2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)

A Dynamic and Scriptable Environment and Framework for Stimulus-Based Cognitive Research in Virtual Reality

Stéven Picard* Jean Botev† Sahar Niknam‡

VR/AR Lab
University of Luxembourg

Figure 1: Example scene in the editor with various assets and experimenter-defined, scriptable task zones in different colors.

Abstract

Time perception is essential to immersive media experience, and particularly virtual reality. With the relevant technology becoming both readily available and affordable in recent years, there has been a corresponding growth in interdisciplinary research on time perception and virtual reality. This paper presents a fully customizable virtual environment and framework devised for such studies, which can furthermore be easily extended to accommodate any stimulus-based cognitive and behavioral research. The different elements of the environment can be defined in simple text-based configuration files to load and deploy new components and experimental setups quickly. Due to the generic architecture, experimenters can also control all elements externally via hardware-agnostic network messages.

Index Terms: Human-centered computing—Virtual reality Human-centered computing—User studies Software and its engineering—Software development techniques—Reusability

1 Introduction

In recent years virtual reality (VR) has become increasingly accessible and revealed itself as an excellent tool for behavioral studies, providing experimenters with an easy way to reproduce fully controlled environments and tasks across participants. However, creating these requires much technical effort, so making this work reusable across different experiments is a valuable resource.

Within the ChronoPilot project [4], which aims at active subjective time modulation, a substantial part of our activities is focused on stimulus-based behavioral studies of the user experience and time perception. Therefore, we developed a dynamic and scriptable environment and framework that, while tailored to the specific needs of our group in terms of time perception studies, can support any type of behavioral research scenario employing VR.

Before focusing on the framework itself, the following section will cover relevant concepts and notions related to time perception and its possible applications, VR behavioral studies, and existing frameworks. We will then discuss the experimental design of our environment, specifically its controls, behaviors, and settings, and explore the technical aspects that enable scriptability and dynamics.

2 Background and Related Work

Time perception can be considered from a design point of view as it is an integral part of the user experience with an application or an activity. Studies have been trying to identify ways to use the non-accurate perception of time to the user's advantage; for instance, a faster-loading animation (such as a rotating circle animation) yields a more compressed time perception than slower ones [24]. A similar example is related to downloading data which, according to Gorn et al.'s study testing fake download web pages that differ in color, that parameter appears to influence relaxation, which in turn influences the perceived download speed [10].

As such, different conditions or stimuli affect time perception aspects. For instance, when it comes to the setting, environments tend to have temporal cues or "zeitgebers" such as clocks or the sun's movement. Schatzschneider et al. investigated the latter's effect on time perception with and without cognitive loads as a zeitgeber in a virtual environment (VE). They found that the absence or presence of sun movement, as well as the presence of a task, affect time perception; however, the zeitgeber's speed did not affect time perception, and if its presence affected time perception, it did not affect task performance [21]. Davydenko and Peetz's work furthermore suggested that walks in nature felt longer than in an urban environment, showing that the environment affects time perception and nature's properties influence mood and behavior [6].Other conditions that can affect time perception can be tied to specific properties of stimuli. In a series of experiments, Droit-Volet et al. observed

*e-mail: steven.picard@uni.lu
†e-mail: jean.botev@uni.lu
‡e-mail: sahar.niknam@uni.lu

