



Scenario-based Testing with BeamNG.tech

Hands-on Training

Dr. Chrysanthi Papamichail

Dr. David Stark

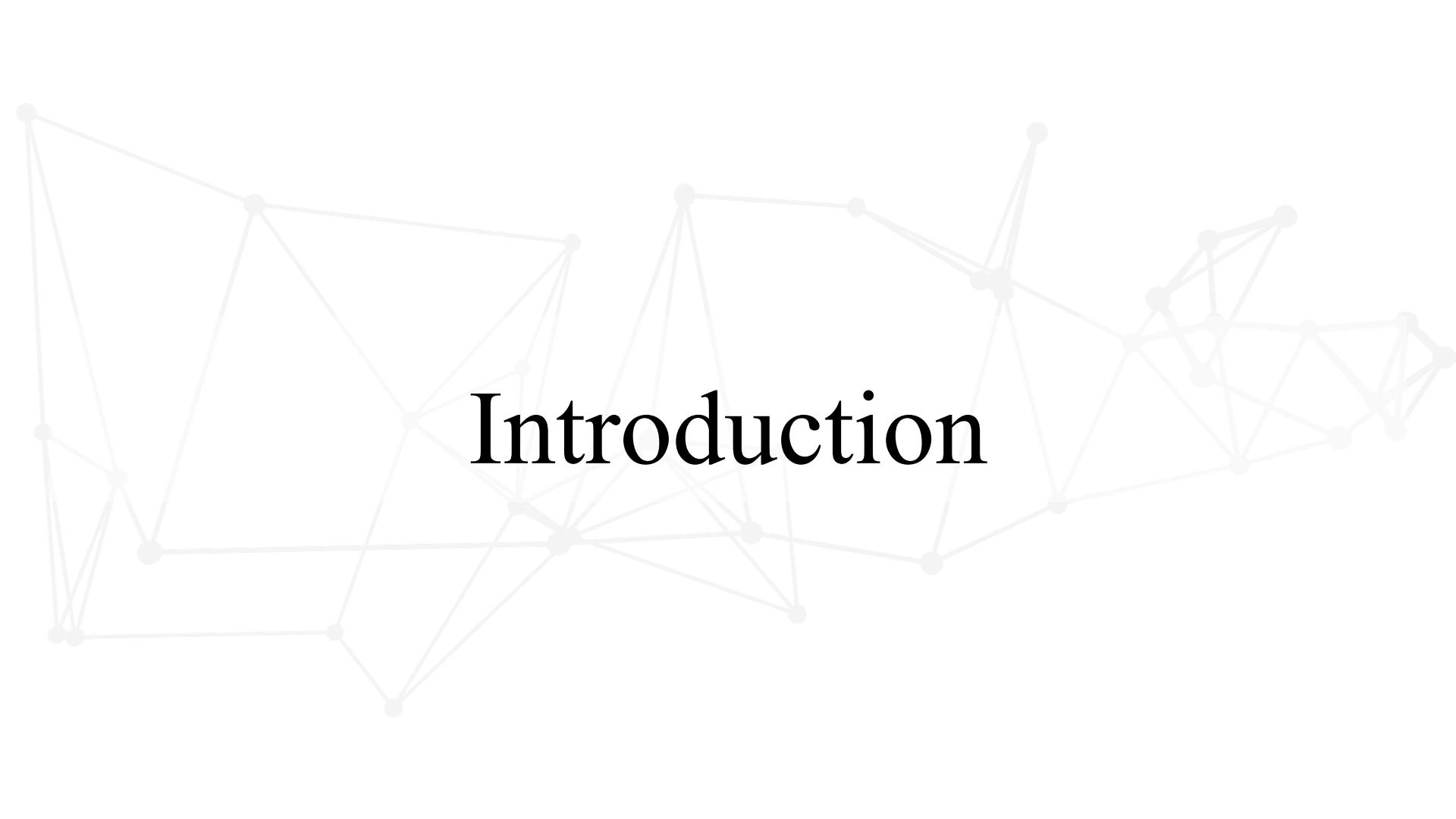
Dr. Lorenzo Bartali

Dr. Alessio Gambi



April 2025

Introduction



ADAS/ADS (Our Focus)

- Safety-critical Cyber-Physical Systems
- Expected to reduce **accidents** and **emissions**, increase **comfort**, and more



SAE J3016™ LEVELS OF DRIVING AUTOMATION™

Learn more here: sae.org/standards/content/j3016_202104

Copyright © 2021 SAE International. The summary table may be freely copied and distributed AS-IS provided that SAE International is acknowledged as the source of the content.

What does the human in the driver's seat have to do?

SAE LEVEL 0™

SAE LEVEL 1™

SAE LEVEL 2™

SAE LEVEL 3™

SAE LEVEL 4™

SAE LEVEL 5™

You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering

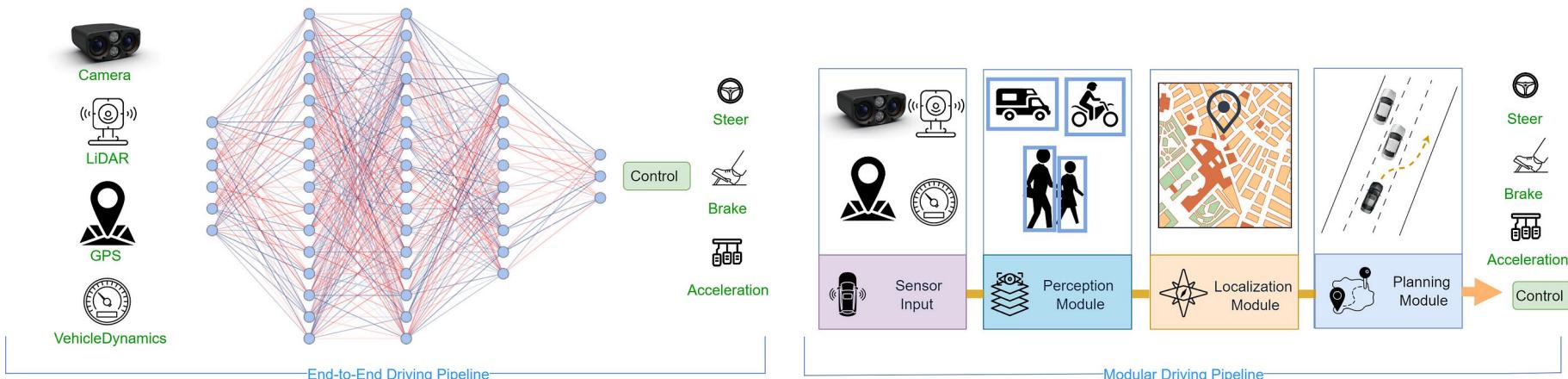
You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety

You are not driving when these automated driving features are engaged – even if you are seated in “the driver’s seat”

When the feature requests,
you must drive

These automated driving features will not require you to take over driving

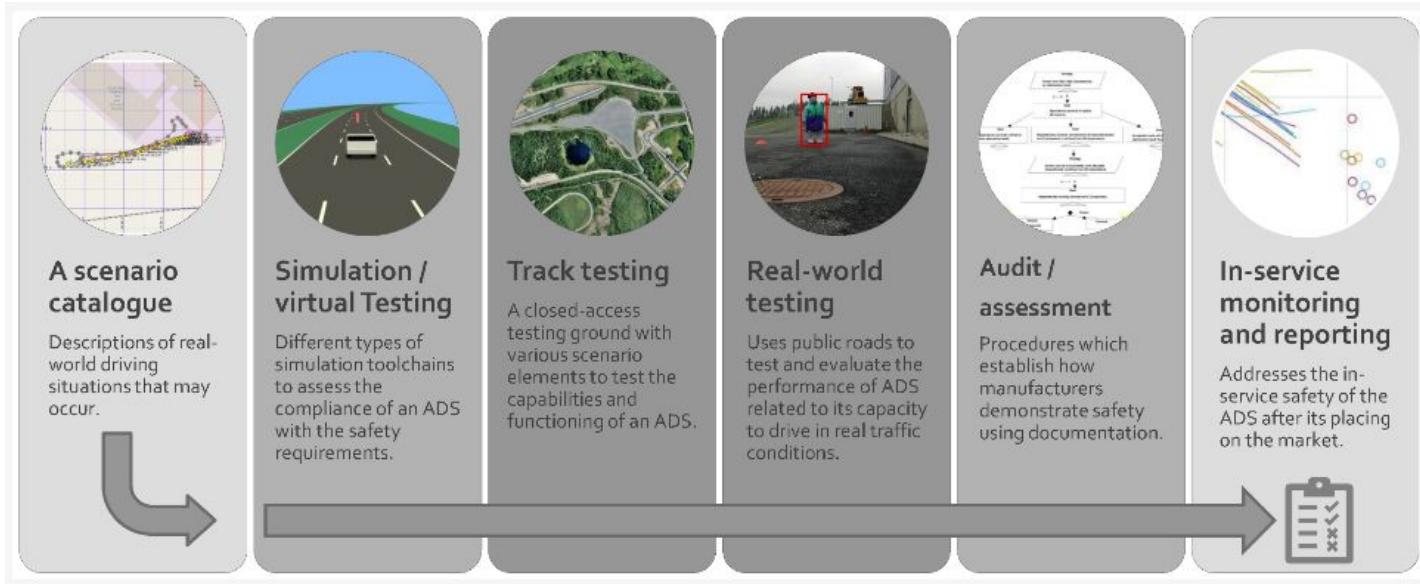
ADAS/ADS High-Level Architectures



Scenario-Based Testing (SBT)

- Testing the **ego vehicles** (system under test) in predefined **traffic scenarios**
- SBT can be done on proving grounds and real-roads (naturalistic field tests) but it can also be done in (driving) simulations
- **No free lunch:** Naturalistic field tests are expensive and dangerous and may be ineffective, whereas simulation-based tests require photorealistic and physically accurate driving simulations that come with a reality gap (generalizability issues)
- In practice, naturalistic field and simulation-based tests are often combined

Scenario-Based Testing (SBT) (cont.)

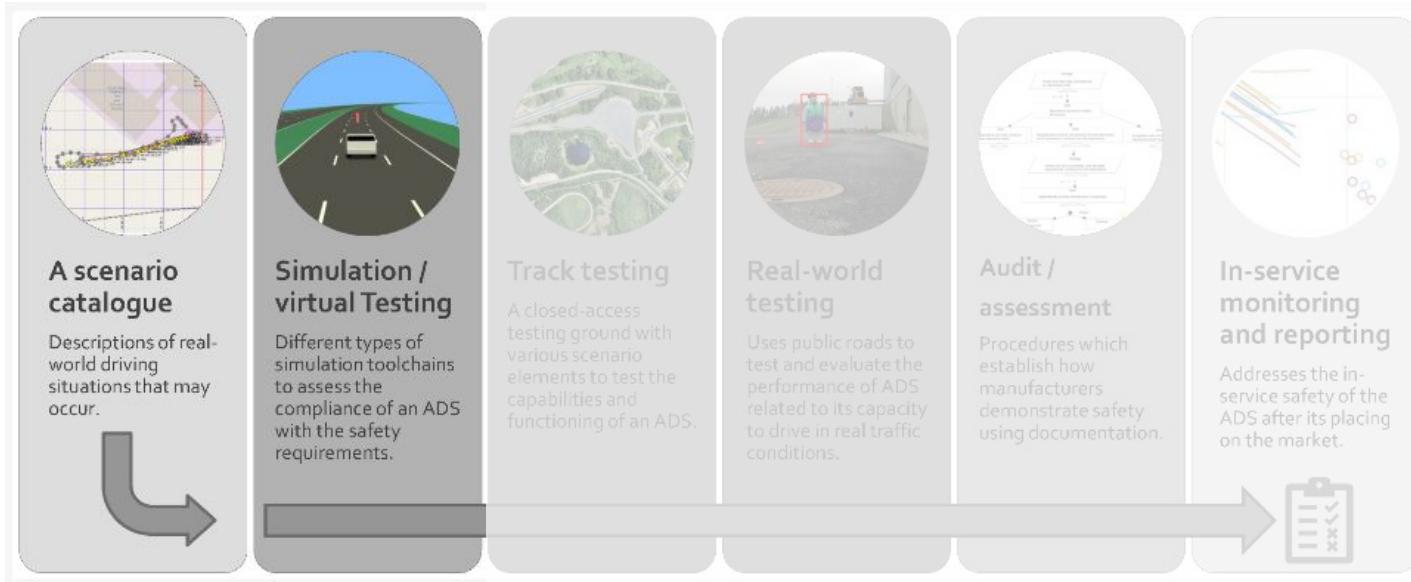


Enhancing Safety Assessment of Automated Driving Systems with Key Enabling Technology Assessment Templates

by Martin Skoglund 1,* , Fredrik Warg 1,† , Anders Thorsén 1,†  and Mats Bergman 2 

(This article belongs to the Special Issue Emerging Transportation Safety and Operations: Practical Perspectives)

Scenario-Based Testing (SBT) (cont.)



Enhancing Safety Assessment of Automated Driving Systems with Key Enabling Technology Assessment Templates

by Martin Skoglund 1,* , Fredrik Warg 1,† , Anders Thorsén 1,†  and Mats Bergman 2 

(This article belongs to the Special Issue Emerging Transportation Safety and Operations: Practical Perspectives)

Scenario Abstraction Levels

Concrete	Logical	Abstract	Functional
User assigns a single value to all attributes	User either assigns a value or provides a fully specified ranges	User can use any constraint on any attribute including cross attributes, location, timing and control flow constraints	Not formal semantics and not machine readable. May include intuitions, motivation, figures, and more

Scenarios for Development, Test and Validation of Automated Vehicles

Authors:  Till Menzel,  Gerrit Bagschik,  and Markus Maurer | [Authors Info & Claims](#)

[2018 IEEE Intelligent Vehicles Symposium \(IV\)](#) • Pages 1821 - 1827 • <https://doi.org/10.1109/IVS.2018.8500406>

<https://www.foretellix.com/how-do-you-get-more-out-of-your-logical-scenarios-and-how-do-abstract-scenarios-take-productivity-and-safety-to-a-new-level/>



Examples of Scenarios

You are an autonomous driving system developer that needs to test an ego vehicle in simulation. List three interesting testing scenarios to start testing the ego vehicle. For each scenario, provide a description at functional, abstract, logical and concrete level

LeGEND: A Top-Down Approach to Scenario Generation of Autonomous Driving Systems Assisted by Large Language Models

Authors: Shuncheng Tang, Zhenya Zhang, Jixiang Zhou, Lei Lei, Yuan Zhou, Yinxing Xue | [Authors Info & Claims](#)

ASE '24: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering • Pages 1497 - 1508
<https://doi.org/10.1145/3691620.3695520>



Examples of Scenarios

You are an autonomous driving system developer that needs to test an ego vehicle in simulation. List three interesting testing scenarios to start testing the ego vehicle. For each scenario, provide a description at functional, abstract, logical and concrete level

Scenario 1. Sudden Cut-in by Another Vehicle

Scenario 2. Unprotected Left Turn at an Intersection

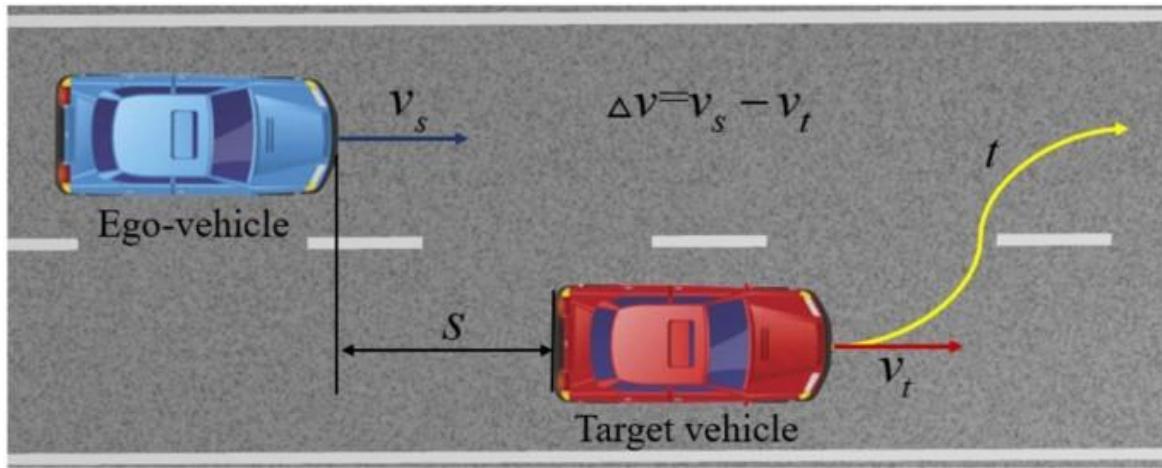
Scenario 3. Pedestrian Crossing Unexpectedly



Sudden Cut-in by Another Vehicle

- **Functional Level:** ego vehicle must maintain safety while minimizing discomfort when another vehicle suddenly merges into its lane
- **Abstract Level:** A vehicle ahead in an adjacent lane merges into the ego vehicle's lane requiring immediate adaptation of speed or trajectory
- **Logical Level**
 - Road Type and Environment: Highway with three lanes, dry pavement.
 - Ego vehicle: speed 96 km/h. Cut-in Vehicle: speed 88 km/h
 - Other Agents: traffic ahead and behind the ego vehicle in its lane.
- **Concrete Level**
 - The ego vehicle follows the lead vehicle at a 2-second gap.
 - Another vehicle (scripted in the simulation) abruptly moves into the ego vehicle's lane, reducing the available headway to less than 1 second.
 - ... additional details omitted ...

