SIMUL 2021
The Thirteenth International Conference on
Advances in System Simulation
October 03, 2021 to October 07, 2021 - Barcelona, Spain

A Photorealistic Rendering Infrastructure for Man-in-the-Loop Real-Time Vehicle Simulation

Alessandro Tasora, Dario Mangoni

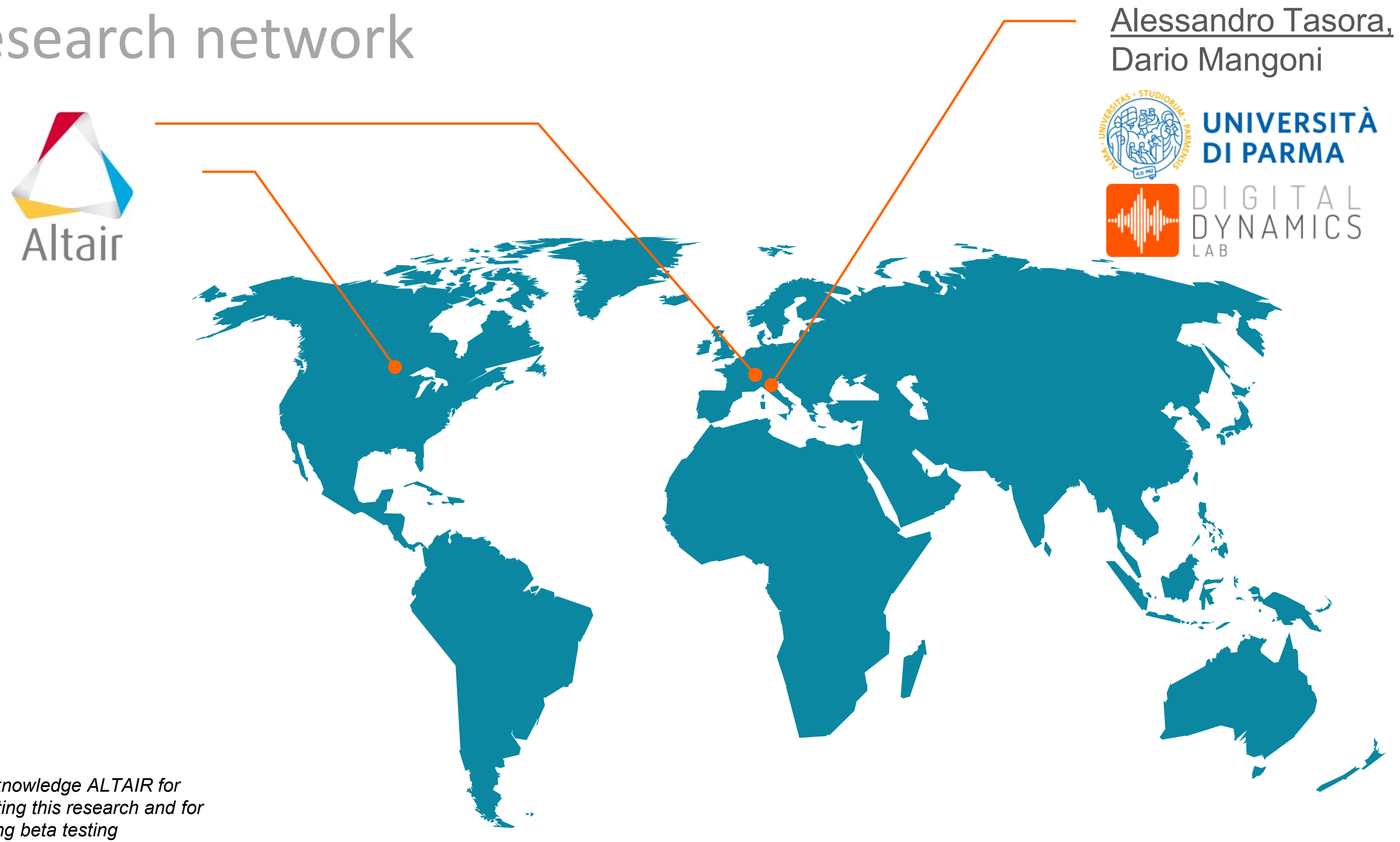
Dept. of Engineering and Architecture
University of Parma
Parma, Italy
email: alessandro.tasora@unipr.it

Presenter bio

Alessandro Tasora

- Professor at the University of Parma, teaching applied mechanics, robotics, vehicle dynamics
- Developer of ProjectChrono C++ multiphysics simulation library
- Member of the ASME
- Director of the Digital Dynamics Lab
- Chair of the International Summer School on Multibody dynamics

Research network



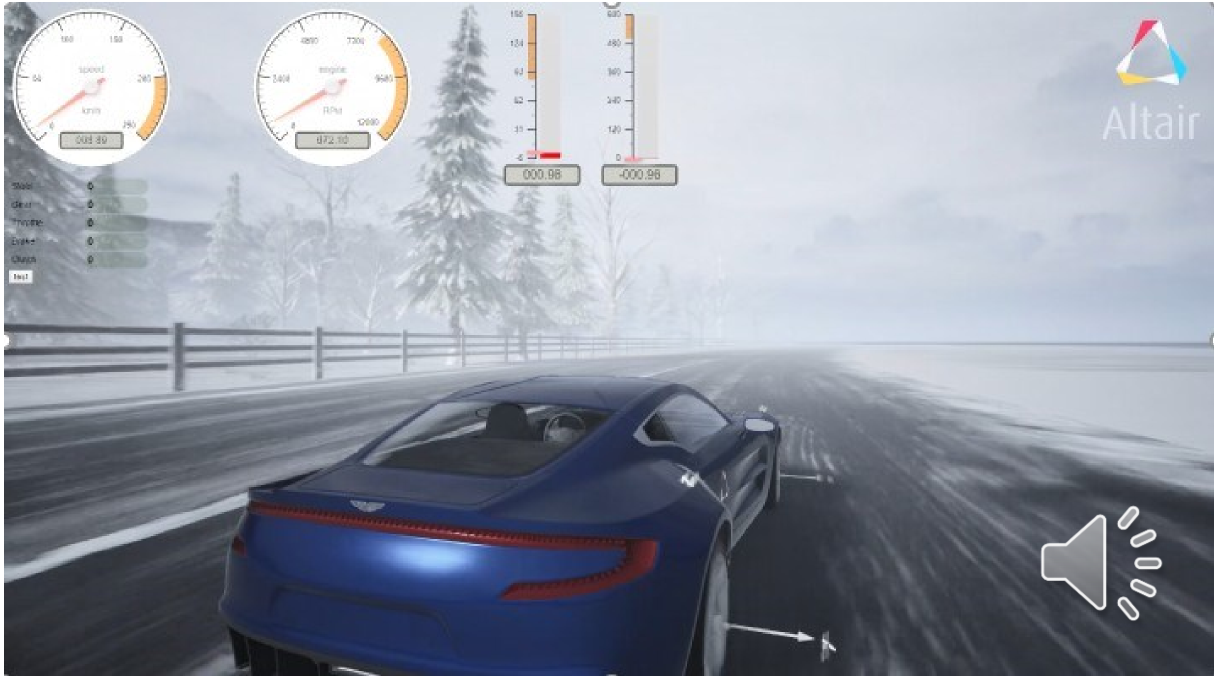
We acknowledge ALTAIR for supporting this research and for providing beta testing

