Dossier de Production

Peugeot E-3008 Vehicle Configurator

Mohamad ITANI
ICAN 5 Animation Temps Réel
2024-2025

Summary

Contents

Project Overview:

Pitch:

An immersive and interactive vehicle configurator developed in virtual reality, showcasing the Peugeot E-3008.

Synopsis:

This project presents a real-time vehicle configurator for the Peugeot E-3008, set within a virtual environment. The user can interact with the vehicle and personalize key features including body paint, trim levels, and wheel types. while experiencing the car at full scale.

Note of Intent:

The aim of this project is to create an immersive vehicle configurator experience in virtual reality. The central objective is to allow users to dynamically discover the configurations offered by Peugeot for their E-3008 in an immersive and interactive manner. In this virtual world, the user can navigate in the showroom and interact with the vehicle. This experience highlights the potential of VR configurators as a powerful tool for automotive applications, particularly in digital showrooms.

Technical Sheet:

Project Theme:	Vehicle Configurator
Vehicle:	Peugeot E 3008
Experience Duration:	~5 minutes
Rendering Software:	Unreal Engine 5.6
Medium:	Virtual Reality (Open XR)
Portability:	Windows Executable (.exe)
Recommended Specs (GPU):	Nvidia RTX 4070 Super
Development:	Visual Scripting (Blueprints)

About the Vehicle: Peugeot E 3008

Unveiled in September 2023, The Peugeot E 3008 represents Peugeot's latest generation of electric vehicles. It features a fast back SUV design and integrates the brand's latest design language, making it a suitable vehicle for showcasing in a VR configurator.

Vehicle Preparation Methodology, Asset Pipeline:

Pre-Unreal Engine Import:

The original 3D model of the vehicle was delivered in CAD format via Deltagen, which is not directly supported by Unreal Engine.

In order to integrate the vehicle into Unreal Engine, the geometry had to be converted from surface-based CAD data to polygonal geometry using tessellation. The vehicle was carefully prepared using Autodesk Maya, ensuring no loss of data, including geometry, materials, and textures.

Post-Unreal Engine Import:

The optimized vehicle model was imported into Unreal Engine using the Interchange Pipeline due to the intuitive control it gives over the build settings of the assets that are being imported. After import, the vehicle's components were organized into modular Blueprints, allowing individual parts to be managed and controlled efficiently.

(DeltaGen)
CAD Vehicle Model

3DEXCITE

-Tessellation

FBX

(Autodesk Maya)
Mesh Vehicle Model

MAYA

-Clean up
-Object optimization
-Material organization

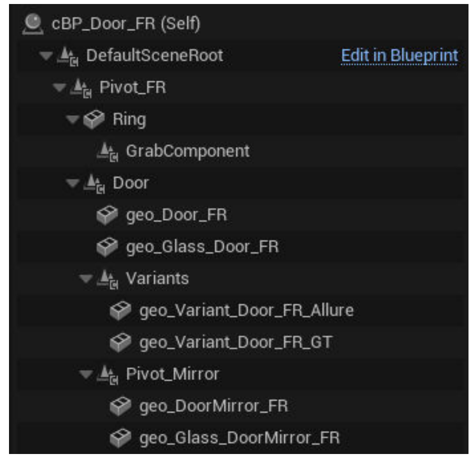
FBX
Interchange Pipeline

(Unreal Engine 5)

-Material correction
-Vehicle Rig

Vehicle Rig Set Up:

In order to set the rig of the vehicle, moving parts of the vehicle were separated in their respective blueprints and were parented to Scene Components (transforms), providing precise control over pivot placement and transformation. Since the vehicle is made up entirely of rigid components, using a skeletal mesh rig would be unnecessary and inefficient, This decision reduced both performance overhead and development time, while maintaining the required functionality.



Materials and Textures:

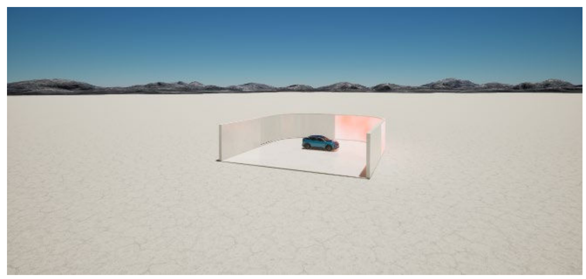
The vehicle model initially contained around 107 materials, most of which were available in the base file, though some were missing. Since the original Deltagen materials used a 3DS PBR workflow, several parameters were lost during the conversion process (from Maya's Phong to Unreal's HLSL). As a result, many shaders required adjustments in Unreal to achieve the intended visual accuracy. Missing textures were also rebuilt where necessary: the HMI (Human-Machine Interface) was recreated in Adobe After Effects, while additional textures were redesigned in Substance Designer.