DOI 10.1109/ISMAR-Adjunct60411.2023.00084

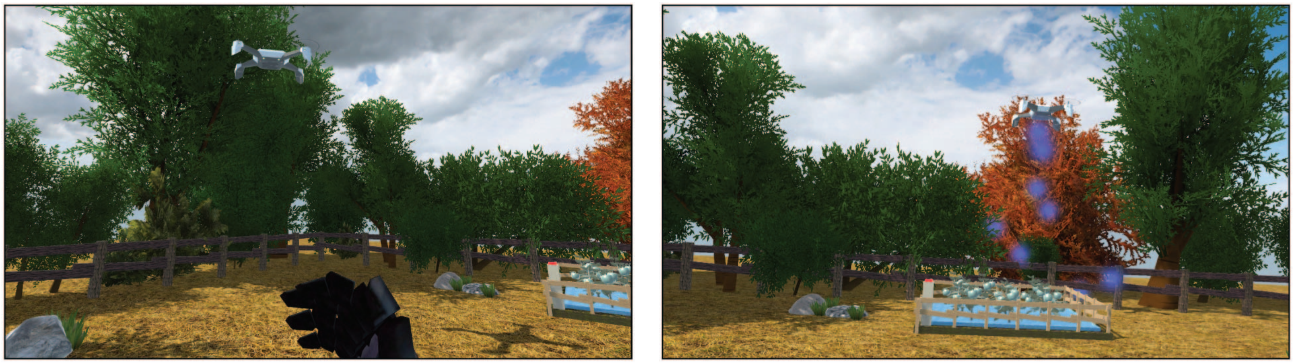

Figure 2: Drone moving towards a zone (left) and watering the plants in the target zone (right).

how higher tempos induce longer subjective time, and emotional valence decreases (but does not suppress) the effect of tempo while affecting time perception [7].A recent study by Hammerschmidt et al. [11] explored different timing evaluations (reproduction, estimation, and subjective rating) of instrumental excerpts of Disco songs at different tempi, revealing interesting tempo properties regarding time perception. Non-rhythmic stimuli properties also affect time perception such as motion. For instance, a study by Fornaciai et al. suggests the adaptation (i.e., the effect of repetition) of fast translation motion compresses time, but this effect was not observed for radial and circular motions [8].

VR, particularly when experienced through fully enclosing head-mounted displays (HMDs), is an ideal medium for psychological studies and testing new stimuli. A notable example of the potential is how Outram et al. employed artificial synesthesia in VR to demonstrate the possibilities of a full-body vibrotactile haptic suit using vibrations and synesthetically linked visuals, audio, and vibrations to increase immersion and satisfaction. The authors suggest that "as being highly enjoyable, the use of such environments may have implications for the exploration of altered states including flow, meditation and ecstasies, and the next stages of related research will be to measure these psychological aspects" [3]. Another example of artificial VR synesthesia can be found in Reif and Alhalabi's study [20], where it was used to control attention and increase immersion in the context of pain therapy. This possibility of immersing the users in fully controllable environments makes VR a perfect tool to investigate stimuli-based perceptual time transformations.

Because creating VEs to study these stimuli is software intensive, various frameworks to facilitate the development of user behavioral studies have emerged in recent years. Open-source frameworks like the Unity Experiment Framework (UXF) [5], the Biomotion-Lab Toolkit for Unity Experiments (bmlTUX) [2], or the vexptoolbox [22], aim to alleviate the coding work for experimenters wanting to use VR-compatible 3D engines such as Unity or Vizard for their experiments. These frameworks primarily provide tools to simplify or automate generic workloads typical to most behavioral studies, such as trial generation and sequencing, independent variables conditions, or data collection. Some frameworks, e.g., the Unified Suite for Experiments (USE) [25], have more advanced features like replaying trials, millisecond precision, and hardware integration. For time perception studies specifically, Landeck et al. created an Unreal Engine 4 framework mainly focused on zeitgebers, i.e., environmental time cues, by giving the option to modify available events on three dimensions (velocity, synchronicity, and density) [12]. These above frameworks aim to either test specific scenarios and stimuli [12] or simplify experimental designs [2,5,22,25]; however, they do not necessarily provide the tools to build a timeline of in-trial events and stimuli, as does our environment framework.

3 Example Design

We will illustrate the framework using an example VE prototype covering an initial precision farming application scenario developed in the context of the ChronoPilot project in order to test various mechanisms and stimuli for subjective time modulation in VR.

3.1 Objectives

The main focus of the VE is on enabling the execution of single-user evaluation tasks. The precision farming scenario naturally integrates time-modulating stimuli and time estimation metrics, emphasizing complex overall problem-solving. Users must perform a series of tasks in the VE characterized by specific deadlines, such as watering, harvesting, and sowing crops. They can assign (virtual) robotic agents, such as drones, to perform these tasks. The various robot types differ in their execution time for tasks, which arrive online, and the user must, for instance, allocate resources to the various tasks to maximize the number of completed tasks. Consequently, cognitive load constitutes one of the main variables in the current scenario, tuned by different parameters such as the number of tasks ready to schedule, the arrival rate of tasks, task criticality, or the number and composition of available resources and solutions. The fundamental idea behind precision farming is to support traditional farming practices with data-driven and AI-enabled technologies (cf. [9], [17], [19], or [27]). With this in mind, precision farming is a promising scenario to realistically test the potential of novel time modulation techniques to improve overall performance and reduce the anxiety of working with advanced technologies altering time perception. The main objectives in developing the VE were its generalizability and scalability while providing an immersive user experience at a practical yet sufficient level.