1. INTRODUCTION

Real-time visualization of simulations in Unreal Engine

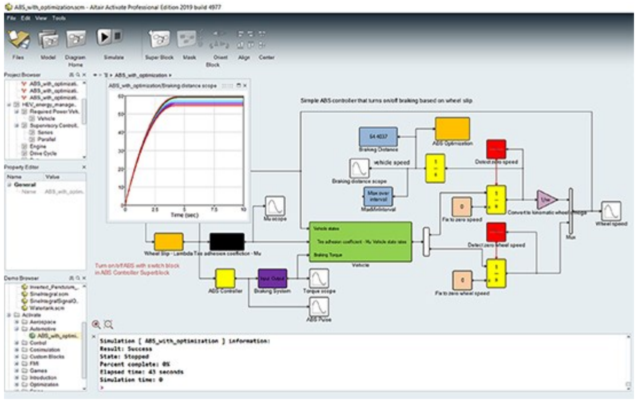
The RTV project

- Goal: real-time visualization of Modelica-based multibody systems
- Development of vehicle simulation tools in collaboration with Altair
- Based on Unreal Engine API for rendering

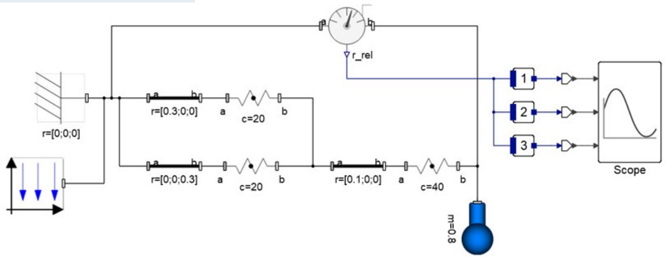


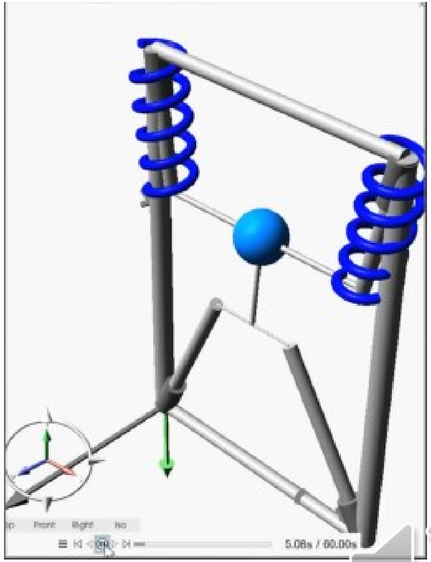
Altair ACTIVATE

- ACTIVATE as a tool to generate FMUs
  - It supports the MODELICA Multibody library
  - 3D visualization already possible, but not photorealistic



```
Simulation [ ABS_with_optimization ] information:
Result: Success
State: Stopped
Percent complete: 0%
Elapsed time: 43 seconds
Simulation time: 0
```





Unreal Engine

- Unreal Engine v.4/v.5 is the state-of-the-art of rendering engines
- Applications can be build on it, using a C++ API

