From Interaction to Cooperation: a new approach for human-machine interaction research for closing the out-of-the-loop unfamiliarity

Farbod N. Nezami[1‡], Maximilian A. Wächter[1‡*], Nora Maleki[1], Philipp Spaniol[1], Lea M. Kühne[1], Anke Haas[1], Johannes M. Pingel[1], Linus Tiemann[1], Frederik Nienhaus[1], Lynn Keller[1], Sabine König[1], Peter König[1, 2*], Gordon Pipa[1*]

‡ Shared First Authorship

* Shared Senior Authorship

Affiliation 1: Institute of Cognitive Science, University of Osnabrück, Osnabrück, Germany, mwaechter@uni-osnabrueck.de

Affiliation 2; Department of Neurophysiology and Pathophysiology, Center of Experimental Medicine, University Medical Center Hamburg-Eppendorf, Hamburg, Germany

Abstract:
With the introduction of autonomous vehicles, drivers will be able to engage in non-related tasks while being driven. But in critical situations the car needs the support of the human driver. How do distracted drivers get back into the control-loop quickly when the car requests a take-over? To investigate effective take-over actions, we developed an interactive virtual reality experiment, that uses premises of the embodied cognition theory. Accordingly, the car should not only provide sensory input, but also help enhance the driver's motor response by interpreting intention and thus helping to accomplish desired actions. This binds humans and machines together in becoming true cooperation partners in joint action. Therefore, we aim for a close monitoring of participants combined with sensorimotor feedforward and feedback. The presented prototype also serves as an open-access, cost-efficient toolkit that enables interested researchers to tailor the presented LoopAR tool to their own needs as part of a previously published toolkit called WestDrive. With the presented work, we hope to shift the paradigm of future research from only visual aids to full sensorimotor integration assistance.

Keywords: VR research, Out-of-the-loop unfamiliarity (OOTLU) Autonomous driving, Human-machine interaction, Take-over Request (ToR)

1. Introduction

The question of what is to be considered for the user-friendly design of automated systems like autonomous cars has been the subject of scientific discussion for decades [1]. One factor for this is that highly automated vehicles will change the demands on the driver's cognitive system in a radical way [2]. The role of humans as physically active decision-makers in vehicles is being replaced gradually by automated systems [3], such as the Lane Departure Warning System (LDWS) [4] or Forward Collision Warning System (FCWS) [5]. With the growing efficiency of technical systems, the aspect of human-centered interaction is becoming increasingly important. Experiences from various domains, such as aviation, where automated systems are already integrated, make it clear that the safety and reliability of such systems cannot be achieved by optimizing technical components alone [6]. Rather, the reliability of automated systems is largely determined by the quality of the interaction between a human and a machine as an intertwined process [7,8]. The automated system should therefore include representations of both interaction partners to ensure cooperation.

This applies in particular to situations where humans have the task of taking over system control in the event of sensor failures or malfunctions summarized under the term "Out-of-the-loop-unfamiliarity" (OOTLU) [9,10]. Thus, investigating the fluent integration of the take-over request (ToR) is of crucial importance for the safety of autonomous cars [11]. At the same time, new skills like system supervision and a new understanding of the system must be learned [12]. This is due to the fact that the human's new roles have to be represented in the autonomous vehicle in such a way that its requirements correspond both to the expectations and needs of humans and to the abilities of the human information processing system. Thus, in critical situations, time and precision in driver actions are of essence to minimize take-over time.

In a take-over request scenario, the driver has to take over immediately, even when not engaged in driving-related activities. Naturally, an orientation phase follows as the human driver has to assess the traffic situation [13]. Unfortunately, the driver's reaction is often too slow in accident situations, as there is only a small time frame of less than four seconds before an impact occurs [14,15]. Even in case of fast reactions within said brief interval, studies have shown merely hectic responses by human drivers, which of course did neither improve the situation nor the outcome [11]. As an alternative to improve and shorten the take-over time and support actions relevant to the situation, we suggest to warn the driver in advance and supply targeted information while discounting potential distractors to ease the orientation phase as well as facilitating the driver's reaction.

Investigating optimized driver-car interaction, however, requires a new toolset. In the presented paper, we developed a human-machine interaction (HMI) that provides the information of a take-over request in audiovisual form as sound system outputs and augmented reality (AR) features in the windscreen. The developed HMI displays the take-over request and highlights critical traffic objects, to enable participants to take over more quickly and recognize the danger faster than without an HMI. Also, we hypothesize that the steering and braking reaction should improve with the information given by the HMI, which should be expressed by less amplitude and frequency in movements during a critical traffic situation.

Current approaches to this problem rely on hints giving additional sensory input for the driver [16,17,18]. To close the full loop, however, relying on additional sensory input for the driver is not sufficient. The unaware driver may react inappropriately, e.g. turning the steering wheel uncontrollably or braking excessively. The framework of embodied cognition suggests that sensory and motor actions are bidirectional [19,20,21]. In the context of take-over requests, this means that the motor response of a driver changes perception and vice versa. Therefore, to effectively target the out-of-the-loop-unfamiliarity, future HMIs have to not only provide additional sensory input, but should also actively promote and interpret the motor response of a driver. The consequence would be that the drivers have the possibility of being fully in control, while also having the machine assist them in their motor response.

Our design will provide a new approach for human traffic behavior research, as we combine VR and eye-tracking investigation in an interactive and immersive scenario, where eye-tracking data serve as an AR feedforward visual stimulus directly into VR. Additionally, we will propose how to close the full control-loop of sensory input and motor response, introducing machine learning-based likelihood calculations in the following section. This setup allows a more valid, cost-efficient and mobile experiment by adding the paradigm of sensorimotor contingencies to future HMI research to foster full sensorimotor integration assistance [22].

Because of COVID-19, it is currently not possible to perform experimental investigations. Therefore, we will present a detailed description of our project design and its features as an open-access toolkit for further VR HMI studies. Since the LoopAR project is based on the project WestDrive [23], all code needed as well as the designed scenes are available in a Github repository. This enables interested researchers to create their own experiments with the WestDrive LoopAR toolkit. In order to fully use the project presented here, a powerful computer, VR glasses, a simulation steering wheel and pedals, as well as Unity as a development program are required. A more detailed list of the requirements and features is given below.

2. Materials and Methods

In the following section, a detailed overview of the to be performed experiment is given. The experiment consists of two separate phases. The first phase of the experiment acts as the training phase of the project, in which behavioral data of the test subjects as well as outcomes in critical traffic situations are recorded. The training phase is focused on a multi-modal audio-visual warning system which not only provides all relevant information to the sensory loop, but also allows close monitoring of the test subjects. The car is controlled by the MotorAI which does not interfere with test subjects' actions but controls the car in non-critical traffic. The experimental procedure will be described under 2.2 in more detail. The resulting data of the test subjects in critical traffic situations will be used to shape the car's AI belief about desired actions and outcomes (see Fig.1).

In the second phase, the car AI will be trained with the behavioral data from phase one, and a new set of participants will go through almost the same experiment as phase one. Additionally to the first phase, the now trained AI belief will - together with the participant's input from phase one - be fed to the AI cognitive module which will to assist the driver by correcting inappropriate responses. This module is called SenseAI which is different from the steady-state MotorAI in phase one (see Fig.1).