Environment Building:

The environment built around the vehicle is inspired by Peugeot's visual branding and common industry practices for vehicle presentation.

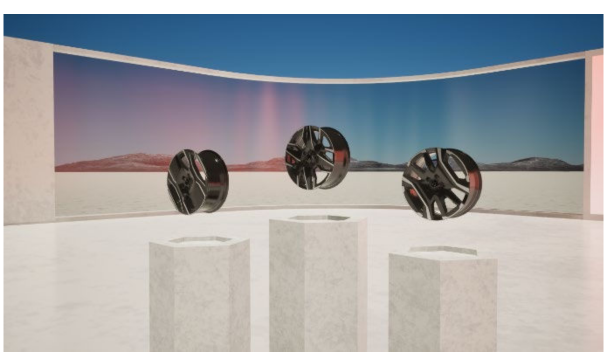
References:

The environment features the iconic Bonneville Salt Flats background that is commonly used in automotive projects for its minimalistic and neutral tones.



The showroom contains a large LED display, projecting a soft pink hue across the vehicle, giving it a contemporary showroom effect.

It also has a glass wall that provides a 360° view of the Salt Flats background from within the showroom, making the environment appear more spacious.



For the variant switching stations, For the Car Paint, Trim Levels, and Wheels. Interactable elements are integrated into retractable pillars placed around the vehicle. Their retractable design ensures that the car remains unobstructed during the experience.

Vehicle Variants:

Car Paint Variant Switcher: *(Index .1)*

The vehicle supports 6 paint variants:

Variant 1: Obsession Blue

Variant 2: Titanium Grey

Variant 3: Ingaro Blue

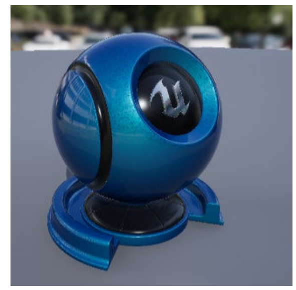
Variant 4: Perla Nera Black

Variant 5: Artense Grey

Variant 6: Okenite White

To create a dynamic and visually engaging transition, a hexagonal dissolve effect was implemented to blend between paint colors.

<ins>Technique Applied:</ins> A dedicated material layer was created for each car paint variant. Then material functions were built to control transition animations between each pair of variants. One master car paint material was created to interpolate between all the variant transition animations. Then the transition animations based on input are applied using blueprints.



<ins>Vehicle Trim Variant Switcher:</ins> *(Index .2)*

Peugeot offers two main trim levels for the Peugeot E-3008, the base Allure trim and a sporty GT trim.

Allure **GT**




Visual transformation Effect:

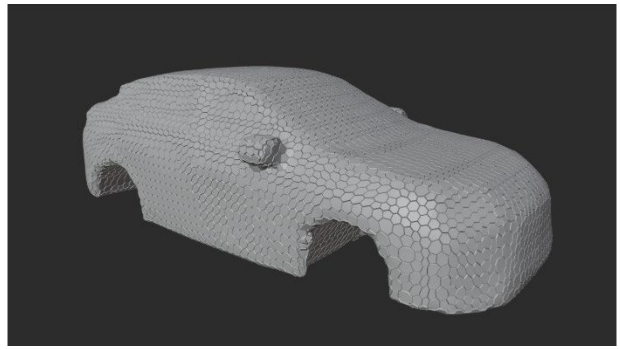
To transition between trims, a particle-based effect was implemented. Particles appear and grow across the areas of the vehicle that change, then disperse to quickly hide the previous trim components. To enhance user anticipation, a highlight effect was added on to the parts of the vehicle that will change.

Technique Applied : The particle effect was designed in Blender using Geometry Nodes, then imported into Unreal Engine as geometry cache. The highlighting was achieved using the Custom Depth Pass, combined with a post-process material in Unreal Engine to emphasize changing components.