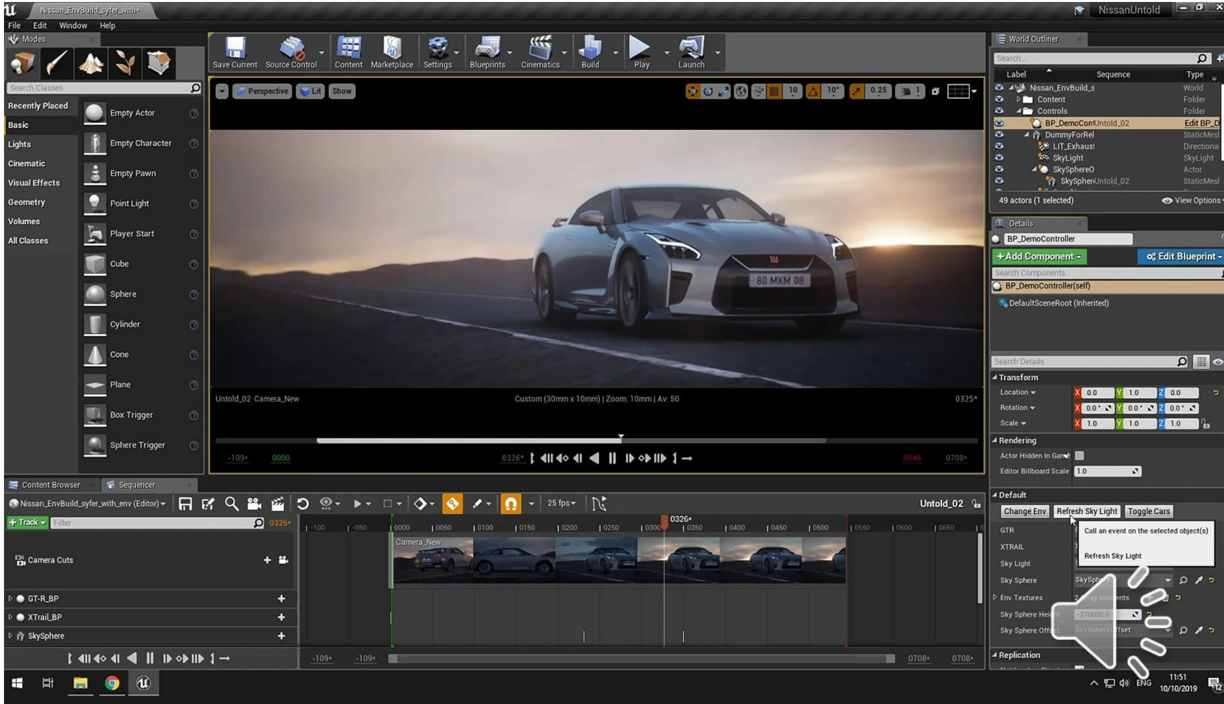
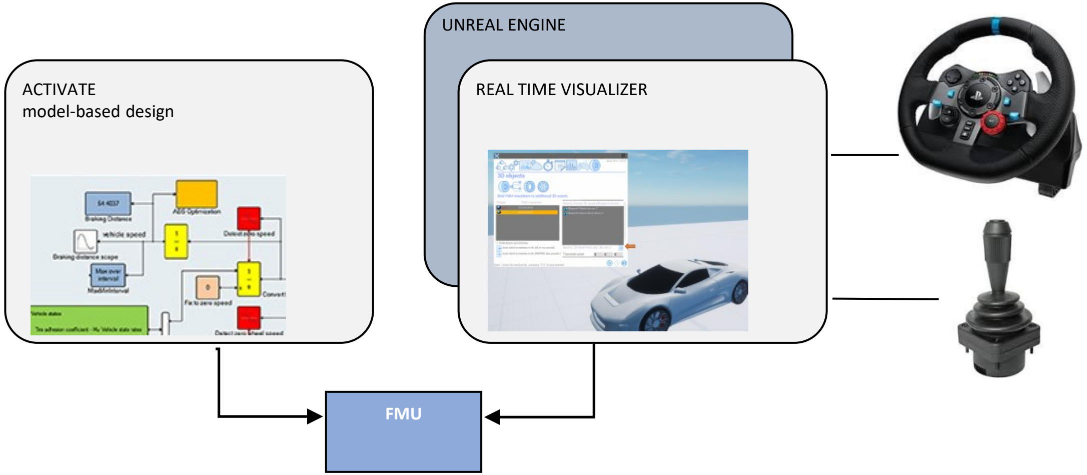


Image courtesy of Untold Studios

Workflow

- Two applications:
  - ALTAIR ACTIVATE uses Modelica model-based design to generate a FMU
  - RTV , based on UnrealEngine, is the real-time visualization tool that:
    - loads the FMU and simulate it
    - performs cyclic I/O
    - handles GUI / HMI
    - Renders the 3D view
- Model and Visualization are separate
  - The model can be hot-swapped



UNREAL ENGINE

ACTIVATE
model-based design

REAL TIME VISUALIZER

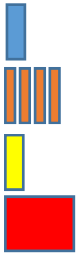
FMU

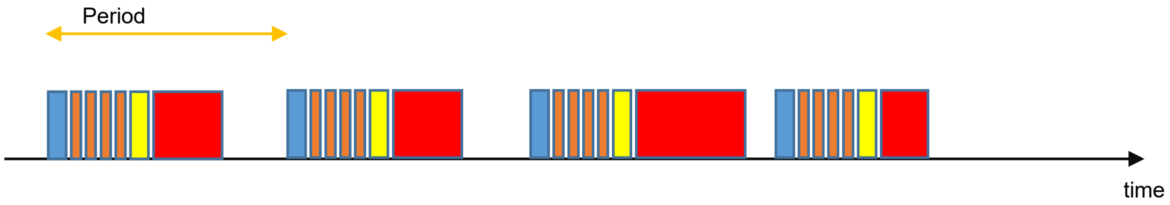
2.
INFRASTRUCTURE HIGHLIGHTS

Main features of the visualization and simulation tool

Real time simulation

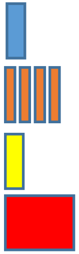
- Typical periodic task:
  - Get state of sensors (ex. steering wheel angle, throttle)
  - Perform N integration time steps
  - Set outputs (ex. 6 DOF platform displacement)
  - Generate realistic rendering for 3D visualization



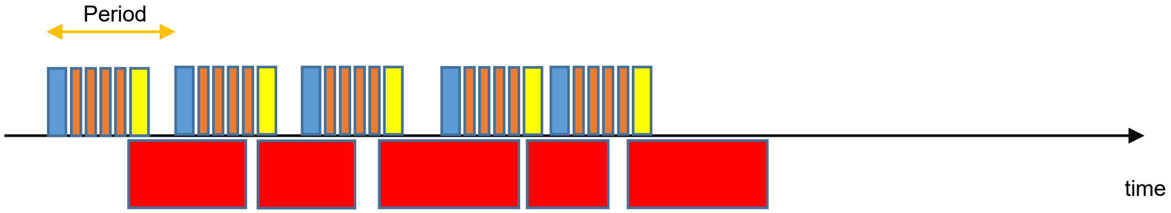


Real time simulation

- Typical periodic task:
  - Get state of sensors (ex. steering wheel angle, throttle)
  - Perform N integration time steps
  - Set outputs (ex. 6 DOF platform displacement)
  - Generate realistic rendering for 3D visualization

