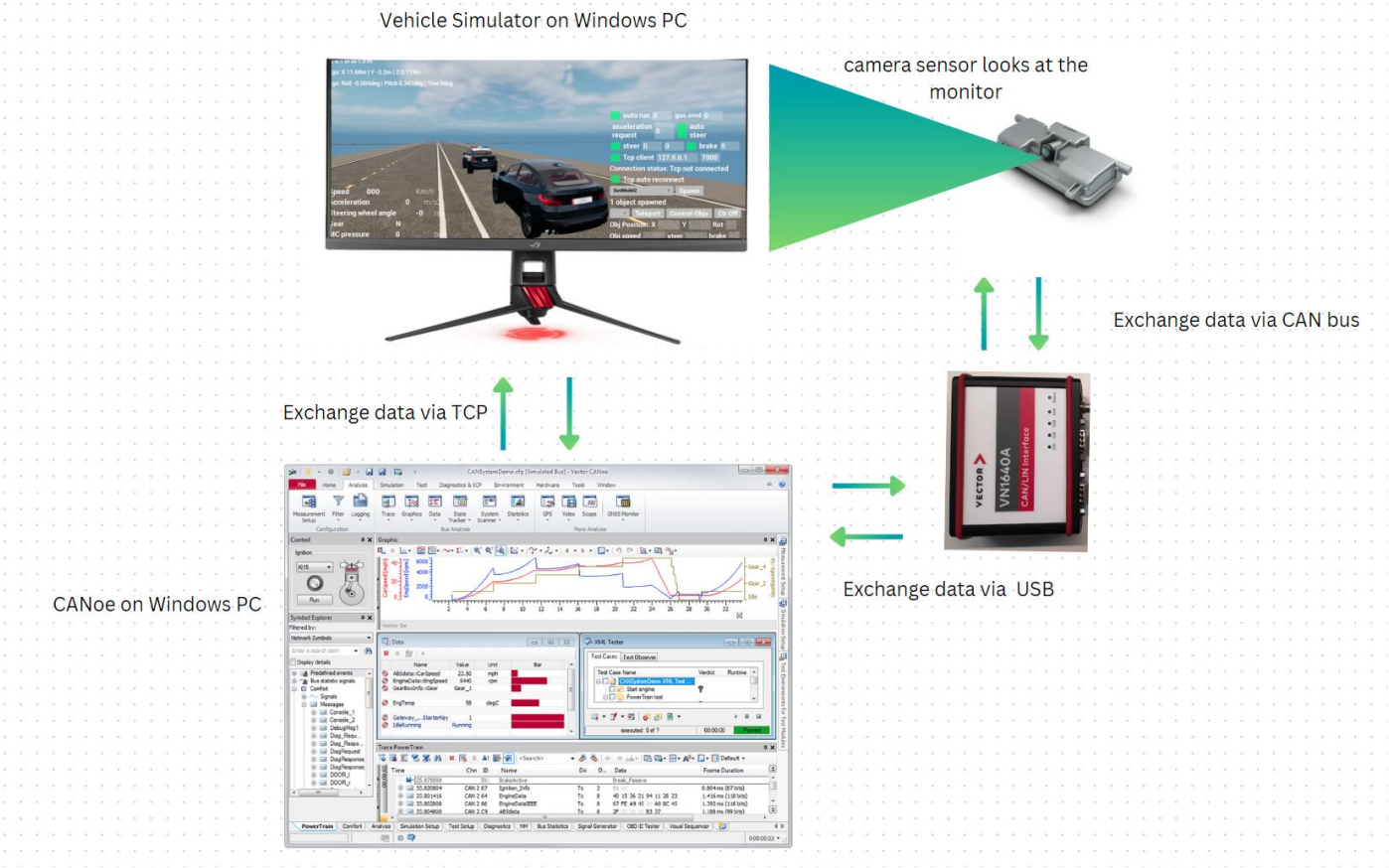
Adas TestSim Manual

Author: Nguyen Anh Duc

Table of Contents

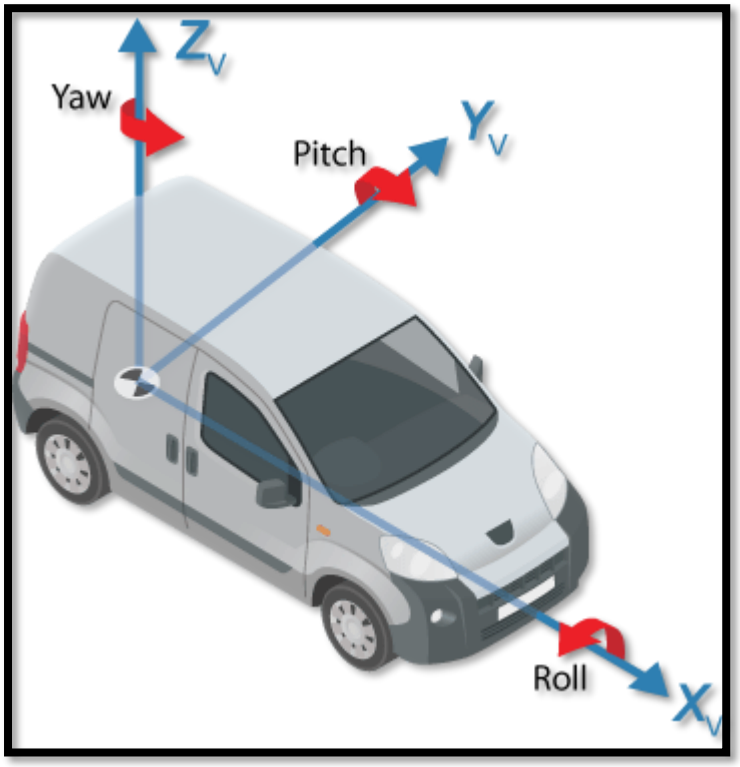
Introduction

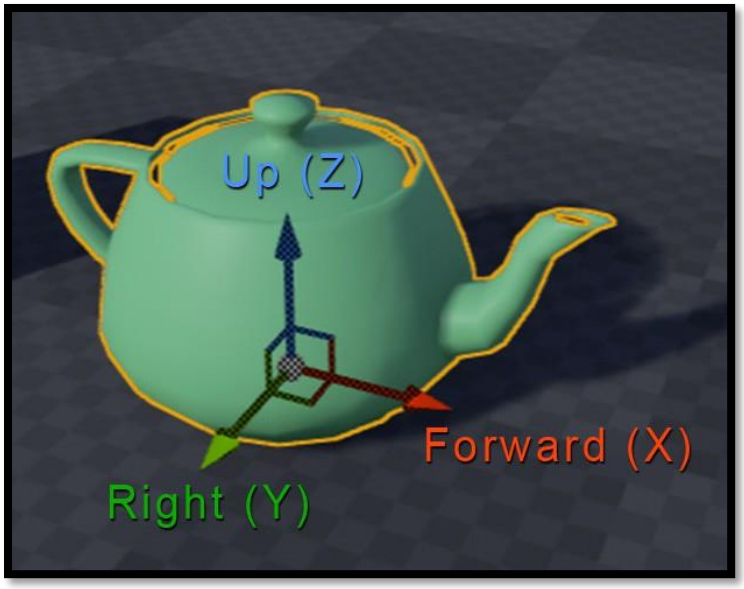
Adas TestSim is an application powered by Unreal Engine. It provides visualization of traffic situations and simple vehicle dynamics which can be used to test ADAS features in lab. The following picture depicts a setup for testing ADAS features implemented on a multi purpose camera ECU.



