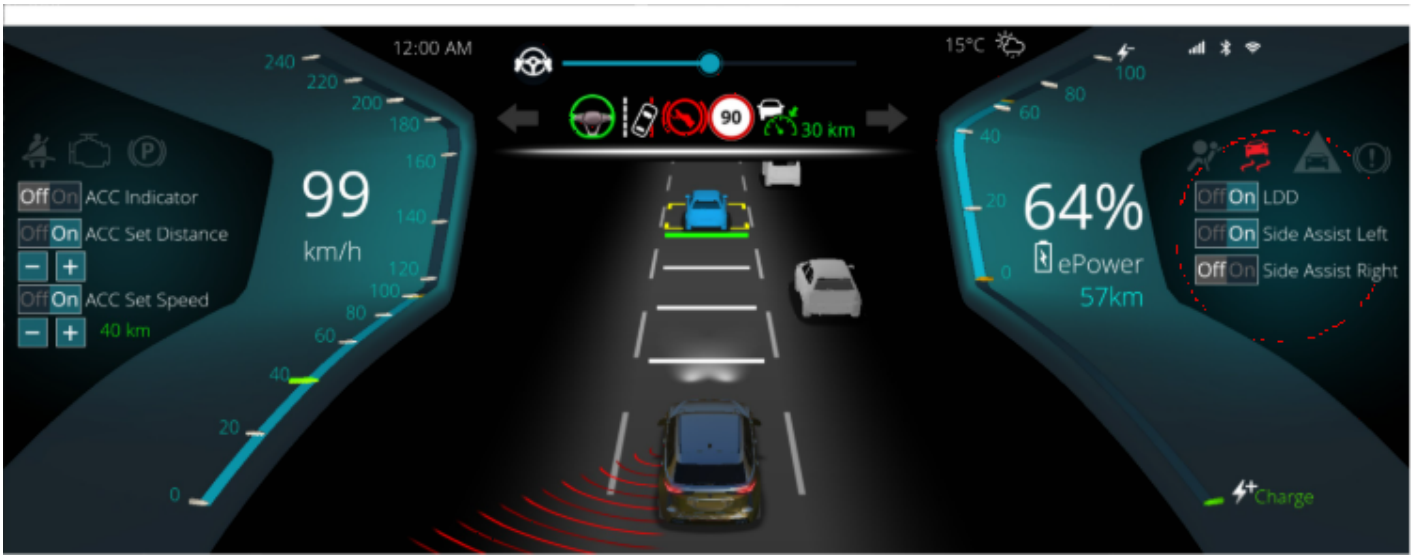
3D ADAS Control Solution

Summary

This solution allows you to check the functions and usage of controls (ADAS-Control) that support the **A**dvanced **D**riving **A**ssistant **S**ystem.



Marcket Demand to ADAS

- **ADAS visualization in the cockpit**: Helps users intuitively receive information about their surroundings and ADAS decisions.
- **Regulations:** Lead to more ADAS components in future cars.
- **Automated driving:** Increases the need for ADAS visualization to inform drivers of car decisions and warnings.
- **Growing demand:** There is a rapid increase in demand for HMI visualization of ADAS systems.