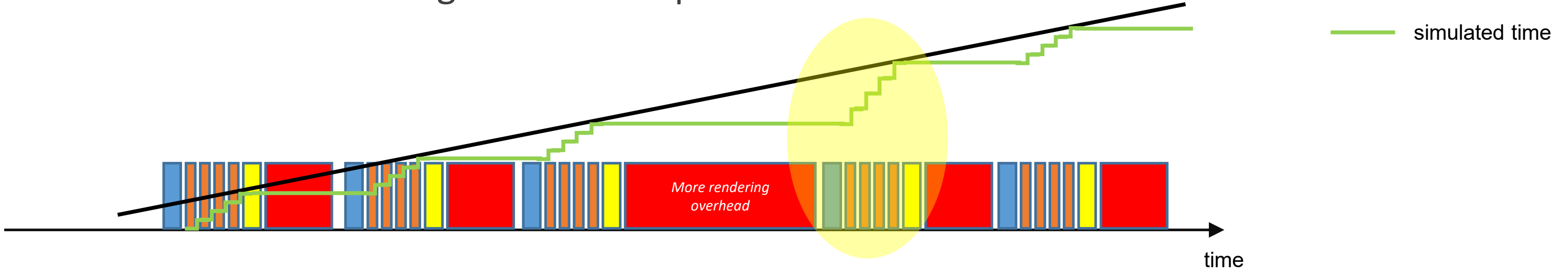


- The 3D rendering can run in parallel
  - Use multithreading (different choices of syncing)
  - Especially with GPUs, the CPU has less burden
  - Allows more complex rendering

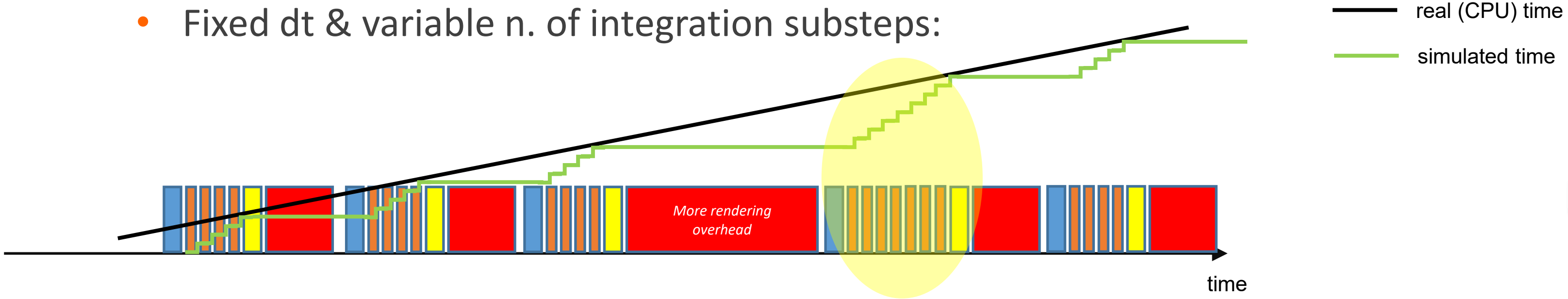


Real time simulation

- Especially in soft-RT, the period could be quite non-uniform, for max performance
  - How to catch up "delays" in simulated time? Various solutions, ex:
    - Fixed n. of integration substeps & variable dt:



    - Fixed dt & variable n. of integration substeps:



Real time simulation

- FMUs can be of “cosimulation” type or “model exchange” type
  - The former has an embedded time integrator generated by Activate: it can’t be bypassed.
  - The latter could allow custom timestepping:

- Time integrators that are fit for real-time:
  - Should be of fixed time step. Ex. dt=1ms for typical car simulations.
  - Must implement fast nonlinear solver loops, or even avoid any solver loops by smart formulations of the FMU model. Modelica compilers can automatically optimize this.
  - Example: Euler implicit. Runge Kutta explicit 2nd order, etc.

RTV highlights

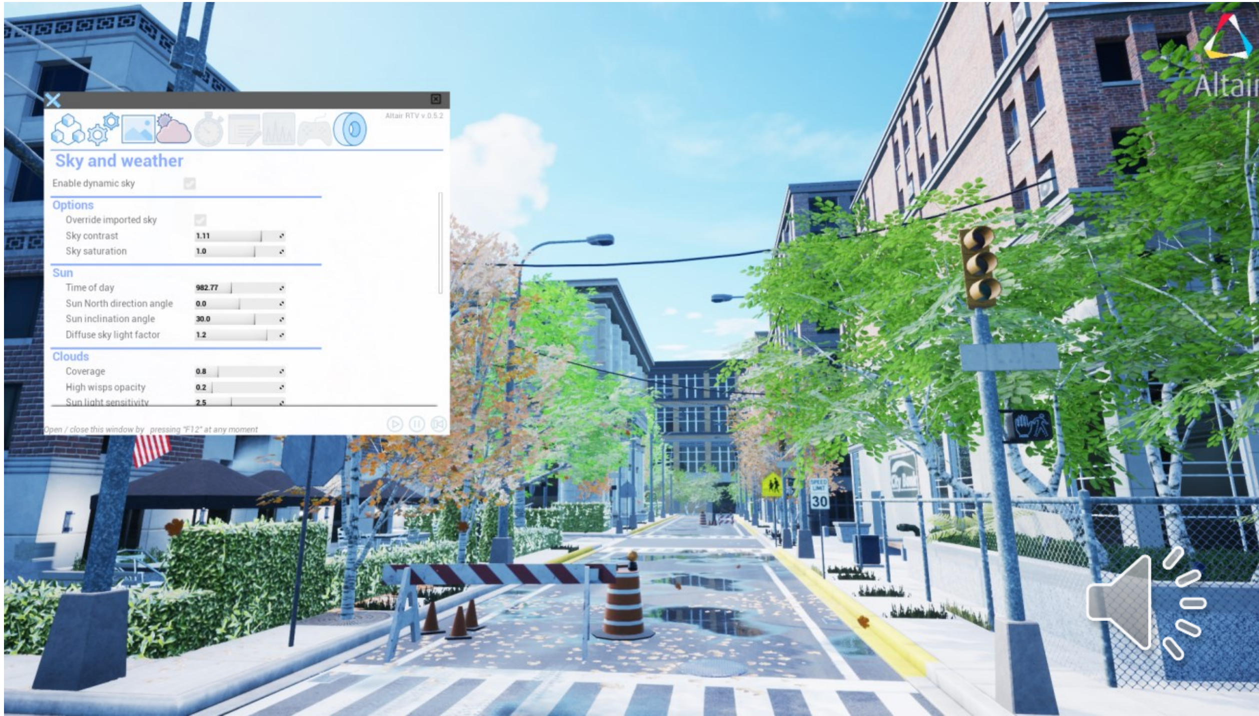
- State of the art rendering quality thanks to Unreal Engine
  - Physically Based Rendering (PBR) photorealistic materials
  - Hard shadows, soft shadows, raytracing, …

RTV highlights

- State of the art rendering quality thanks to Unreal Engine
  - Physically Based Rendering (PBR) photorealistic materials
  - Hard shadows, soft shadows, raytracing, …
  - Cinematic cameras (color grading, bloom, DOF, motion blur, lens flares, …)
  - Atmospheric effects (weather, clouds, atmosphere scattering, fog, …)

RTV highlights

- Landscapes can be authored using the Unreal Engine Editor
    - Streets, roads, traffic lights
    - Use Blueprint UE visual scripting to script building/traffic/etc.