Coordinate System

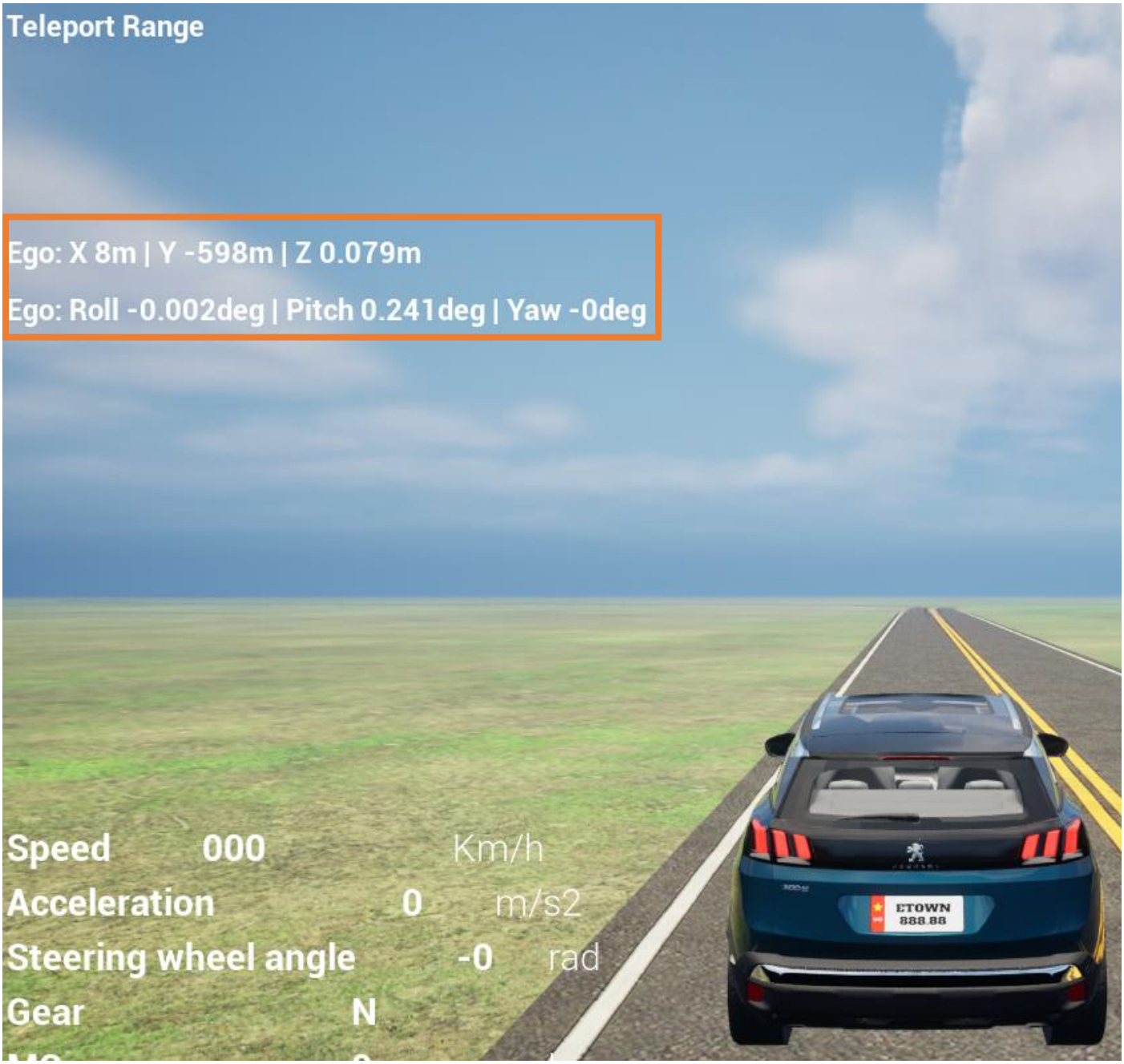
There is a difference between vehicle coordinate system (y to the left) and Unreal Engine coordinate system (y to the right). All locations described in maps section are presented in Unreal Engine coordinate system. Please remember to change the sign of y, yaw and roll in case you use these locations to put vehicles in a map.





Another way to know where to put vehicles is to drive the ego vehicle in a map. Once you get to the desired location, look at the top left of the screen,

you will have information of the desired location. The location coordinate is already converted to vehicle coordinate system.

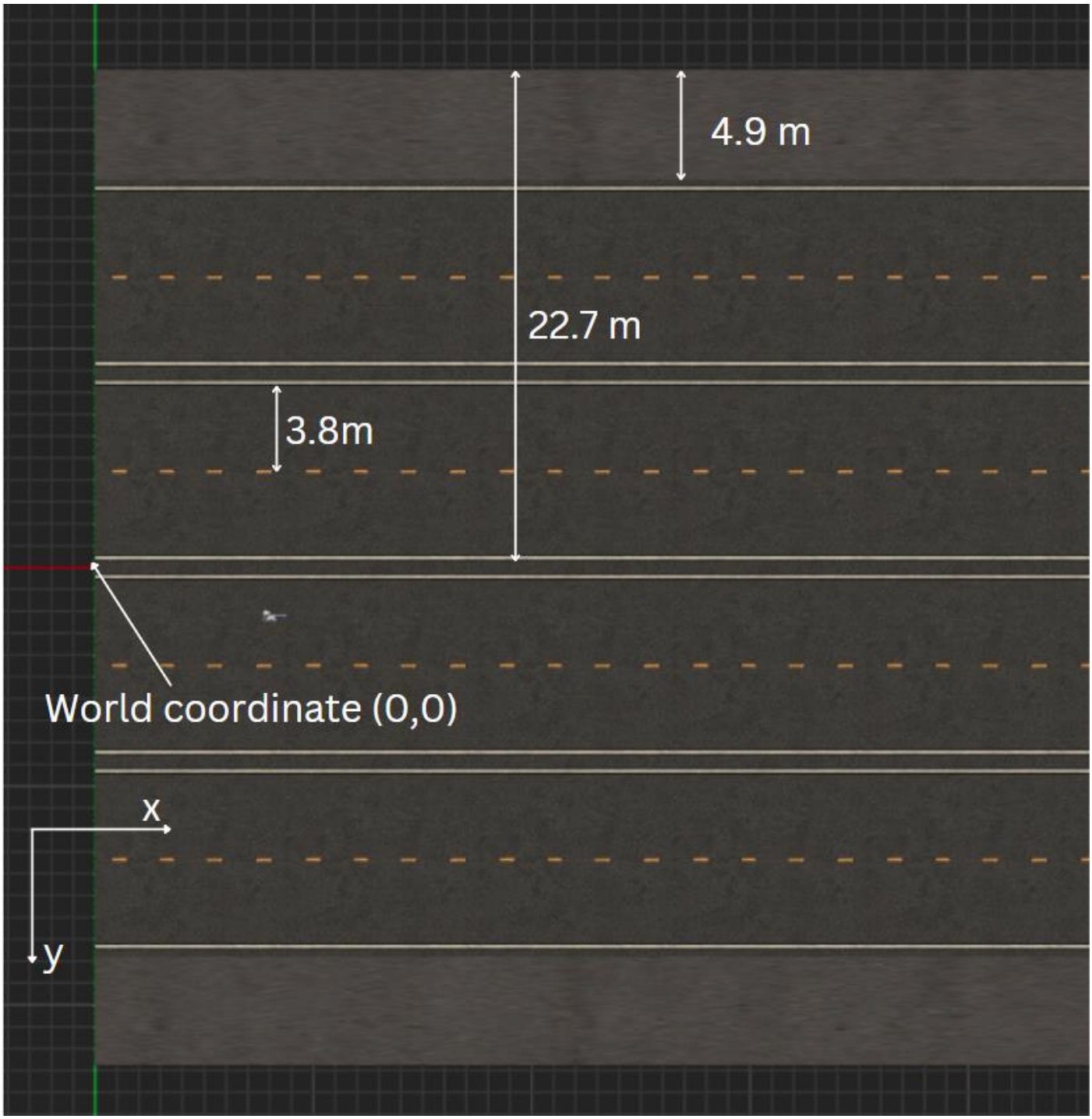


Maps

Don't try to spawn/teleport vehicles too near to the border of a map, it may cause error.