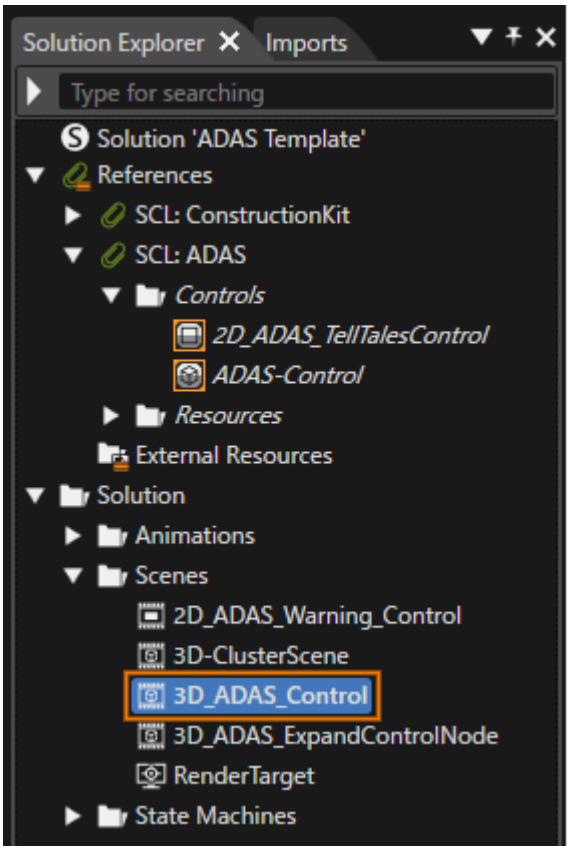
Configuration of ADAS-Control

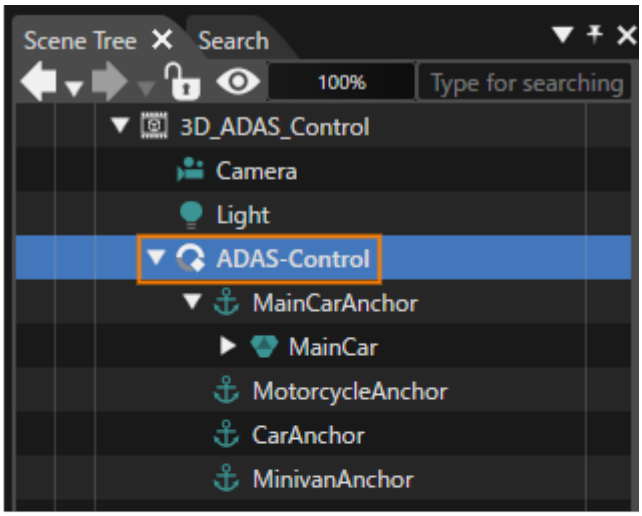
- Adaptive Cruise Control (ACC)
- Lane Departure Detection (LDD)
- Blind Spot Detection
- Drowsiness Detection
- Collision Warning
- Traffic Sign Recognition

Description

Open 3D ADAS Control Solution

Open "ADAS Template" from [Samples and Templates] in the startup panel. ADAS-Control is used in the 3D_ADAS_Control scene.





ADAS-Control Properties:

In the properties window of the 3D ADAS control, user can activate or deactivate ADAS features and adjust the properties of each feature separately. The following properties are available:

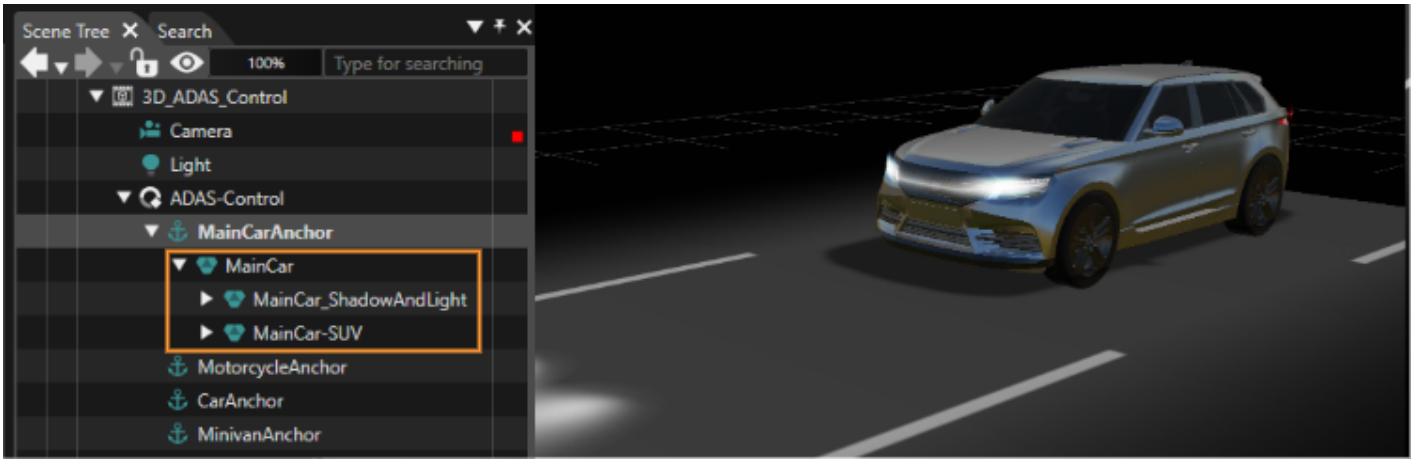
Property Name	Description
Main Car Position (Float type)	Accepts values between 0 and 100. This property adjusts the Road texture u_OffsetU position to simulate the car moving along the X-axis. The main car remains static while the road texture moves left and right based on this value.
LDD (Boolean type)	Accepts values between 0 and 100. This property adjusts the Road texture u_OffsetU position to simulate the car moving along the X-axis. The main car remains static while the road texture moves left and right based on this value.
Side Assist (Boolean type)	Activates Blind Spot Detection (BSD). This system uses sensors to detect vehicles in the driver’s blind spot and alerts the driver with visual warnings. BSD is useful for changing lanes, merging, and overtaking. There are two attributes for Side Assist, one for the left side and one for the right side.