Altair RTV v.0.5.2

Sky and weather

Enable dynamic sky ☑

Options

Override imported sky ☑

Sky contrast	1.11
Sky saturation	1.0

Sun

Time of day	982.77
Sun North direction angle	0.0
Sun inclination angle	30.0
Diffuse sky light factor	1.2

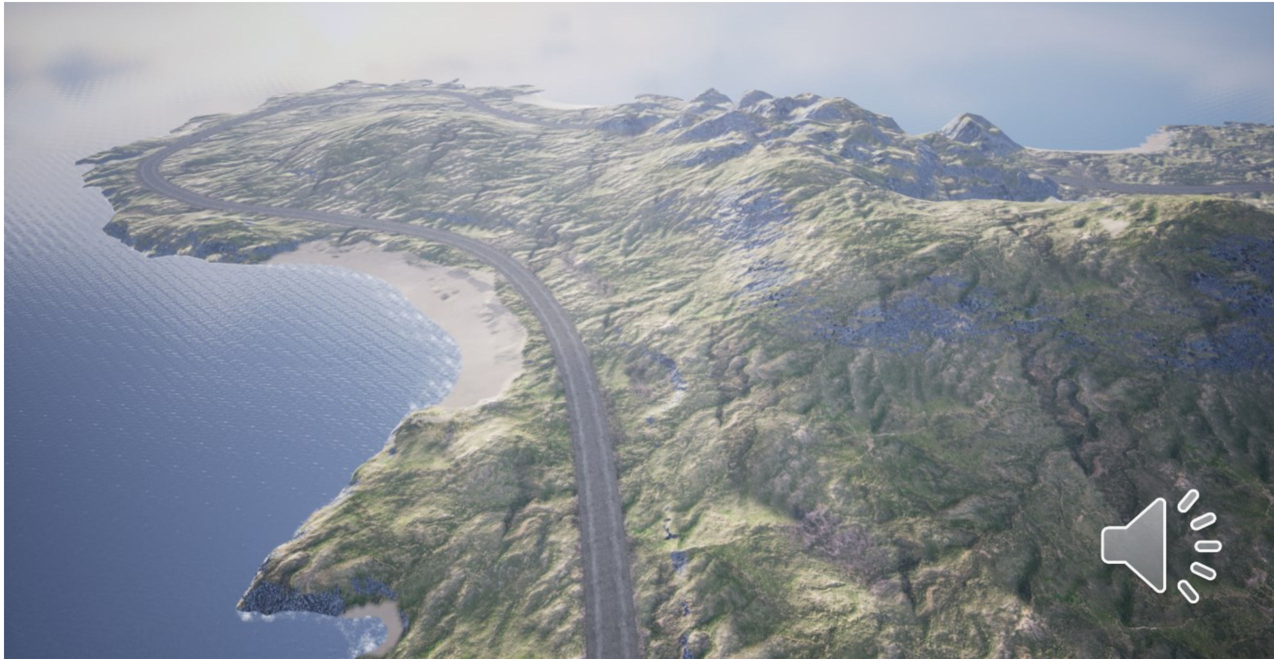
Clouds

Coverage	0.8
High wisps opacity	0.2
Sun light sensitivity	2.5

Open / close this window by pressing "F12" at any moment

RTV highlights

- Landscapes can be authored using the Unreal Engine Editor
  - Streets, roads, traffic lights
  - Use Blueprint UE visual scripting to script building/traffic/etc.
  - Terrain sculpting

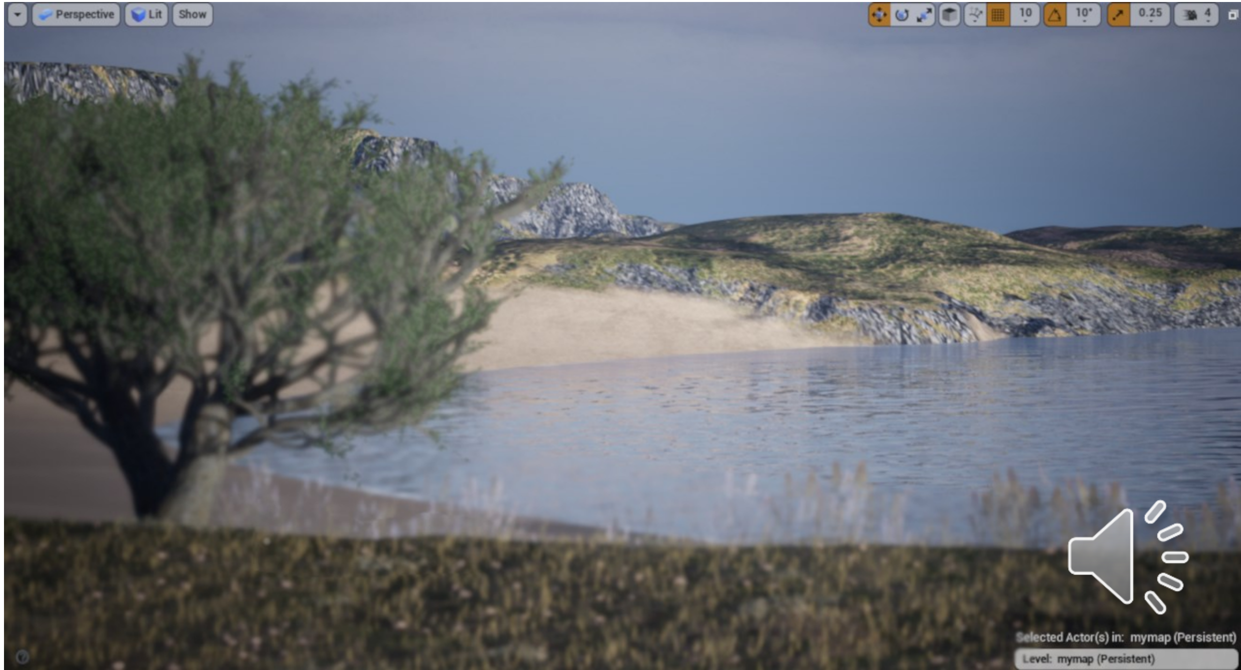


RTV highlights

- Landscapes can be authored using the Unreal Engine Editor
  - Streets, roads, traffic lights
  - Use Blueprint UE visual scripting to script building/traffic/etc.
  - Terrain sculpting
  - Foliage, trees, grass