Sudden Cut-in by Another Vehicle (Possible Visualization)

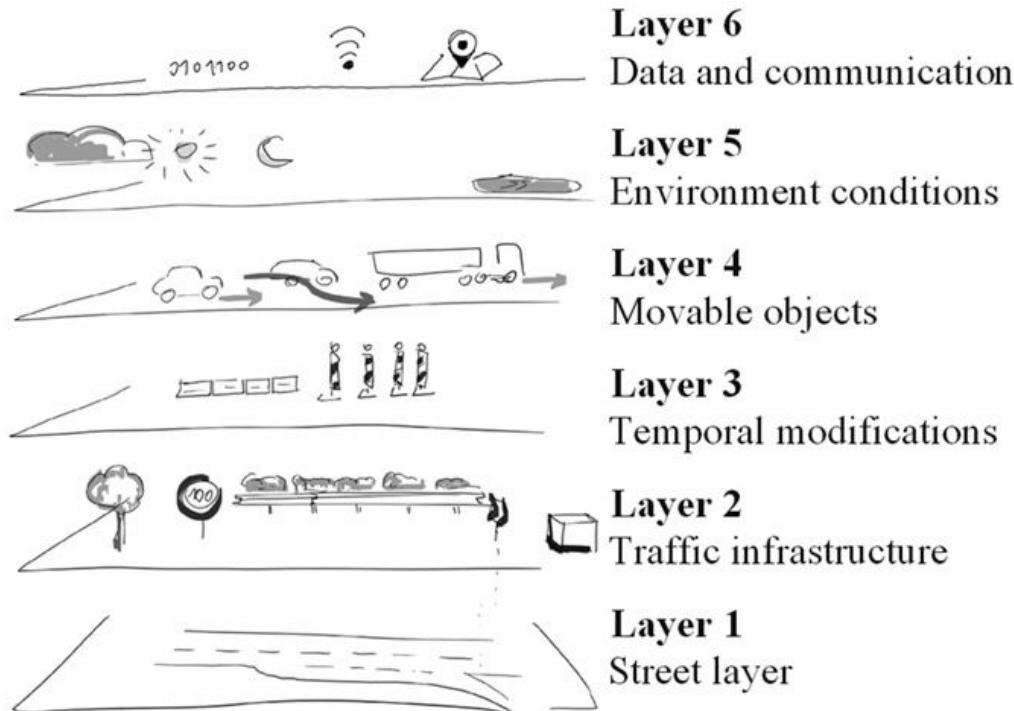


Generalization of cut-in pre-crash scenarios for autonomous vehicles based on accident data

[Pingfei Li](#), [Xinyu Zhu](#), [Yao Ren](#), [Zhengping Tan](#), [Wenhao Hu](#)✉, [You Zhang](#) & [Chang Xu](#)

[Scientific Reports](#) **14**, Article number: 17664 (2024) | [Cite this article](#)

Logical Scenarios Definition



Bock et al. (2019). A framework for definition of logical scenarios for safety assurance of automated driving. *Traffic Injury Prevention*. 20. S65-S70. 10.1080/15389588.2019.1630827.

This Tutorial Goals

Effective and efficient SBT in simulation requires generating scenarios quickly, accessing **rich** content, and executing **photorealistic and physical accurate** simulations automatically

This tutorial illustrates how to achieve that using  **BeamNG.tech** and focusing on manual scenario generation (central to SBT but often neglected)

Additionally, it aims to stimulate further research, promote best practices, and reflect **BeamNG.GmbH's commitment** to fostering the innovation and development of new technologies for mobility.

Target Audience

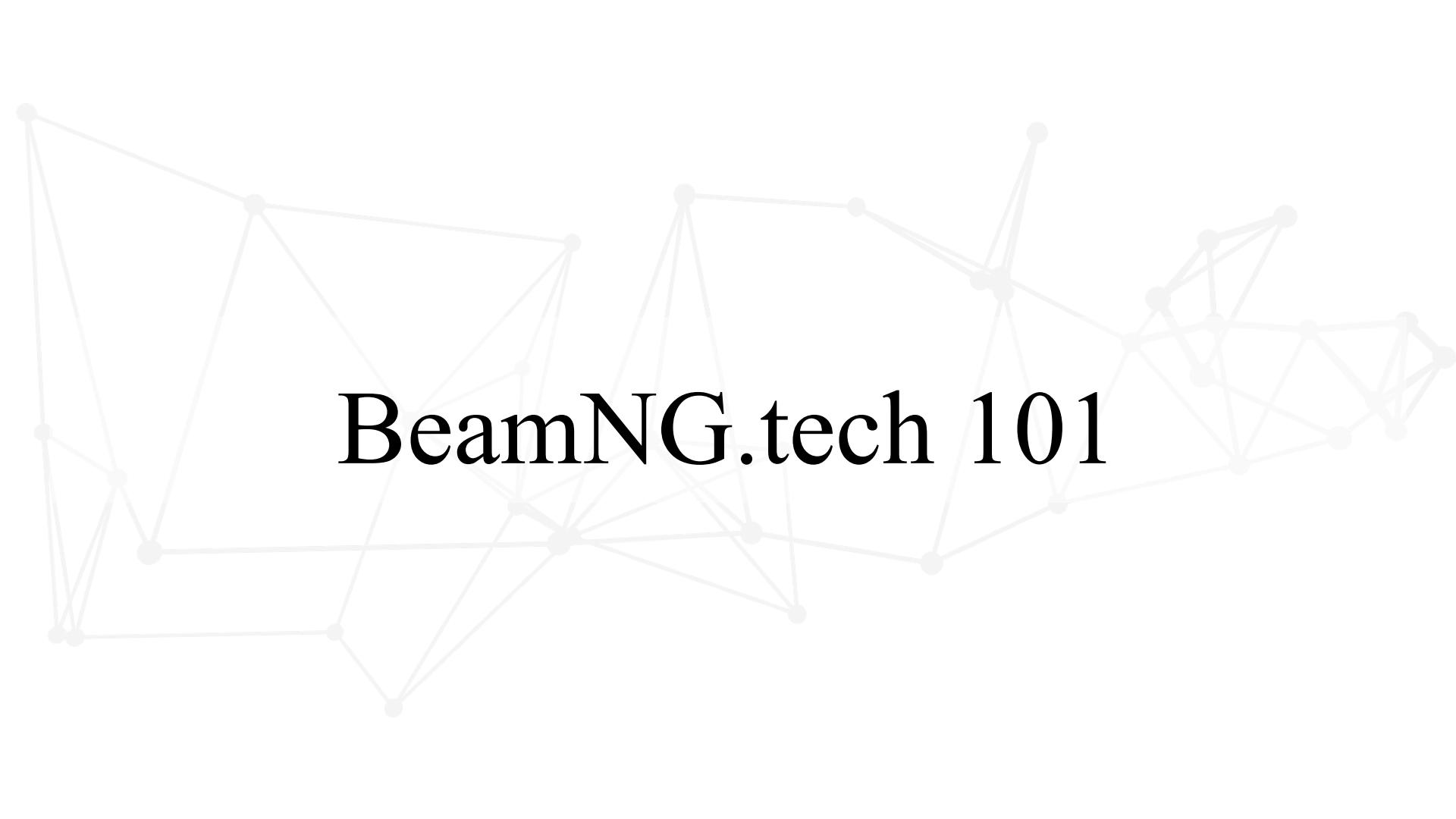
Anyone interested in exploring, learning, developing, and applying scenario-based testing methodologies, including

- Experts in validation and verification
- Students approaching ADS development and validation
- Educators who plan to include scenario-based testing in class
- AI/ML experts who need large collections of labeled data
- and (probably) more !

Useful Links

- Tutorial GitHub:
 - <https://github.com/BeamNG/scenario-based-testing-tutorial>
- BeamNG.tech Homepage:
 - <https://beamng.tech/>
- BeamNG.tech Distro (*minimized* for this tutorial) and *temporary* license key:
 - <https://nextcloud.beamng.com/s/KC4ztXbfmxsbF8/authenticate/showshare>
 - Password distribution via email
- BeamNG's Python API (tutorial branch)
 - <https://github.com/BeamNG/BeamNGpy/tree/icst-2025>
- Contacts:
 - Dr. Chrysanthi Papamichail: cpapamichail@beamng.gmbh
 - Dr. Alessio Gambi: alessio.gambi@ait.ac.at
 - BeamNG.tech non-commercial license: <https://register.beamng.tech/>
 - BeamNG.tech commercial license: licensing@beamng.gmbh

BeamNG.tech 101

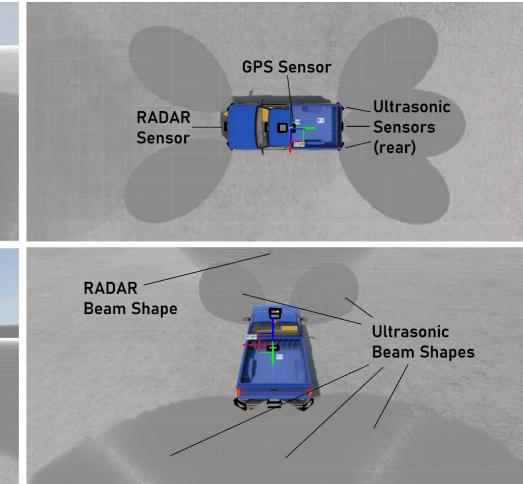
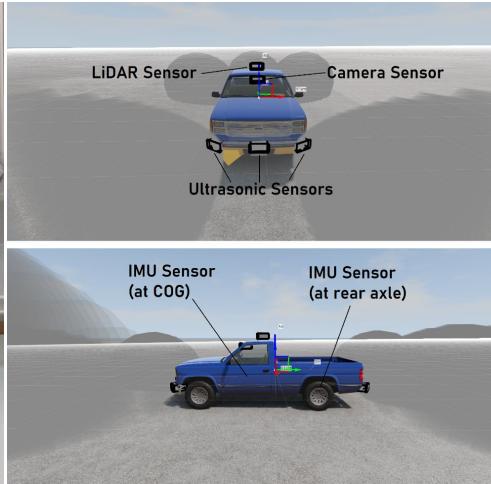


Extensive, Experimental, Advanced Simulation Framework

- **Several tools** for scenarios setup
- Realistic and ground-truth **sensors**
- Plenty of **content** (maps, props, vehicles,...)
- Accurate **soft-body** Physics simulation
- Photorealistic graphics
- (**Faster than**) Real-time execution
- Interoperability (CAN, ROS, Simulink, ...)
- (**Semi**) Open-source
- Extensible via its mod interface



BeamNG.tech Applications



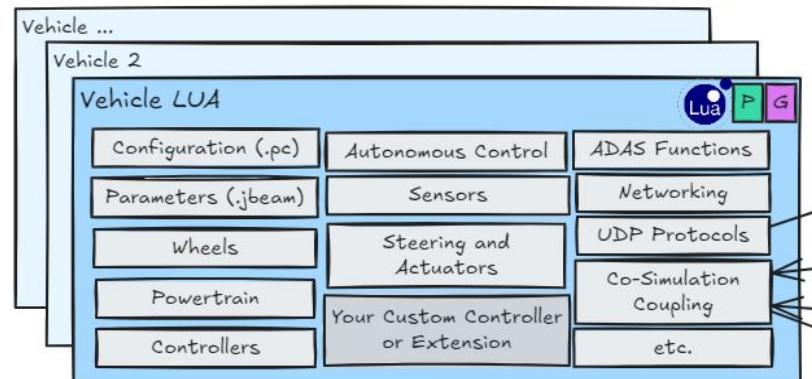
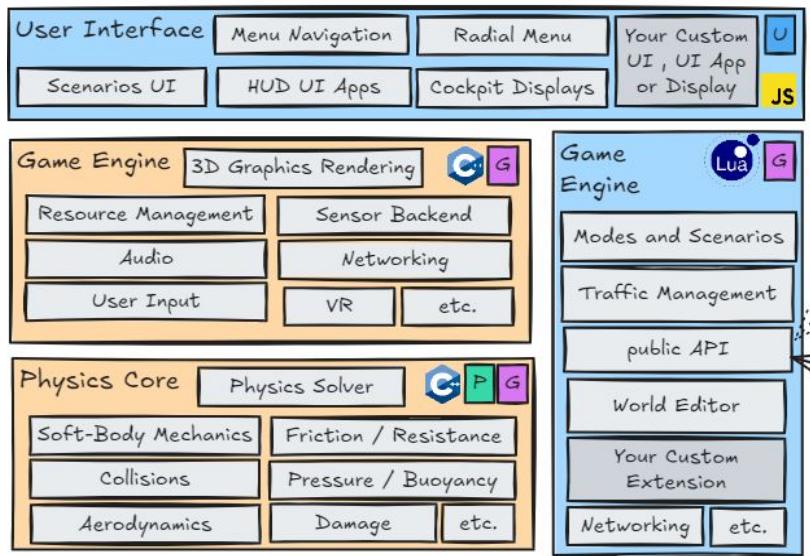
- Crash-simulation, X-in-the-loop testing, vehicle/wheel dynamics, driver training, scenario generation, and more.



