Beyond Coding: Empowering HRI Researchers with an Authoring Tool for Simplified VR Studies

André Helgert
Institute of Computer Science
University of Applied Sciences Ruhr West
Bottrop, Germany
andre.helgert@hs-ruhrwest.de

Sabrina C. Eimler
Institute of Computer Science
University of Applied Sciences Ruhr West
Bottrop, Germany
sabrina.eimler@hs-ruhrwest.de

Tom Gross
Human-Computer Interaction Group
University of Bamberg
Bamberg, Germany
hci@uni-bamberg.de

Abstract—**Although Virtual Reality (VR) is a promising tool in today's Human-Robot Interaction (HRI) research, the technical hurdle of creating high-quality and customized VR applications for one's own research prevents its comprehensive use. To address this problem, we present a VR authoring tool for HRI research, allowing non-technical researchers to create individual study designs in a simplified manner with predefined environments, robots, and measurement methods. Furthermore, the authoring tool has a highly modularized structure so that it can be adapted to a wide range of use cases. Technical details are presented, covering UI/UX decisions in the study design process and the visual programming interface for the creation of the interaction scheme of the robot, as well as the data management system that transfers researchers' study setups into a suitable VR application. The VR authoring tool is intended to bridge the gap between the practical applicability of VR as a research tool and the required technical implementation expertise for VR applications.**

Index Terms—**virtual reality, authoring tool, human-robot interaction, robots**

I. INTRODUCTION

In the field of Human-Robot Interaction (HRI), Virtual Reality (VR) is increasingly utilized as a research tool [1]. One reason for this is that, unlike field and laboratory studies, VR offers an ecologically valid platform with a controllable and reproducible study environment [2]–[4]. It also provides the opportunity to customize and manipulate all virtual objects, such as robots, enabling study setups that are difficult or currently impossible in real-world interactions [5]. These advantages, among others, contribute to the fact that VR studies are an useful research tool for many use cases.

However, developing VR environments and the behavior of virtual robots is a complex process [6]. Environments often need to be modeled from scratch with realistic textures and lighting to ensure authentic look [7]. Additionally, it is crucial to accurately implement the program logic and behavior of the robots, ensuring that both robots and environments are detailed and function comparably to real-world counterparts. This process is highly cost- and resource-intensive, requiring developers with expertise in game engines like Unity or Unreal [6]. As a result, researchers face a significant technical hurdle [8], limiting the broader adoption of VR as a research tool.

Since there are no ready-to-use tools enabling HRI researchers to use VR for their research in a simple and customizable way, regardless of their technical implementation skills, a VR authoring tool was developed. The requirements of the tool were collected through expert interviews with HRI researchers in a previous iteration [8]. This tool stands out for three key reasons: First, it simplifies complex implementation processes, such as robot navigation or voice output, by offering these functionalities through intuitive, clickable buttons. This means that researchers can leverage advanced features without needing any coding expertise, significantly reducing the development effort typically required. Second, the tool provides access to high-quality virtual environments, various robots, and other assets, which are usually the result of months-long design processes. These resources are readily available for immediate use. Third, the tool offers modularity in both its functions and the virtual environments and objects, allowing for a wide range of application scenarios within the highly interdisciplinary field of HRI research. This flexibility ensures that various options, such as navigation modes or voice output, can be customized to meet specific needs. Together, these features make the VR authoring tool a valuable asset, transforming complex technical processes into an intuitive, user-friendly experience for researchers. This technical paper demonstrates the components and architecture of the VR authoring tool and discusses the limitations and future directions on a technical level.

II. RELATED WORK

VR is increasingly being used as a research tool in various HRI scenarios. These range from virtual applications for training and simulation that are dangerous in real environments [9] to studies that focus on social interactions between robots and humans [10]. Furthermore, recent studies emphasize the potential of VR for robot teleoperation [11] or the possibility of exploring human-robot collaboration in safe virtual environments [12]. This wide range of applications demonstrates the versatility of VR as a research tool in HRI. However, a common challenge in all these applications is the need for customized solutions. This circumstance requires specialist technology skills to create and maintain virtual environments. Inadequate visual and auditory augmentation of the elements can compromise core principles of VR, such as immersion and presence, potentially leading to lower data validity and limiting the overall results of the study.