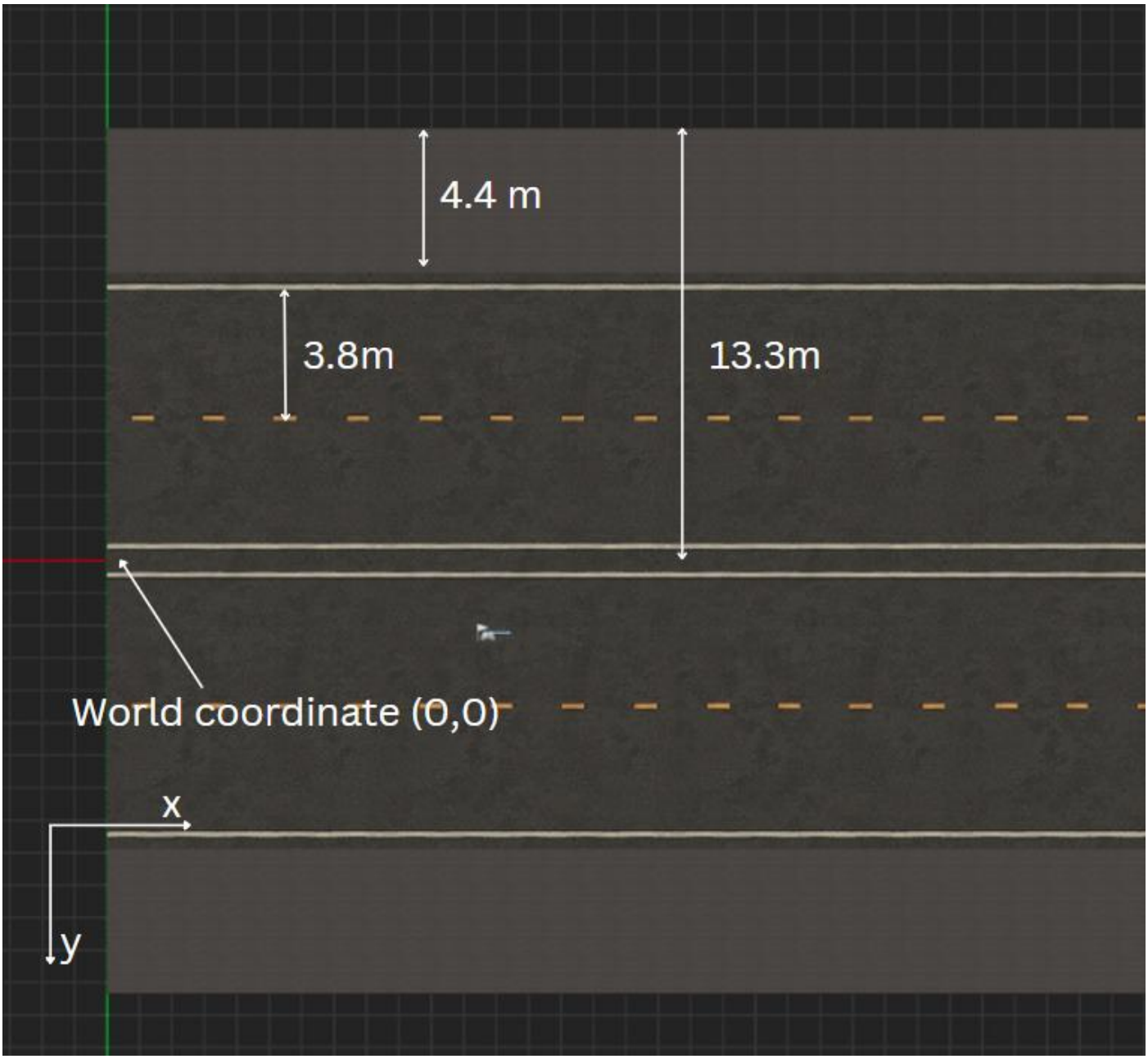
1. Straight Road Map

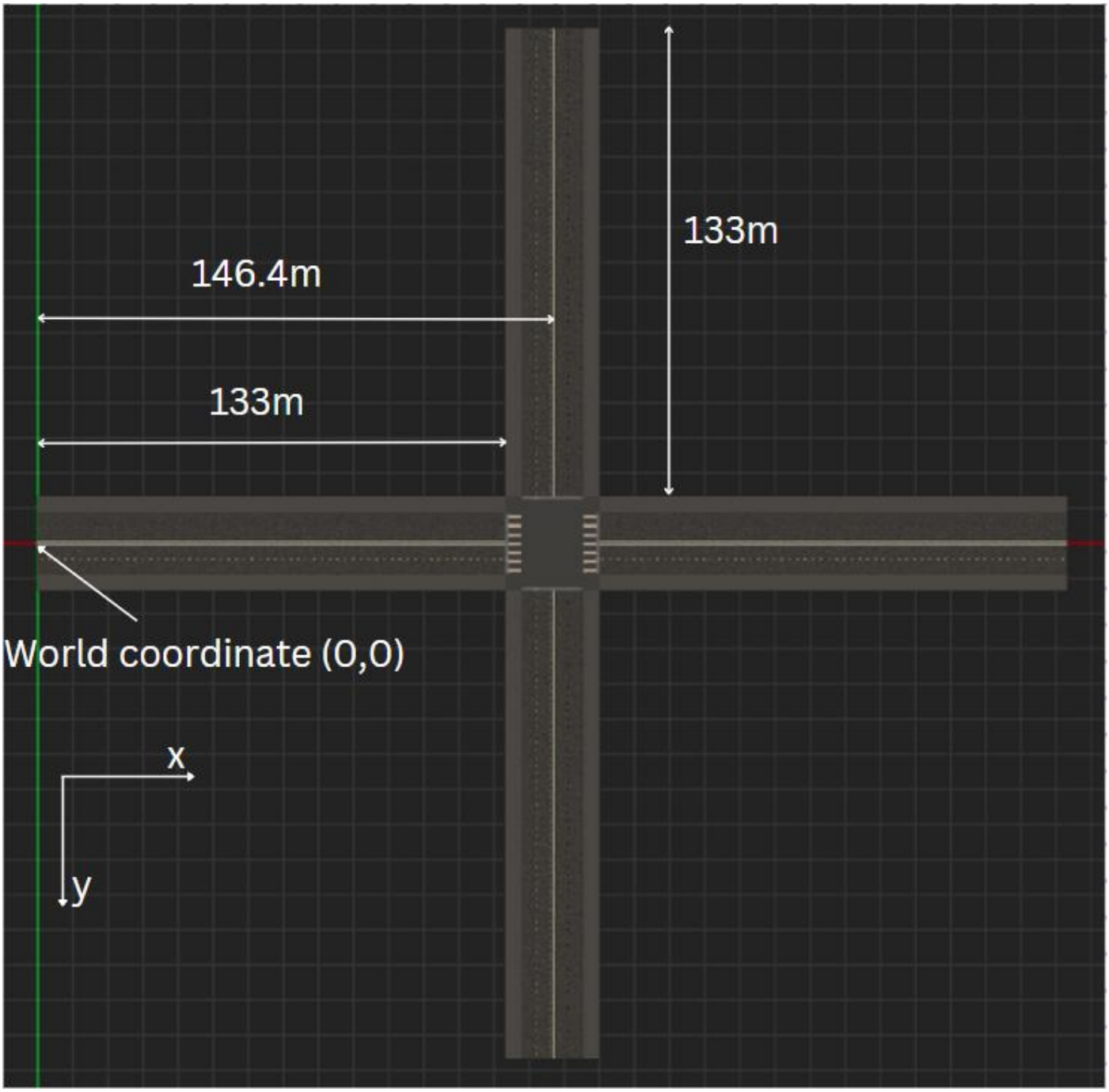
8 lanes, width ~45.4 m, length ~2220 m, default ego position (x: 8m, y: 2.22m)



2. Cross Road Map

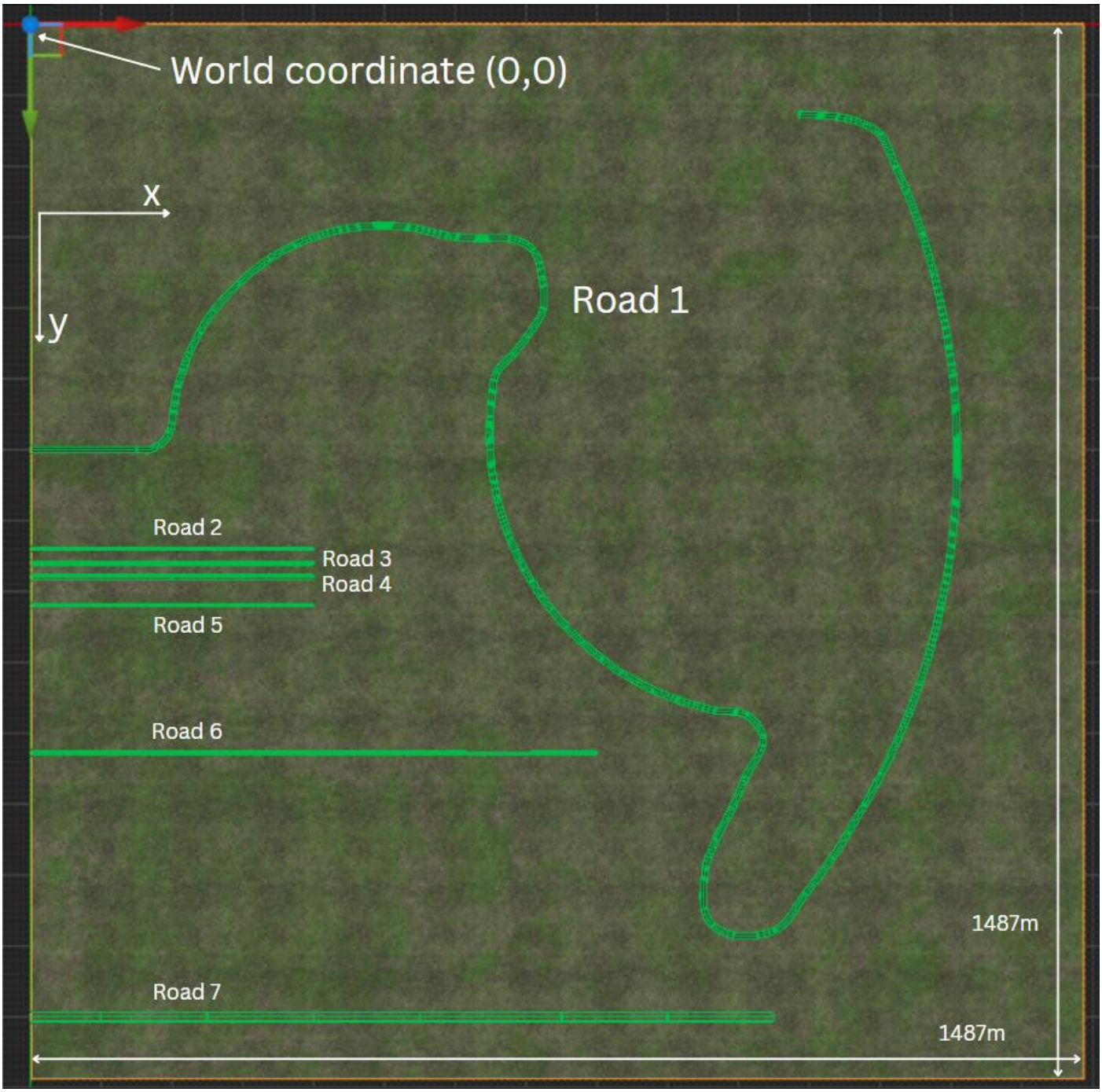
default ego position (x: 11.68m, y: 2.2m)