- Since 2018
- 60+ papers
- 20+ theses
- 6+ Challenges on Test generation, Test Selection
- 3 Tutorials (including the current one)

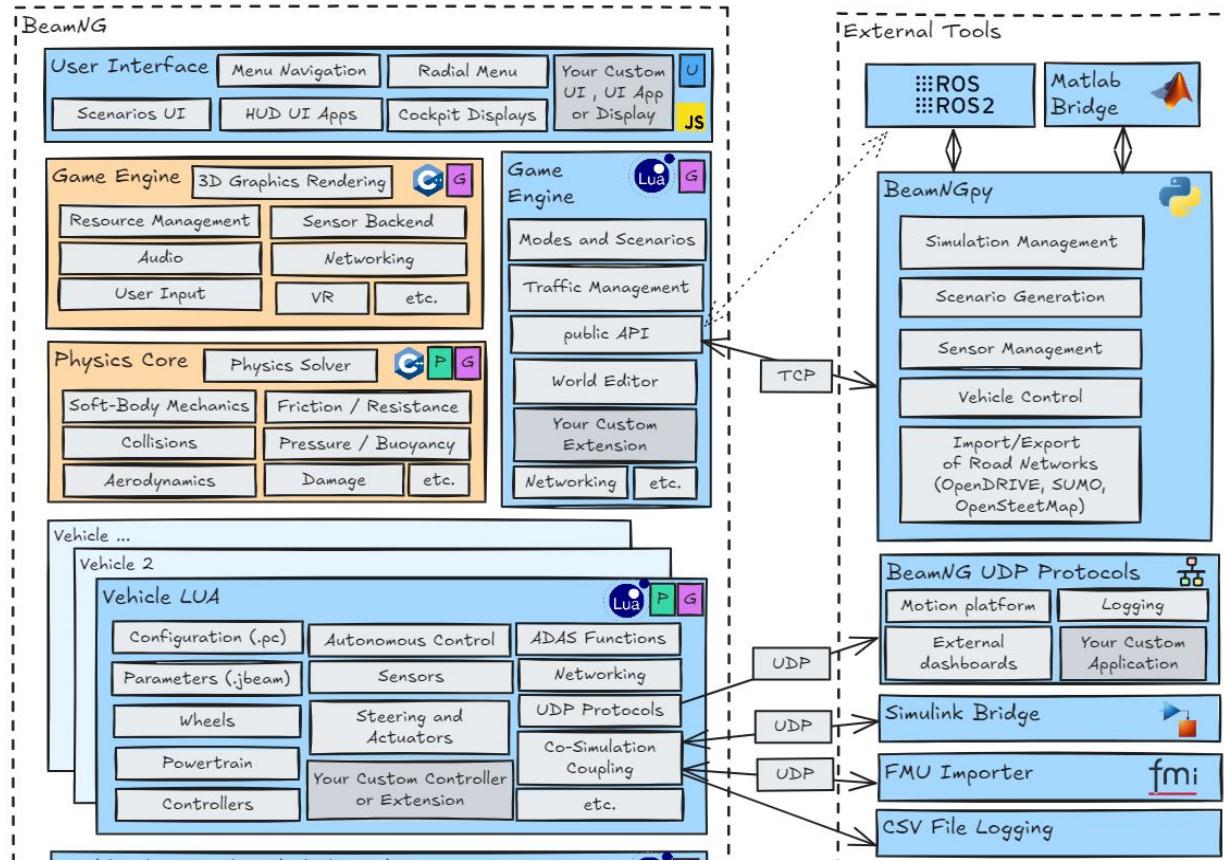
<https://beamng.tech/research/>

Architecture Overview (Core Components)



Legend:
P: Open Source
G: Closed Source
U: GUI update rate

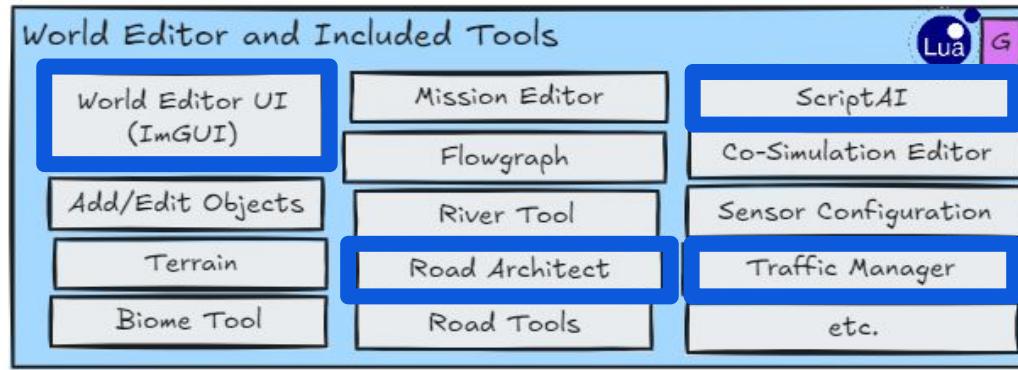
Architecture Overview (Extended Components)



Architecture Overview (Tooling)



Architecture Overview (Tooling)

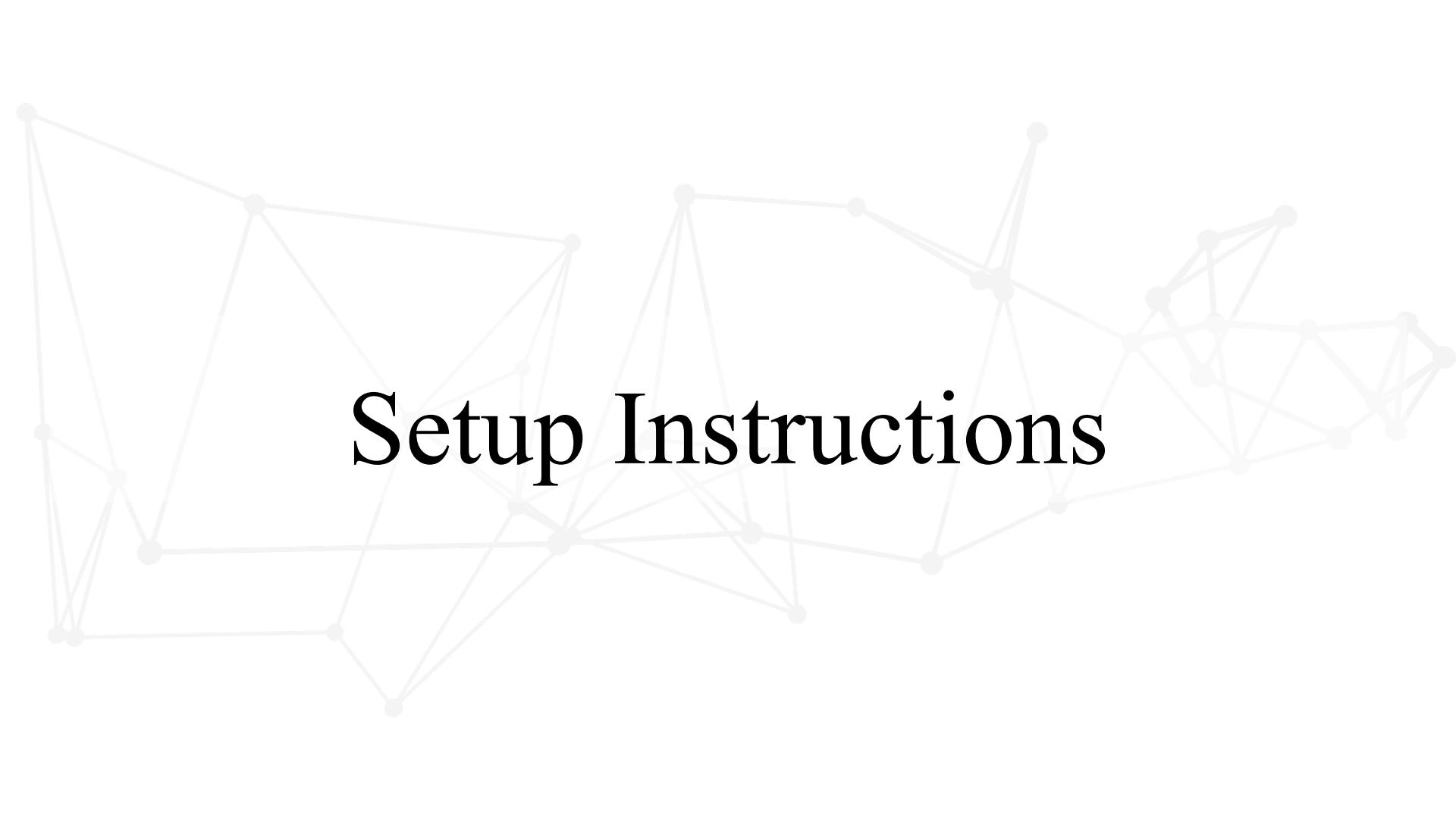


Our focus today

Basic Concepts

- BeamNG.tech implements and simulates/executes **scenarios**
- Scenarios take place in **levels** (or **maps**) that include a **terrain, roads, vegetation, props, and other vehicles**
- “Vehicles” is a broad term to identify everything that needs to be **simulated** using **soft-body** physics
- Various **sensors** can be attached to each vehicle
- A **camera** let us navigate the 3D environment (Use “C” to toggle)
- **Almost everything is configurable (using the appropriate tool)**

Setup Instructions



Download and Install BeamNG.tech

- Download the BeamNG.tech.zip and tech.key (license file) files from <https://nextcloud.beamng.com/s/KC4ztXbfmxsbF8/authenticate/showshare>
 - Password distribution via email.
- Or copy the package from the USB key
- **Note: Temporary License Key**
 - The tech.key you get for this tutorial will expire in **ten days**
 - You can apply for a non-commercial license at: <https://register.beamng.tech/>
- Extract all the files under **C:/BeamNG** or any other folder that **do not have a space in the path**

Download and Install BeamNG.tech

Extract the files in the BeamNG.tech.zip under **C:/BeamNG** or any other folder that **do not have a space in the path**

- Check that you have the same layout shown here

Name	Date modified	Type	Size
Bin64	3/31/2025 12:08 PM	File folder	
BinLinux	3/31/2025 12:08 PM	File folder	
content	3/31/2025 12:20 PM	File folder	
flowgraphEditor	3/31/2025 12:20 PM	File folder	
gameplay	3/31/2025 12:20 PM	File folder	
inspector	3/31/2025 12:20 PM	File folder	
locales	3/31/2025 12:20 PM	File folder	
lua	3/31/2025 12:21 PM	File folder	
renderer	3/31/2025 12:21 PM	File folder	
replays	3/31/2025 12:21 PM	File folder	
roadArchitect	3/31/2025 12:21 PM	File folder	
<input checked="" type="checkbox"/> settings	3/31/2025 12:21 PM	File folder	
shaders	3/31/2025 12:21 PM	File folder	
tech	3/31/2025 12:21 PM	File folder	
trackEditor	3/31/2025 12:21 PM	File folder	
ui	3/31/2025 12:22 PM	File folder	
EULA	3/31/2025 12:08 PM	Adobe Acrobat D...	175 KB
gameengine	3/31/2025 12:08 PM	Compressed (zipp...	149,553 KB
icon-beamng	3/31/2025 12:08 PM	Icon	36 KB
integrity	3/31/2025 12:08 PM	JSON Source File	1,722 KB
licenses	3/31/2025 12:08 PM	Textdokument	280 KB
PrivacyPolicy	3/31/2025 12:08 PM	Adobe Acrobat D...	173 KB
PrivacyPolicy-tech	3/31/2025 12:08 PM	Adobe Acrobat D...	114 KB
startup.default	3/31/2025 12:08 PM	Configuration sett...	1 KB
startup	3/31/2025 12:08 PM	Configuration sett...	1 KB
support	3/31/2025 12:08 PM	Application	4,551 KB
tech.key	3/31/2025 12:04 PM	KEY File	1 KB
thirdpartyFilter	3/31/2025 12:08 PM	Configuration sett...	1 KB
thirdpartyFilter_cef	3/31/2025 12:08 PM	Configuration sett...	1 KB

Download and Install BeamNG.tech

Extract the files in the BeamNG.tech.zip under **C:/BeamNG** or any other folder that **do not have a space in the path**

- Check that you have the same layout shown here



Name	Date modified	Type	Size
Bin64	3/31/2025 12:08 PM	File folder	
BinLinux	3/31/2025 12:08 PM	File folder	
content	3/31/2025 12:20 PM	File folder	
flowgraphEditor	3/31/2025 12:20 PM	File folder	
gameplay	3/31/2025 12:20 PM	File folder	
inspector	3/31/2025 12:20 PM	File folder	
locales	3/31/2025 12:20 PM	File folder	
lua	3/31/2025 12:21 PM	File folder	
renderer	3/31/2025 12:21 PM	File folder	
replays	3/31/2025 12:21 PM	File folder	
roadArchitect	3/31/2025 12:21 PM	File folder	
<input checked="" type="checkbox"/> settings	3/31/2025 12:21 PM	File folder	
shaders	3/31/2025 12:21 PM	File folder	
tech	3/31/2025 12:21 PM	File folder	
trackEditor	3/31/2025 12:21 PM	File folder	
ui	3/31/2025 12:22 PM	File folder	
EULA	3/31/2025 12:08 PM	Adobe Acrobat D...	175 KB
gameengine	3/31/2025 12:08 PM	Compressed (zipp...	149,553 KB
icon-beamng	3/31/2025 12:08 PM	Icon	36 KB
integrity	3/31/2025 12:08 PM	JSON Source File	1,722 KB
licenses	3/31/2025 12:08 PM	Textdokument	280 KB
PrivacyPolicy	3/31/2025 12:08 PM	Adobe Acrobat D...	173 KB
PrivacyPolicy-tech	3/31/2025 12:08 PM	Adobe Acrobat D...	114 KB
startup.default	3/31/2025 12:08 PM	Configuration sett...	1 KB
startup	3/31/2025 12:08 PM	Configuration sett...	1 KB
support	3/31/2025 12:08 PM	Application	4,551 KB
tech.key	3/31/2025 12:08 PM	KEY File	1 KB
thirdpartyFilter	3/31/2025 12:08 PM	Configuration sett...	1 KB
thirdpartyFilter_cef	3/31/2025 12:08 PM	Configuration sett...	1 KB

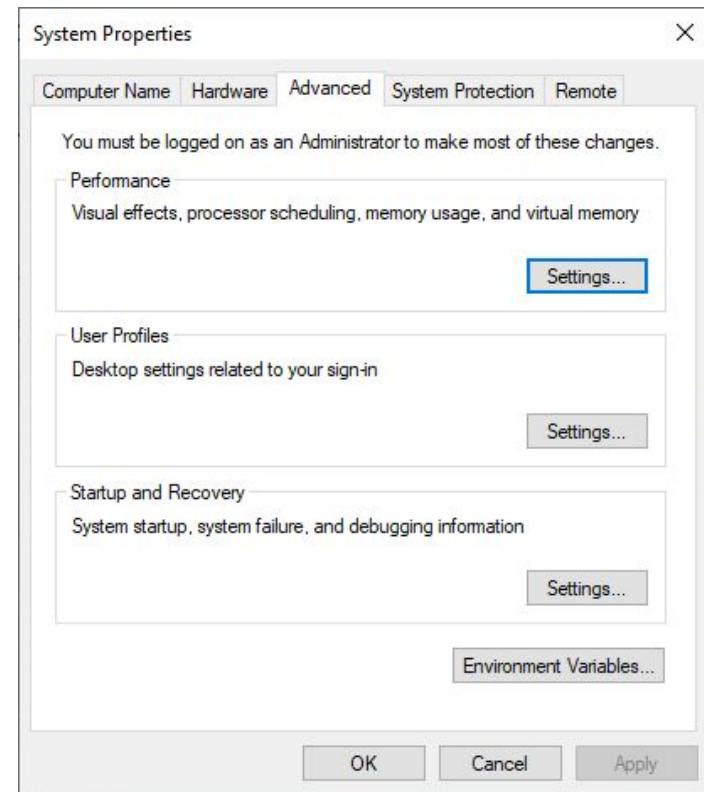
Configure Your Windows Environment

From start type: environments

Select “Edit the system environment variables”

Click on “Environment Variables ...”

