

Building a Driving Simulator with UX at the Forefront – Part 1

7月 26, 2022

By Gabriel Haas, Senior User Experience Project Manager

Have you ever wondered how we build and refine our products in realistic driving situations without putting real drivers at risk? In this two-part series, I will provide insights into the Cereence DRIVE Lab and give a snapshot of some of the ways we are creating amazing user experiences for our automaker customers. As our team wears many hats, we bring a lot to the table, including creating rapid prototypes of new products, showcasing our current products, and collecting user feedback. One such example is how we use our driving simulators to demonstrate Proactive AI, including the skills that were required and applied by our DRIVE Lab team to build this real-world driving environment, and what we'll be doing next.

The virtual environment and driving simulation

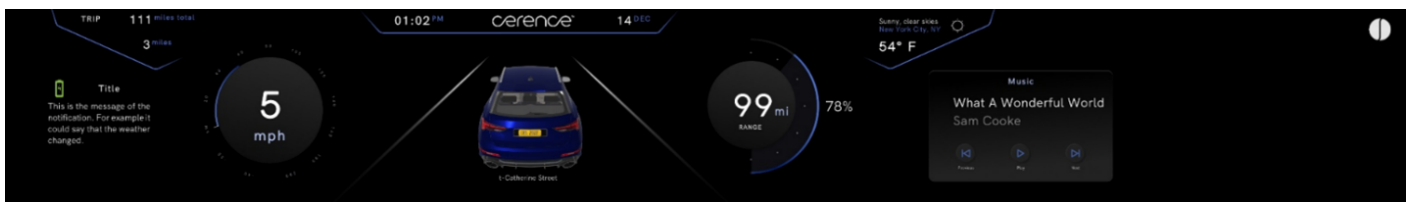
Our setup includes a physical driving simulator and a virtual environment in which the car can be freely driven around. This environment was created with the Unity Engine; originally developed for games, this engine is now widely used for various real-time 3D applications ranging from virtual product configurators and digital twins to building the metaverse in virtual reality. We used it to create a virtual copy of the roads in a section of Manhattan (from 88th to 93rd street and from Central Park Avenue to Broadway), an ideal selection as it accurately represents an urban environment with a lot of traffic and challenging situations. This allowed us to integrate external navigation solutions and send the "real" GPS position of our virtual car to the navigation software. We added virtual buildings, trees, and a realistic traffic simulation including cars, traffic lights, and street signs to bring the virtual city to life. This environment provided diverse traffic situations and included several intersections, one-way streets, and multi-lane roads. Not only did we copy Manhattan, but we also added some interactive elements such as electric charging stations and underground parking for evaluating or showcasing our products' features.

In some cases, we were able to rely on ready-made modules, but in others, we had to develop our own solutions. This was only possible because our [DRIVE Lab team](#) has a wide range of skills and expertise that, among many other things, includes 3D-content generation, and software development. For instance, to be always in control of the virtual environment, we had to create a remote controller capable of changing and manipulating almost any aspect of the environment in real-time. With a mere press of a button, we could change from sunny weather to a snowstorm, drain the car's battery, display messages to the driver, or even place the whole car in a different location.

Integration of sensors and components in a flexible network architecture

To have the previously described possibilities, each of the simulator's components must be talking to each other. The remote controller is, as any other component, connected via MQTT, a network protocol for machine-to-machine communication that is managed by a central broker. This means that the sender and receiver do not have to know each other. The sending components just publish their data and all recipients with interest in that specific topic get the data delivered because they have registered as subscribers. This gives us the flexibility to integrate or exchange all kinds of devices into the overall system very quickly and easily. Try to affect the mood of the driver? Add some RGB-LED strips. Want to experiment with gesture controls? Add a Leap Motion controller. It's that simple.

In addition to our sensors (steering wheel and pedals, microphones for natural language understanding, eye trackers to track where a driver is looking, and cameras for driver monitoring), the dashboard and navigation tablet were also connected to the driving simulation via MQTT. To display navigation instructions on the tablet, we installed an application that received the GPS position of our virtual car. This allowed us to start the navigation in any navigation software such as Google Maps, Here, TomTom, or other providers. As the main method to communicate with the driver, the dashboard was one of the most important components of the driving simulator. It displayed the driving-related data of the vehicle such as the speed and remaining battery range, other relevant information such as time and weather, infotainment with a music player, and notifications related to the drive.



The interactive dashboard, built by our well-rounded UX team.

These bits of information can be textual, but as an example, we also experimented with messages spoken to the driver by a virtual avatar. This dashboard was entirely conceived, designed, and brought to life by our DRIVE Lab team. Again, this in-house development (or even in-team development) gives us the flexibility to create novel driver-vehicle interfaces for today and the future.

Key assets of a creative and strong UX team and strategy

These abilities within the DRIVE-Lab and the power to adapt and customize a driving simulator to our needs in both, software, and hardware, enables three major use cases I'll cover in more detail in part two of this blog:

- Showcasing our products in a realistic and tangible driving experience
- Rapid prototyping of new products and components
- High fidelity user studies in a controlled environment

In addition, we are in constant contact with our R&D and product departments to ensure that our findings are incorporated into our end products and

that we are addressing practical and realistic applications. This way, the entire company benefits directly and indirectly through increased visibility to the outside world, accelerated product development, and the early integration of real user feedback.