3.2 Tasks and Task Types

In the experiments and user studies conducted with the current VE, users perform control and learning tasks. The two major VE task types, production and identification, reflect these categories. Experimenters can generally define arbitrary task types depending on what is needed in the specific experimental environment. Tasks can be cumulative, i.e., multiple tasks may be active simultaneously, which allows for adjusting the cognitive load during the experiment and inducing stress or boredom.

The main goal of the production task is to meet a plant's needs and harvest it when it is fully grown. In this task type, the needs of the plants are known a priori, making it essentially a resource-allocation task, with the users deploying drones remotely via a specialized, arm-based interface (see Section 3.2) and assigning the desired parameter values, e.g., to water plants (see Figure 2). Currently, the production task only allows the deployment of drones to irrigate plants, but it can be extended to more dimensions and to include further steps (e.g., powering up the drones).

The command to harvest a crop is given by pressing a nearby button (see Figure 3), with the crop being replaced as soon as the users lift their hand, which forces them to move and navigate within the VE instead of sending drones during the growth phase.

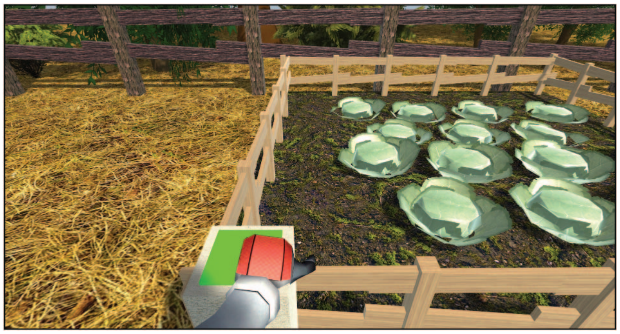

Figure 3: Manual activation of the harvesting sequence.

The user's primary goal regarding the identification task type is to explore and find information about an unknown plant. The retrievable information can be the necessary water levels for a plant to grow or the plant's health status. For this purpose, the user has access to manual controls affecting the properties of the plant; in the present example, they allow to increase or decrease the water level. The plant state can be "reset" by harvesting and planting a new seed. Once the user feels confident knowing the plant, they can answer a questionnaire to complete the task (see Figure 4).

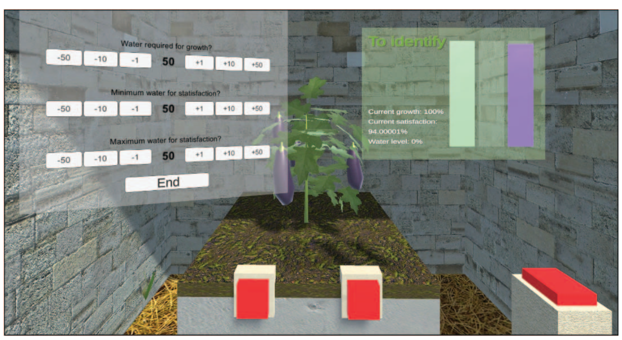

Figure 4: Prefab with plant status panel and questionnaire.

3.3 User Interface and Experience

To provide a compelling, immersive, and seamless user experience (UX) in VR, several aspects must be considered in the user interface (UI) design. Due to the three-dimensional nature of the environment, it is essential to provide spatially consistent feedback that matches the user's position and orientation in the VE. In general, the VE and its embedded UI should be intuitive and easy to navigate, with clear affordances, i.e., cues about where the user should look and where they can interact, employing VR-appropriate interaction methods such as gaze-based selection, hand gestures, and physical controllers. During navigation, relevant contextual information and feedback related to the user's location and actions in the VE must be presented with sufficiently large and clearly visible text and graphics to ensure readability [13, 23]. A specific goal in designing the UI for our test environment is to fully realize the desired level of interaction while making the UI cognitively and physically comfortable to use.