Set **BNG_HOME** as new env. variable



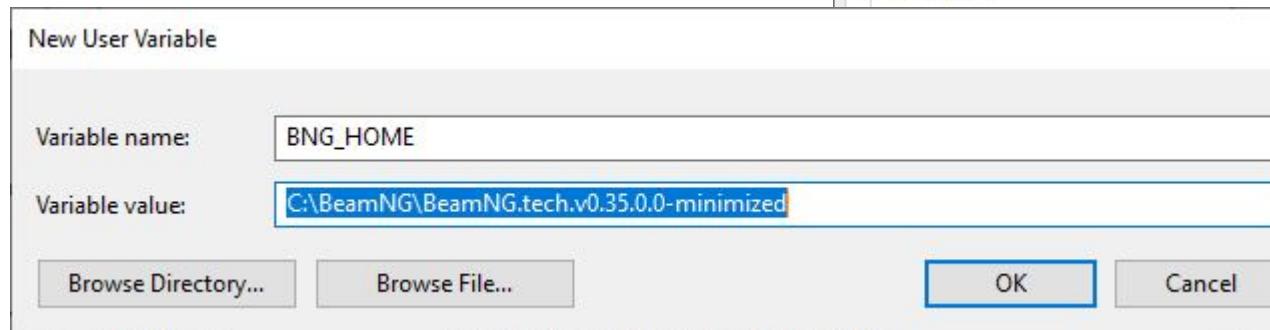
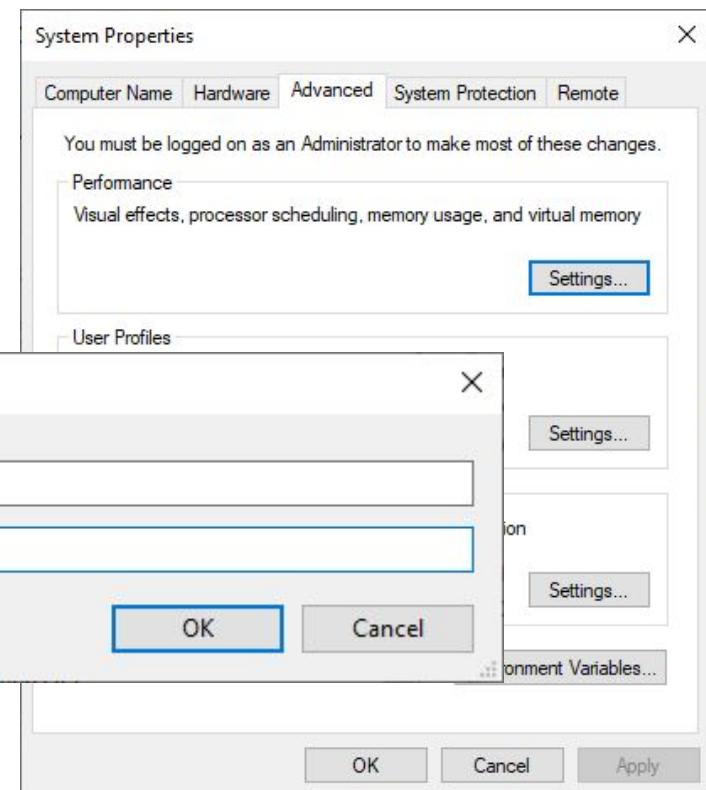
Configure Your Windows Environment

From start type: environments

Select “Edit the system environment variables”

Click on “Environment Variables ...”

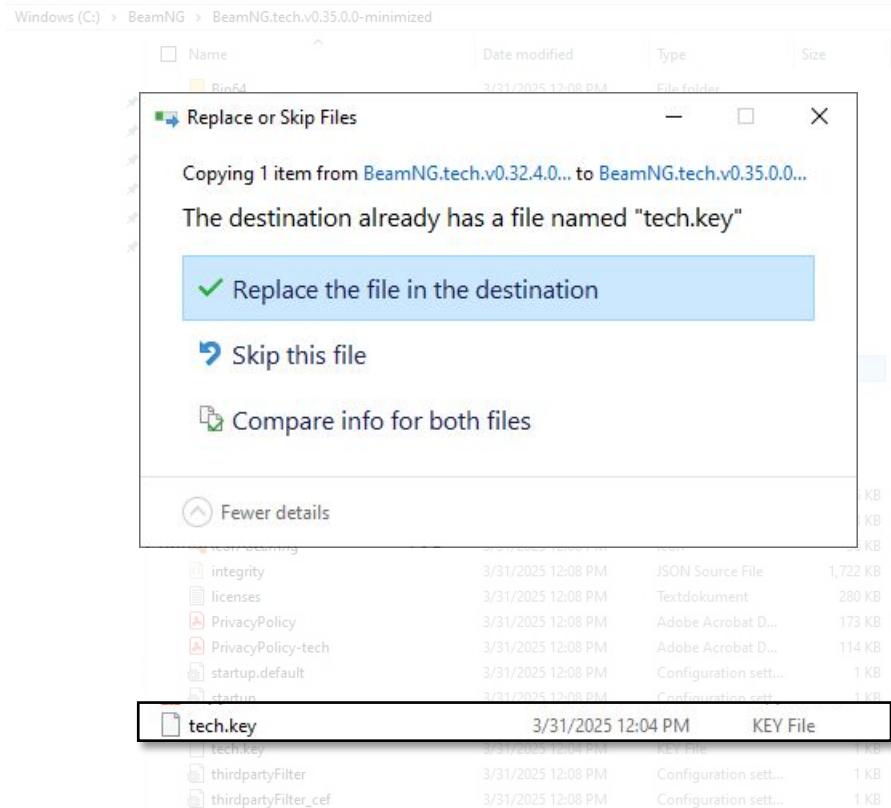
Set **BNG_HOME** as new env. variable



Download and Install BeamNG.tech

Extract the files in the BeamNG.tech.zip under **C:/BeamNG** or any other folder that do not have a space in the path

Replace the tech.key file (0Kb) with the one you downloaded before in the **BNG_HOME** folder



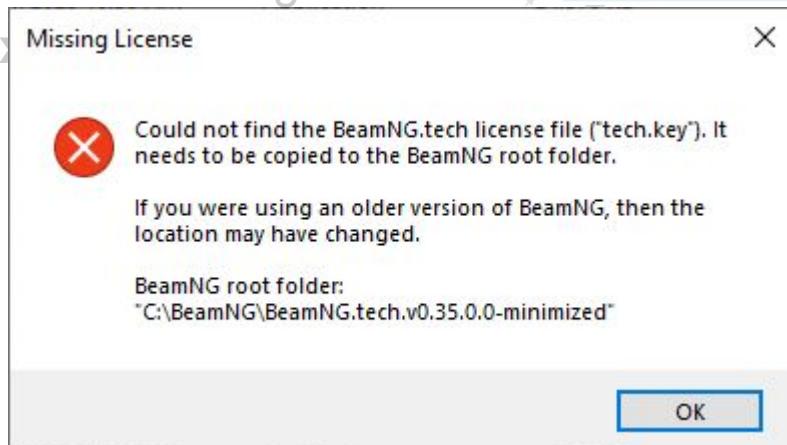
Start BeamNG.tech

Start the simulator by double-clicking
the file **BeamNG.tech.x64** under
BNG_HOME/Bin64

Name	Date modified	Type	Size
BeamNG.tech.x64	3/31/2025 10:30 AM	Application	32,292 KB
console.x64	3/31/2025 10:30 AM	Application	3,910 KB
CrashSender	3/31/2025 10:30 AM	Application	1,369 KB
perf_graph_viewer	3/31/2025 10:30 AM	Application	3,362 KB
VC_redist.x64	3/31/2025 10:30 AM	Application	24,578 KB
assimp.dll	3/31/2025 10:30 AM	Application exten...	6,055 KB
BeamNGBlendLayerDSP.dll	3/31/2025 10:30 AM	Application exten...	188 KB
chrome_elf.dll	3/31/2025 10:30 AM	Application exten...	1,395 KB
crash rpt.dll	3/31/2025 10:30 AM	Application exten...	242 KB
d3dcompiler_47.dll	3/31/2025 10:30 AM	Application exten...	4,802 KB
draco.dll	3/31/2025 10:30 AM	Application exten...	1,510 KB
dxcompiler.dll	3/31/2025 10:30 AM	Application exten...	20,596 KB
dxi.dll	3/31/2025 10:30 AM	Application exten...	1,474 KB
EOSSDK-Win64-Shipping.dll	3/31/2025 10:30 AM	Application exten...	17,839 KB
fmodDistanceFilter.dll	3/31/2025 10:30 AM	Application exten...	219 KB
fmodGain.dll	3/31/2025 10:30 AM	Application exten...	227 KB
libbeamng.x64.dll	3/31/2025 10:30 AM	Application exten...	1,584 KB
libcef.dll	3/31/2025 10:30 AM	Application exten...	220,237 KB
libEGL.dll	3/31/2025 10:30 AM	Application exten...	490 KB
libGLESv2.dll	3/31/2025 10:30 AM	Application exten...	8,085 KB
libtools.x64.dll	3/31/2025 10:30 AM	Application exten...	5,123 KB
logi_fmod.dll	3/31/2025 10:30 AM	Application exten...	133 KB

Start BeamNG.tech

Start the simulator by double-clicking
the file **BeamNG.tech.x64**
BNG_HOME/Bin64



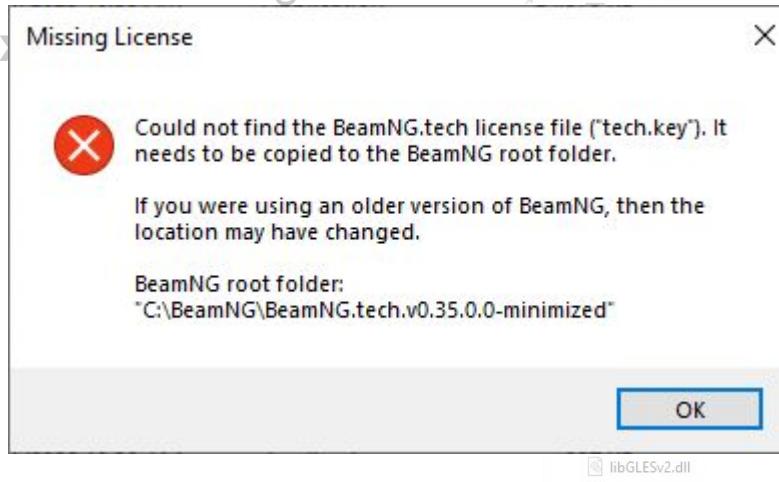
Windows (C:) > BeamNG > BeamNG.tech.v0.35.0.0-minimized > Bin64

	Name	Date modified	Type	Size
	<input checked="" type="checkbox"/> BeamNG.tech.x64	3/31/2025 10:30 AM	Application	32,292 KB
		3/31/2025 10:30 AM	Application	3,910 KB
		3/31/2025 10:30 AM	Application	1,369 KB
		3/31/2025 10:30 AM	Application	3,362 KB
		3/31/2025 10:30 AM	Application	24,578 KB
		3/31/2025 10:30 AM	Application exten...	6,055 KB
		3/31/2025 10:30 AM	Application exten...	188 KB
		3/31/2025 10:30 AM	Application exten...	1,395 KB
		3/31/2025 10:30 AM	Application exten...	242 KB
		3/31/2025 10:30 AM	Application exten...	4,802 KB
		3/31/2025 10:30 AM	Application exten...	1,510 KB
		3/31/2025 10:30 AM	Application exten...	20,596 KB
		3/31/2025 10:30 AM	Application exten...	1,474 KB
		3/31/2025 10:30 AM	Application exten...	17,839 KB
		3/31/2025 10:30 AM	Application exten...	219 KB
		3/31/2025 10:30 AM	Application exten...	227 KB
		3/31/2025 10:30 AM	Application exten...	1,584 KB
		3/31/2025 10:30 AM	Application exten...	220,237 KB
		3/31/2025 10:30 AM	Application exten...	490 KB
		3/31/2025 10:30 AM	Application exten...	8,085 KB
		3/31/2025 10:30 AM	Application exten...	5,123 KB
		3/31/2025 10:30 AM	Application exten...	133 KB

libGLESv2.dll
libtools.x64.dll
logi_fmod.dll

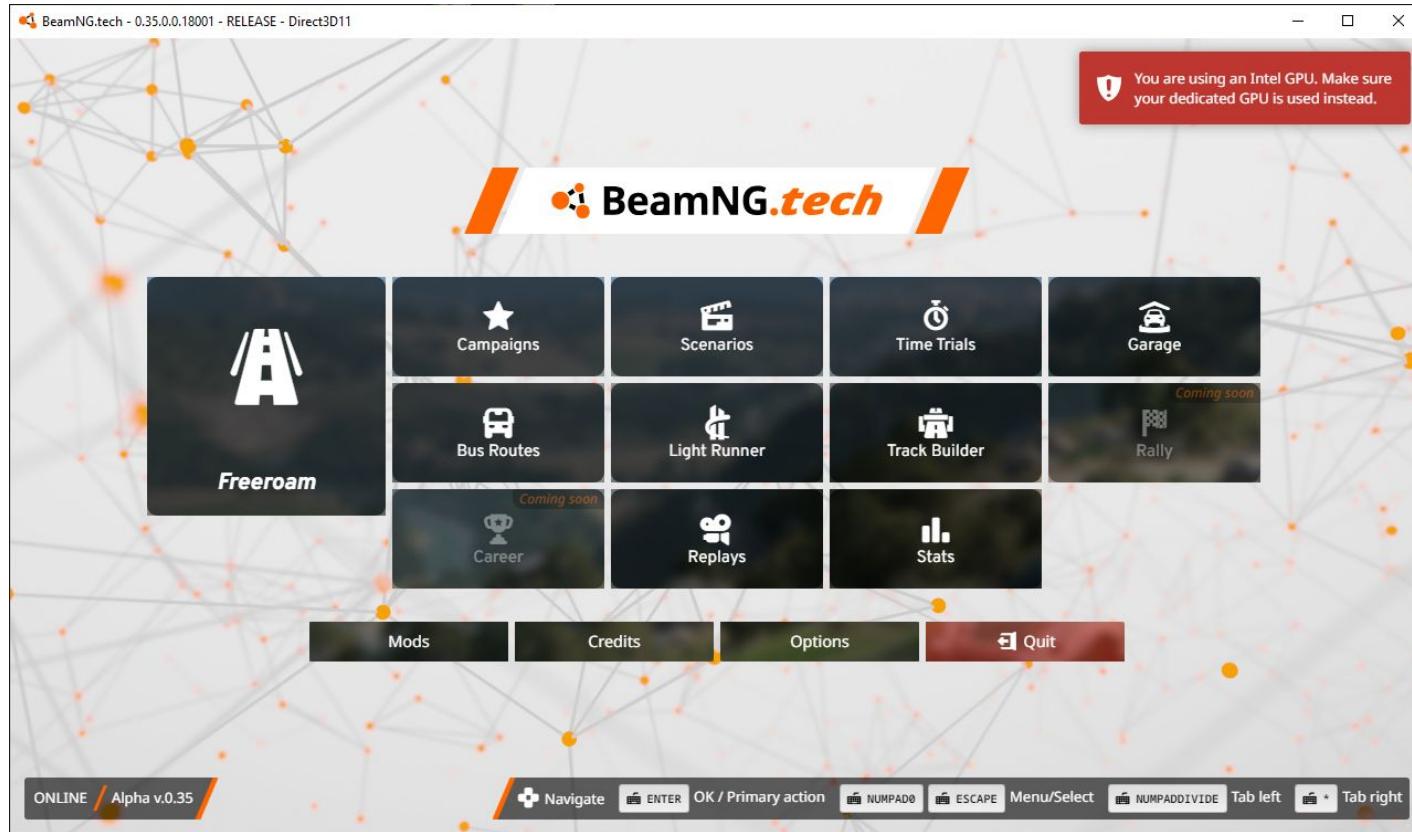
Start BeamNG.tech

Start the simulator by double-clicking
the file **BeamNG.tech.x64**
BNG_HOME/Bin64