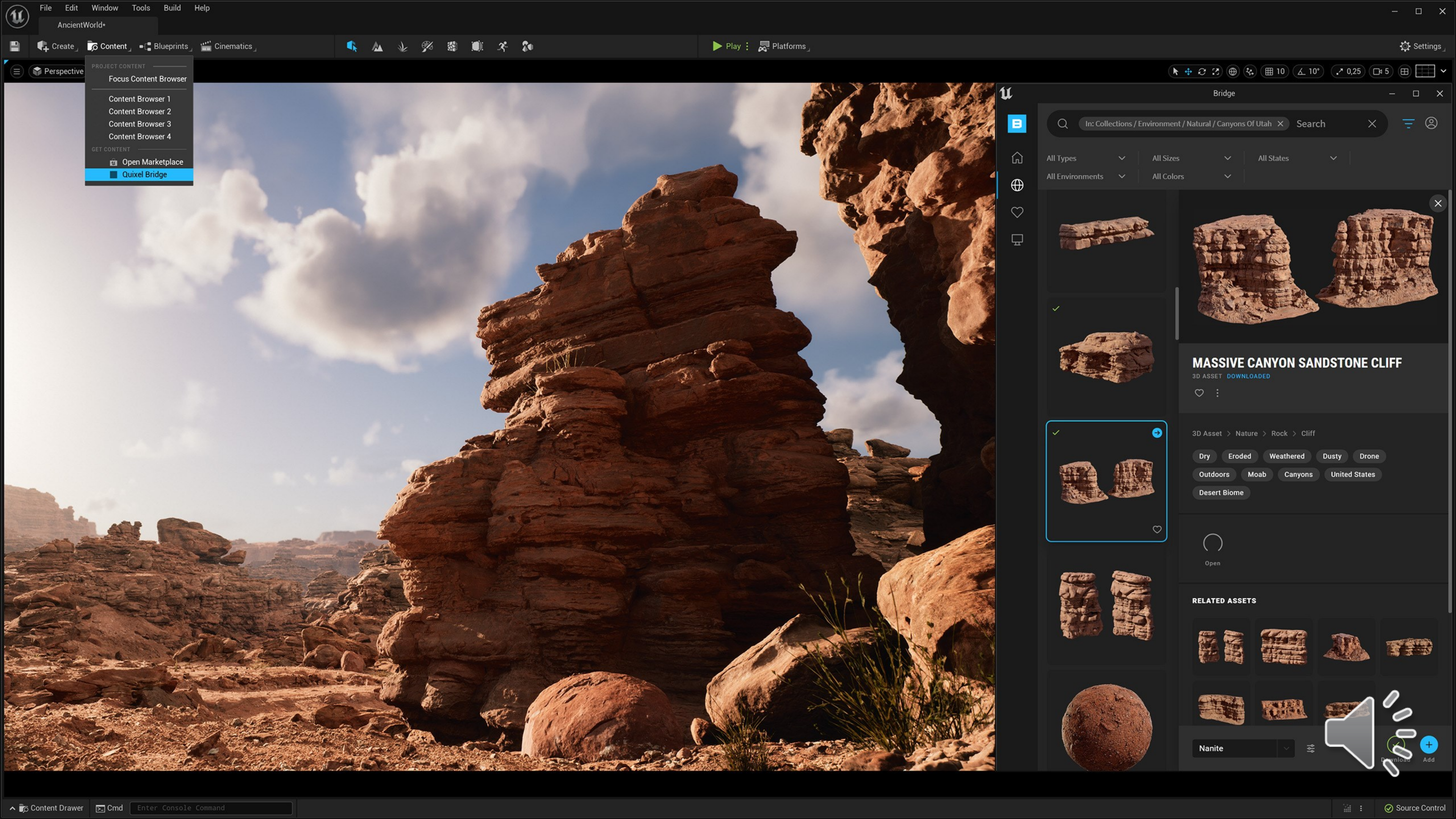
RTV highlights

- Landscapes can be authored using the Unreal Engine Editor
  - Streets, roads, traffic lights
  - Use Blueprint UE visual scripting to script building/traffic/etc.
  - Terrain sculpting
  - Foliage, trees, grass
  - Use 3D scanned assets, ex. The Megascan library (see example in next slide)