Two main categories of interaction are implemented in the VE: integrated objects in the environment that function as actuators, such as harvest buttons next to the production zones, and a control panel to operate or assist agents and obtain information. Recent human-computer interaction (HCI) studies show that implementing UI elements as integrated objects in the environment reduces cognitive complexity and is consistent with natural modes of behaviour [1]. Consequently, part of the interaction with the environment happens through integrated objects. However, to meet the perspectives of the precision farming scenario, participants need immediate access to the environment's status and remote control of the assisting agents. For this reason, we have devised a control panel that can be activated with a hand gesture performed with an outstretched arm and the open palm facing upward (see Figure 5).While dropping the arm deactivates the control panel, wrist flexion serves as a reset.

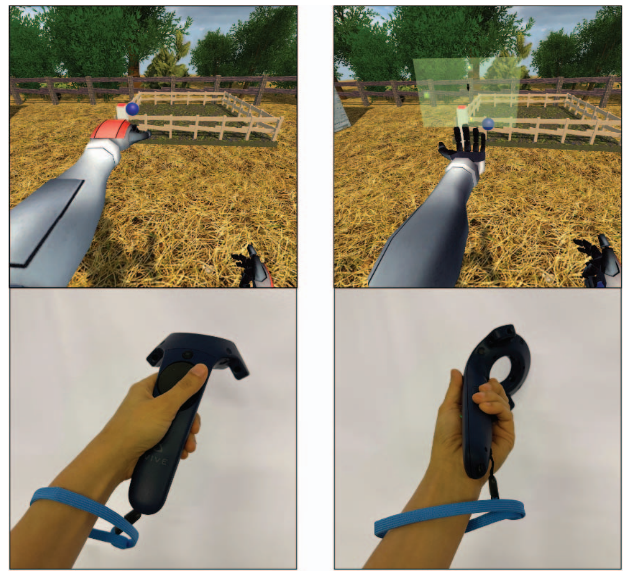

Figure 5: Control panel activation via stretch-and-turn hand gesture.

When designing the panel, instead of having interfaces pinned to the screen that are uncomfortable for the eyes, especially in the corners or during longer interactions, we defined the interface panel as a semi-transparent object that attaches to the (virtual) wrist at a comfortable distance from the eyes [1]. With hand gestures, we wanted to make the overall UX more natural and follow research-supported UI trends and best practices [14–16, 26]. The control panel rolls out progressively to keep complexity manageable, starting with a main map marking the production zones and the user's location. The user can select a zone with a virtual tap. In this case, selected data, such as the plant specifications and soil in the zone, as well as live plant statistics, are first displayed in an attached, smaller panel. Selecting the zone also activates a set of sliders representing the provided resources, such as water, fertilizer, and pesticides. Considering the plant specifications and the live statistics of the zone, the user can then allocate available resources to the selected zone. Selecting a value on at least one of the sliders activates the drone button, which can be tapped to send out one of the drones for resource distribution.

Different levels of UI complexity can be implemented with the current design in line with the experimental requirements. For instance, increasing the volume of presented task-related information is possible, which constitutes the basis of decision-making for the subject, while direct visual feedback can be given, for instance, by animating assets, e.g., to display the plant growth and representing plant states (see Figure 6 with a corn plant as an example).

Figure 6: Animated corn plant asset states; from healthy seed to fully grown (left) and sick (right) plant.

4 Virtual Environment Definition

To allow customization of the VE by different experimenters, we provide a system for flexible environment definition, which we call scene description. Experimenters can create scenes in their Unity project based on our VE and change the environment's layout by moving or replacing visual elements and defining custom zones.

Figure 1 shows an example scene in the editor with different experimenter-defined zones (production tasks in blue, an identification task in green, and another task type in yellow). Zones define physical spaces within the environment and have types and identifiers (cf. Figure 7). The type tells the VE what can happen in the respective zone, while the identifier is mainly used by events (that we will discuss later in this section) to select a specific zone. Once a scene is defined, part of the contents of the scene can be specified at application runtime by user-defined description files that can be divided into the two main subtypes of scene elements and events.