Initial Tests (Unapplied):

During development, an alternative transformation was explored using a hexagonal mesh overlay that enveloped the vehicle during switching. While technically functional, this approach was discarded for several reasons:

- From a distance, the effect made the transition between trims difficult to notice as the entire vehicle was hidden during the transition
- The hexagonal mesh appeared like a reptilian skin covering the vehicle, inconsistent with the intended design identity.



Wheel Variant Switcher: *(Index .3)*

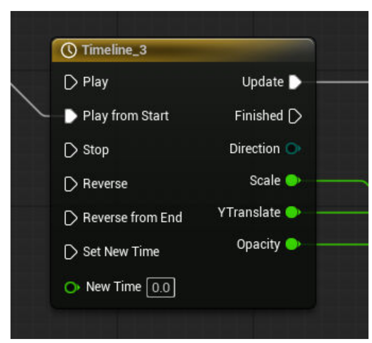
The vehicle has 3 main wheel variations:

Variant 1: Yari

Variant 2: Fuji

Variant 3: Yosemite

Technique Applied: The wheel change animation is managed solely by using the Timeline node in Unreal Engine blueprints, mixing transform and opacity animations to achieve a seamless transition.



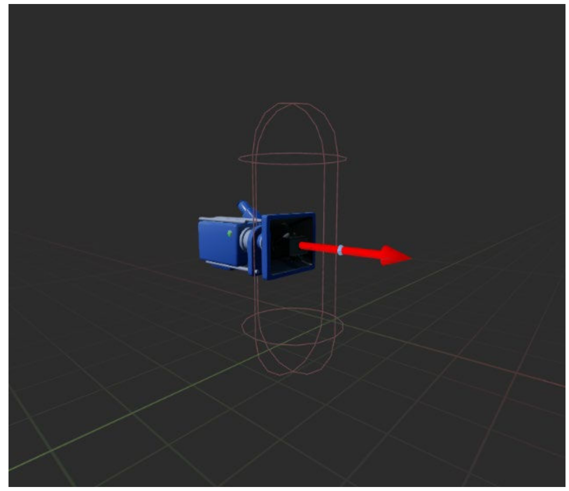
Cinematic Development:

The cinematic teaser for this project was produced using Unreal Engine’s Sequencer. To achieve a higher visual fidelity than the VR experience, I employed the deferred shading pipeline and increased the resolution of shadows and reflections, and enabled Temporal Super Resolution (TSR) for enhanced anti-aliasing and image quality.. The cinematic was rendered using the Movie Render Queue and subsequently composited in Adobe After Effects.

Character Set-up:

Main Pawn: *(Index .7)*

The VR Pawn is a Character class pawn blueprint, meaning it contains a root capsule component, activating physics and allowing it to send and receive collision inputs. The choice of adopting a character class pawn is to allow the user to move in smooth locomotion + room scale, making the experience more comfortable and immersive.



User Input:

Since the project is packaged using the OpenXR API, it offers support for various HMD's including Meta, Varjo, and Vive HMDs. While the inputs mentioned below may vary in name or shape, the interaction remains universal.

The inputs are the following:

Input	**Action**
R Joystick - XY	XY Translation
L Joystick - X	LR Rotation
LR Trigger	Line-Trace Interaction
LR Grip	Grab Interaction

Interaction Set up:

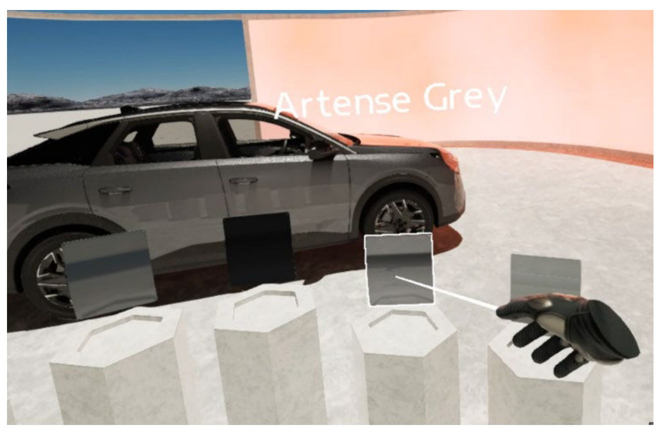
The interaction for this experience is divided into three main methods, Grab interactions, Line Trace interactions, and Collision interactions.

Grab Interaction:

Grab interactions are applied to elements such as the doors and trunk.