4.4 m
3.8m
13.3m
World coordinate (0,0)
x
y

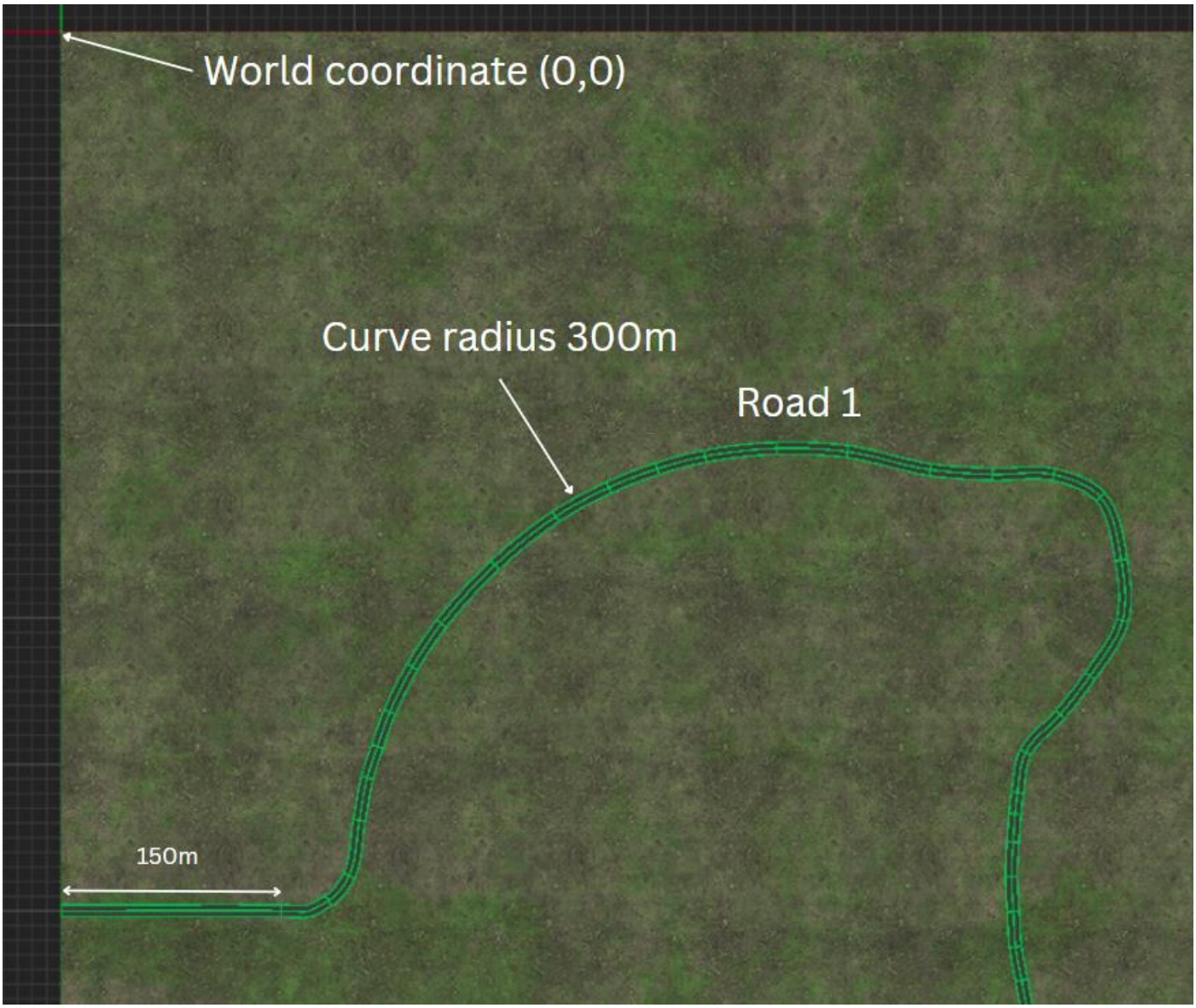


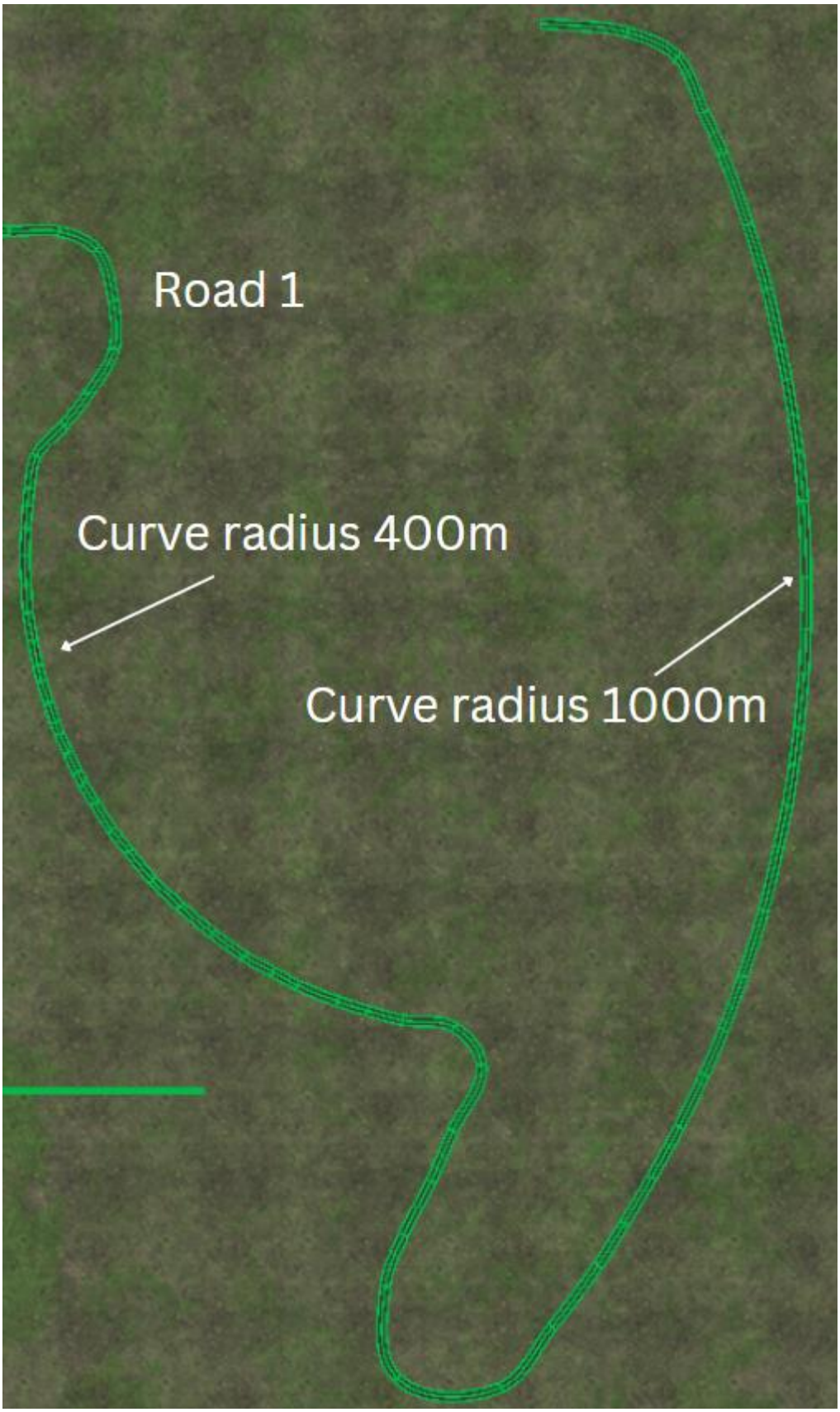
3. Mixed Roads Map A

This map contains 7 roads, default ego position (x: 8m, y: 598m, on road 1)