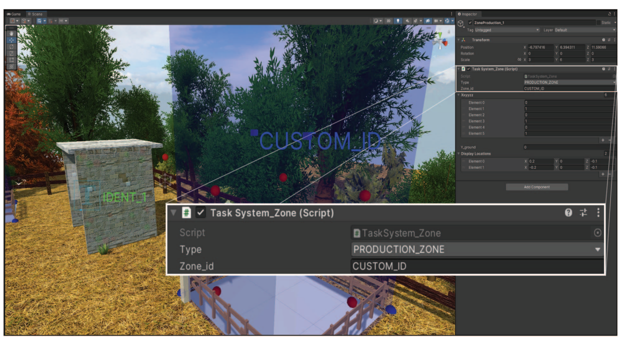

Figure 7: Example zone configuration in editor.

There are three scene element types in the example VE: plants, agents, and stimuli. The description files for plants define a plant template (e.g., how it looks and how much water it needs to grow) that can be used for a task. The agent description files define a user-controlled agent (e.g., a drone for watering plants) created in the scene. Finally, the stimuli description files define a stimulus template (behavior, parameters like frequency or intensity, and so on). All scene element files (see examples in Figure 8) follow a similar structure in which the experimenter describes the scene element identifier (e.g., "id:Plant_1"), the main behavioral specification used with its parameters (e.g., "main:plantwater"), and any sub-behaviors (e.g., "sub:plantbasevisualizer, visualizer:cabbage_production").

Events constitute a more abstract concept. The main idea is that after the scene is loaded, all events are waiting to start; an event can be anything an experimenter wants to happen in the scene. Events produce logs during their lifecycle, which are a combination of a log type and the id of the event that produced it. These logs are a core component of the event system since events have, as a start condition, a list of logs that must be produced before their start. The currently available log types are *START*, *END*, *SUCCESS* (also produces *END*), and *FAIL* (also produces *END*). One log is not produced by an

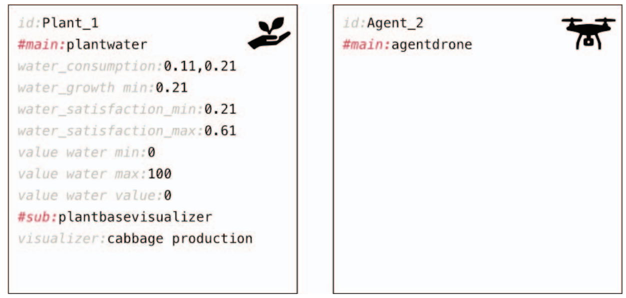

Figure 8: Example descriptions for a plant and a drone.

explicit user-defined event, which is the *TIMELINE_START* once the scene has finished loading. Like scene elements files, experimenters specify an identifier in an event description file and select the main behaviour of the event. In addition, the log values are listed, which represent the start conditions. Standard event behaviours include a timer, production and identification tasks, and starting or stopping a stimulus. While events can refer to other events via the logs that are part of their start conditions, the behaviour of both scene elements and events can refer to other described elements as well as the zones defined in the Unity scene. For example, a "production task" event can start when a timer produces its "end" and employ a user-defined "plant" in a particular "zone" (see Figure 10). Currently, the number of possible behaviors for each element is limited. What is provided is primarily a template or basic framework that can be expanded according on the experimenter's needs.

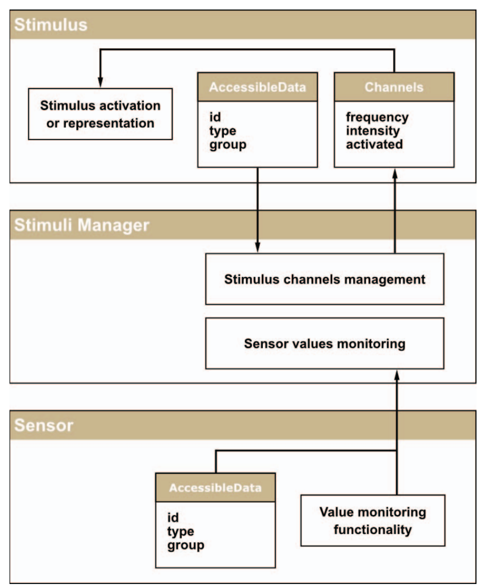

Figure 9: Framework component structure.