The underlying idea stems from the use of a Bayesian inference (I). It presupposes that the system has prior knowledge of the probability distribution of certain parameters. In the example presented here, the parameters include possible actions (e.g. braking or steering) of the driver or the autonomous car. In Bayes' theorem these actions are defined as the prior. We use the test subject's inputs of the first phase of the experiment in order to define the prior. In the second phase these inputs will be used to update the behavior of the AI car. The AI belief module (see Fig.1) calculates the probability (post-hoc) of a certain driver action p(action) in a critical event. It uses the previous test subject inputs as an update factor (prior) and constantly multiplies it with the drivers input (likelihood) during a critical situation. Since the input is multiplied every instance the participant provides further input, the probability of the action, which is in line with the participant's input, becomes more and more pronounced during the critical situation. Thus, the general update step follows Bayes formula:

$P(action| \ participant \ input) \approx P(participant \ input \mid action) * P(action)$

We hypothesize that in a time-sensitive event, subjects will not be able to respond immediately and appropriately. We assume, for example, that a steering movement will be performed too excessively or too inert. Please note, that is does not imply that the drivers actions are random. Far from that, these overshooting actions allow the AI to differentiate between the more plausible action alternatives. That is, the AI-belief including the participant's data in the first phase will calculate the likelihood of a certain outcome and refine it against the current driver input to estimate a desired outcome. This result will be a slightly smoothed probability distribution, which is then used by the SenseAI module to adjust the driver reaction towards a probable outcome. The SenseAI does not take action on its own, but rather infers the intended action based on its data-driven prior distribution on possible actions, which we called AI belief, and current participant's input.

In the experimental phase, the behavioral output together with a questionnaire will then be used to investigate the effectiveness of this cooperation, checking the outcomes of critical traffic events against the intended outcomes of the participants.

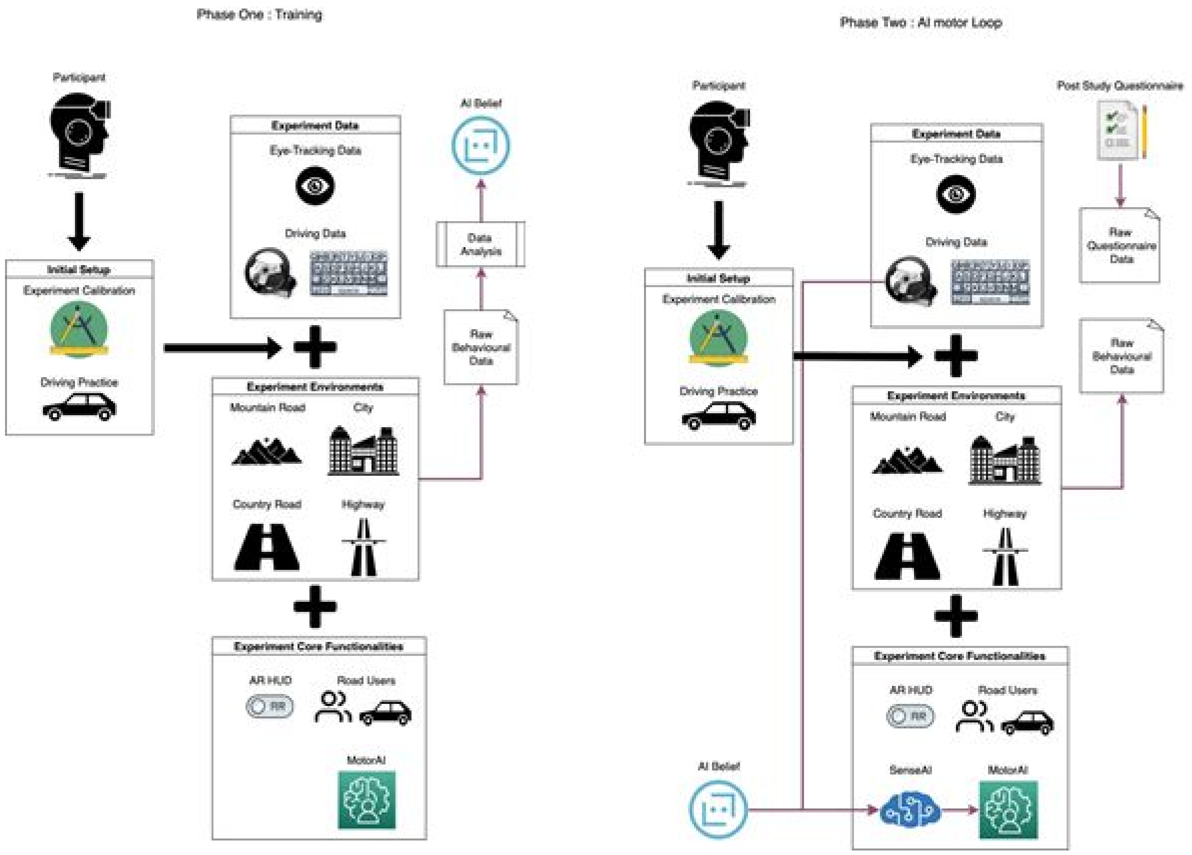


Fig 1. Board view of the two planned phases of the experiment including a training phase for collecting participants' data and a motor loop experiment that includes the training data to shape the output of the SensAI module.

2.1. Apparatus and Setup

For the technical apparatus, we suggest a computer equipped with an Intel(R) Xenon® E5-1607 v4, equivalent or better, 32 GB RAM memory with a 64-bit operating system Windows 10 Professional. An Nvidia GeForce RTX 2080Ti, served as a graphics processing unit. We used the HTC Pro Eye including the HTC base stations 2.0 as a VR setup. To enhance the immersion of participants, there will be a physical model of a car for participants to sit in together with a Fanatec CLS Elite steering wheel with pedals as an input device.

The setup used and presented here has already been tested intensively in other VR studies and has proven to be very efficient in two large studies with over six months of continuous operation. This makes it a cost-efficient and very mobile replacement for maintenance-intensive, rigid and expensive driving simulators. A further advantage is the freedom regarding the selected components. The only requirement for operation is granting the computing power for the entire system. As a virtual reality device, we are using the HTC Vive Pro Eye with an integrated Tobii Eye Tracker. It is a cable bound head-mounted-display that enables the participant to transfer movements into virtual reality. This is done passively, meaning the device is not tracking the surroundings by itself, but relying on laser-emitting “base stations” version 2.0, also provided by HTC. Although we are using the Vive Pro exclusively at our department, the LoopAR experiment is not dependent on this specific VR device and can be used with other setups.

The recommended components of the setup include:

GPU: NVidia GeForce GTX 1080 ti, equivalent or better

CPU: Intel(R) Xenon RE5-1607 v4, equivalent or better

RAM: 32 GB

Video Output: HDMI 1.4, DisplayPort 1.2 or newer. USB Port: 1x USB 2.0 or better port

Operating System: Windows 8.1 or later, Windows 10

VR HMD: Vive Pro -Eye with built-in Tobii Eye Tracker

Game ready Fanatec CSL Elite Steering Wheel and pedals

The achieved frame rates with this setup stay between 45-60

The presented setup allows data collection outside of research facilities by using only a computer with VR glasses. A more detailed description of all included functionalities can be found in the repository mentioned at the data availability statement.