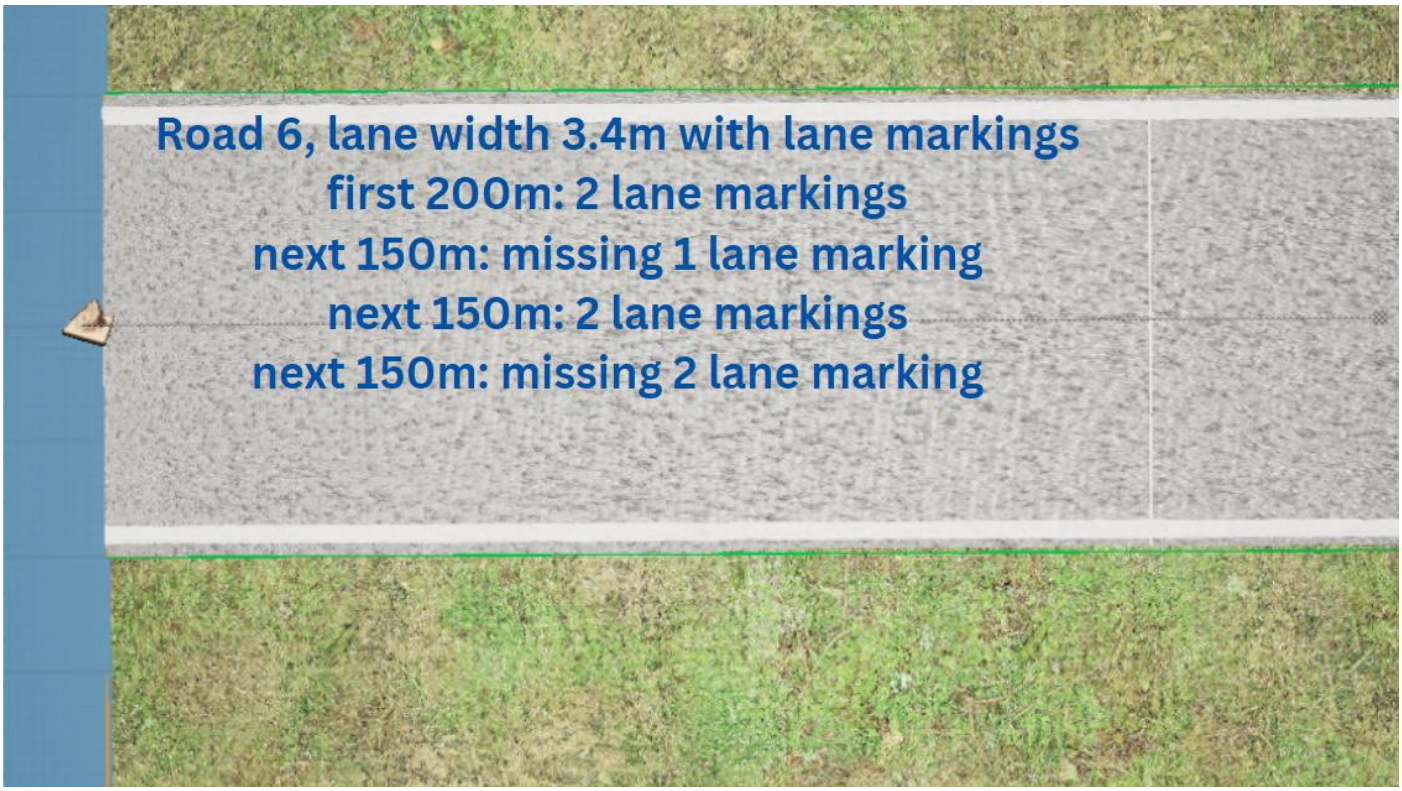
Road 1 is a 2-lanes road, lane width 3.4m. Road 1 consists of 3 main segments: 150m straight, 300m curve radius, 400m curve radius, 1000m curve radius. Reference location on curve radius 300m to place vehicles (x: 205.3m, y: 515.5m), (x: 209m, y: 516.4m)

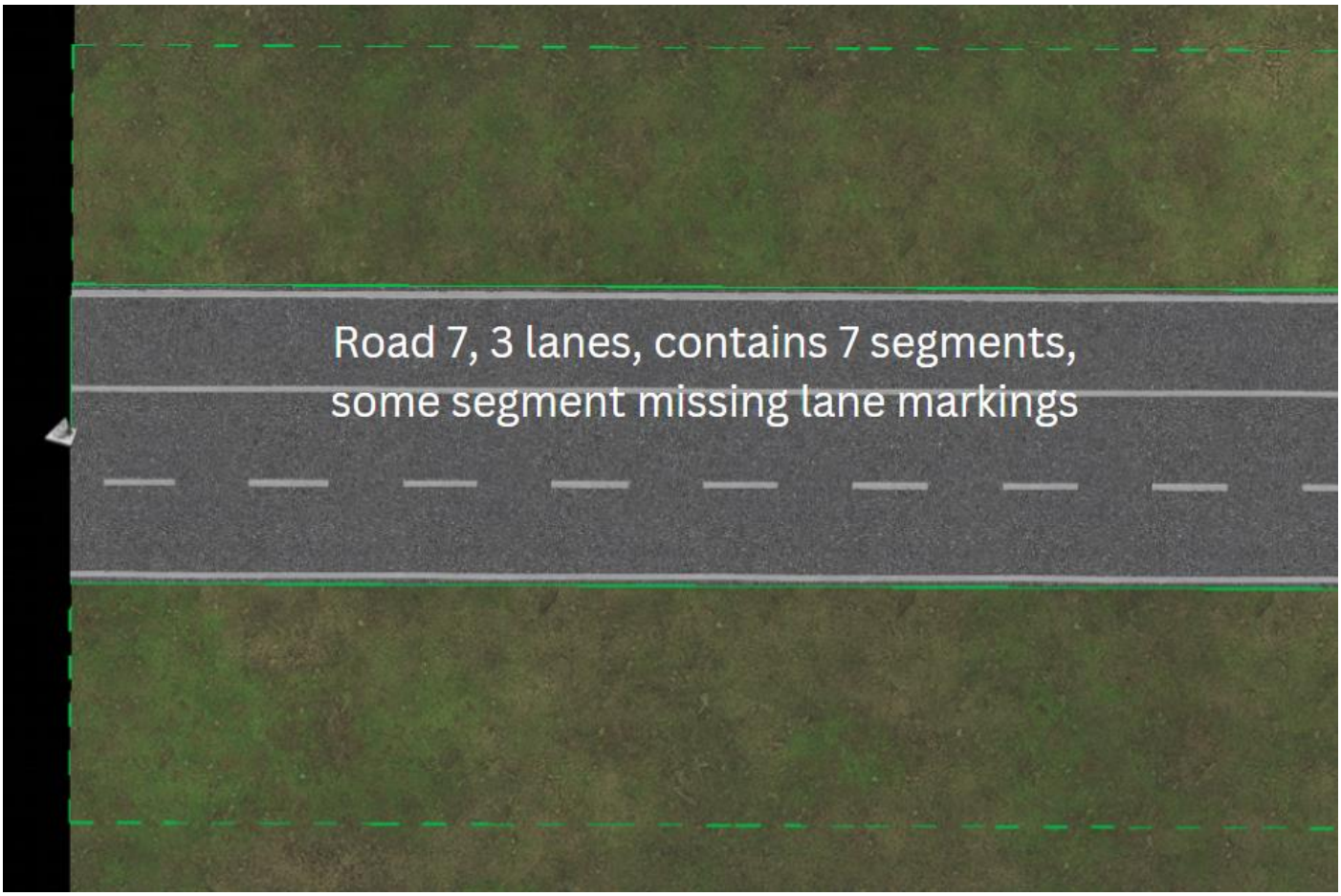

World coordinate (0,0)
Curve radius 300m
Road 1
150m


Road 1
Curve radius 400m
Curve radius 1000m


Road 2, 2-lanes, lane width 2m
Road 3, 2-lanes, lane width 3m
Road 4, 2-lanes, lane width 3.5m