5 Component Architecture

To enable the implementation of reusable modulation approaches beyond the current VE and for the entire project, we have developed an underlying component architecture. In order to achieve the desired level of extensibility and flexibility, we opted for a tripartite structure, as shown in Figure 9, consisting of (1) a stimulus component that represents a time-modulating stimulus (e.g., (2) a sensor component that generates values that can be used to modify

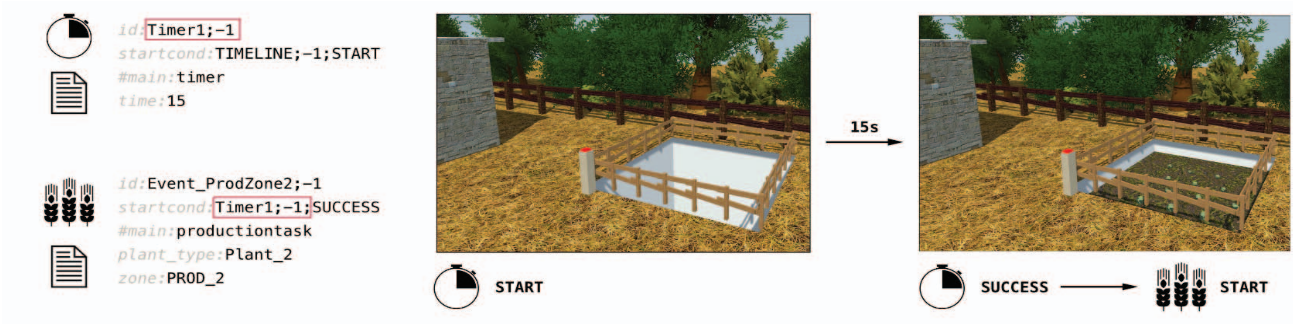

Figure 10: Event-based start of production task from a timer; after 15 seconds, a new production task is initiated.

the stimuli channels (from physiological sensors to more abstract modes such as task progress or time remaining), and (3) a stimuli manager component that processes the values from the sensors to affect the stimuli channels. In other words, the stimuli manager only defines the processing of the sensor values and how this processing will affect the stimuli.

Figure 11 shows an example of the applied component architecture with three stimulus patterns, each using a different modality (audio, visual, or haptic). The patterns have (at least) two common channels: Intensity and Frequency. The visual pattern could, for instance, be a golden flashing screen, as illustrated in Figure 12, which is a modulator type we recently employed in a study related to audio-visual rhythmic stimuli effects on subjective time perception [18]. The frequency would then correspond to the time interval between flashes, while the intensity would indicate the color transparency of the flash. Furthermore, there are three sensors, two of which record physiological values while another monitors task progress, stressing that a sensor component is not necessarily tied to a physical sensor but anything that can produce data output. The two stimuli managers must process and apply these sensor values, each affecting different channels of overlapping stimuli based on the various values.

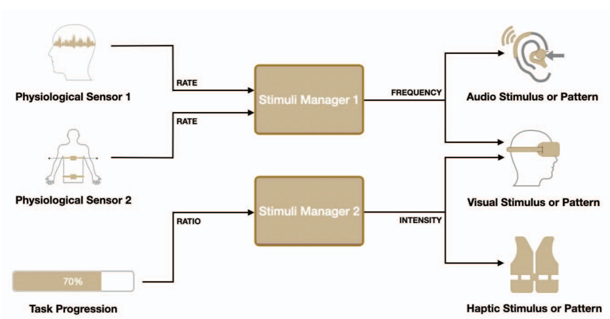

Figure 11: Application example for the generic component architecture; the rates of two different physiological sensors are processed by Stimuli Manager 1 to change the frequency of an audio-visual output stimulus (top), while a task progression ratio feeds into Stimuli Manager 2 resulting in an intensity change of a visio-haptic output stimulus (bottom).

5.1 External Control

Components of the architecture can be accessed externally to control their parameters and influence the VE in real-time. In the current VE, they can be accessed via a UDP interface. Experimenters can connect their own custom software to the VE to retrieve data from Sensors or send their own data "remotely" into the environment. The VE can then interpret and use this new data to control the environmental stimuli.

For example, the screenshot in Figure 12 shows a console application implemented in Python that is connected to the VE, sending sensor data constituting parameters of a color-flashing stimulus, as discussed before. The sensor data received from the console application is then processed by a stimuli manager within the VE, altering the stimulus (intensity/duration) according to the received user input.

Figure 12: External VE control via example Python console application controlling a visual stimulus (golden flashing).