ACC (Boolean type)	Activates Adaptive Cruise Control. This system uses radar and cameras to maintain a safe distance from the vehicle ahead and automatically adjusts the speed to keep a safe following distance. ACC is ideal for highway driving, reducing driver fatigue, and helping to prevent rear-end collisions caused by sudden braking.
Set ACC Distance (Float type)	Allows the user to set a safe distance from the front vehicle, ranging from 1 to 4.
ACC Safe distance Indicator (Boolean type)	Activates the rendering of a mesh with a dynamic texture to demonstrate the safe distance to the front vehicle.
Normal Lane Texture	Texture for normal road conditions.
Highlight Center Lane Texture	Optional texture for highlighting the center part of the road lane.
Left Red Lane Texture	Changes the left lane texture to red when LDD is on and the car crosses the left lane.
Right Red Lane Texture	Changes the right lane texture to red when LDD is on and the car crosses the right lane.
Left Yellow Lane Texture	Changes the left lane texture to yellow when LDD is on and the car is close to the left lane.
Right Yellow Lane Texture	Changes the right lane texture to yellow when LDD is on and the car is close to the right lane.
Motorcycle Default Model	Deactivates the rendering of the default motorcycle 3D model, allowing users to assign their own 3D models to the ADAS control anchors.
Minivan Default Model	Deactivates the rendering of the default minivan 3D model, allowing users to assign their own 3D models to the ADAS control anchors.
Car Default Model	Deactivates the rendering of the default car 3D model, allowing users to assign their own 3D models to the ADAS control anchors.
Car Speed (Float type)	Sets the speed of the main car.
Car Minimum Speed (Float type)	Sets the minimum speed of the main car.
Car Maximum Speed (Float type)	Sets the maximum speed of the main car.
Forward Car Speed Value	Sets a target node in a 2D or 3D scene to receive the Car speed value.
Set ACC Speed Value (Float type)	Changes the ACC speed.
ACC Minimum Speed (Float type)	Sets the ACC minimum speed.

ACC Maximum Speed (Float type)	Sets the ACC maximum speed.
Forward the ACC Speed value	Sets a target node in a 2D or 3D scene to receive the ACC speed value.

ADAS-Control Anchors:

Users can import their 3D models as FBX or glTF files and attach them to the anchors to replace the default ADAS 3D models. For example, a user can import a new 3D mesh for the main car and make it a child of the MainCarAnchor, as shown in the screenshot below.



ADAS-Control 3D models and textures:

Users can use the default vehicle 3D models (FBX files) as a bounding box and import them into DCC tools like 3Ds Max, Cinema 4D, Blender, etc., to create their 3D meshes at the same scale as the default models.

Textures can be exported from the solution explorer window or reimported with the same name and path to replace the existing ADAS control textures.

ADAS-Control Data Source:

A simple data source is created for ADAS control based on the ADAS properties. Each property in the ADAS control is bound to the target data type. The data source can be written in XML format and stored in an .xhcdl file. It can be extended by users based on their use cases.