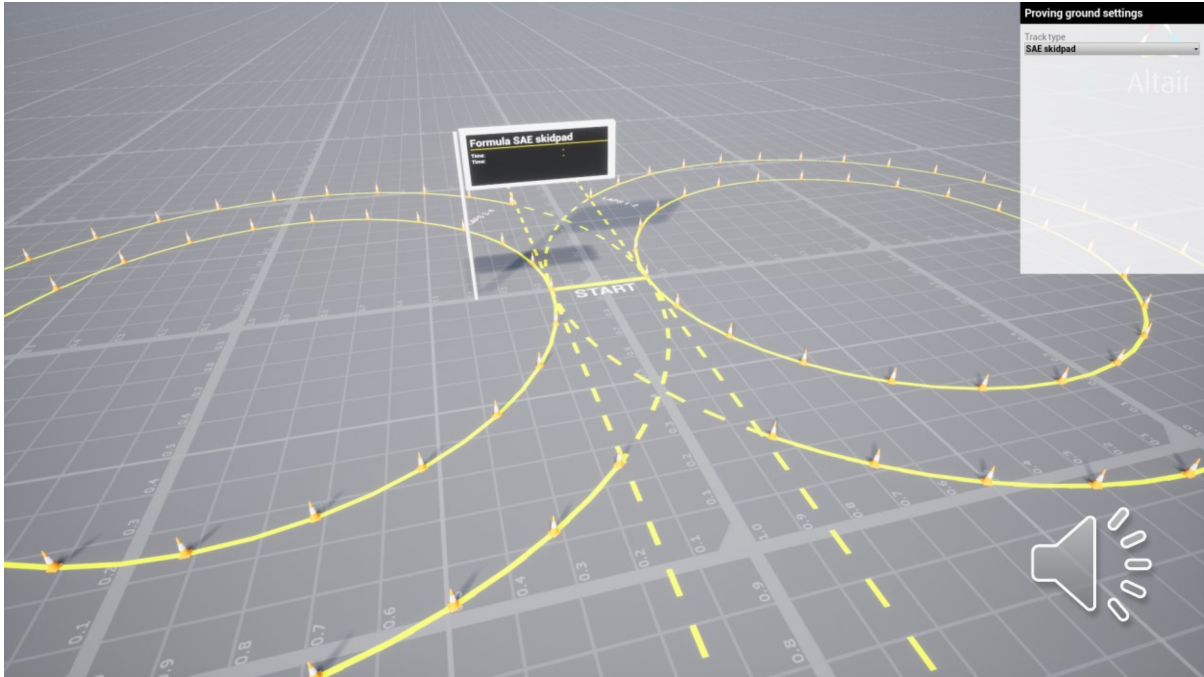
Selected Actor(s) in: mymap (Persistent)

Level: mymap (Persistent)


File
Edit
Window
Tools
Build
Help
AncientWorld*
Create
Content
Blueprints
Cinematics
Play
Platforms
Settings
Perspective
PROJECT CONTENT
Focus Content Browser
Content Browser 1
Content Browser 2
Content Browser 3
Content Browser 4
GET CONTENT
Open Marketplace
Quixel Bridge
10
10°
0,25
5
Bridge
In: Collections / Environment / Natural / Canyons Of Utah
Search
All Types
All Sizes
All States
All Environments
All Colors
MASSIVE CANYON SANDSTONE CLIFF
3D ASSET DOWNLOADED
3D Asset > Nature > Rock > Cliff
Dry
Eroded
Weathered
Dusty
Drone
Outdoors
Moab
Canyons
United States
Desert Biome
Open
RELATED ASSETS
Nanite
Download
Add
Content Drawer
Cmd
Enter Console Command
Source Control

RTV highlights

- Landscapes can be authored using the Unreal Engine Editor
  - Streets, roads, traffic lights
  - Use Blueprint UE visual scripting to script building/traffic/etc.
  - Terrain sculpting
  - Foliage, trees, grass
  - Use 3D scanned assets, ex. The Megascan library (see example in next slide)
  - Custom GUI and scripting ex. for tracks etc.

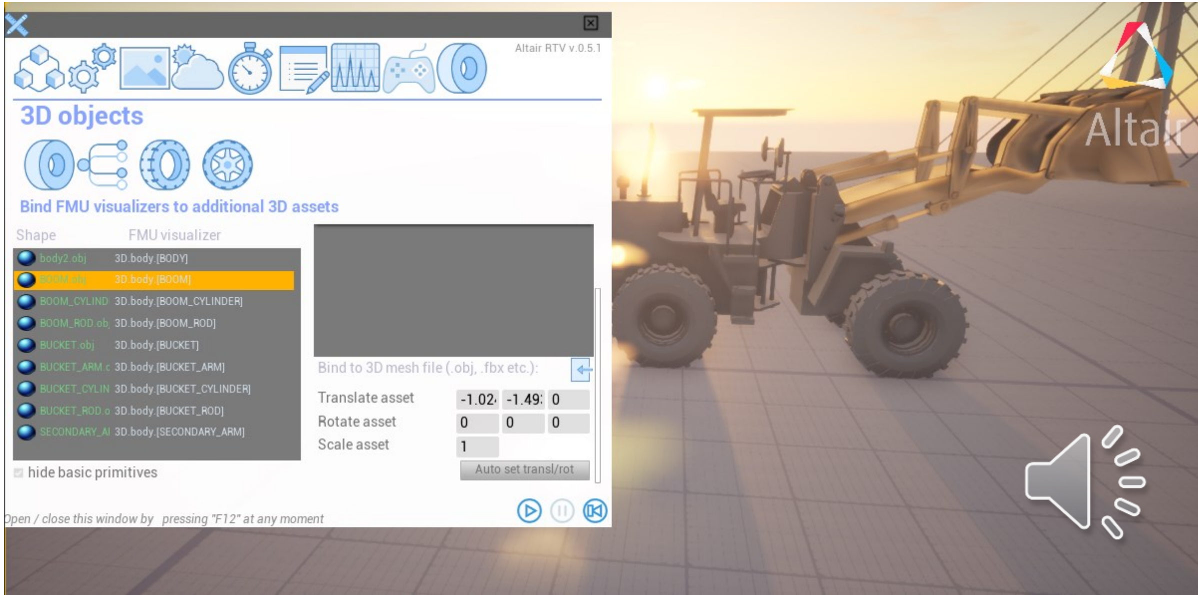


Proving ground settings

Track type: SAE skidpad

RTV highlights

- 3D assets for FMU objects
  - Load assets (cars, wheels, etc) from OBJ or 3DS or FBX or other files



Altair RTV v.0.5.1

3D objects

Bind FMU visualizers to additional 3D assets

Shape	FMU visualizer
body2.obj	3D.body.[BODY]
BOOM.obj	3D.body.[BOOM]
BOOM_CYLIND	3D.body.[BOOM_CYLINDER]
BOOM_ROD.ob	3D.body.[BOOM_ROD]
BUCKET.obj	3D.body.[BUCKET]
BUCKET_ARM.c	3D.body.[BUCKET_ARM]
BUCKET_CYLIN	3D.body.[BUCKET_CYLINDER]
BUCKET_ROD.o	3D.body.[BUCKET_ROD]
SECONDARY_AI	3D.body.[SECONDARY_ARM]