6 Data Collection and Logging Capabilities

As the VE constitutes a testbed, it must integrate ways to collect and record various data in the background related to experimenters' needs. Within a scene of our VE, there are several points where data can be stored: the events system (including explicit data from questionnaires), virtual sensors, and custom behaviors. As described in Section 4, event systems store logs that associate a particular type with an event, as well as an optional message. After a log is processed, it is still available in memory and can then be written out cumulatively at the end of a session and stored in text form. Another possibility is to use a virtual sensor. The experimenter defines these sensors, which are then placed in the scene, and the system automatically retrieves their data. Technically, a virtual sensor is just an object that implements an interface. Virtual sensors can listen to data from real sensors, e.g., external physiological sensors and the eye-tracking data of the VR HMD, or from mechanisms in the environment, e.g., the user's position in the VE or the status of a plant in a zone. Finally, since the environment definition (cf. Section 4) is tied to implementing custom behaviors, these can also be the subject of user-defined data storage points. For instance, experimenters may add a questionnaire at the end of a task. Another example is when experimenters implement a custom stimulus, where it is possible to also add data monitoring in the stimulus behavior.

7 Conclusion

This paper presented a fully customizable VR environment and framework that, while initially developed for time perception studies, can be easily adapted to accommodate any stimulus-based cognitive research and behavioral studies. Time perception and other cognitive aspects are integral to user experience in VR, and such behavioral studies are crucial to understanding the intricacies and cross-effects, ultimately helping in the design of more immersive experiences and purposeful systems. The environment definition can be reused to produce any standard experimental setup and extended with custom tasks, while the external control system allows environment- and hardware-agnostic control of the VE.

We have been successfully employing the framework within the ChronoPilot project to investigate different factors related to time perception, and the framework substantially reduces the coding effort usually associated with creating and deploying new VR-based behavioral studies. Since the framework may be of great interest to other research groups, we are open to collaboration and open-sourcing it in the near future.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 964464.

References

[1] S. Alves, A. Callado, and P. Jucá. Evaluation of Graphical User Interfaces Guidelines for Virtual Reality Games. In *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pp. 71–79, 2020. doi: 10.1109/SBGames51465.2020.00020

[2] A. O. Bebko and N. F. Troje. bmlTUX: Design and Control of Experiments in Virtual Reality and Beyond. *i-Perception*, 11(4):2041669520938400, 2020. doi: 10.1177/2041669520938400

[3] O. Benjamin, K. Yukari, S. Aria, S. Reiko, M. Kouta, S. Ayahiko, and M. Tetsuya. Crystal Vibes feat. Ott: A psychedelic musical virtual reality experience utilising the full-body vibrotactile haptic synesthesia suit. IEEE, 2017. doi: 10.1109/vsmm.2017.8346269

[4] J. Botev, K. Drewing, H. Hamann, Y. Khaluf, P. Simoens, and V. Argiro. ChronoPilot – Modulating Time Perception. In *Proceedings of the 4th IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR 2021)*, pp. 215–218. IEEE, 2021. doi: 10.1109/AIVR52153.2021.00049

[5] J. Brookes, M. Warburton, M. Alghadier, M. Mon-Williams, and F. Mushtaq. Studying human behavior with virtual reality: The Unity Experiment Framework. *Behavior Research Methods*, 52(2):455–463, 2020. doi: 10.3758/s13428-019-01242-0

[6] M. Davydenko and J. Peetz. Time grows on trees: The effect of nature settings on time perception. 54:20–26, 2017. doi: 10.1016/j.jenvp.2017.09.003

[7] S. Droit-Volet, D. Ramos, J. L. O. Bueno, and E. Bigand. Music, emotion, and time perception: the influence of subjective emotional valence and arousal? *Frontiers in Psychology*, 4:417, 2013. doi: 10.3389/fpsyg.2013.00417

[8] M. Fornaciai, R. Arrighi, and D. C. Burr. Adaptation-Induced Compression of Event Time Occurs Only for Translational Motion. 6(1), 2016. doi: 10.1038/srep23341

[9] R. Gebbers and V. I. Adamchuk. Precision Agriculture and Food Security. *Science*, 327(5967):828–831, 2010. doi: 10.1126/science.1183899

[10] G. J. Gorn, A. Chattopadhyay, J. Sengupta, and S. Tripathi. Waiting for the Web: How Screen Color Affects Time Perception. 41(2):215–225, 2004. doi: 10.1509/jmkr.41.2.215.28668