This is a sample data source file "ADAS.xhcdl" that can be used with the 3D ADAS Control solution:

```
<?xml version="1.0" encoding="utf-8"?>

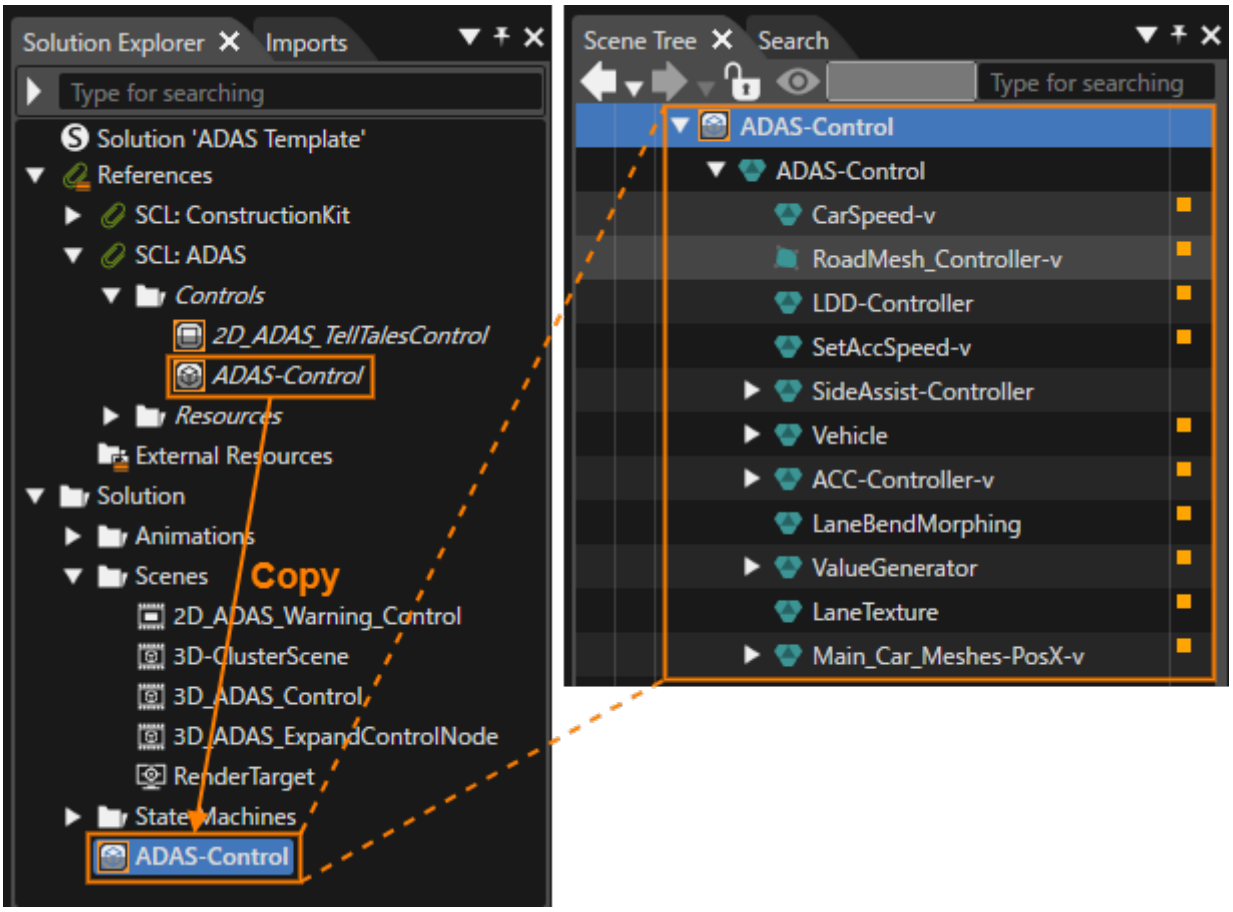

<definition
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="../../../../cgi_studio_cit/schema/HmiContract.xsd">
```

```
    <!-- Data binding
============================================================================-->
    <types>
        <!--Dummy type to force the generation of code -->
        <compound name="IntRange" uid="{67DA817B-A748-4DF5-92FF-8193C1C595C6}">
            <item name="Min" type="::Courier::Int32" uid="{E7AFB591-6B09-4DD7-9396-
DE75D36297F6}" defaultValue="0"/>
            <item name="Max" type="::Courier::Int32" uid="{7F9A827D-B690-46FA-BCD0-
89C1FF0259D4}" defaultValue="0"/>
        </compound>
  </types>
    <bindingSources>
        <!-- Sample application specific messages -->
        <!-- Dynamic list specific messages -->
        <bindingSource name="ADAS" access="asynchronous" dataContextItem="true"
uid="{86f39d72-8d42-43eb-87e7-475cdf384734}">
□□    <item name="MainCarPosition" type="FeatStd::Float" uid="{7a55879f-0a01-41bf-9d96-
d3048280b79f}" />
            <item name="LaneDepartureDetection" type="bool" uid="{f6ce6e18-00fc-45b7-a3fa-
575179642ded}" />
□□□<item name="LeftSideAssist" type="bool" uid="{c973d596-d54a-4718-9b88-4c474e848a7b}" />