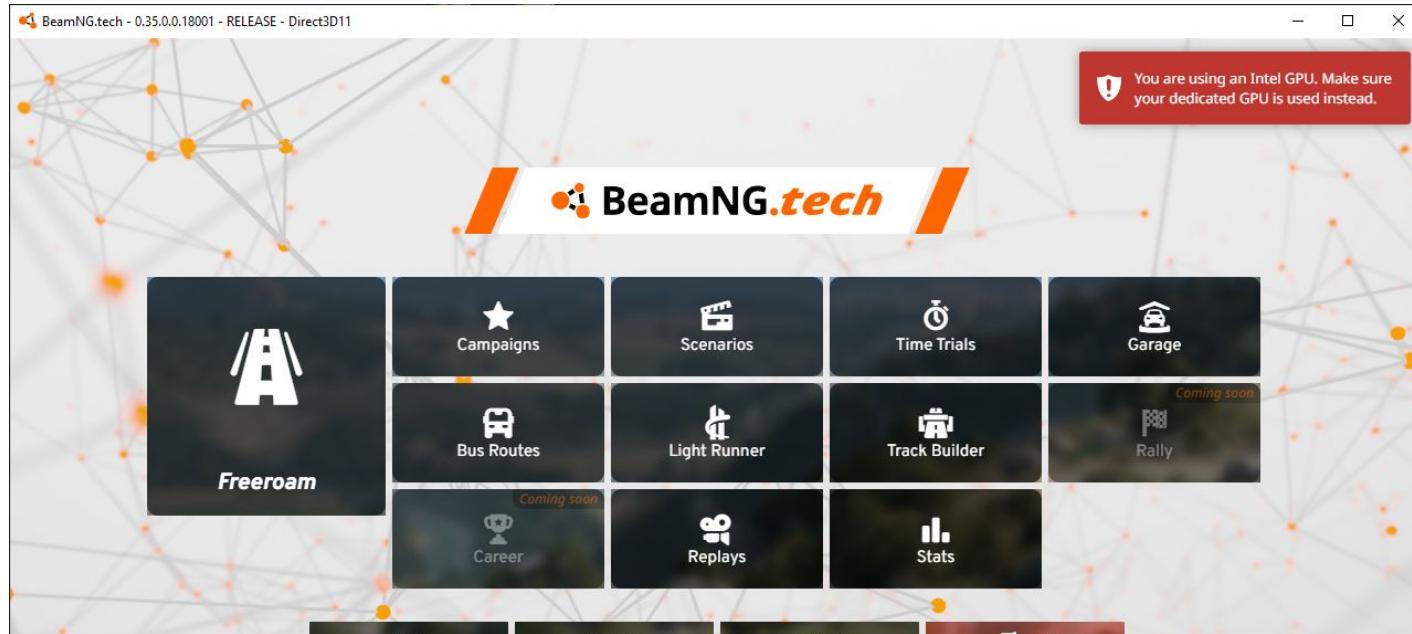


Check again if **tech.key** is in the correct folder:
BNG_HOME

Start BeamNG.tech



Start BeamNG.tech



Success !



Exercise: Load a Level and Navigate through it (Part 1)

- Start BeamNG.tech
- Load the **Automation Test Track** level (Freeroam)
- Choose the **Highway** spawn point



Exercise: Load a Level and Navigate through it (Part 2)

- Drive the vehicle (using the arrow keys) until past the junction ahead
- Toggle the camera to *driver mode* and *hood mode* while driving (press “c”)

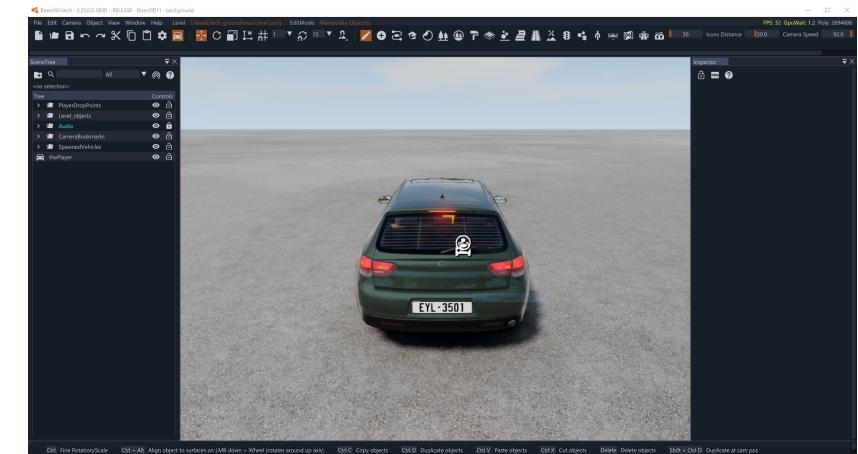


World Editor Tools



World Editor

- After loading a **level**, you can toggle the World Editor by pressing **F11**.

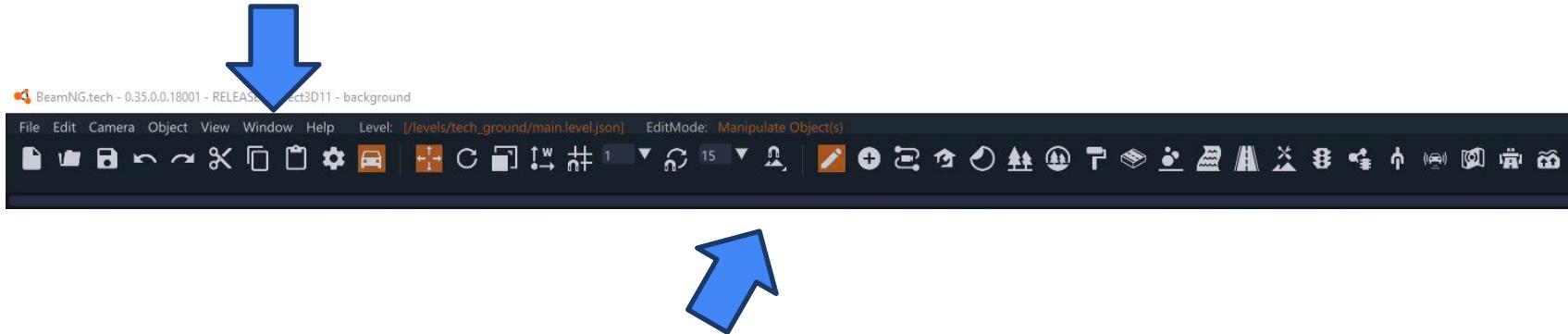


World Editor Tools

- After loading a **level**, you can toggle the World Editor by pressing **F11**.
- World Editor it's the GUI that allows you to **setup** the static and dynamic elements of scenario and most of the existing **tools** of BeamNG.tech
- Possible actions are:
 - **Scenario management:** save, load, store configurations, levels, maps, scenarios
 - **Inspection:** navigate the map, list the objects currently loaded, show their details
 - **Environment setup:** set time-of-day, configure weather
 - **Static element setup:** place/move/rotate/remove vehicles, props, and vegetation.
 - **Dynamic element setup:** drive vehicles, record/replay trajectories, etc.
 - **Sensor setup:** place/move/rotate/remove and configure sensors (cameras, LiDARs, etc.)

World Editor Tools

- Tools can be activated from the **button bar** or from the **Window** menu



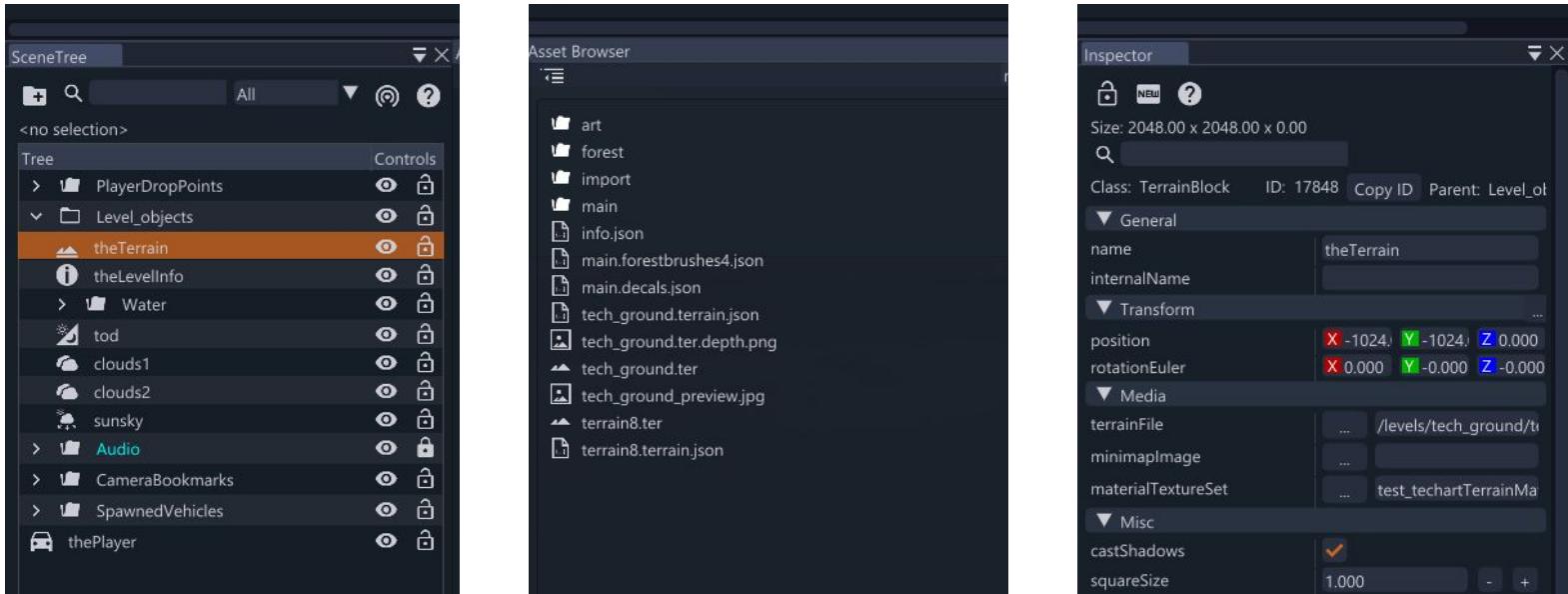
World Editor Tools

- Today we focus on a subset of the tools (but feel free to explore the others!)



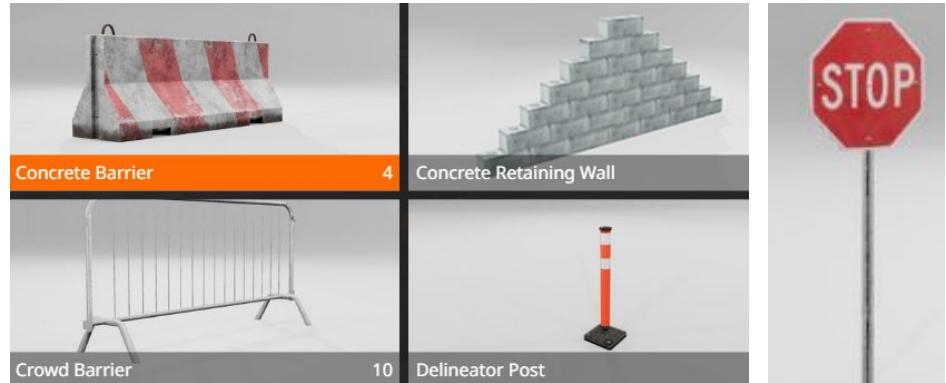
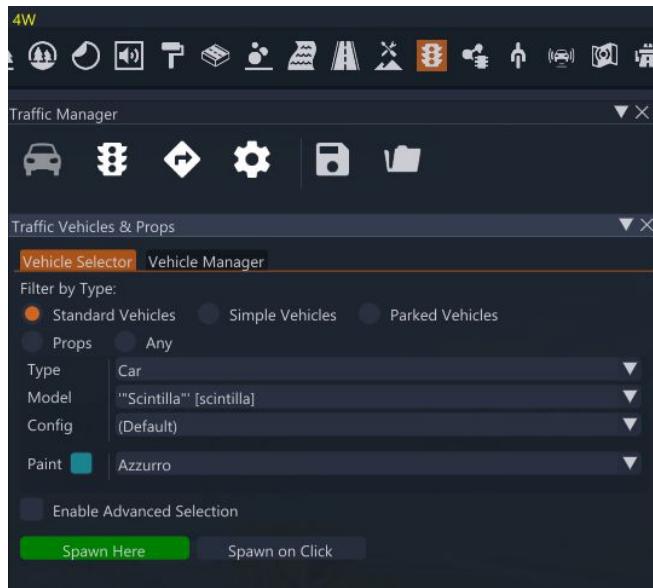
- Tools for inspection (Scene Tree, Asset Browser, Inspector)
- Traffic Manager
- ScriptAI Editor
- Sensor configurations for vehicles and maps
- Road Architect

World Editor Tools: Tools for Inspection



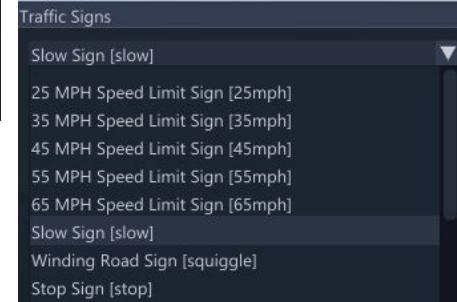
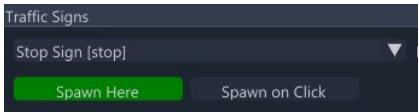
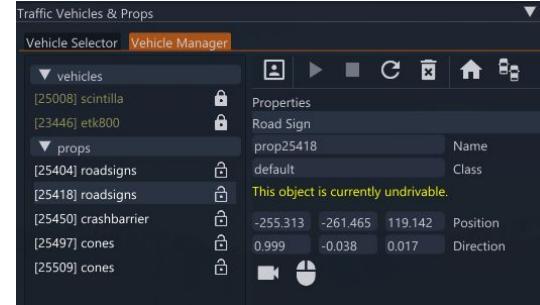
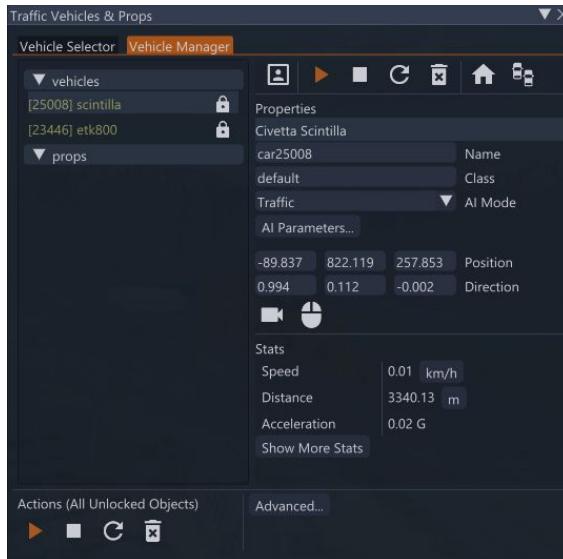
- Show the content loaded in the scenario, all the available content, and details of a specific element

World Editor Tools: Traffic Manager



- Place vehicles and objects in the map
- Select simulation accuracy for vehicles (regular, simplified, parked)

World Editor Tools: Traffic Manager (cont.)