Road 5, lane width 3.4m, no lane marking


Road 6, lane width 3.4m with lane markings
first 200m: 2 lane markings
next 150m: missing 1 lane marking
next 150m: 2 lane markings
next 150m: missing 2 lane marking

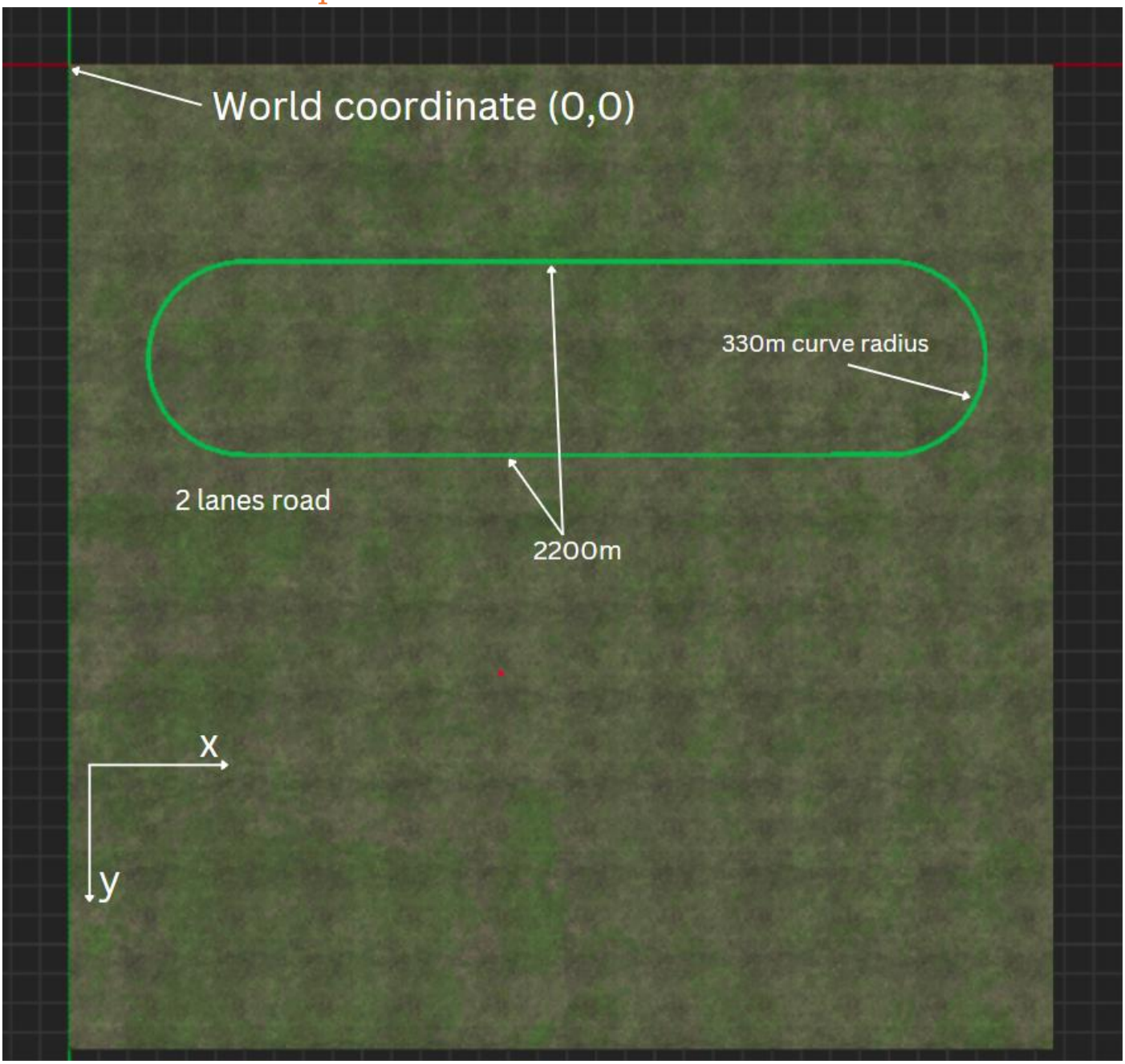


4. Mixed Roads Map B

Same size with Mixed Roads Map A. Contains 9 roads. Road 1 and 2 contain guiding lines so ego vehicle and other traffic participants can stay inside lane automatically. For the remaining 7 roads, I haven't implement guiding lines so there is no lane keeping function (auto steer) for vehicles.

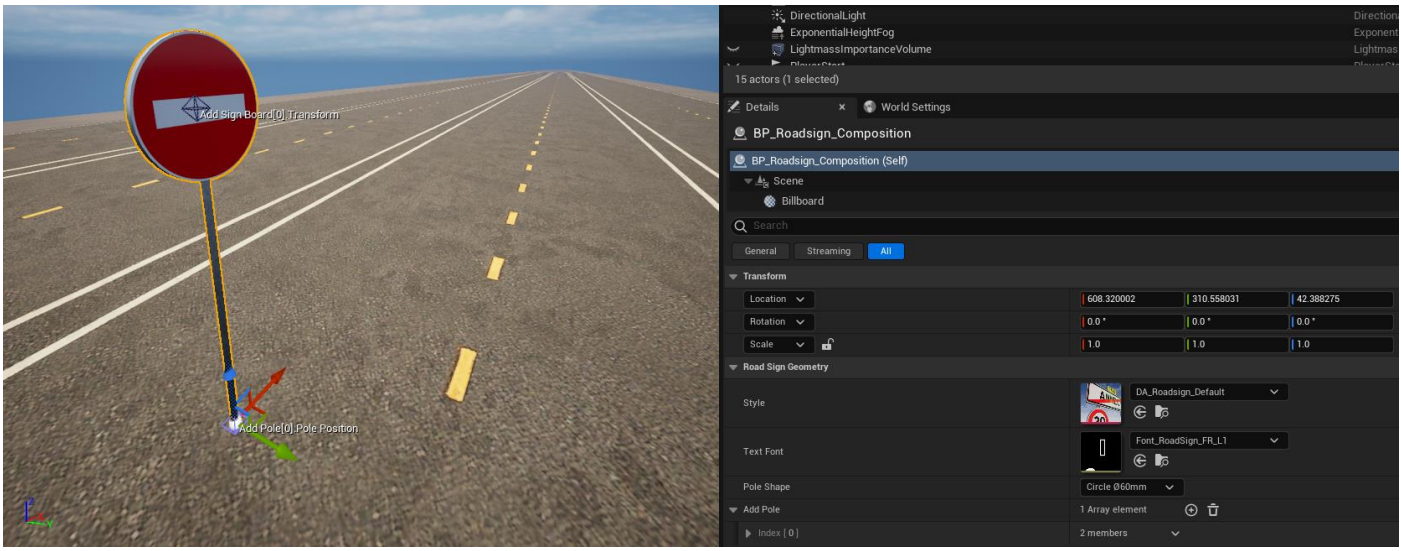
Remaining 7 roads with different lane markings

5. Test Track Map A



Traffic Signs

Please use the exact name of traffic signs image as described in this section to spawn them in runtime. Default orientation of a traffic sign model in Unreal Engine as follows:


Add Sign Board[0] Transform
Add Pole[0] Pole Position
DirectionalLight
ExponentialHeightFog
LightmassImportanceVolume
15 actors (1 selected)
Details
World Settings
BP_Roadsign_Composition
BP_Roadsign_Composition (Self)
Scene
Billboard
Search
General
Streaming
All
Transform
Location
608.320002
310.558031
42.388275
Rotation
0.0 °
0.0 °
0.0 °
Scale
1.0
1.0
1.0
Road Sign Geometry
Style
DA_Roadsign_Default
Text Font
Font_RoadSign_FR_L1
Pole Shape
Circle Ø60mm
Add Pole
1 Array element
Index [0]
2 members