□□□<item name="RightSideAssist" type="bool" uid="{66f36b0a-2a4a-474b-8baf-d3dcd48708f3}" />
□□□<item name="AdaptiveCruiseControl" type="bool" uid="{e28b3baa-af17-49ff-920d-ea767bad4ef1}"
□□□<item name="SetAccDistance" type="FeatStd::Int32" uid="{31fbcacd-9efe-4f42-ac2c-db68876103a0
/>
□□□<item name="AccSafeDistanceIndicator" type="bool" uid="{b0fa6306-e6a4-47ad-9412-e4880ddcf1ea
/>
□□□<item name="MotorcycleDefualtModel" type="bool" uid="{a62c4ed7-d20a-4e8e-ace7-942a60f4b71f}"
/>
□□□<item name="MinivanDefualtModel" type="bool" uid="{492abae1-dfbe-47ef-acc1-aee625b075af}" />
□□□<item name="CarDefualtModel" type="bool" uid="{5cb01d1b-9b9b-4982-99b2-4c82a4823d16}" />
□□□<item name="CarSpeed" type="FeatStd::Float" uid="{aaef9ff1-57f2-4a28-9955-4062c53773cd}" />
□□□<item name="CarMinSpeed" type="FeatStd::Float" uid="{AF002E21-2A9D-4617-B39B-B47B078A0C0D}"
□□□<item name="CarMaxSpeed" type="FeatStd::Float" uid="{BEAF3A1F-ACB9-4EAF-A781-BB1D2094CDFA}"
□□□<item name="RoadSpeed" type="FeatStd::Float" uid="{d9e793ad-fd4a-4edc-8a18-f122a8a0ce0e}" />
□□□<item name="AccSpeed" type="FeatStd::Float" uid="{4a6dca6e-4ad9-4f10-bc14-1bf7185a393f}" />
□□□<item name="AccMinSpeed" type="FeatStd::Float" uid="{C3096715-35A9-459E-9B00-9010C6090462}"
□□□<item name="AccMaxSpeed" type="FeatStd::Float" uid="{B2D24547-7D01-4404-BEF9-746444CB6026}"
        </bindingSource>
  </bindingSources>
```