- Inspect and modify objects properties (position, rotation, etc.)
- Configure automated driving*

*some driving modes are not available on BeamNG.tech original maps

Exercise: Spawn and Control a Vehicle (optional)

- Start BeamNG.tech, load Test Automation Track, select Highway spawn points
- Open Traffic Manager, spawn vehicles on the adjacent or the same lane, spawn props, then visualize all the vehicles (advanced option)
- Set the ego vehicle to (-285, -257), save the Home position of the vehicle
- Edit the position and direction of the other vehicles and save the Home position
- Assign driving mode to the vehicles
- Run the simulation



File Edit Camera Object View Window Help Level: /levels/automation_test_track/main.level.json* EditMode: Manipulate Object(s)



Ctrl Fine Rotation/Scale

Ctrl + Alt Align object to surfaces on LMB down + Wheel (rotates around up axis)

Ctrl C Copy objects

Ctrl D Duplicate objects

Ctrl V Paste objects

Ctrl X Cut objects

Delete Delete objects

Shift + Ctrl D Duplicate at cam pos

File Edit Camera Object View Window Help Level: /levels/automation_test_track/main.level.json EditMode: Traffic Manager



World Editor Tools: Script AI Editor



- Configure NPC movement by “**attaching**” trajectories to them
- Draw** trajectories, **record/replay** trajectories, **store/load/mutate** trajectories
- Control one vehicle at the time (incremental setup) and all together (play)
- Multiple** vehicle - to - vehicle interactions: Scene vehicles connected with scene trajectories, Recording / replay
- Trajectories are saved in the userfolder, at the software version folder
(C:\Users\User\AppData\Local\BeamNG.drive\0.35)

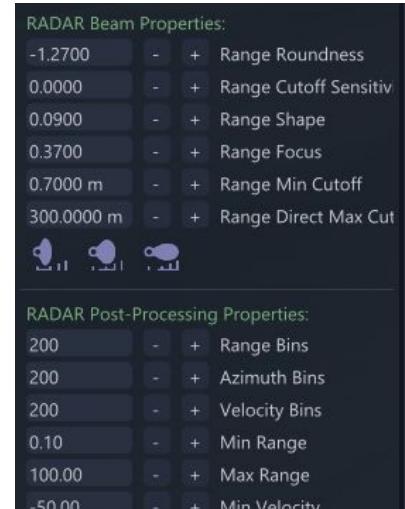
Exercise: Drive Vehicles Using Scripted Trajectories

- Spawn two vehicles on Automation track map, Highway, e.g., (-273.4, -260.4) & (-283, -258)
- Record a trajectory of a stationary vehicle which enters the lane
- Record a trajectory of a vehicle moving on its lane
- Run the simulation



Combination of Traffic Manager & ScriptAI

World Editor Tools: Vehicle's Sensors



- Choose, configure, and attach sensors to vehicles
- More than 16 types of sensors available (camera, lidar, damage, radar, etc.)

World Editor Tools: Lists of Sensors

- ‘Classical’ (Ground Truth) sensors
 - Physical State
 - Electrics
 - Timer
 - Damage
 - Gforces
 - Annotated Images
- ‘Automated’ sensors:
 - Mesh Info Sensor
 - Advanced IMU
 - Camera
 - Powertrain
 - RADAR
 - Lidar
 - Ultrasonic
 - Ideal RADAR
 - Roads sensor
 - GPS
 - Inertial Measurement Units

Exercise: Attach Sensors to a Vehicle (optional)

Exercise

1. Launch the Vehicle sensors configuration tool
2. Select one among the spawned vehicles
3. Attach sensors on the vehicle; such as a LiDAR 360 deg, ultrasonics and a RADAR
4. Edit the sensors parameters
5. Save the configuration, as 'vconfig1.json'

Coffee Brake!

To those who need help for setting up the suite and Python:
Visit the tutors during the break

World Editor Tools: Map/Infrastructure Sensors



- Camera, Radar, LiDAR and Ultrasonic sensors can be deployed on the map, thus simulating smart infrastructures and traffic monitoring solutions
- Their working is the same as vehicle's sensors

Exercise: Attach sensors to vehicle and Infrastructure (Optional)

Upon level option and vehicle spawning :

2. Launch the Map sensors configuration tool
3. Select and attach sensors on the map
4. Edit the sensors parameters
5. Save the configuration
6. Launch the Vehicle sensors configuration tool
7. Select and attach sensors on the vehicle
8. Edit the sensors parameters
9. Save the configuration

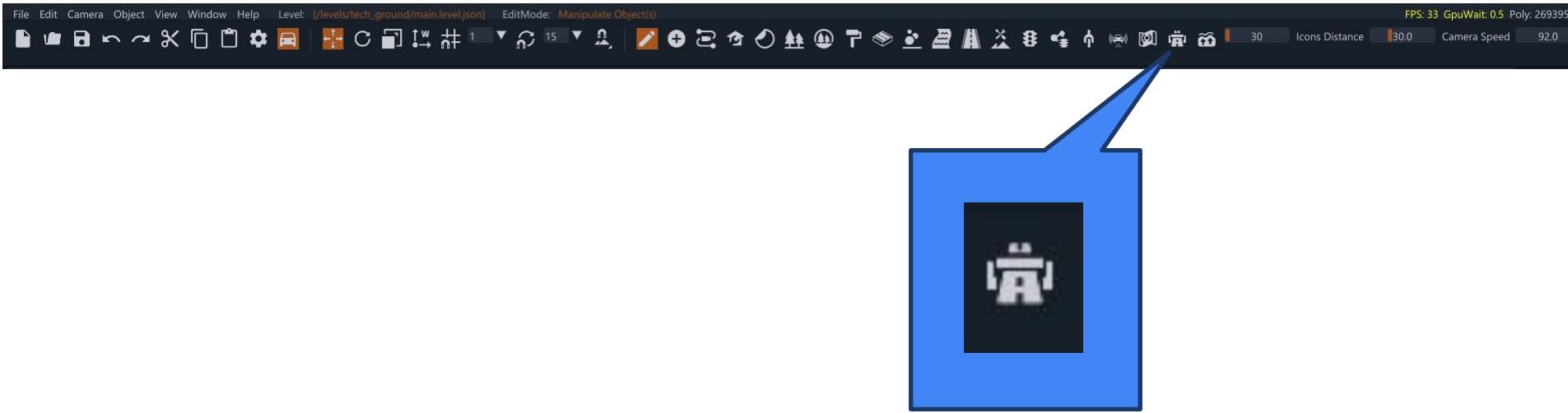
The Road Architect



The Road Architect



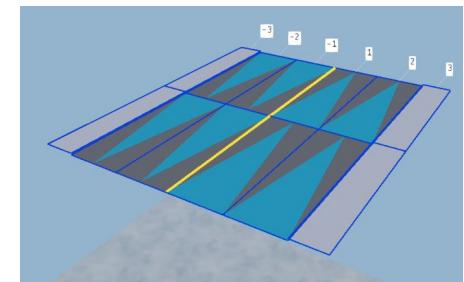
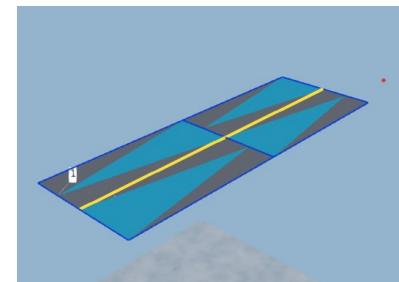
The Road Architect





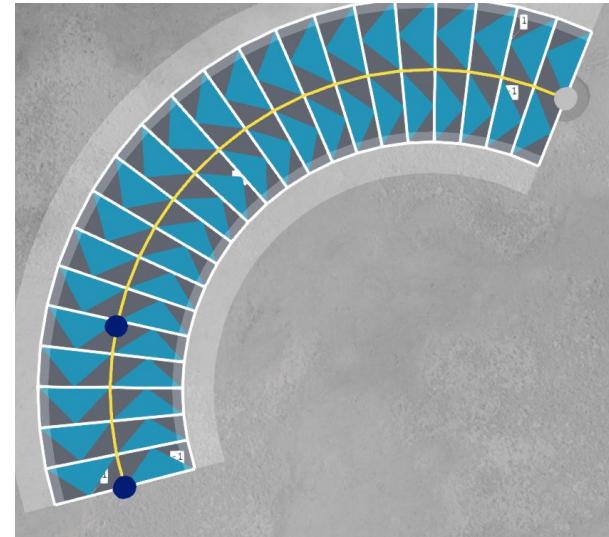
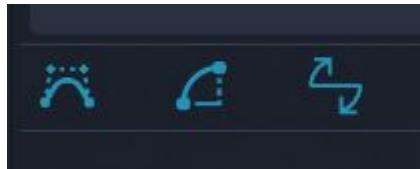
The Road Architect

- Allows users to do terraforming (sculpting the terrain), import roads (e.g., OpenDrive), draw and mutate roads, join roads, and import complex elements (junction, roundabouts, etc.)
- Allows users to assign **profiles** to road segments which include pavements, materials, lamps, etc.
- Road can be saved to level or scenario and reloaded



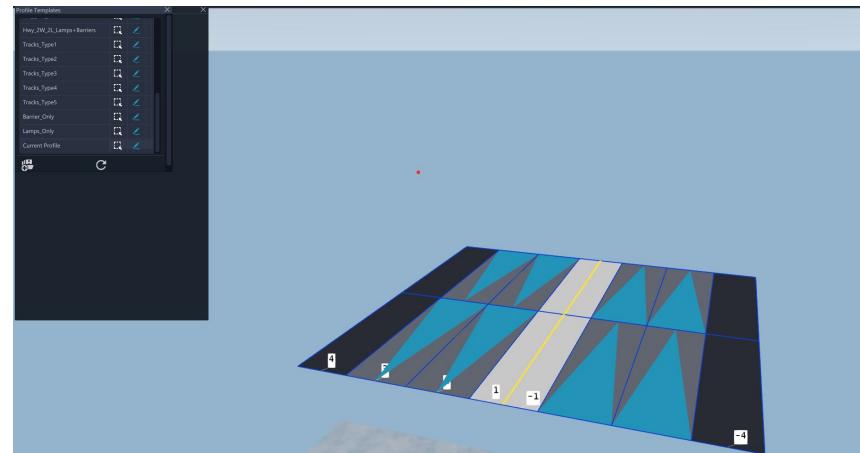
Simple Roads

- Simple roads can be drawn on the map by placing **road points**
- The way BeamNG.tech interpolates the road points is defined by the road type (arcs, splints, clothoids)

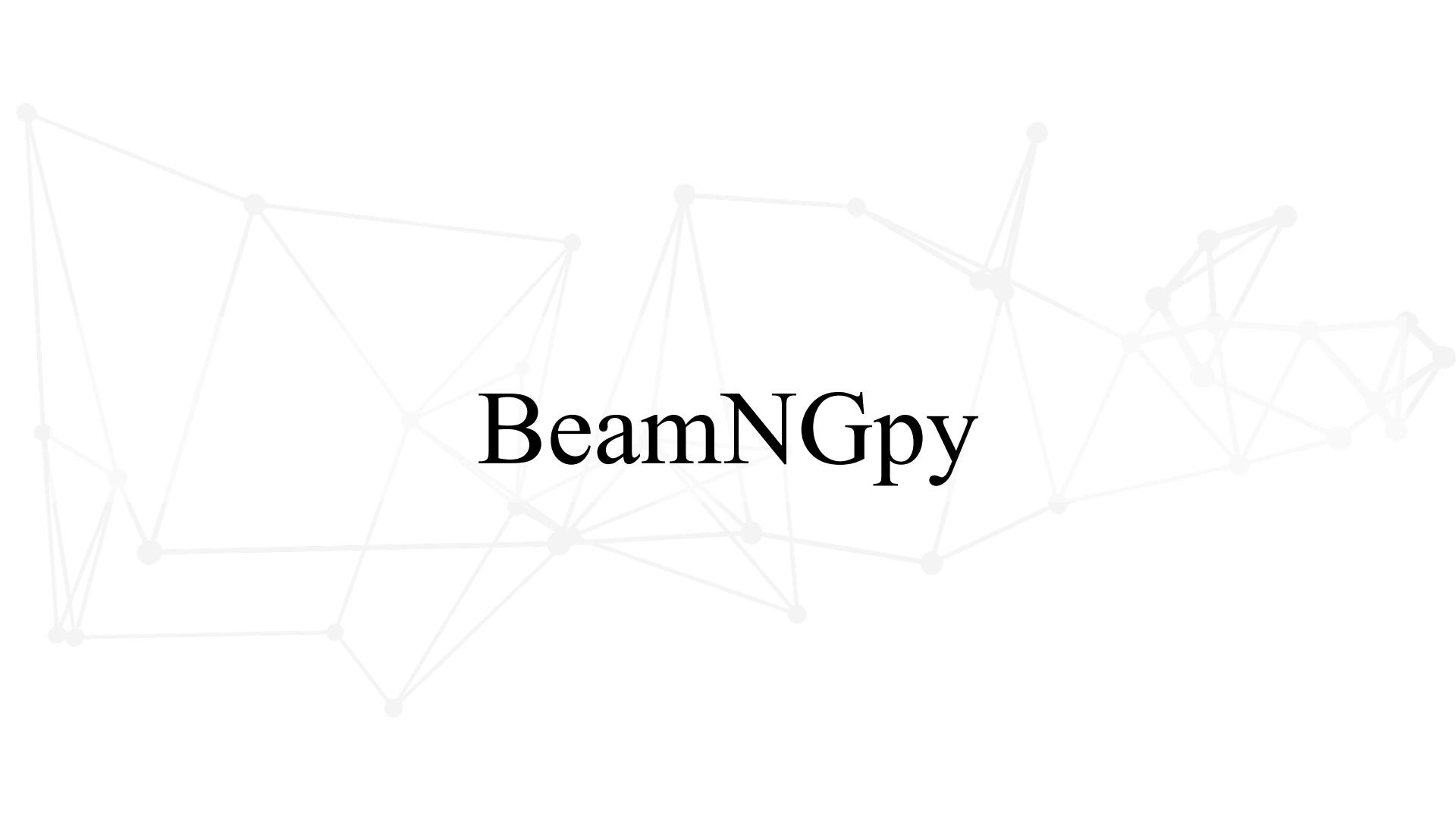


Exercise: Draw Your First Road

- Select the Tech_ground map
- Launch the Road Architect tool
- Select Cubic spline type
- Draw a highway
- Optional : combine with a junction type
- Optional : use a bridge
- Optional : add overlays
- Save the session



BeamNGpy



BeamNGpy Overview

A set of Python APIs to control
BeamNG.tech

Makes BeamNG.tech a client-server
system

Open source (on GitHub) and
available on pypi.org

The image shows two screenshots of the BeamNGpy project. The top screenshot is the GitHub repository page for 'BeamNGpy' (Public), showing a list of recent commits from 'aivora-beaming'. The bottom screenshot is the PyPI project page for 'beamngpy 1.31', which is the latest version released on Dec 12, 2024. The PyPI page includes sections for navigation (Project description, Release history, Download files), project description (BeamNGpy), documentation, verified details, maintainers, and a table of contents.

