POLY-VERIFICATION USER GUIDE

Welcome to PolyVerif

This user guide offers a comprehensive step-by-step tutorial for developing a customized Scenic script to execute scenarios within the OSSDC simulator.

- **Set the map parameters and declare the model to be used in the scenario:**

```
param map = localPath('maps/BorregasAve.xodr')
param lgsvl_map ='BorregasAve'
```

- **Define the simulator model by choosing between two available options: the LGVL simulator model or the driving domain model in Scenic.**

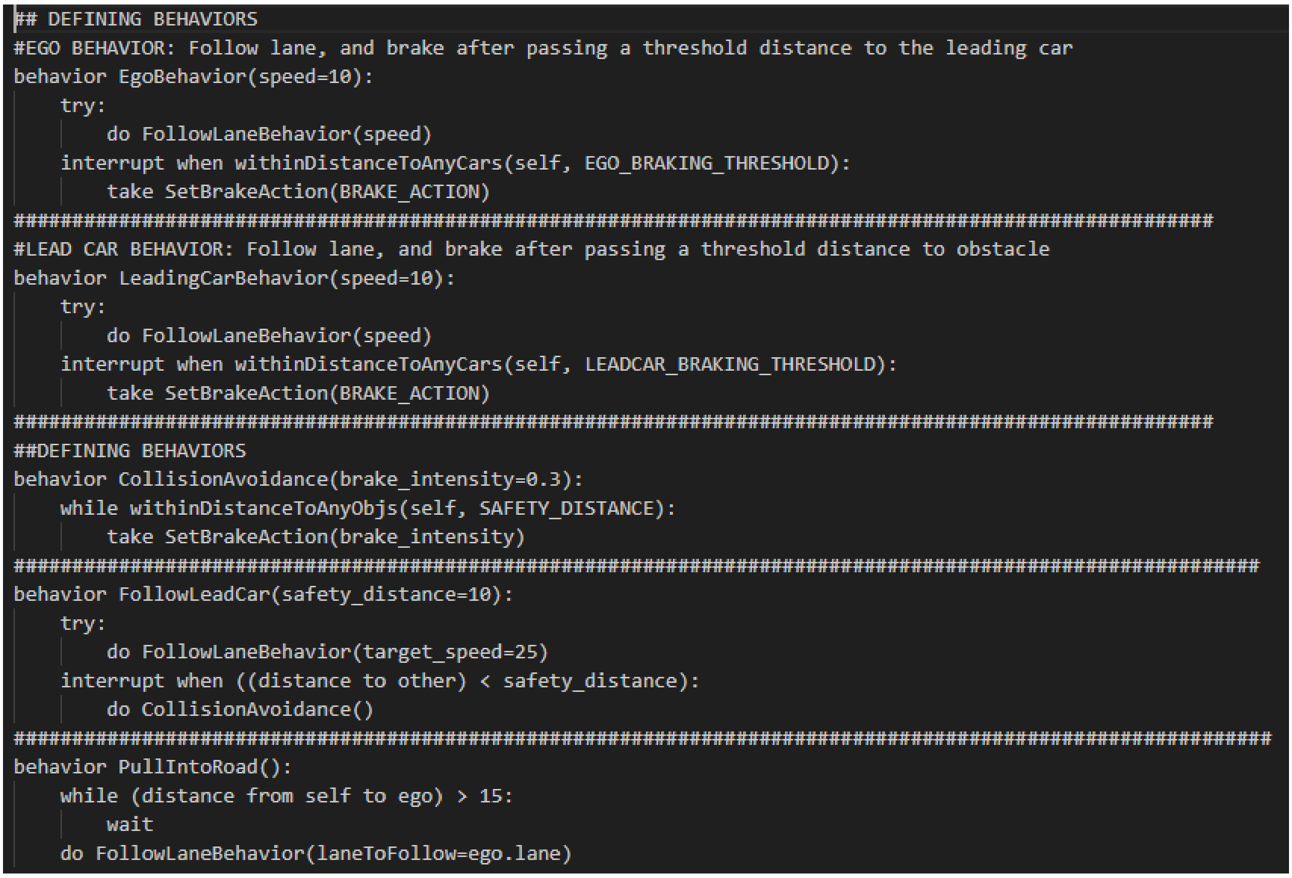
```
1. Lgsvl model - model scenic.simulators.lgsvl.model
2. Driving Domain - model scenic.domains.driving.model
```

- **Define the constants to be used in the scenario. (If any)**

```
#CONSTANTS
Ego_Speed = 10
Lead_Car_Speed = 10
Brake_Action = 1.0
Throttle_Action = 0.6
```

- **Define the scenario behaviours**

Define behaviours for the ego vehicle, such as (FollowLaneBehaviour, AvoidingCollisionBehaviour, etc.), choosing from the available list of behaviours shown below. Use any behaviour that suits your requirements.



```
## DEFINING BEHAVIORS
#EGO BEHAVIOR: Follow lane, and brake after passing a threshold distance to the leading car
behavior EgoBehavior(speed=10):
    try:
        do FollowLaneBehavior(speed)
    interrupt when withinDistanceToAnyCars(self, EGO_BRAKING_THRESHOLD):
        take SetBrakeAction(BRAKE_ACTION)
#######################################################################################################
#LEAD CAR BEHAVIOR: Follow lane, and brake after passing a threshold distance to obstacle
behavior LeadingCarBehavior(speed=10):
    try:
        do FollowLaneBehavior(speed)
    interrupt when withinDistanceToAnyCars(self, LEADCAR_BRAKING_THRESHOLD):
        take SetBrakeAction(BRAKE_ACTION)
#######################################################################################################
##DEFINING BEHAVIORS
behavior CollisionAvoidance(brake_intensity=0.3):
    while withinDistanceToAnyObjs(self, SAFETY_DISTANCE):
        take SetBrakeAction(brake_intensity)
###########################################################################################################
behavior FollowLeadCar(safety_distance=10):
    try:
        do FollowLaneBehavior(target_speed=25)
    interrupt when ((distance to other) < safety_distance):
        do CollisionAvoidance()
############################################################################################################
behavior PullIntoRoad():
    while (distance from self to ego) > 15:
        wait
    do FollowLaneBehavior(laneToFollow=ego.lane)
```

Define the ego vehicle with the specified behaviours mentioned above

```
ego = Car following roadDirection from leadCar for EGO_TO_LEADCAR,
        with behavior EgoBehavior(EGO_SPEED)
```

- **Generate the road network**

The Scenic roads library is used to generate the road network geometry and traffic information. The road network is represented by an instance of the **Network** class and is generated from the **.xodr** file defined at the beginning of the script.

```
##DEFINING SPATIAL RELATIONS
# 'network' is the 'class Network' object in roads.py
# Make sure to put '*' to uniformly randomly select from all elements of the list, 'network.
lanes'
lane = Uniform(*network.lanes)
```

- **Set the scene by defining the position of Objects, EGO, NPC or Pedestrian.**

```
##OBJECT PLACEMENT
obstacle = NPCCar on lane

npc1 = NPCCar visible
```

- **Set an end point so the script knows when the scene is finished.**

```
terminate when ego.lane is None
terminate when other.lane is None
```

Note: Familiarity with Python is crucial for scripting in Scenic, as Scenic shares similarities with the Python language.

For more in-depth information on scenarios, consult the scenic scripts found in the Detection_Scenic folder within the Test_case/(particular map) directory. Keep in mind that, as we are using Scenic 2.0 with the deprecated OSSDC simulator, some functions may not be accessible.

Learn more:

- Scenic documentation: scenic
- Further insights and references are available in the provided links.