Japan – JPN


psd
EndOfRestriction.png
8 – 20
EventTimeA.png
30
MinimumSpeedLimit30.png
50
MinimumSpeedLimit50.png
NoCrossingCenterLineToEnter.png
NoEntry.png
横断禁止
NoJaywalking.png
NoOvertake.png
NoParkingOrStop.png
NoTurn.png
NoUTurn.png
OnlyStraightAndLeftTurn.png
OnlyStraightAndRightTurn.png
RestrictionBegins.png
RestrictionEnds.png
RoadClose.png
RoadNarrowsOnBothSides.png
RoundAbout.png
安全速度
30
SafetySpeed30.png
SlipperyRoad.png
徐行
SLOW
SlowDown.png
10
SpeedLimit10.png
20
SpeedLimit20.png
30
SpeedLimit30.png
40
SpeedLimit40.png
50
SpeedLimit50.png
60
SpeedLimit60.png
70
SpeedLimit70.png
80
SpeedLimit80.png
90
SpeedLimit90.png
100
SpeedLimit100.png
110
SpeedLimit110.png
120
SpeedLimit120.png
止まれ
STOP
TomareStop.png
TurnLeft.png
TurnRight.png
WatchForLargeAnimals.png

China – CHN

Germany – DEU

Manual Controls

W: throttle

A, D: left, right steering

S: brake

Spacebar: handbrake

N, R: Neutral, Reverse gear

1, 2, 3: Back, Front, Top view camera

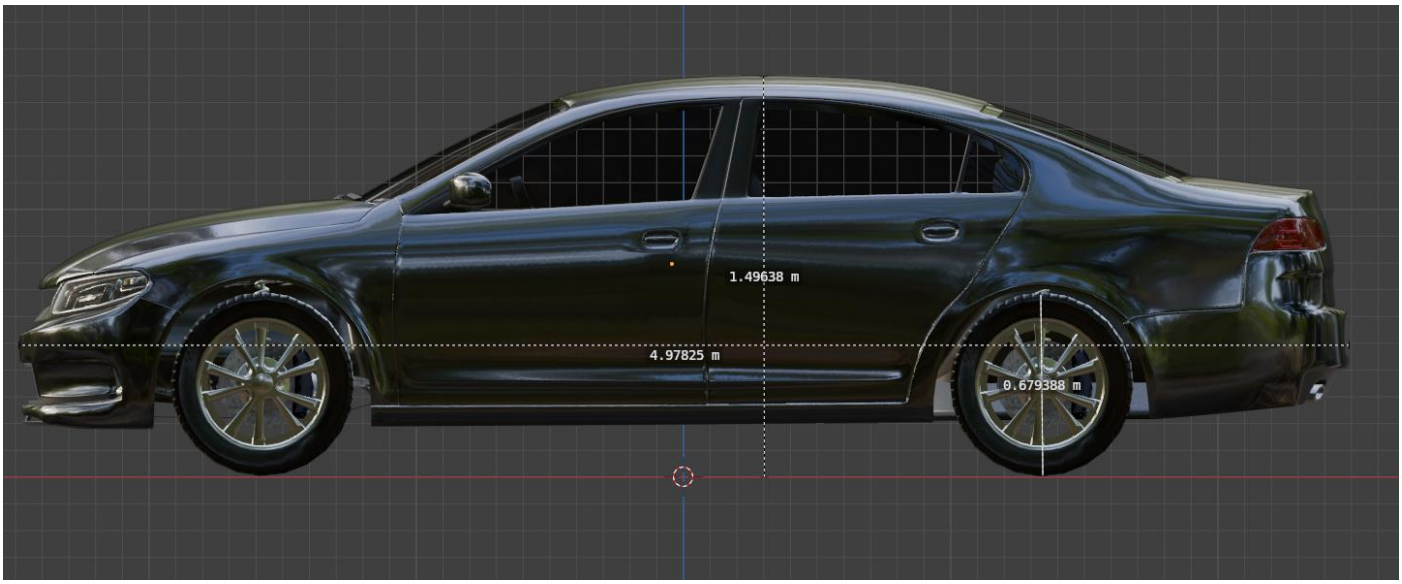
Numpad 1, 2: change ego model

P: Pause

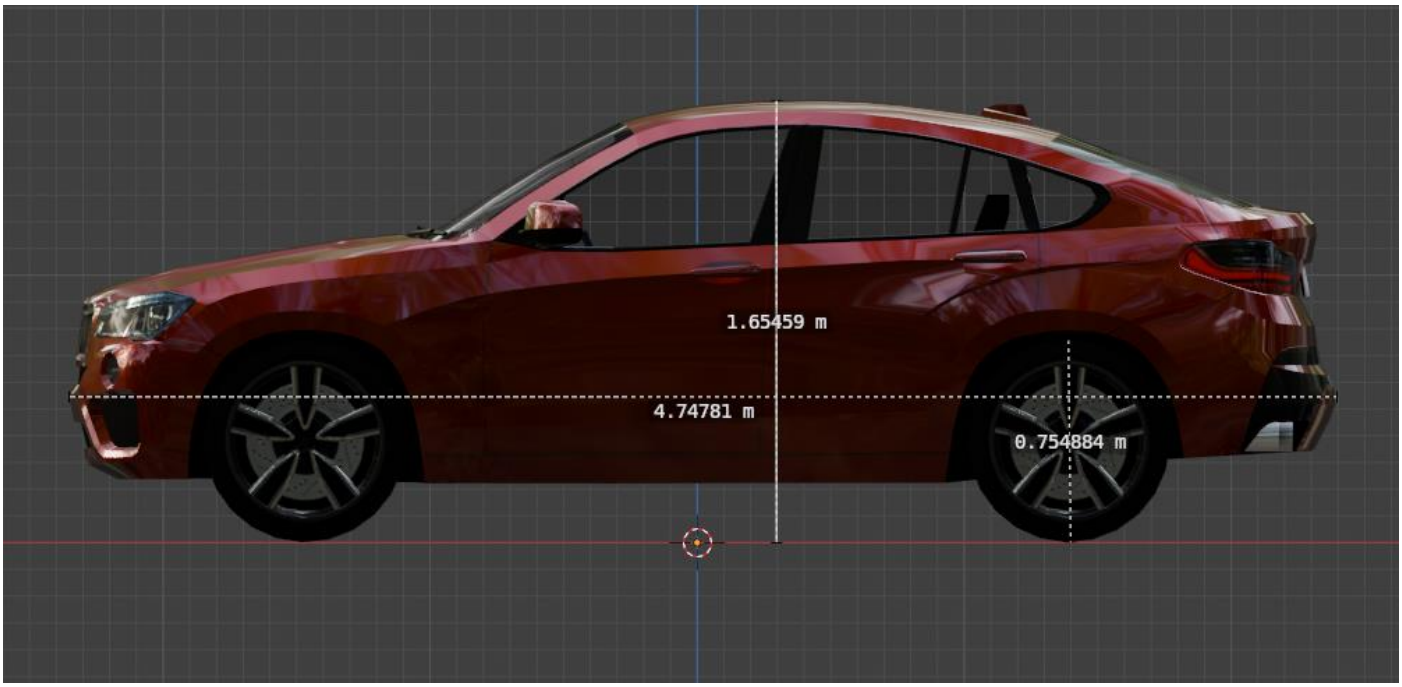
Vehicle 3D Models

1. Ego Vehicles

Sedan model 1



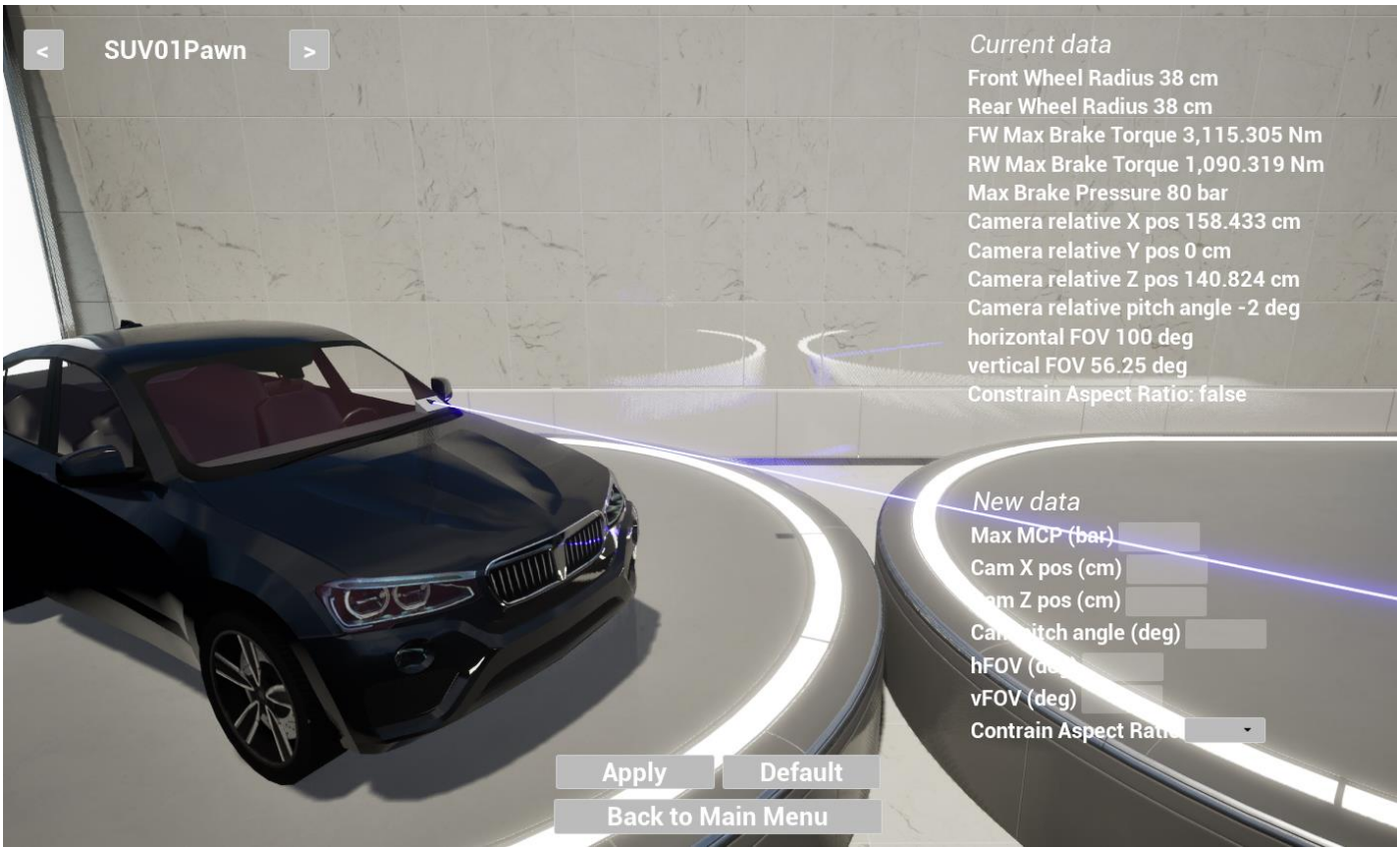
SUV model 1



2. Customize some data of Ego Vehicles

Following attributes can be modified:

- Maximum master cylinder brake pressure (calculate based on this video)
- Position of front camera (relative to ego vehicle) in x-axis, z-axis, pitch angle
- Horizontal and vertical Field of view of front camera in degree, in case constrain aspect ratio is yes then if the current monitor aspect ratio (width/height) is different from ratio of camera, black bars will be added to the view



Communication with Adas TestSim via TCP

A. Commands to control Adas TestSim

1/ StartSimulation -> Adas TestSim starts to send out vehicle data

2/ SetEgoSpeed_x -> x in km/h

3/ SetEgoSteering_x_y -> x in rad, range -3.14 to 3.14, y is steering angle speed in rad/s, range 0 to 6.283, set 0 to change steering angle instantly

4/ SetEgoBrake_x -> x range 0..1

5/ SetEgoGas_x -> x range 0..1

6/ EnableEgoAutoSteer -> set ego steering angle to 0 before using this command, only works with maps that have guiding spline

7/ DisableEgoAutoSteer -> only works with maps that have guiding spline

8/ SetEgoAcceleration_x -> x in m/s2

9/ SetEgoPosition_x_y_rot -> x in m, y in m, z in degree

10/ SpawnObject_x -> x in [SedanModel1, PickupModel1, SuvModel1, SuvModel2, Motorbike, Bus, Truck, Bicycle, Bicycle1, PedestrianMan, PedestrianWoman, PedestrianChild, Dog]

11/ SetObjectPosition_id_x_y_rot -> id is object spawn order, first spawn is 0, second spawn is 1, etc., 255 is invalid

12/ EnableControlObject_id_x_y_z -> id is object spawn order, x is desired speed in km/h, y is desired steering in rad, z is brake in 0..1

for characters (PedestrianMan, Dog): there is no brake implementation, if you want to stop a character, the only way is to set speed to 0

for characters (PedestrianMan, Dog): don't use rotation to change movement direction, use desired steering instead

13/ DisableControlObject -> free all controlled objects

14/ RestartMap -> restart current map

15/ OpenMap_x -> x is map name: StraightRoadMap, CrossRoadMap, MixedRoadsMap_A, MixedRoadsMap_B, TestTrackMap_A

16/ FrontCamera -> change to front view