BeamNGpy Public

master 3 Branches 55 Tags

aivora-beaming Update AUTHORS.rst 2416441 · 4 months ago 907 Commits

debug Fix beamngpyDissector 4 months ago

docs Merge branch '0.33' 6 months ago

examples Set drive_in_lane where appropriate 4 months ago

media Add Wireshark debug plugin 7 months ago

beamngpy 1.31

Latest version

Released: Dec 12, 2024

pip install beamngpy

Python API to interact with BeamNG.tech.

Navigation

- Project description
- Release history
- Download files

Project description

BeamNGpy

Documentation

BeamNGpy is an official library providing a Python API to [BeamNG.tech](#), the academia- and industry-oriented fork of the video game [BeamNG.drive](#). BeamNGpy and BeamNG.tech are designed to go hand in hand, both being kept up to date to support each other's functions, meaning using the latest versions of both is recommended.

It allows remote control of the simulation, including vehicles contained in it. See [Features](#) or go through the [Feature Overview](#) Jupyter notebook.

Verified details

These details have been [verified by PyPi](#)

Maintainers

 aivora

Table of Contents

Download and Install Python and BeamNGpy

Download the installer of Python 3.11 from the official repository and install it under C:/Python/Python311 (use Custom Installation to select the folder)

- <https://www.python.org/downloads/release/python-3110/>

Clone or download the **icst-2025 branch** from the BeamNGpy repository

- <https://github.com/BeamNG/BeamNGpy>

Clone or download the **main branch** from the ICST Tutorial repository

- <https://github.com/BeamNG/scenario-based-testing-tutorial>

Setup a virtual environment (using Visual Studio Code)

Create the folder **C:/BeamNG/icst2025** (or anything without no spaces in path)
extract BeamNGpy and the tutorial repo under that

Windows (C:) > BeamNG > icst2025			
Name	Date modified	Type	
BeamNGpy-icst-2025	3/31/2025 1:59 PM	File folder	
scenario-based-testing-tutorial-main	3/31/2025 4:45 PM	File folder	

Windows (C:) > BeamNG > icst2025 > BeamNGpy-icst-2025			
Name	Date modified	Type	Size
debug	3/31/2025 1:59 PM	File folder	
docs	3/31/2025 1:59 PM	File folder	
examples	3/31/2025 1:59 PM	File folder	
media	3/31/2025 1:59 PM	File folder	
<input checked="" type="checkbox"/> src	3/31/2025 1:59 PM	File folder	
tests	3/31/2025 1:59 PM	File folder	
.coveragerc	3/31/2025 1:59 PM	COVERAGERC File	1 KB
.gitignore	3/31/2025 1:59 PM	Git Ignore Source ...	2 KB
.gitlab-ci	3/31/2025 1:59 PM	Yaml Source File	1 KB
readthedocs	3/31/2025 1:59 PM	Yaml Source File	1 KB
AUTHORS	3/31/2025 1:59 PM	Restructured Text ...	1 KB

Windows (C:) > BeamNG > icst2025 > scenario-based-testing-tutorial-main				
Name	Date modified	Type	Size	
examples	3/31/2025 4:45 PM	File folder		
ICST25_Hands_on_Training	3/31/2025 4:45 PM	Adobe Acrobat D...	1,428 KB	
PC_Cheat_Sheet	3/31/2025 4:45 PM	Adobe Acrobat D...	62 KB	
RoadsArchitectTool_demo	3/31/2025 4:45 PM	MP4 Video	21,974 KB	

Setup a virtual environment (using Visual Studio Code)

Open VSC, then Open Folder, select **C:/BeamNG/icst2025**

Open the terminal tab: **C:\Python\Python311\python.exe -m venv .venv**

Activate the virtual environment: **..\.venv\Scripts\activate**

Check the Python version: **python.exe -V**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - BeamNGpy-icst-2025 ▲ + ×

PS C:\BeamNG\icst2025> python.exe -V
python.exe : The term 'python.exe' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ python.exe -V
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (python.exe:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

● PS C:\BeamNG\icst2025> C:
● PS C:\BeamNG\icst2025> C:\Python\Python311\python.exe -m venv .venv
● PS C:\BeamNG\icst2025> ..\venv\Scripts\activate
● (.venv) PS C:\BeamNG\icst2025> python.exe -V
Python 3.11.8
```

Install BeamNGpy from source (using Visual Studio Code)

With the active virtual environment

Update pip: **python.exe -m pip install --upgrade pip**

Change directory to BeamNGpy-icst-2025: **cd BeamNGpy-icst-2025**

Install BeamNGpy from source: **pip install .**

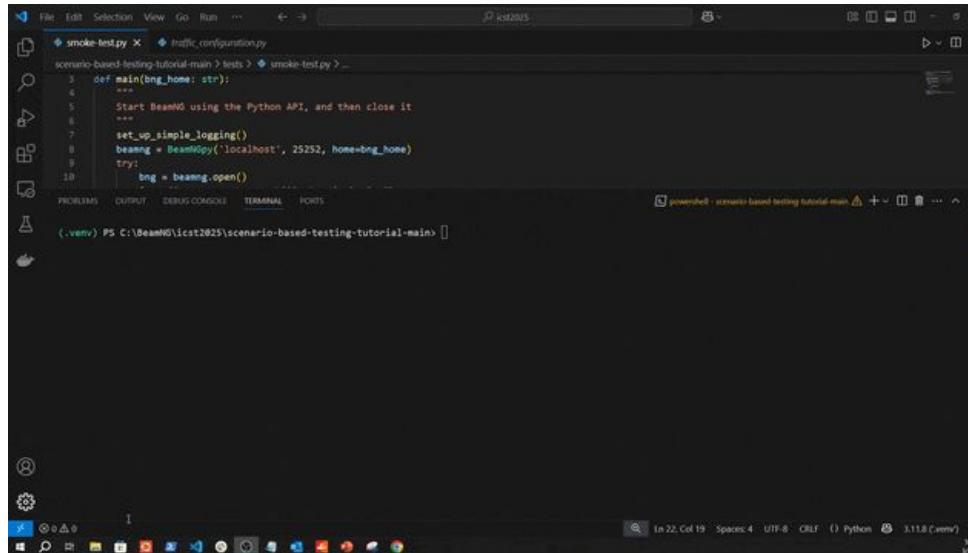
```
● (.venv) PS C:\BeamNG\icst2025> cd ..\BeamNGpy-icst-2025\
● (.venv) PS C:\BeamNG\icst2025\BeamNGpy-icst-2025> pwd
Path
-----
C:\BeamNG\icst2025\BeamNGpy-icst-2025

● (.venv) PS C:\BeamNG\icst2025\BeamNGpy-icst-2025> pip install .
Processing c:\beamng\icst2025\beamngpy-icst-2025
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Collecting numpy (from beamngpy==1.31+icst2025)
    Downloading numpy-2.2.4-cp311-cp311-win_amd64.whl.metadata (60 kB)
        60.8/60.8 kB 544.1 kB/s eta 0:00:00
Collecting msgpack (from beamngpy==1.31+icst2025)
    Downloading msgpack-1.1.0-cp311-cp311-win_amd64.whl.metadata (8.6 kB)
Collecting Pillow (from beamngpy==1.31+icst2025)
    Downloading pillow-11.1.0-cp311-cp311-win_amd64.whl.metadata (9.3 kB)

Successfully built beamngpy
Installing collected packages: pytz, tzdata, six, pyparsing, Pillow, packaging, numpy, msgpack, MarkupSafe, kiwisolver, fonttools, cycler, scipy, python-dateutil, Jinja2, contourpy, pandas, matplotlib, seaborn, beamngpy
Successfully installed Jinja2-3.1.6 MarkupSafe-3.0.2 Pillow-11.1.0 beamngpy-1.31+icst2025 contourpy-1.3.1 cycler-0.12.1 fonttools-4.56.0 kiwisolver-1.4.8 matplotlib-3.10.1 msgpack-1.1.0 numpy-2.2.4 packaging-24.2 pandas-2.2.3 pyparsing-3.2.3 python-dateutil-2.9.0.post0 pytz-2025.2 scipy-1.15.2 seaborn-0.13.2 six-1.17.0 tzdata-2025.2
```

Test the installation (using Visual Studio Code)

With the active environment, go to the
C:/BeamNG/icst2025/scenario-based-testing-tutorial-main folder and run
python.exe tests/smoke-test.py



```
File Edit Selection View Go Run ... < > icst2025
smoke-test.py * traffic_configuration.py
scenario-based-testing-tutorial-main > tests > smoke-test.py ...
3 def main(bng_home: str):
4     """
5         Start BeamNG using the Python API, and then close it
6     """
7     set_up_simple_logging()
8     beamng = BeamNGpy('localhost', 25252, home=bng_home)
9     try:
10         bng = beamng.open()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(.venv) PS C:\BeamNG\icst2025\scenario-based-testing-tutorial-main>
```



Putting All Together

Automated Execution of Scenarios

We created some scenarios featuring maps, vehicles, obstacles, sensors **manually**, and **manually** we can execute them from the World Editor

However, thanks to **BeamNGpy** we can also execute them **automatically**

The general approach is to implement Python scripts that:

- Start BeamNG
- Load the scenario
- Load all the needed configurations (sensors, vehicles, etc.)
- Start the execution
- Collect sensor data
- Evaluate assertions (goal reached, collision, etc.)

BeamNGpy examples

feature_overview.ipynb

road_definition.py

vehicle_logging.py

west_coast_IMU.py

scenario_control.ipynb

```
1  from time import sleep
2
3  from beamngpy import BeamNGpy, Scenario, Vehicle, set_up_simple_logging
4  from beamngpy.sensors import Mesh
5
6
7  def main():
8      set_up_simple_logging()
9
10     # Start up the simulator.
11     bng = BeamNGpy("localhost", 25252)
12     bng.open(launch=True)
13     # Create a vehicle.
14     vehicle = Vehicle("ego_vehicle", model="etki", licence="PYTHON", color="Red")
15     # Create a scenario.
16     scenario = Scenario("industrial", "mesh_test", description="Mesh analysis")
17     # Add the vehicle to the scenario.
18     scenario.add_vehicle(vehicle, pos=(408.36, -336.56, 35.54), rot_quat=(0, 0, 0.99, -0.17))
19     scenario.make(bng)
20     # Set simulator to 60hz temporal resolution
21     bng.settings.set_deterministic(60)
22     bng.scenario.load(scenario)
23     bng.ui.hide_hud()
24     bng.scenario.start()
25
26     # Get the mesh data for the vehicle.
27     mesh = Mesh("mesh1", bng, vehicle)
28
29     print("Driving around, polling the mesh data every 5 seconds...")
30     vehicle.ai.set_mode("traffic")
31     for _ in range(3):
32         sleep(5)
33         data = mesh.poll()
```

Python API exercise (1/2)

On traffic control and sensors configuration

- Preliminary : setup traffic and sensors configuration with the World Editor
- Launch Python API
- Use the Traffic API with the to control traffic
- Use the Vehicle sensors API to collect sensors data
- Run the scenario

```
from beamngpy import BeamNGpy, set_up_simple_logging
from beamngpy.sensors import VehicleSensorConfig
from beamngpy.tools import TrafficConfig

SCRIPT_DIR = Path(__file__).parent.resolve()

def copy_traffic_config_files(dest_dir: Path):
    files = [
        "data/traffic_config_example.json",
        "data/lanechange2.json",
        "data/veh_sensors2.json",
    ]
    dest_dir.mkdir(parents=True, exist_ok=True)
    for file in files:
        shutil.copy(SCRIPT_DIR / file, dest_dir)

def main():
    set_up_simple_logging()
    bng = BeamNGpy("localhost", 25252)
    bng = bng.open()

    # we need to copy the configuration files to the user path
    destination_path = Path(bng.user_with_version)
    copy_traffic_config_files(destination_path)
    print(f"Copied traffic configuration to {destination_path}.")
    # Change version number (0.35) to X.XX if it is different from the current one.
    traffic = TrafficConfig(bng, "0.35/traffic_config_example.json")
    # With TrafficConfig.vehicles[vehicle name set from traffic configuration] you can access the related Vehicle object
    veh_sensors = VehicleSensorConfig(
        "vehicle_sensors",
        bng,
        traffic.vehicles["thePlayer"],
        "/veh_sensors2.json",
    )
}
```

Python API exercise (2/2)

On Road networks and sensors configuration

- Launch Tech_ground map
- Introduce a road network (make or import)
- Drive a vehicle and record the trajectory
- Setup map sensors and vehicle sensors configuration
- Save the level
- On Python API, control the vehicle and the sensors configuration
- Run the scenario

```
from beamngpy import BeamNGpy, Scenario, Vehicle, set_up_simple_logging
from beamngpy.sensors import VehicleSensorConfig
from beamngpy.sensors import MapSensorConfig

import json

# Load the JSON file
SCRIPT_DIR = Path(__file__).parent.resolve()

folder_path = str(SCRIPT_DIR)

# Copy the entire folder and its contents
def copy_level_folder(dest_dir: Path):
    file = "data/tech_ground"
    if dest_dir.exists():
        shutil.rmtree(dest_dir)

    # Copy the entire source folder to the destination
    #dest_dir.mkdir(parents=True, exist_ok=True)
    shutil.copytree(SCRIPT_DIR / file, dest_dir)