To indicate the grabbable components, A circular outline appears on hand proximity, with opacity managed proportional to distance. On grab input, an event launches its corresponding action (Opening, Holding).

Line Trace Interaction: *(Index .6)*



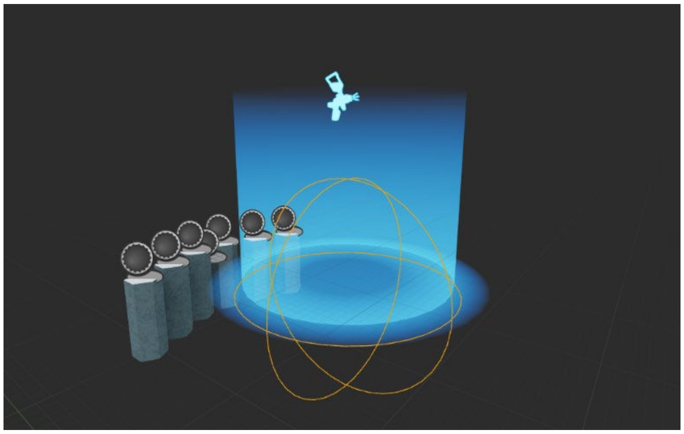
Line trace interactions are applied mainly for variant switching, such as wheel switching, car paint switching, and trim switching.

This interaction is visualized by a ray that is projected from the hand component to the interactable element. The collision of the ray with the element outlines the variant, and upon trigger, the respective effect is launched to simulate the variant change.

Collision Interactions:

Collision based interactions are used to activate the pillars that hold the configuration elements.

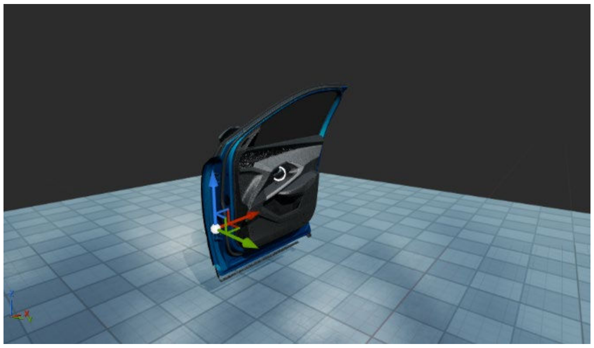
Since the VR Pawn contains a capsule collision, physics can be applied to the character. Colliders are placed near the pillars to launch pillar animations based on pawn collision overlapping.



Vehicle Interaction:

Door Opening System: *(Index .4)*

The door-opening system is central to vehicle interaction, as it also serves as the gateway to the entry system. On door input, the door opens and the user is given the option to teleport into the vehicle.



Vehicle Entry: *(Index .5)*

Once the door is open, the entry input teleports the VR pawn to the respective seat. Player movement is temporarily disabled to avoid collision conflicts. The same technique is applied when exiting the vehicle

Pawn Possess (Unapplied):

An alternative system for vehicle entry was tested using two separate VR pawns: one for outside the vehicle and one for inside, with the user switching possession upon entry. While functional, this approach proved unnecessarily complex and less efficient than the teleport-based solution and was therefore discarded.

Sound Design:

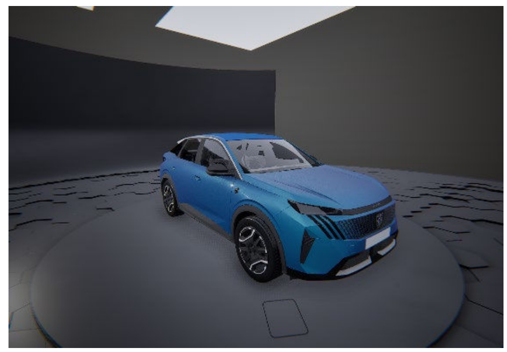
To enhance immersion, sound was integrated into the experience. Sound design was implemented primarily through Unreal Engine's Play Sound at Location and Play Sound 2D nodes in Blueprints. Basic sound compositing was performed in Adobe Premiere Pro before importing the audio assets into Unreal Engine. Sounds were applied to the ambience and key interactions such as door opening, variant switching, and UI feedback, contributing to a more engaging VR experience.