2.2. Experiment Procedure

Our experiment contains four different VR environments, which are described in section three. After the initial eye-tracking and seat calibration, a training trial scene is presented. Here the test subjects are introduced to the virtual car and the HUD (see Fig. 2). Participants are asked to take over manual control and maneuver through an obstacle course to complete the training. After successfully completing the training scene, the experimental scenes sequence starts. At the beginning of each scenario, the car drives fully autonomous within the legal speed limits, which are displayed in the head-up display (HUD) (see Fig.2, picture 1). In correspondence to the car speed, the scenes vary in length from 1,200m in the city scene to 3,600m on the highway.

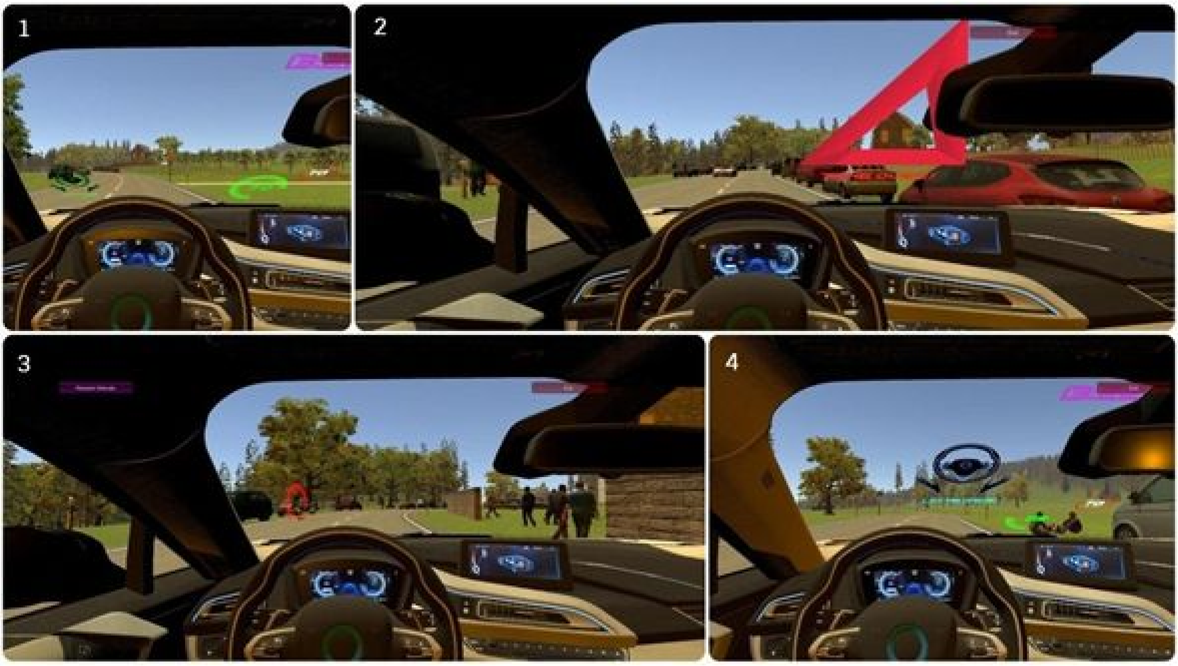


Fig.2: Pictures of the different stages of the HUD in a critical traffic event 1. fully automated driving condition, 2. Warning the driver and requesting a take-over 3. Highlighting the potentially dangerous obstacle while discounting distractors, 4. Switching to automated driving.

Each of the scenes is loaded separately to enable a recalibration of the eye-tracker in case the accuracy decreases due to participants' head movements [25]. Every scene entails three critical traffic situations, in which the test subjects are required to take over control of the car. After completing the trials in all four scenes, participants are asked to fill out a questionnaire on demographic data, the level of immersion, their previous VR experience and their previous hours of gaming, which concludes the experiment.

Another group will be recruited to complete the trials without the audiovisual warnings in the head-up display as a control group. In this condition, participants will be warned once via a beeping sound 2.5 seconds before the onset of a critical traffic event.

2.3. Stimuli and Critical Traffic Events

Stimuli

As stimuli we use information within the augmented reality head-up display provided by the autonomous vehicle (see Fig.2). To decrease the take-over time of the driver, we give visual and auditory warning signals. The stimuli appear, when a take-over request from the car is announced. Therefore, the take-over request starts with a beep followed by a verbal warning [24] accompanied by a red warning triangle (ISO 3864) on the windscreen (see Fig.2, picture 2). Following the initial warning. For a higher saliency, the visual warning sign is blinking for a period of two seconds to cue the attention of the driver [26,27]. While the warning sign is flashing, a lettering informs the driver of the take-over request in a written form. The warning sign disappears after two seconds to give the driver an unobstructed view of the situation, only marking possible obstacles (see Fig.2, picture 3). Finally, the car will grant control to the human driver 0.5 seconds later. The take-over request is finalized and a small icon in the lower-left corner indicates that the driver is in full control (see Fig.2, picture 1).

Critical Traffic Events

To assess the test subject’s performance in terms of take-over time and reaction precision, we investigate driver reactions in a total of 12 critical traffic events. A take-over request is demanded when a car detects a critical situation that an artificial agent cannot resolve correctly without human intervention. This could be a running deer, jumping in front of the car, or a broken car blocking the road. These events depict situations, in which the car is not able to solve the situation within the legal boundaries. One example of a legal boundary is that an autonomous car is not allowed to cross a continuous line on the road, which might be necessary in case an obstacle blocks the car's lane. In this case, the control should be handed into the human driver. In both of our experimental phases, we will, therefore, compare the outcomes of the critical traffic events as well as the quality and time of the driver’s response. Once the critical traffic event is overcome, the car suggests a reinitialization of the autonomous driving mode in the form of a clearly visible pictogram accompanied by text in the windscreen (see Fig.2, picture 4). Another beep sound (with lower frequency) indicates a successful transmission of control back to the car. The event zone prefab is designed to be easily customizable for different critical traffic event scenarios and environments. For more detail on that see section 3.6 and 3.7.

3. Virtual Environments and Main Functionalities

In order to test our new approach in human-machine interaction, an interactive and realistic scenario is needed. LoopAR aims at a fully immersive experience of an autonomous car encountering critical traffic events, where eye-tracking data serves as an AR feedforward visual stimulus. Additionally, to test the machine learning-based likelihood calculations described in section 2, we created a broad variety of scenes and traffic events. In the following section, the decisions regarding the environment design will be elaborated together with a short description of the four experimental scenes. This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

3.1 Environment Design

The LoopAR environment is based on real geographical information of the city of Baulmes in the Swiss Alps. The region was selected due to its variety in the terrain, including a small village, a country road, a mountain pass, as well as a region suitable for adding a highway section, totaling in around 1000 hectares of environments and an 11km drive on different roads (see Fig.3). We decided to include these environments because urban areas, rural areas and highways are the three main areas of car transportation. These areas demand different driving skills to an automated driving vehicle as well as to a human driver to react in dissimilar situations with different conditions according to a feature of the landscape and traffic rules. To make the region accessible in Unity, we used the collaborative project OpenStreetMap (OSM) (OpenStreetMap Foundation). We used the open-source 3D software Blender [28] to import the region and to extract the data. This provided information about vegetation, different street types, bodies of water and buildings on the terrain, which we could use as references to build a close-to-reality, natural scenery.

3.2 Environmental Scenes

To follow an efficient workflow, we sliced the terrain into four areas (see Fig.3). Due to the road network design, these separate environments can be merged. Developing terrain parts independently from each other decreases the computational overhead for design and experiment purposes significantly.