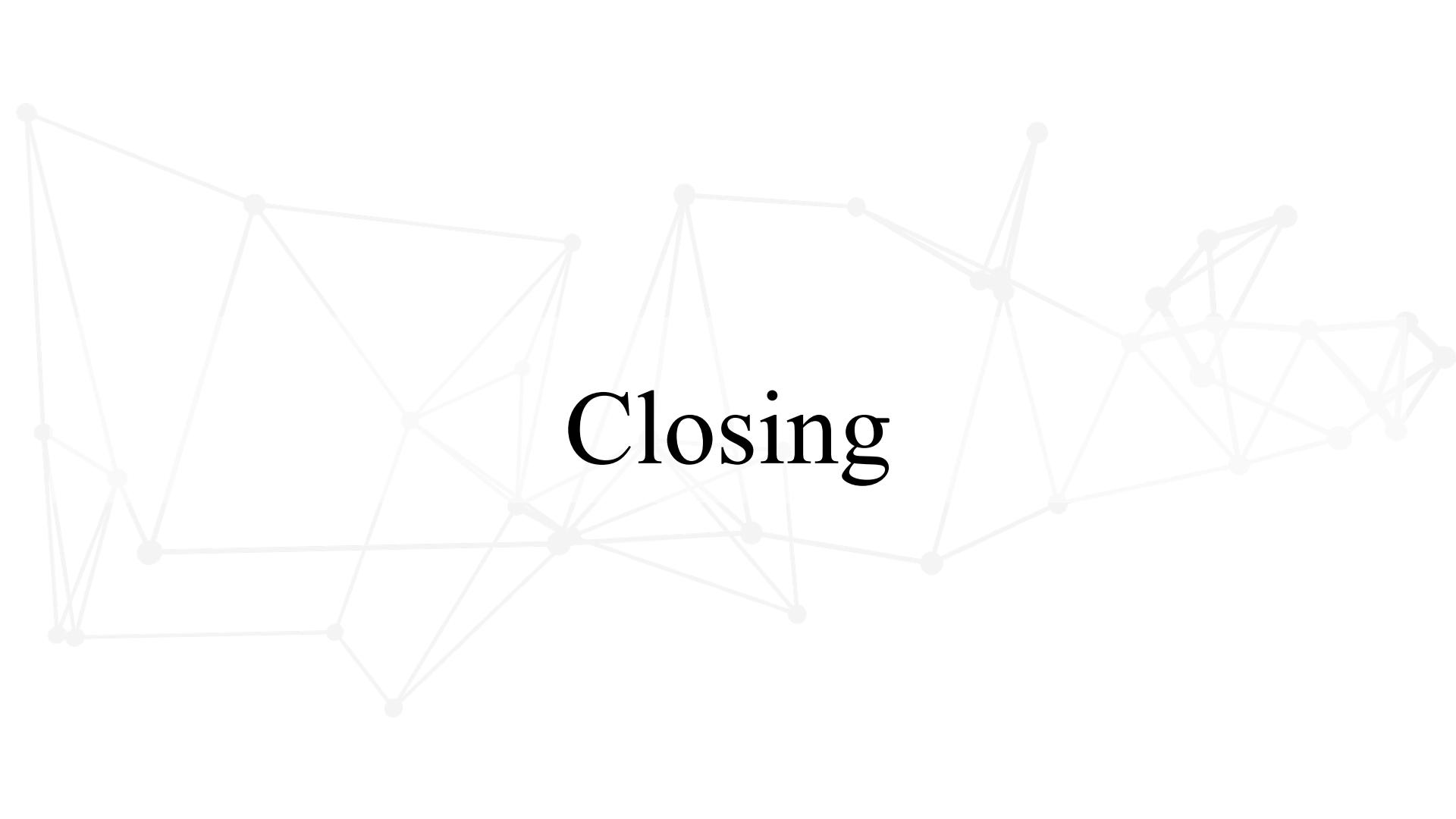
    file_path = str(SCRIPT_DIR) + "/data/scriptedturn.json"
```

Automated Generation of Scenarios

Although, we do not show automated generation in this tutorial, thanks to BeamNGpy one can also

- Generate scenarios from scratch (setting up roads, configuring vehicles, sensors)
- Mutate existing scenarios (e.g., mutate roads, update trajectories)

Feel free to explore and contribute!



Closing

Community Engagement and Contacts

BeamNG.tech is a work in progress and will keep growing and improving.
You **can be part of this** by inventing new tools, identify use cases, contributing patches and fixes, promoting the cool research you do with it, and more!

Contact :

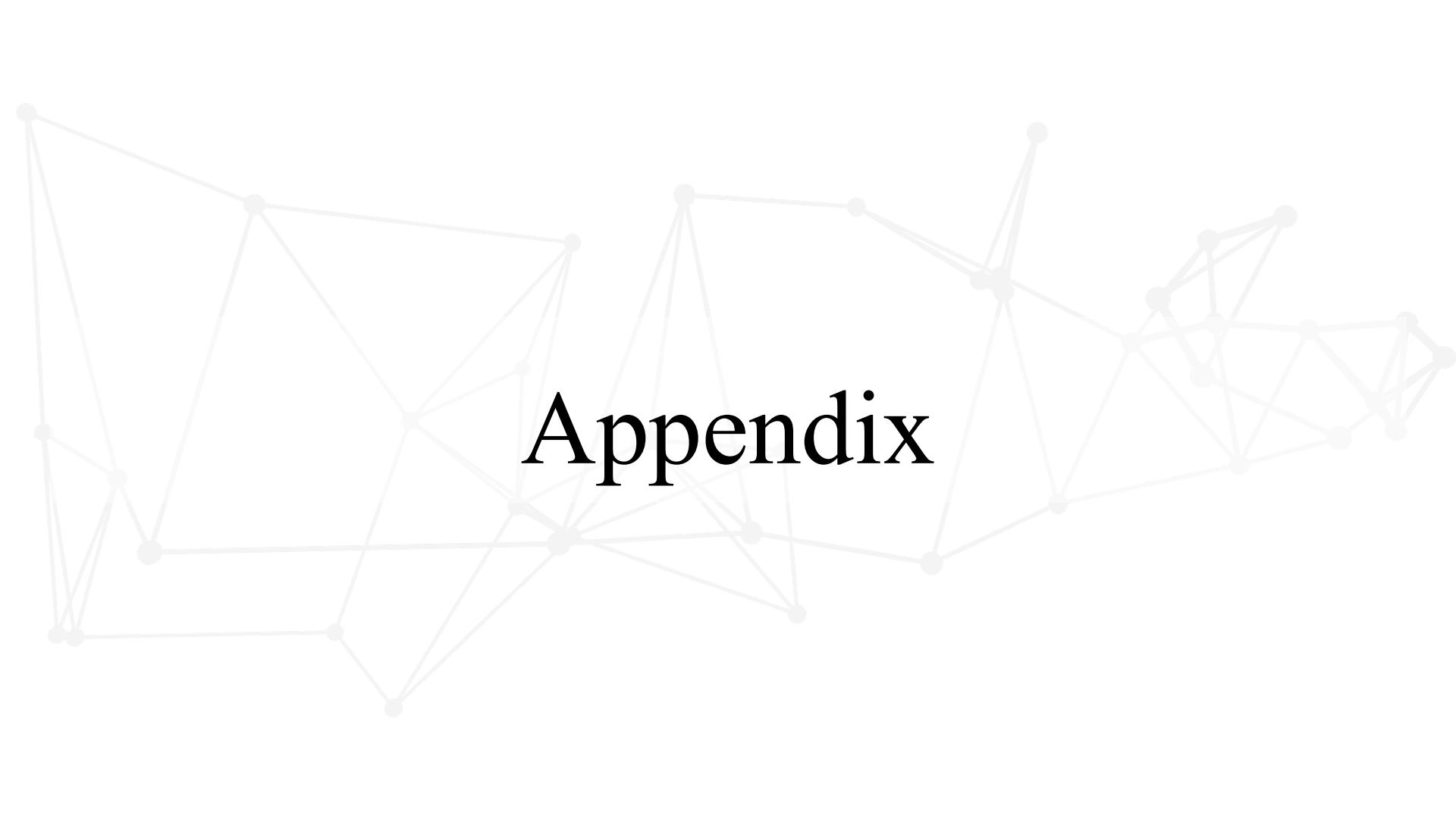
<https://documentation.beamng.com/support/troubleshooter/> - for technical support

<https://register.beamng.tech/> - non-commercial license requests

<https://github.com/BeamNG/scenario-based-testing-tutorial> - additional material

<https://github.com/BeamNG/BeamNGpy/issues> - issues on BeamNGpy

Appendix



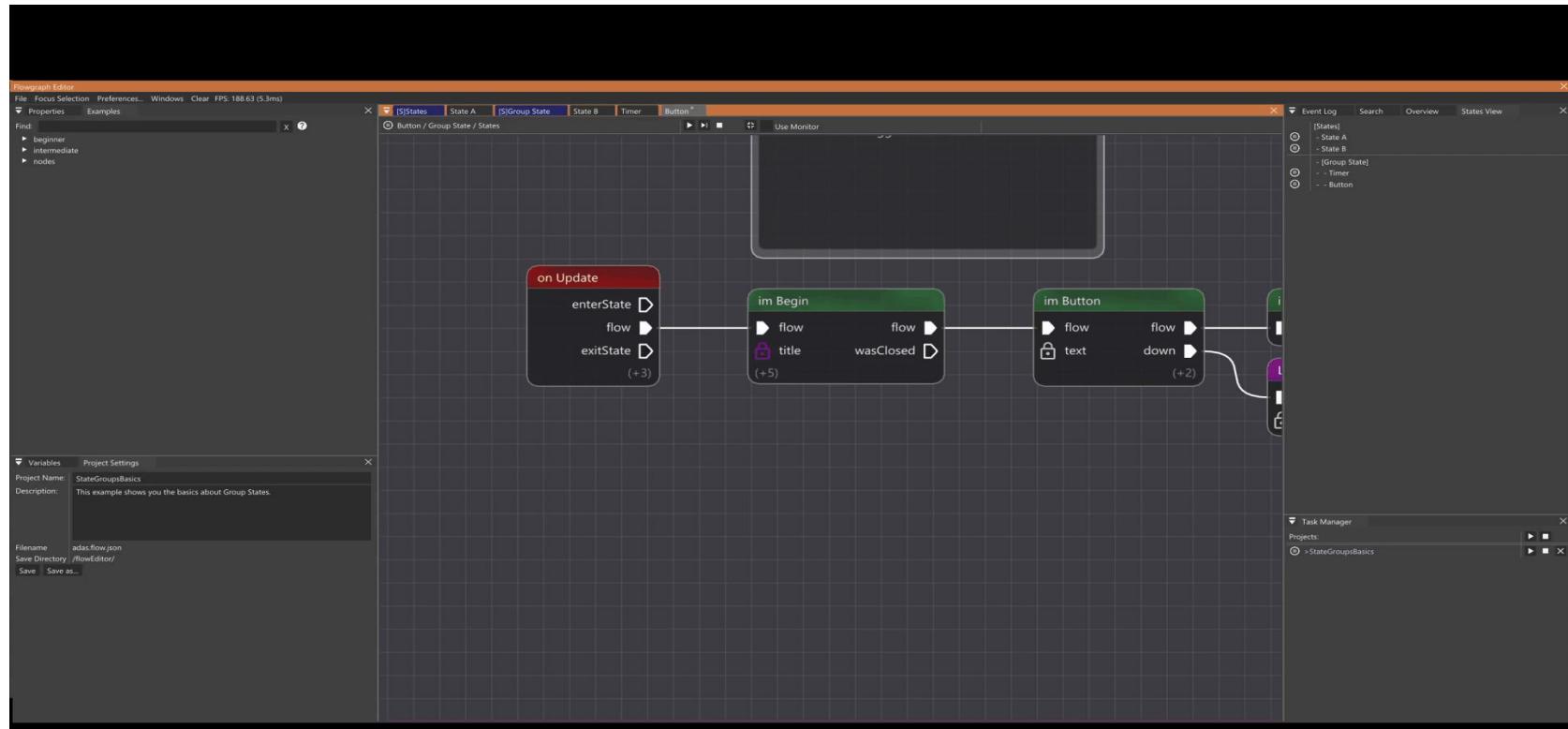
Tools Compatibility

BeamNG.tech features many tools; however, not all of them are currently fully interoperable. The following table shows the current situation:

Tools compatibility	Traffic Manager	ScriptAI	Vehicle sensors configuration tool	Map sensors configuration tool	Roads Architect Tool	Python API	Flowgraph
Traffic Manager	+(WIP)	+	+	+	+(WIP)	+(partially)	- redundant
ScriptAI	#	+ (WIP)	+	+	+	+	+
Vehicle sensors configuration tool	#	#	+ (WIP)	+	+	+	-
Map sensors configuration tool	#	#	#	+	+	+	-
Roads Architect Tool	#	#	#	#	+ (WIP)	+	+
Python API	#	#	#	#	#	+	+
Flowgraph	#	#	#	#	#	#	+

Flowgraph : Node-based scripting language - Visual interface

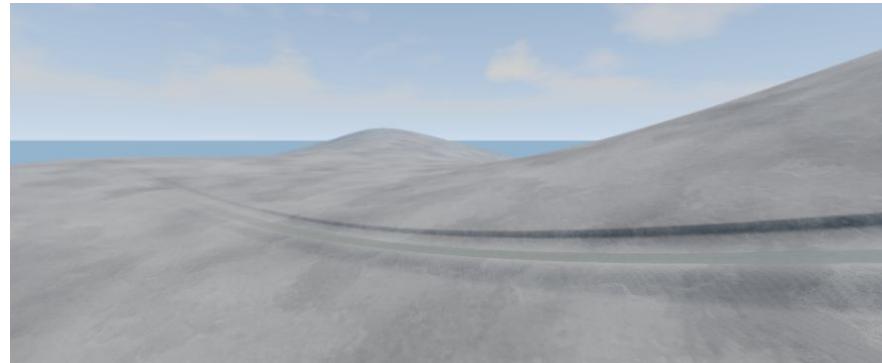
F11 : Window -> Gameplay -> Flowgraph Editor, in the menu header



Terrain & Road importer tool

1. It imports terrains,
2. then imports roads,
3. then terraforms the terrain wrt the imported roads.

Compatible with tech_ground map -
image resolution constraints.



Biome tool

It generates **vegetation** in automated way, with natural distribution of elements.



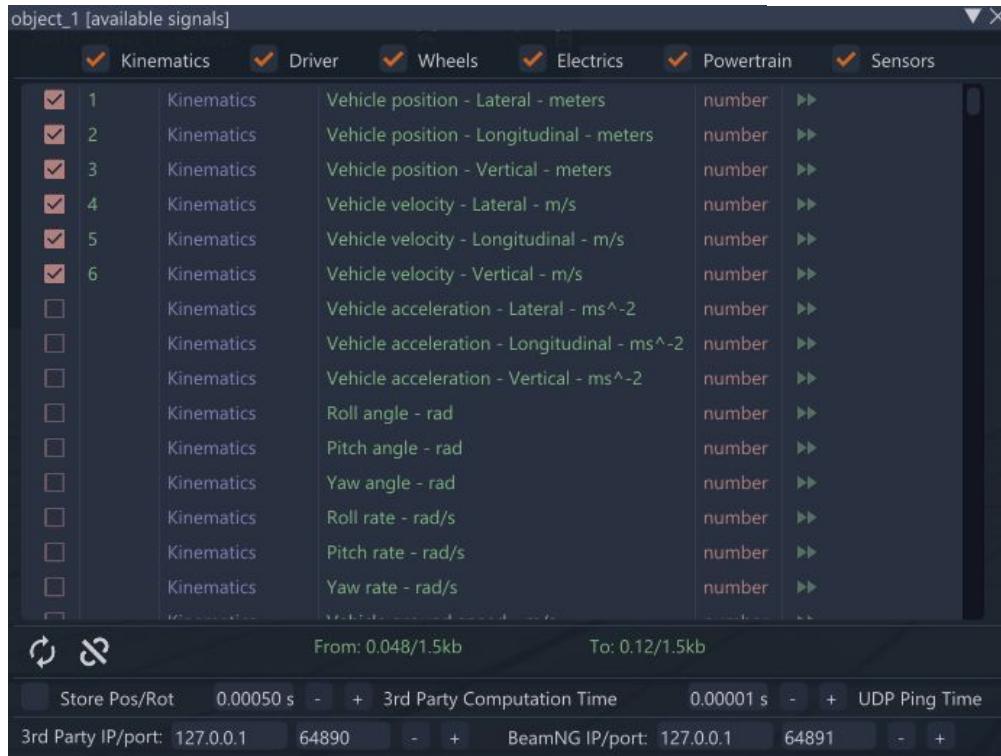
Mathworks co-simulation

BeamNG.tech Support for MATLAB and Simulink

Version 1.3.0 (3.09 MB) by BeamNG GmbH

Connects MATLAB and Simulink to BeamNG.tech

<https://beamng.tech/>



MATLAB ↔ BeamNGpy ↔ BeamNG.tech

Simulink ↔ BeamNG.tech

UDP protocol

S-function