☑ hide basic primitives

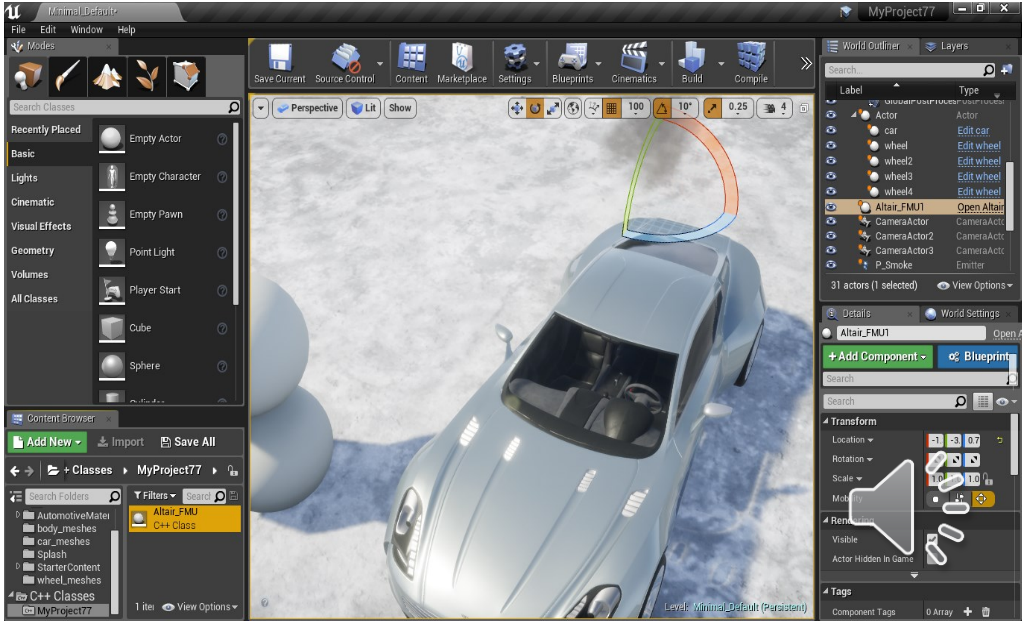
Bind to 3D mesh file (.obj, .fbx etc.):

Translate asset	-1.02	-1.49	0
Rotate asset	0	0	0
Scale asset	1		

Open / close this window by pressing "F12" at any moment

RTV highlights

- 3D assets for FMU objects
  - Load assets (cars, wheels, etc) from OBJ or 3DS or FBX or other files
  - Or prepare assets using Unreal Engine editor, for more possibilities
    - Add scripting in Blueprint for controlling engine sound, tire smoke, exhaust smoke, etc
    - Add photorealistic PBR materials and handle millions of polygons via dynamic LOD

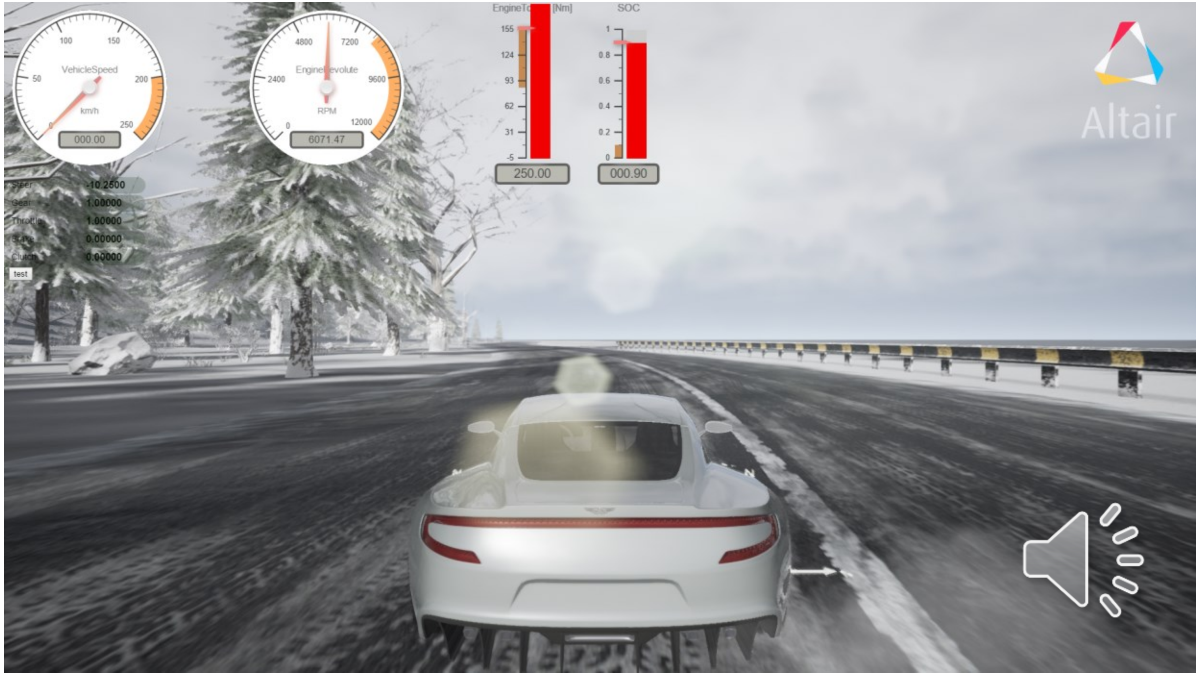


RTV highlights

- 3D assets for FMU objects
  - Load assets (cars, wheels, etc) from OBJ or 3DS or FBX or other files
  - Or prepare assets using Unreal Engine editor, for more possibilities
    - Add scripting in Blueprint for controlling engine sound, tire smoke, exhaust smoke, etc
    - Add photorealistic PBR materials. Handle millions of polygons via dynamic LOD

RTV highlights

- I/O
  - Define GUI and HMI panels (via custom widgets, or via HTML5 overlay)



Steer -10.2500
Gear 1.00000
Throttle 1.00000
Brake 0.00000
Clutch 0.00000

RTV highlights

- I/O
  - Define GUI and HMI panels (via custom widgets, or via HTML5 overlay)
  - Connect
    - RawInput devices,
    - Joysticks
    - steering wheels
    - keyboard, …

Conclusions

- Exploit FMU for model exchange
- Use MODELICA as a tool to generate FMU
- Our RTV tool visualizes FMU simulations in real-time
- Use UnrealEngine for photorealistic rendering
- Connect to I/O for man-in-the-loop