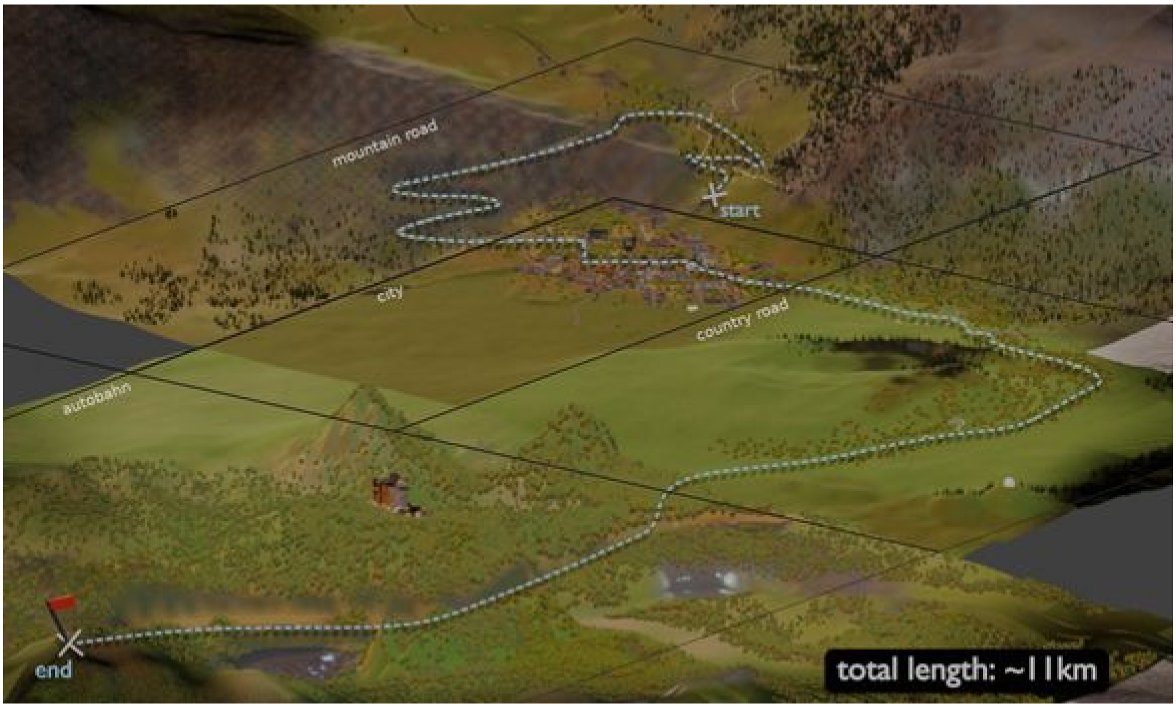


Fig.3: LoopAR map preview: mountain road 3,4km, city 1,2km, country road 2,4km, highway 3,6km.

The first scene the participants will drive through is the mountain road scene (see Fig. 4a), which includes curvy roads winding through a forest and steep serpentines running down a mountain. This requires a variety of different driving speeds (from 30km/h or slower, up to 100km/h on straight stretches). The overall traffic density is low. The scene features three critical traffic events: a deer crossing the street, rocks blocking the path after falling onto the road and dense fog, which impede the sensors of the car as well as the sight for the human driver.

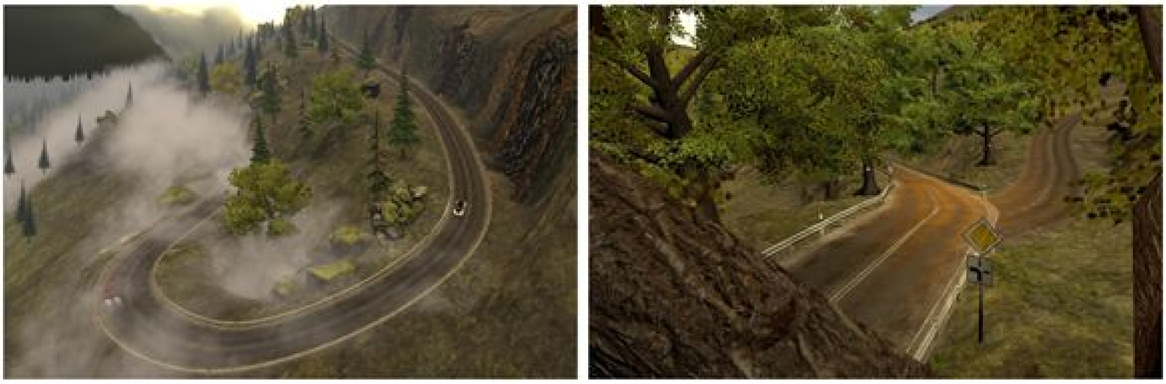


Fig.4a: Pictures of the different scenes: Mountain Road

After the Mountain Road the participants will drive through the village “Westbrück” (See Fig. 4b). Here we are able to test possible events in a more inhabited environment. Critical events in the village environment include an ambulance car with sirens, people playing next to the road and a construction site.

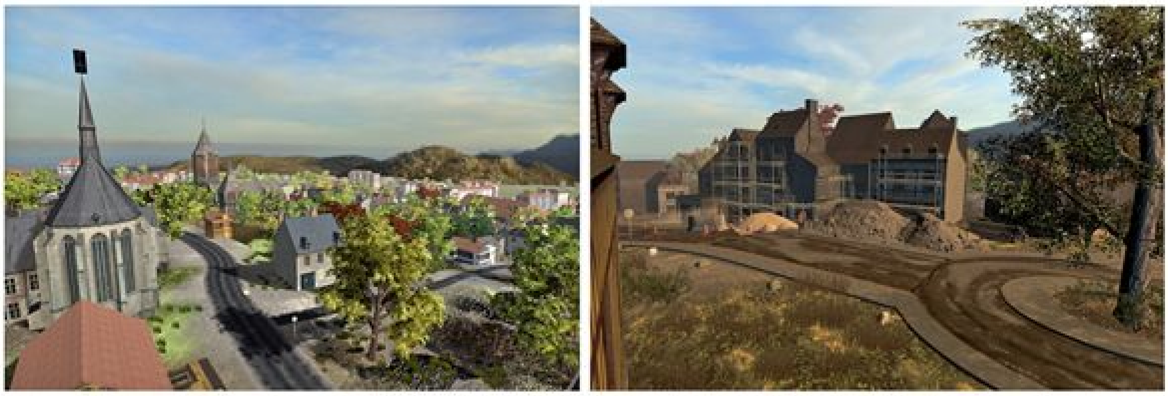


Fig.4b: Pictures of the different scenes: Village “Westbrück”

The third scenario is the country road scene (see Fig. 4c) which was designed for medium to high speed (~70km/h), medium traffic density and features three critical traffic events. Firstly, the participant is required to avoid hitting a person who is illicitly walking onto a street by steering onto the neighboring lane. Secondly, they must avoid running over a cyclist violating traffic laws by not giving the right of way. Lastly, they have to dodge a motorcyclist who is blocking the autonomous car´s lane by coming to a halt on a riverbank.