17/ In order to spawn a traffic sign at runtime, need 3 steps:

a) Create a pole with one traffic sign using command SpawnTrsBegin
b) In case you need more than one sign on a pole, keep adding one sign a time using SpawnTrsContinue
c) Inform Adas TestSim to spawn a traffic signs with given information using SpawnTrsEnd

SpawnTrsBegin_CountryCode_TrafficSignName_TrafficSignShape_PoleX_PoleY_PoleZ_PoleYaw_PolePitch_PoleRoll_TrafficSignZ_ TrafficSignScaleX_ TrafficSignScaleY_ TrafficSignScaleZ

SpawnTrsContinue_ TrafficSignName_TrafficSignShape_TrafficSignZ_ TrafficSignScaleX_ TrafficSignScaleY_ TrafficSignScaleZ

SpawnTrsEnd

TrafficSignShape is enum -> 0: Round, 1: Triangle, 2: Square, 3: Diamond, 4: Octagon, 5: Rectangle

CountryCode follows ISO 3166-1 alpha-3

Example:

```
"SpawnTrsBegin_JPN_SpeedLimit60_0_10_-5_0.1_-90_0_0_2.4_1_1_1"
"SpawnTrsContinue_RestrictionEnds_5_1.8_1.6_1.0_0.4"
"SpawnTrsEnd"
```

The above commands will spawn this traffic sign

B. Decode the data sent from Adas TestSim

Data encoding rule

Data type length: float 4 bytes, integer 4 bytes

Order

1 - ego speed (float, km/h)

2 - ego gear (integer)

3 - ego engine RPM (float)

4 - ego master cylinder pressure (float, bar)

5 - ego longitudinal acceleration (float, m/s2)

6 - ego lateral acceleration (float, m/s2)

7 - ego yaw rate (float, deg/s)

8 - steering angle (float, rad)

9 - steering angle speed (rad/s)

10 - number of objects (1 byte)

11 - object ID (1 byte)

12 - object type (1 byte; enum type -> 2: Car, 3: Motorbike, 4: Truck, 5: Bus, 6: Bicycle, 7: Pedestrian)

13 - dx relative to ego (float, meter)

14 - dy relative to ego (float, meter)

15 - dz relative to ego (float, meter)

16 - vx relative to ego (float, m/s)

17 - vy relative to ego (float, m/s)

18 - yaw angle relative to ego (float, deg)

19, 20, 21, … -> repeat data from 11 to 18

Appendix

3D Models Source and Unreal Marketplace assets

"Lowpoly Generic SUV Police" (https://skfb.ly/oQJFT) by mk2design is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

"Lowpoly Generic SUV" (https://skfb.ly/oQJPW) by mk2design is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

"Generic civil service vehicles pack" (https://skfb.ly/orDHz) by Comrade1280 is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

"Generic passenger car pack" (https://skfb.ly/6sUFy) by Comrade1280 is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

"1990 Fatboy inspired Cruiser Motorcycle" (https://skfb.ly/otv7o) by Alpha One is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

“Scanned 3D People Pack” (https://renderpeople.com/free-3d-people/) by Renderpeople

"Generic Sedan Car" (https://skfb.ly/oIOJC) by Márcio Meireles is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

https://www.unrealengine.com/marketplace/en-US/product/tcp-socket-plugin

https://www.unrealengine.com/marketplace/en-US/product/roadsign-bp

https://www.unrealengine.com/marketplace/en-US/product/17c2d7d545674204a7644c3c0c4c58ba

https://www.unrealengine.com/marketplace/en-US/product/twinmotion-construction-vehicles

Vehicle dynamics

Torque curve creator:
https://www.lamertonsimulation.com/torque_curve_input.php

Gear shift

https://www.blocklayer.com/rpm-gear

Average deceleration

https://www.sae.org/publications/technical-papers/content/2023-01-0616/