展示，同时可以对场景的天气状态效果进行处理展示。自定义摄像机可通过功能对象管理工具

（FEToolsManager）创建，也可以通过菜单栏 FETools—CamController—CamPlayer 创建。

功能对象 Camplayer 的漫游模式，驾驶模式，巡线模式等展示模式分别使用了<MouseLookAdvance><DriveMode>和<MoveAlongRoad>功能脚本。脚本可编辑内容见图 13。



图 13 CamPlayer 功能对象可配置内容

- DriverMode

Path: 用于驾驶模式选择车辆对象。其值可引用 UTS Pro 创建的交通流对象。

- MoveAlongRoad

Roads: 道路列表。其内容可填写待展示路径的桩号生成管理器，其路径的顺序应与交互界面对象（DemoCanvas）中的 Direction 下拉框中的内容对应。如：下拉框为位置 0 为上行路径，则 Roads 变量位置 0 则填写上行路径对应的桩号生成管理器。

Moveable: 摄像机可移动标识，勾选此值时，摄像机可在巡线模式下根据交互界面的滑动条取值变化移动。

Offset: 摄像机位置偏置。

#### 1.4.8 摄像机展示模式交互界面

摄像机展示模式控制界面如图 14 所示，图中展示了摄像机的三种展示模式。从左至右分别为漫游模式、驾驶模式、巡线模式的展示效果。其中，漫游模式可以上帝视角漫游查看场景的任意对象；驾驶模式可跟随指定路径上的任意车辆，以司机视移动查看场景；巡线模式可通过拖动滑动条查看道路任意位置场景。示例界面可通过功能对象管理工具创建，也可以通过菜单栏 FTTools—UI—DemoUI 创建。



图 14 摄像机展示模式效果

示例场景摄像机控制模式界面如图 15 所示，大多数变量为 UI 功能对象，已提前配好，可直接使用。需要根据项目配置的变量为：

- MainCam：资源包自定义摄像机对象；
- WeatherSetUI：有天气控制效果时，此值需要引用天气控制界面。



图 15 示例界面 DemoCanvas 可编辑内容

1.4.9 Binder 绑定控制器

为了减小场景设计中设备布设的工作量，可将多个设备进行绑定控制，此方式需要用到 Binder 绑定控制器。Binder 控制器可通过菜单栏 FETools—ToolManager—

Binder 创建。每个主从控制关系设备群组均需一个 Binder 绑定控制器。如隧道照明系统以回路为单位进行控制，因此每个照明回路均需要一个 Binder 绑定控制器。



图 16 Binder 控制器可编辑变量

绑定控制需要通过<BinderControl>脚本实现，需要设置 Master 和 Slavers。其中，Master 为主控设备，Slavers 为从控设备，即对主控设备进行控制，与主控设备绑定的从控设备会根据主控设备控制信息变化。从控设备可根据需要设置一个或多个。【注：并非所有设备类型均支持绑定控制，目前仅支持自动栏杆机（BarrierMachine），照明灯具（Light），声光报警器（WarnLight）和防火卷帘门（ElectricDoor），即主控/从控设备都必须为上述设备类型】。

1.4.10 LampDeploy 灯具布设器

隧道内照明控制按照是以回路为单位进行控制的，因此单个照明灯具的布设需要按照回路设置布设。灯具布设器可通过图 18 为 LampDeploy 脚本可编辑内容。

Road: 可引用桩号生成器功能对象，即挂载有<RoadMileageSetting>和<SingleDeviceDeploy>脚本的功能对象。

Lamp: 待实例化的灯具预制体。

Offset: 灯具布设偏置。隧道照明灯具布设时会安装在左右两侧衬砌墙面上，此处可仅设置灯具在一侧安装位置的偏移（具体按照哪一侧与灯具模型的朝向有关，安装位置应能保持灯具在不旋转的前提下将光照投射到隧道路面上）。另一侧安装偏置可通过程序计算得到，即照明回路生成时，会同时生成左右两侧的灯具，其中一侧灯具回路为 XX\_A 回路，另一侧灯具回路为 XX\_B 回路。

Start, End: 隧道的起点桩号和终点桩号。此信息会用于不同照明区域灯具布设的位置计算。

Parent: 生成灯具实例化对象的存放父节点对象，方便管理。

Group\_list: 照明区域设置。不同照明区域对灯具布设的长度、间隔设置不同，灯具布设前需要提前按照需要设置照明区域。照明区域设置时，按照从入口到出口

的顺序，依次将照明区域信息录入 Group\_list。Group\_list 记录中的变量：**Name:** 照明区域名称，如：入口基本段、入口过渡段、基本段、出口过渡段、出口基本段。**Class\_code:** 照明区域类型编码，此类型主要用于灯具布设的位置计算。灯具布设时，入口段的灯具以隧道起点为基点按照区域顺序依次计算，出口段的灯具以隧道终点为基点按照照明区域的反向顺序逆序依次计算，入口段和出口段计算完成剩余的区域为基本段。**Length:** 照明区域的长度，单位 m。**space:** 灯具的布设间隔，单位 m。

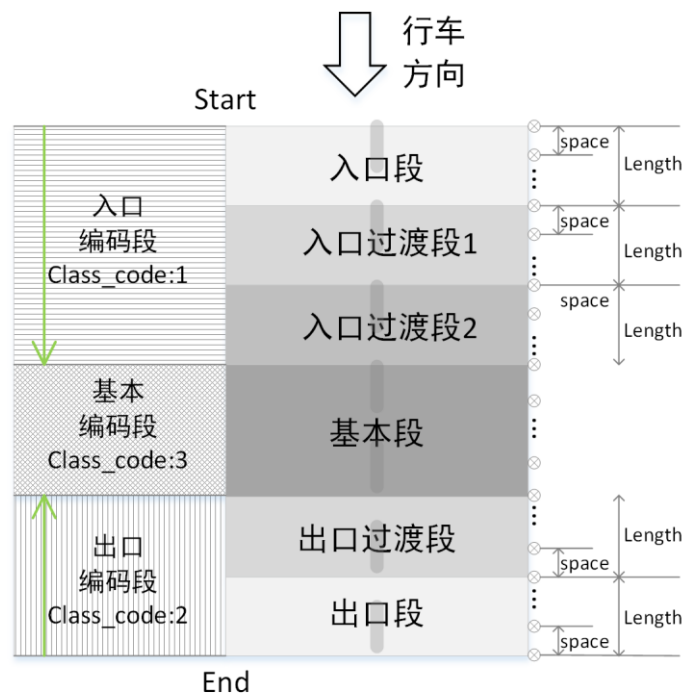


图 17 照明区域设置参数示意图



图 18 LampDeploy 脚本可编辑变量

全部设置完成后，点击“生成照明回路”，脚本会按照设置自动实例化照明灯具并布设至隧道场景内。【注：脚本功能有限，仅能实现简单的灯具布设逻辑，即照明区域不交叉的布设逻辑，实际工程中，可能大部分并不按照此逻辑布设，待完善】

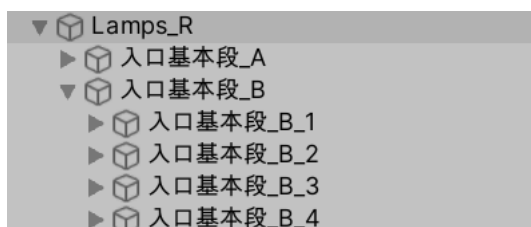


图 19 灯具布设器生成灯具层级