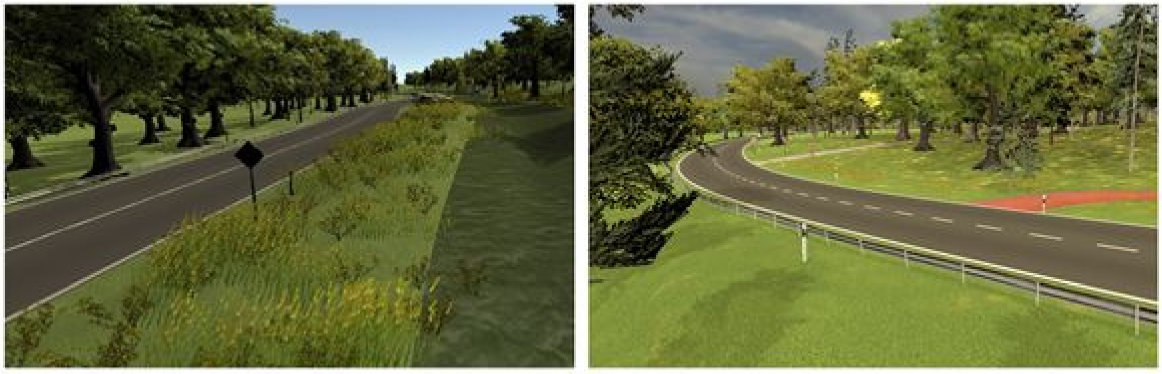


Fig.4c: Pictures of the different scenes: Country Road

The last scenario for the participants is the highway scene (see Fig. 4d), enabling critical traffic events with a higher speed and medium traffic density. The scenario involves one critical event in which the human driver has to take over control in order to quickly avoid another driver headed in the wrong direction. The scene features two more events in which the autonomous car will warn the driver of danger, but the participant is not required to take over the controls, such as a broken-down car on the side lane and a construction site.

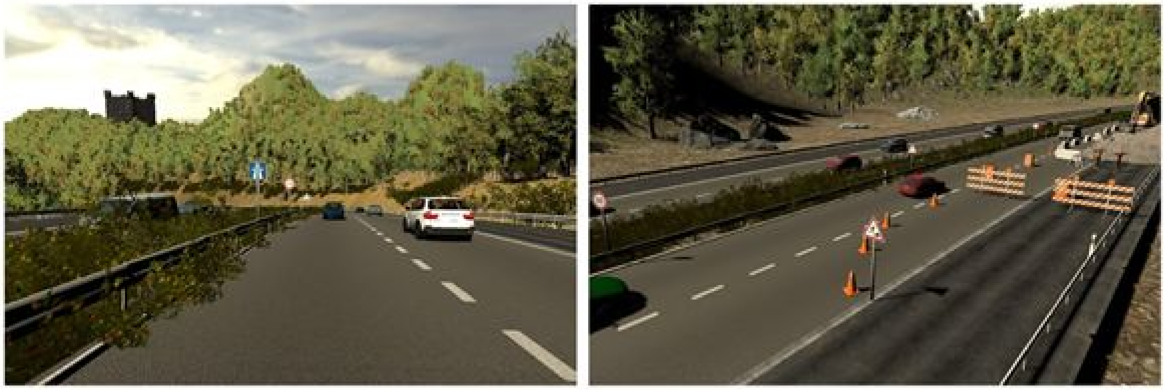


Fig.4d: Pictures of the different scenes: Highway

3.3 Adaptation Possibilities

There is an entire network of streets that connects the scenes. A large part of the assets used are from Unity's own asset store and the 3D platforms Sketchfab and Turbosquid. Each of the four scenes can be customized anytime. It is therefore possible to change the number, size and shape of all objects in each scene. Like previous work of the WestDrive project, LoopAR was developed for a special investigation but serves as a foundation for a variety of VR research. One application for this can be spatial navigation tasks in VR. In essence, "Spatial navigation is the process by which organisms use multiple cue sources such as path integration, magnetic cues, landmarks, and beacons to determine the route to a goal and then travel that route." (Brodbeck, 2012). Although it is not the focus of our experiment, we included aspects like landmarks to the virtual environment, which will allow us to use the environment for different experiments such as pointing tasks, route recall tasks, or distance estimation tasks.

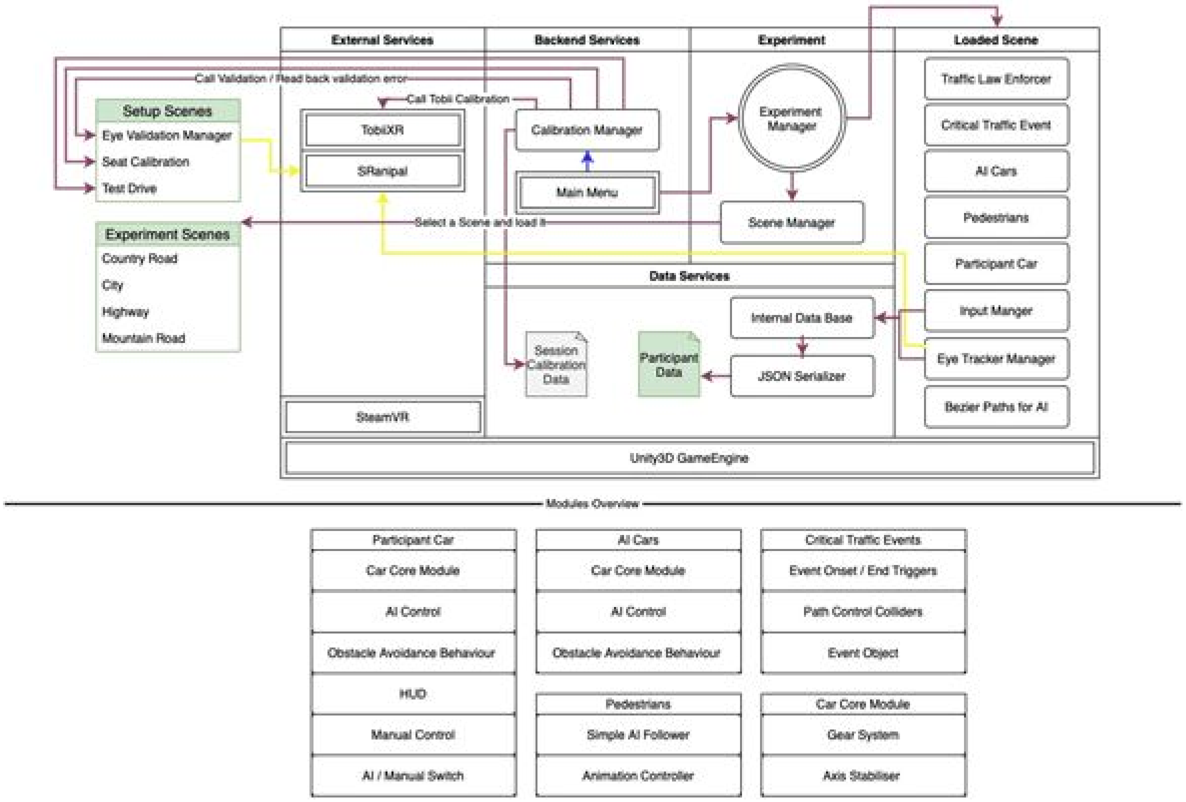


Fig.5: Scheme of the LoopAR functionalities and components illustrating the interaction of the different services and manager scripts within the Unity environment.