Real-Time Rendering Pipeline:

Graphic Intentions:

For this project, my objective was to achieve a convincing level of visual realism in order to enhance immersion and highlight the vehicle's car paint materials. However, balancing realism with performance presented significant challenges. VR rendering is inherently demanding, as it requires computing two high-definition views simultaneously. The following section details how I addressed these challenges through the rendering pipeline.

Forward Shading Tests (Unapplied):



My initial tests used the forward shading pipeline; a lightweight method often applied in VR due to its support for MSAA and reliance on baked lighting and shadows. However, this approach disables key features such as Global Illumination, Ray Tracing, Lumen, and Nanite—essential for dynamic lighting, reflective materials, and handling the vehicle's high-polygon geometry. While the pipeline delivered acceptable performance (35–45 fps), the visual quality fell short of automotive visualization standards.

Deferred Shading Implementation:

The deferred shading pipeline was then applied, enabling advanced features such as Global Illumination, Ray Tracing, Lumen, and Nanite. This significantly improved visual quality but came at a heavy cost: performance dropped to 7–14 fps.

Optimization + DLSS Integration:

To improve performance, rendering parameters were optimized and Nvidia DLSS (Deep Learning Super Sampling) was integrated, an AI-driven upscaling and anti-aliasing technology. DLSS delivers higher visual fidelity without rendering at full native resolution. The result was a great boost in performance, reaching 72 fps.

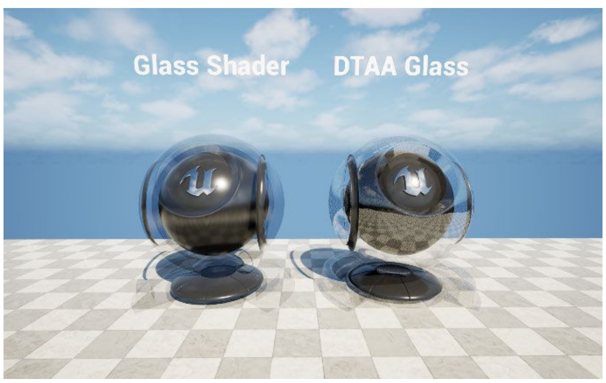
Optimization Techniques and Considerations:

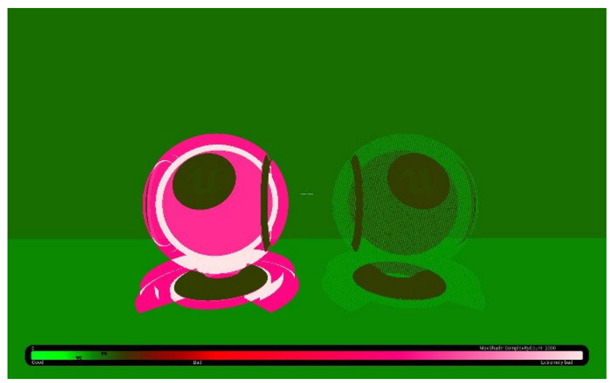
Techniques Applied:

Optimization focused on reducing draw calls, material complexity, VRAM usage, light complexity, and Lumen cost. Unreal Engine 5.6 offers several features that were leveraged, including Nanite, MegaLights, and Virtual Textures. Additional optimizations included the use of unlit materials where appropriate and trim textures to minimize the number of unique materials.

Dither Temporal Anti-Aliasing (Unapplied):

Dither Temporal Anti-Aliasing replaces translucent materials with lighter masked materials by using small, dithered holes to simulate transparency and reflections. While efficient for fading elements, it was not applied for the vehicle windows: the dithering reduced light transmission into the interior, forcing fake lighting that negated the performance benefits.





Index:

1. Material Switcher:

-https://blueprintue.com/blueprint/33ctxa0o/

-https://blueprintue.com/blueprint/8j242b1-/

2. GT-Allure Switcher:

-https://blueprintue.com/blueprint/vvg41wcg

3. Wheel Switcher:

-https://blueprintue.com/blueprint/ej3i68zy/

-https://blueprintue.com/blueprint/zt4l91jf/

4. Door Setup:

-https://blueprintue.com/blueprint/xtuiis4m/

5. Vehicle Entry:

-https://blueprintue.com/blueprint/m2ompto_/

6. Line Trace Interaction:

-https://blueprintue.com/blueprint/h7525za2/

-https://blueprintue.com/blueprint/0_xl29q2/

7. VR Pawn

-https://blueprintue.com/blueprint/e0x83343/