[11] D. Hammerschmidt, C. Wöllner, J. London, and B. Burger. Disco Time: The Relationship Between Perceived Duration and Tempo in Music. *Music & Science*, 4:2059204320986384, 2021. doi: 10.1177/2059204320986384

[12] M. Landeck, F. Unruh, J.-L. Lugrin, and M. E. Latoschik. Metachron: A framework for time perception research in VR. In *26th ACM Symposium on Virtual Reality Software and Technology*. ACM, 2020. doi: 10.1145/3385956.3422111

[13] J. J. LaViola, D. A. Bowman, E. Kruijff, I. Poupyrev, and R. P. McMahan. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, 2 ed., 2017.

[14] U. Lee and J. Tanaka. Finger Identification and Hand Gesture Recognition Techniques for Natural User Interface. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, p. 274–279. Association for Computing Machinery, 2013. doi: 10.1145/2525194.2525296

[15] X. Lou, X. A. Li, P. Hansen, and P. Du. Hand-adaptive user interface: improved gestural interaction in virtual reality. *Virtual Reality*, 25(2):367–382, 2021. doi: 10.1007/s10055-020-00461-7

[16] K. Lupinetti, A. Ranieri, F. Giannini, and M. Monti. 3D Dynamic Hand Gestures Recognition Using the Leap Motion Sensor and Convolutional Neural Networks. In *Augmented Reality, Virtual Reality, and Computer Graphics*, pp. 420–439. Springer International Publishing, 2020. doi: doi.org/10.48550/arXiv.2003.01450

[17] A. McBratney, B. Whelan, T. Ancev, and J. Bouma. Future Directions of Precision Agriculture. *Precision Agriculture*, 6(1):7–23, 2005. doi: 10.1007/s11119-005-0681-8

[18] S. Picard and J. Botev. Rhythmic Stimuli Effects on Subjective Time Perception in Immersive Virtual Environment. In *Proceedings of the 14th International Workshop on Immersive Mixed and Virtual Environment Systems*, p. 5–11. Association for Computing Machinery, 2022. doi: 10.1145/3534086.3534330

[19] F. J. Pierce and P. Nowak. Aspects of Precision Agriculture. vol. 67, pp. 1–85. Academic Press, 1999. doi: 10.1016/S0065-2113(08)60513-1

[20] J. Reif and W. Alhalabi. Advancing Attention Control Using VR-Induced Multimodal Artificial Synesthesia, 2018. doi: 10.20944/preprints201808.0134.v1

[21] C. Schatzschneider, G. Bruder, and F. Steinicke. Who turned the clock? Effects of Manipulated Zeitgebers, Cognitive Load and Immersion on Time Estimation. 22(4):1387–1395, 2016. doi: 10.1109/tvcg.2016.2518137

[22] I. Schuetz, H. Karimpur, and K. Fiehler. vexptoolbox: A software toolbox for human behavior studies using the Vizard virtual reality platform. *Behavior Research Methods*, 55(2):570–582, 2023. doi: 10.3758/s13428-022-01831-6

[23] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 6 ed., 2016. doi: 10.5555/3033040

[24] U. Söderström, M. Bååth, and T. Mejtoft. The Users' Time Perception: The Effect of Various Animation Speeds on Loading Screens. In *Proceedings of the 36th European Conference on Cognitive Ergonomics*. Association for Computing Machinery, 2018. doi: 10.1145/3232078.3232092

[25] M. R. Watson, B. Voloh, C. Thomas, A. Hasan, and T. Womelsdorf. USE: An integrative suite for temporally-precise psychophysical experiments in virtual environments for human, nonhuman, and artificially intelligent agents, journal = Journal of Neuroscience Methods. 326:108374, 2019. doi: doi.org/10.1016/j.jneumeth.2019.108374

[26] B. Yoo, J.-J. Han, C. Choi, K. Yi, S. Suh, D. Park, and C. Kim. 3D User Interface Combining Gaze and Hand Gestures for Large-Scale Display. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, p. 3709–3714. Association for Computing Machinery, 2010. doi: 10.1145/1753846.1754043

[27] Q. Zhang. *Precision Agriculture Technology for Crop Farming*. CRC Press, 2015. doi: 10.1201/b19336