3.4 Platform

LoopAR is built in Unity Editor 2019. 3. 0f3 (64bit). This software is a game engine platform based on C# by Unity Technologies which supports 2D, 3D, AR & VR applications. The Unity editor and the Unity Hub run on Windows and Mac, as well as Linux (Ubuntu and CentOS) and built applications can be run on nearly all commercially usable platforms and devices. Unity also provides a large variety of available application programming interfaces and is compatible with numerous VR and AR devices (Juliani et al., 2018). Additionally, Unity enables the developers to access the Unity asset store, which is a large library of various assets such as 3D Objects and scripts that can be used to create versatile projects.

The backend code of the project LoopAR was developed entirely using C# within Unity3D Monobehaviour scripting API (see Fig5). The backend comprises functionalities including dynamic loading of the environment, AI car controls, pedestrian controls, cutscene event controls, car windshields augmented reality controller, data serialization and eye-tracking connection. LoopAR code has been developed with modularity in mind to avoid complicated and convoluted code. These functionalities can be enabled or disabled individually using UnityEditor's graphical interface based on need. In the following section, the authors give an overview of various modules implemented in LoopAR. Figure 5 provides an outline of all functionalities and components. The following paragraphs will be a list of the most important components developed by the team.

3.5 Cars and Traffic Behavior

All cars used are based on the Unity standard wheel collider systems of the Unity3D physics engine. In the Car Core Module, user input is translated towards the motor control of the participant's car. The input consists of the motor torque, brake torque and steering, which were applied to the wheels. In addition to that, the Car Core Module can be accessed similarly by the AI Control Module. This allows a seamless transition from autonomous to manual driving. Furthermore, a gear system simulates varying torque on the wheels, increasing stability in steep road sections.

Separate AI behavior modules handle the traffic behavior of the cars. The AI is able to follow predefined paths. Speed limit triggers inside the scene manipulate the AI's aimed speed, which in turn is handling the input propagated to the Car Core Module. Another behavior allows the AI cars to keep distance from each other. Currently not implemented is an obstacle avoidance behavior. The goal is to create an easily configurable and interchangeable traffic AI for multiple study designs. With these measures we maximize the realism of the car physics and the traffic simulation while ensuring easy adjustments.

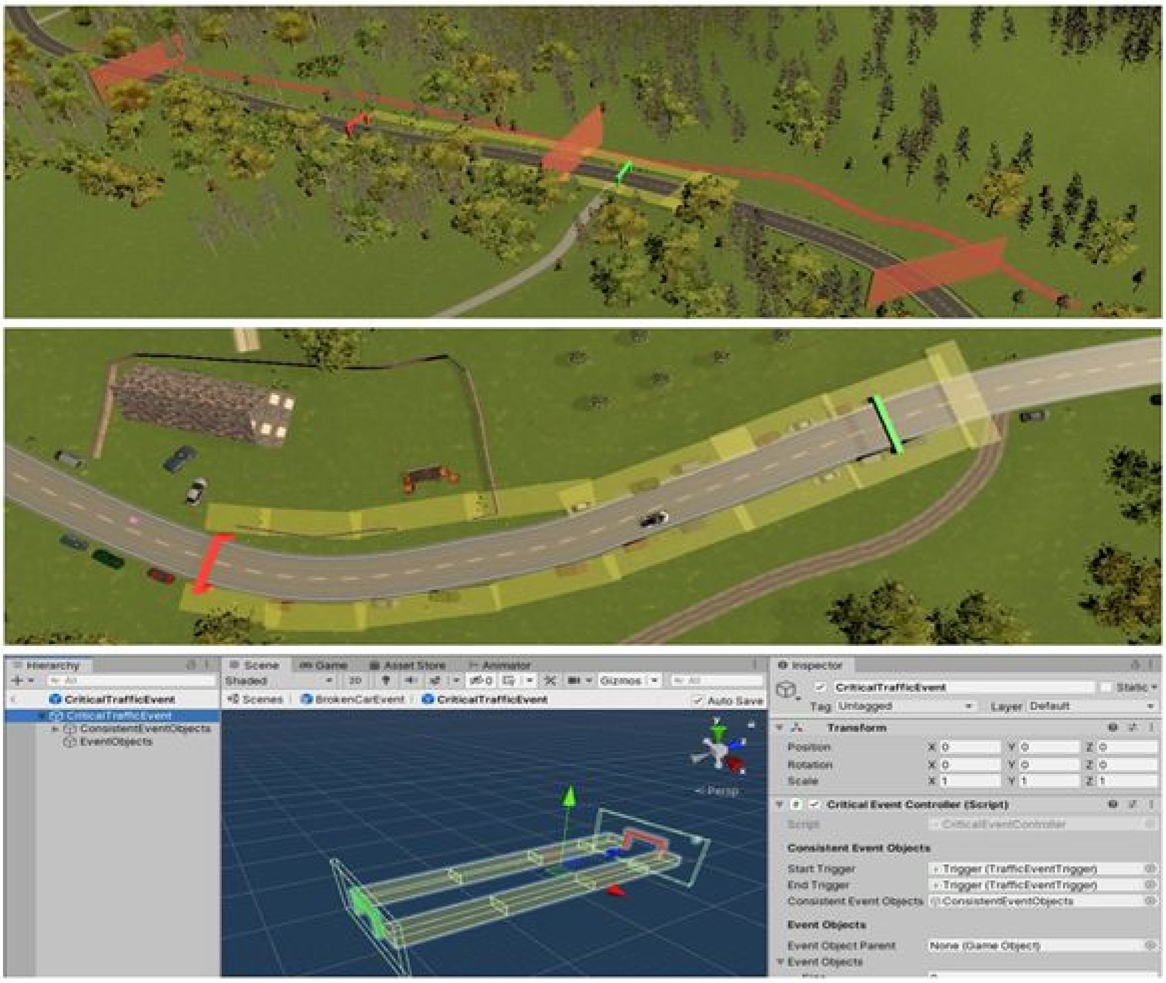
Paths followed by AI Cars as well as walking pedestrians were defined by mathematical bezier curve paths [29] which were realized by a creation tool from previous westdrive Projects. The appropriate tool is currently under refinement and will be part of the LoopAR toolset

3.6 Experiment Management and Initial Adjustments

The Experiment manager starts and ends the experiment and will handle scene relevant information and settings shortly before and during the real experiment phase. It handles different Camera settings, the information given by triggers inside the scene, and handles the respawn of the participants in case of failure. Before the Experiment starts, initial Adjustments were made which consists of Eye Calibration, Eye Validation, Seat Calibration as well as a Test scene which prepares the participant for the experiment. For a detailed look, consider the provided supplementary material

3.7 Critical Traffic Events

The critical traffic events are realized by several trigger components. These independent triggers are activated when the participant's car enters the start trigger (Fig.7: green gate), The event zone is restricted within triggers (Fig.7: yellow boxes). These triggers get activated if the car crashes into them which is considered as participant's failure and leads to a black screen followed by respawning the car at a point after the event (Fig.7: pink box) and giving back the control to the car. Furthermore hitting the marked objects by the HUD also leads to a respawning. The successful scenario is when the participant enters the end trigger (Fig.7: red gate) without crashing. These triggers are all visible in editor mode and invisible to the participant.