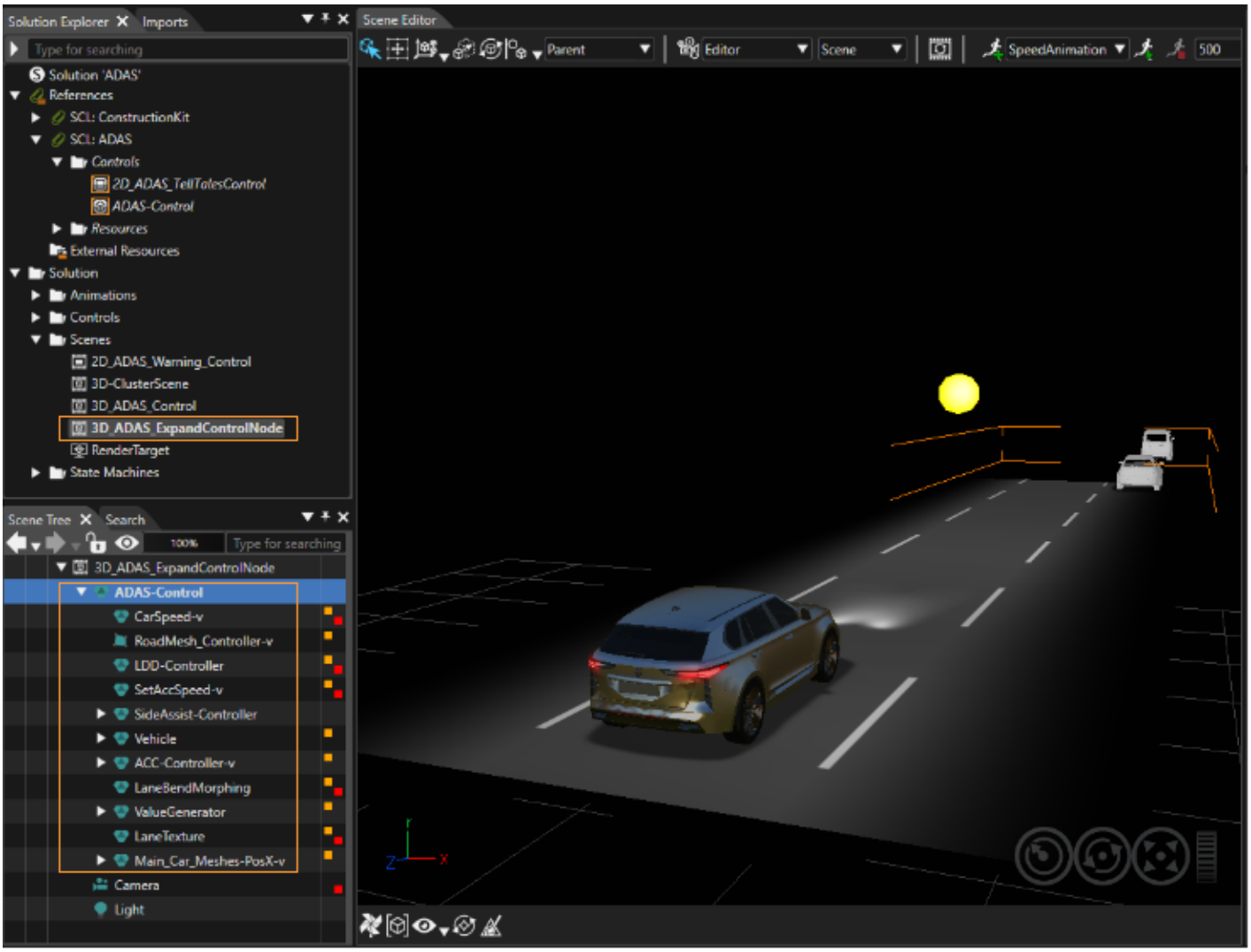
```
</definition>
```

ADAS-Control details:

ADAS-Control is created using Behaviors and Effects from the Scene Composer Toolbox, so you can extend this control based on different use cases by copying ADAS-Control under the Solution folder in the Solution Explorer and double-clicking it.



Additionally, there is a scene called 3D_ADAS_ExpandControlNode in the scenes folder, which provides an expanded version of the ADAS control. You can also edit this and save it as their own control.



ADAS-Control Shaders:

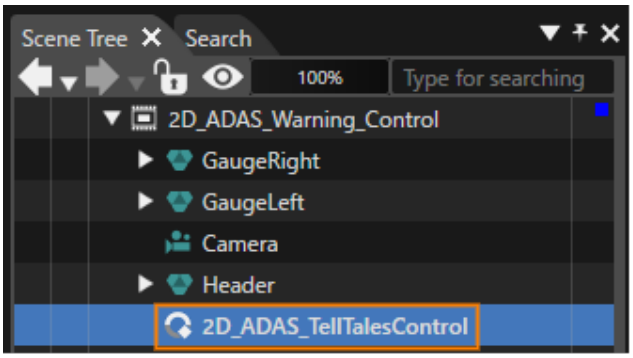
Three shader program is written for ADAS control:

1. **TextureProperties-Shader:** Used in 3D meshes to add texture properties (Tiles and Offsets).
2. **VehicleShader:** Applied to vehicles to change their colors based on the main car's position.
3. **BendMesh_2TexA_Shader:** Accepts two textures—one normal texture and one for masking.

2D_ADAS_TellTalesControl:

This 2D control includes telltales controls and text value. It triggers the rendering of 2D bitmap icons according to the ADAS control conditions, such as altering the Traffic Signs.

This control is optional and can be utilized alongside the 3D ADAS control in a 2D scene.





Revision #15
Created 2024-06-11 00:21:54 UTC by Kanai Tomoaki
Updated 2026-08-19 02:04:34 UTC by Tsuyoshi.Kato