CriticalTrafficEvent

Tag Untagged Layer Default

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Script CriticalEventController

Consistent Event Objects

Start Trigger Trigger (TrafficEventTrigger)

End Trigger Trigger (TrafficEventTrigger)

Consistent Event Objects ConsistentEventObjects

Event Objects

Event Object Parent None (Game Object)

Event Objects

Fig.6: Critical traffic event prefab with an implementation example in the country scene

3.8 Resource Management and Computational Performance

The size of the project requires resource management at changing scenes. This means that not all 3D objects are visible all the time for a test subject in the experiment to reduce the computational load on the hardware. A dynamic scene loading component eases transitions between various environments, which is currently under refinement. Several components like the Experiment Manager or the LOD system are related to the loading of resources. The delayed loading ensures the aimed stable frame rate of 45 frames per second (FPS), as well as disabling unnecessary compartments.

In order to improve the computational performance, we have not only limited the regions of the used OSM map, but adjusted the used 3D objects within every scene as well. High frame rates are of special interest for eye-tracking research and also function as a preventive measure of motion sickness [30]. Adding details like 3D objects, such as high-resolution buildings, cars, vegetation and humanoid avatars gives the scenes a more vivid and realistic look, and potentially more immersion. However, such details are prone to drop the frame rate. It was therefore important to find a balance between getting the best possible performance and providing a highly realistic environment. To achieve this, we made use of different Levels of Detail (LOD). The LOD technique is a built-in Unity functionality using “several Meshes and optionally a Billboard Asset, which all represent the same GameObject with decreasing detail in the geometry” (Unity Technologies) with larger distances of the object to the camera. A billboard follow the gaze of the camera it gives an impression of three-dimensionality. This reduces the computational load on hardware, increasing the rendering output. In our case, a custom billboard shader has been developed and was used on trees within the environment, making it possible to achieve higher fidelity without a notable sacrifice in performance.

3.9 Eyetracking

The Eyetracking Component comprises an eye-tracking calibration, validation and online gaze ray-casting, which is able to record necessary gaze data during the experiment. The component is built for the Tobii HTC Vive Pro Eye device, but is intended to keep the VR component interchangeable. Currently it is intended as a simple connector to tap into SRanipal and the Tobii XR SDK (see Fig. 6). The eye calibration is performed with the built-in Tobii eye calibration tool. The validation is set in the corresponding validation scene which provides a simple scenario with a fixation cross. Validation fails if the validation error angles exceed an error angle of 1° or the head was moved by 2” from the fixation cross. During the experiment, the information about the eye orientation, position and collider hits will be stored with a calculated gaze ray of both eyes. Currently it is set to receive information about any object inside these rays, to prevent the loss of viable information by objects covering each other.

3.10 Behavioral Data / Input

Gathered experiment data consists of input data of the participant’s input source, as well as eye-tracking data. Scene relevant information like the number of failed critical traffic events will be added shortly. All data is stored using generic data structures and Microsoft Linq, serialized into JavaScript Object Notation (JSON) and saved with a unique ID at the end of each experimental block. The generic data structure used in the project ensures flexibility, as different data types can be added or removed from the serialization component. A calibration manager-script stores information about the seat calibration, eye validation error, and relevant test-drive results of every participant. Data from participants’ inputs during the human drive as well as eye-tracking information during the complete course of the experiment are stored using generic data structures and Microsoft Linq, being serialized into JavaScript Object Notation (JSON) data format and saved with a unique ID at the end of each experimental block. The generic data structure used in the project ensures flexibility, as different data types can be added or removed from the serialization component. This approach guarantees the highest compatibility with different analysis platforms such as R or Python for the data gathered with LoopAR.

4. Discussion

One of the most important features in a VR experiment is the feeling of presence in virtual environments, which refers to a subjective perception of illusion (e.g. place illusion or plausibility illusion [31]). The foundation to achieve this is a stable and high frame rate, which also avoids motion sickness for subjects The desired optimum for the experiments would be a stable frame rate matching the fixed rate of 90Hz that is used by the manufacturers HTC and Oculus. Our current frame rate in the different scenes varies between 40-60 frames per second due to the large, highly detailed scenes and the nature of Unity as a game engine and its extensive workaround. Nevertheless our experimental design is fully functional with its current frame rate, even though we strive to improve it for an even better experience in VR.

The concept of presence plays another important role for the validity of VR studies. Recent VR technology potentially allows a high degree of presence, but still differs fundamentally from real life (Pan et. al 2018). The feeling of presence can manipulate the participant's behavior, and thereby harm the results of this research. For example, the participants might not feel integrated well enough into the scene and thereby might not induce the appropriate action. In addition, presence can differ, depending on the experience with virtual environments of the user (e.g. differences between gamers and non-gamers in emotional responses [32]. To keep this on track the demographic questionnaire will check for VR/gaming familiarity. Even though differences to real-world experience have to be considered, investigations in virtual environments provide more natural experimental conditions than classical lab experiments.

Another aspect of the experiment that needs to be considered is the handling of objects and the possible replications of the visual stimuli for each participant. In the preceding experiments of WestDrive, we used kinematics for all dynamic objects to create a lower computational load, a stable visual input over each experiment and to facilitate changes in the scene. Due to the interactions necessary for the current experiment, we needed to access the physics engine of Unity, which does not allow the same amount of control as using kinematics for dynamic objects, but provides on the other hand a more realistic environment, which improves the feeling of presence.

Even though there are some limitations, we can still provide a functional experimental design, able to test the efficiency of our designed HUD in critical events, where a take-over request is needed.

5. Outlook

In the presented paper, we developed LoopAR as an independent experiment, to test the take over of control in critical traffic situations from autonomous cars to human drivers. The emphasis is on a realistic environment with high immersion and the AI supporting and interpreting the sensory as well as the motor leg of driver – car interaction. In its current state and design it provides a promising new approach to human-machine interaction based on the theory of embodied cognition. Therefore, it not only gives the test subject optimal sensory input, but also assists the participant in the motor behavior to ensure fast and precise response of the human driver in case of a take-over request.

LoopAR aims to present a new approach for further development of HMIs, by combining VR and eye-tracking in an interactive and immersive scenario, where eye-tracking data serves as an AR feedforward visual stimulus directly into VR. Additionally, we proposed an experimental design on how to introduce machine learning-based likelihood calculations that enable motor aid. It is thought to fundamentally change research of safe and effective communication between car and driver. This is not only beneficial in terms of safety for the passengers, but could also increase customer acceptance of highly automated vehicles, since up until now malfunctions are key concerns of possible customers [33]

In addition, this project serves as a fully functional toolkit that can be used to easily adapt the current code and 3D environments to create own experiments in VR. Due to the open-access functionalities of the project, it can have a significant influence on the way future VR studies are conducted. This accounts especially for the new functionalities regarding the motor assistance, but also for the provided HUD inside the car. With the newly implemented code, it is not only possible to simulate a large and highly realistic VR environment, but it is also possible to create specific stimuli for the driver, enabling researchers to investigate the human-machine interaction in a valid, mobile and very cost-efficient setup.

The datasets for the asset foundation and scripts can be found in the WestDrive repository

https://gitlab.com/farbod69/project-westdrive

The dataset for the LoopAR project can be found under:

https://github.com/farbod69/Westdrive-LoopAR-Public

Supplementary Materials:

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions:

FNN and MAW wrote this paper. Both authors designed the project. SK, PK and GP supervise the project. Building, testing and proofreading was done by our supervised students. Furthermore: NM developed major parts of the AI and Functional Modules and User Interfaces. JP was involved in scene building and the HUD functionalities. LK designed the mountain road scene and provided performant assets. LT developed large parts of the software architecture and acted as a software engineer for functional compartments of the project, AH designed the highway scene and provided additional assets. LMK designed the country road scene, provided assets and contributed to HUD related literature background. PS was involved in designing the city scene, managing and creating assets. FN developed and designed the TestDrive Scene.

Funding:

This work is funded by the University of Osnabrück in cooperation with the graduate college “Vertrauen und Akzeptanz in erweiterten und virtuellen Arbeitswelten” (FNN), as well as from the GMH foundation https://www.stiftung-stahlwerk.de/home/ (MAW)

Conflicts of Interest:

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Norman, D. A. (1990). The ‘problem’ with automation: inappropriate feedback and interaction, not ‘over-automation’. Philosophical Transactions of the Royal Society of London. B, Biological Sciences, 327(1241), 585-593.

2. Li, S., Blythe, P., Edwards, S., Goodman, P., & Hill, G. (2019). Investigation of the influence of multitasking on drivers’ takeover performance in highly automated vehicles. In 26th Intelligent Transport Systems World Congress. Newcastle University.

3. Lindgren, T., Fors, V., Pink, S., & Osz, K. (2020) Anticipatory experience in everyday autonomous driving. Pers Ubiquit Comput. https://doi.org/10.1007/s00779-020-01410-6

4. Kim, H. J., & Yang, J. H. (2015). Design of LDWS based on performance-based approach considering driver behaviors. Journal of Institute of Control, Robotics and Systems, 21(11), 1081-1087.

5. Abe, G., Itoh, M., & Yamamura, T. (2011). Effective and acceptable forward collision warning systems based on relationships between car-following behaviour and reaction to deceleration of lead vehicle. In Human modelling in assisted transportation (pp. 155-164). Springer, Milano.

6. Maurer, M., Gerdes, J. C., Lenz, B., & Winner, H. (Eds.). (2015). Autonomes Fahren: technische, rechtliche und gesellschaftliche Aspekte. Springer-Verlag.

7. Marberger, C., Mielenz, H., Naujoks, F., Radlmayr, J., Bengler, K., & Wandtner, B. (2017, July). Understanding and applying the concept of “driver availability” in automated driving. In International Conference on Applied Human Factors and Ergonomics (pp. 595-605). Springer, Cham.

8. Altendorf, E., Schütz, R., Canpolat, Y., Weßel, G., & Flemisch, F. (2017, July). A Study on the Human and the Automation in Automated Driving: Getting to Know Each Other. In International Conference on Applied Human Factors and Ergonomics (pp. 672-683). Springer, Cham.

9. Endsley, M. R., & Kiris, E. O. (1995). The out-of-the-loop performance problem and level of control in automation. Human factors, 37(2), 381-394.

10. Merat, N., Seppelt, B., Louw, T., Engström, J., Lee, J. D., Johansson, E., ... & McGehee, D. (2019). The “out-of-the-loop” concept in automated driving: Proposed definition, measures and implications. Cognition, Technology & Work, 21(1), 87-98.

11. Melcher, V., Rauh, S., Diederichs, F., Widlroither, H., Bauer, W., (2015). Take-Over Requests for automated driving. In 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015

12. Bainbridge, L. (1983). Ironies of automation. Automatica, 19(6), 775-779.

13. Merat, N., Jamson, A. H., Lai, F. C., Daly, M., & Carsten, O. M. (2014). Transition to manual: Driver behaviour when resuming control from a highly automated vehicle. Transportation research part F: traffic psychology and behaviour, 27, 274-282.

14. Summala, H. (2000). Brake reaction times and driver behavior analysis. Transportation Human Factors, 2(3), 217-226.

15. Green, M. (2000). "How long does it take to stop?" Methodological analysis of driver perception-brake times. Transportation human factors, 2(3), 195-216.

16. Laughery, K. R., & Wogalter, M. S. (2006). Designing effective warnings. Reviews of human factors and ergonomics, 2(1), 241-271.

17. Large, D. R., Kim, H., Merenda, C., Leong, S., Harvey, C., Burnett, G., & Gabbard, J. (2019). Investigating the effect of urgency and modality of pedestrian alert warnings on driver acceptance and performance. Transportation research part F: traffic psychology and behaviour, 60, 11-24.

18. Cohen-Lazry, G., Katzman, N., Borowsky, A., & Oron-Gilad, T. (2019). Directional tactile alerts for take-over requests in highly-automated driving. Transportation research part F: traffic psychology and behaviour, 65, 217-226.

19. Clark, A. (2006). Vision as dance? Three challenges for sensorimotor contingency theory.

20. Noë, A. (2009). Out of our heads: Why you are not your brain, and other lessons from the biology of consciousness. Macmillan.

21. O'Regan, J. K. (2011). Why red doesn't sound like a bell: Understanding the feel of consciousness. Oxford University Press.

22. Kaspar, K., König, S., Schwandt, J., & König, P. (2014). The experience of new sensorimotor contingencies by sensory augmentation. Consciousness and cognition, 28, 47-63.

23. Nezami, F. N., Wächter, M. A., Pipa, G., & König, P. (2020). Project WestDrive: Unity City With Self-Driving Cars and Pedestrians for Virtual Reality Studies. Frontiers in ICT, 7, 1.

24. Veitengruber,J. E., Boucek, G. P., Smith, W. D. (1977). Aircraft alerting systems criteria study, Volume 2

25. König, S. U., Keshava, A., Clay, V., Ritterhofer, K., Kuske, N., & König, P. (2020). Embodied Spatial Knowledge Acquisition in Immersive Virtual Reality: Comparison of Direct Experience and Map Exploration. bioRxiv.

26. Okada, T., Miyamoto, Ti, Ito, S., Sato, K., Ikeda, N., Sueda, O., & Fujisawa, S. (2018). Research on Blinking-Luminescence Travel Support for Visually Impaired Persons. 319-324. 10.1007/978-3-319-60441-1_32.

27. Endsley, M.R., Bolte, B., & Jones, D. (2003). Designing for Situation Awareness: An Approach to User-Centered Design

28. Blender Foundation (2002). Blender Homepage. Retrieved May 14. 2020. from https://www.blender.org/

29. Choi,J. & Elkaim, G. H (2007). Bezier Curve for Trajectory Guidance, Autonomous Systems Lab, Computer Engineering Department, University of California, Santa Cruz.

30. LaViola, J. J. (2000). A discussion of cybersickness in virtual environments. ACM SIGCHI Bulletin, 32(1), 47–56. https://doi.org/10.1145/333329.333344

31. Slater, M., Perez-Marcos, D., Ehrsson, H. H., & Sanchez-Vives, M. V. (2009). Inducing illusory ownership of a virtual body. Frontiers in Neuroscience, 3(SEP), 214–220. https://doi.org/10.3389/neuro.01.029.2009

32. Geslin, E., Bouchard, S., & Richir, S. (2009). Gamers' versus non-gamers' emotional response in virtual reality. Journal of CyberTherapy & Rehabilitation, 4(4), 489–493. Retrieved from https://www.researchgate.net/publication/291095383_Gamers'_versus_non-gamers'_emotional_response_in_virtual_reality

33. Howard, D., & Danielle, D. (2014). Public perceptions of self-driving cars: The case of Berkeley, California. Transportation research board 93rd annual meeting. Vol. 14. No. 4502.