To address these challenges, we reviewed recent research to identify effective tools that can help overcome existing issues and enable non-technical researchers to develop their own VR studies. Our examination reveals a significant lack of modular and simplified tools for creating custom VR studies (particularly in the field of HRI), which is crucial for enabling researchers, regardless of their technical implementation skills, to effectively use the tool for their individual research. While there are general approaches outside of HRI aiming to modularize technical functionalities—such as Giglioli et al.'s VSA framework, which evaluates psychological attributes in real-time during VR scenarios under stress [13], and Brookes et al.'s UFX framework (directly integrated into the Unity development environment) for gathering participant data and study timestamps [14] — there is still a need for user-friendly tools designed specifically for HRI.

Other significant frameworks include Grübel et al.'s EVE, which facilitates data collection, export, and presentation throughout VR experiments [15], and Vasser et al.'s VREX tool, which simplifies the process of creating experimental VR environments for researchers with limited technical expertise [16]. Additionally, the RosUnity integration combines Unity's advanced 3D visualization features with ROS's comprehensive robot simulation tools [17] as a plugin, which requires technical implementation skills to use it. These innovations make it clear that there is a need to simplify complex technical processes in the field of virtual reality. However, many of the current solutions still do not provide tools for data acquisition and observation methods in HRI, nor ease of use for non-technical researchers.

Therefore, we set out to develop an authoring tool that would make the development of VR study environments in HRI more usable and customizable by providing simplified processes for complex implementation tasks and a modularity that enables broad usability. To this end, expert interviews were conducted with HRI researchers to generate a catalog of requirements for this tool [8]. The main requirements include an intuitive, clickable interface for utilizing measurement methods in HRI, the use of block-based programming for interaction schemes and the provision of documentation, videos, and demos. The catalog of requirements was implemented in a technical application, which is explained in detail in the next chapter.

III. TECHNICAL IMPLEMENTATION

The VR authoring tool is built in Unity, incorporating various SDKs such as OpenXR, Windows Mixed Reality, and the Pico Unity Integration SDK to ensure broad compatibility with different VR hardware. The authoring tool features two versions: a “Beginner“ version, offering a simplified user interface with a compiled version, which is particularly useful for researchers who have not yet had any experience with VR development, and an “Expert“ version, available as a Unity plug-in, which includes a comprehensive set of features on developer level. This paper focuses on the “Beginner“ version, as we anticipate it will be the preferred choice for most researchers.

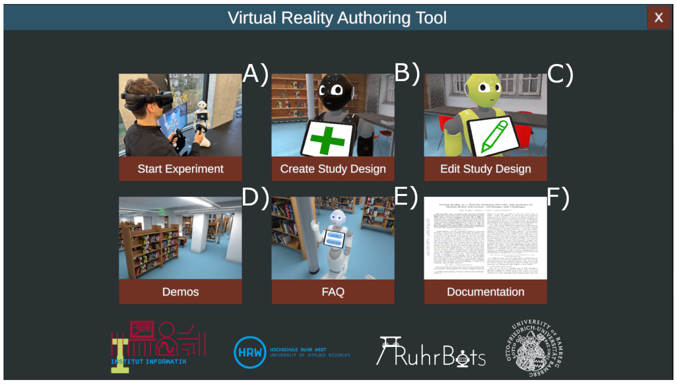

Fig. 1. Menu of the VR authoring tool with the different selection options

A. UI/UX Design

The VR authoring tool is intended to be a simplified toolbox for setting up one's own study designs using VR as a research instrument. In the menu (see Figure 1) the toolbox offers the possibility to (A) execute already created study designs, e.g. for actual studies, (B) create a new study design from scratch, (C) adapt an existing study design, (D) test executable demos as an illustrative aid, (E) call up an FAQ and as a last point (F), look up written documentation and explanations of the functions of the authoring tool.

In order to make the process of creating own studies as simple and intuitive as possible, care was taken to ensure that the process is as sequential as possible. Thus, the study creation process (B) consists of two different parts: The first part provides researchers with some configuration options, which are defined based on the research process in HRI according to Bartneck et al. [18] (see Figure 2 for the example of selecting the environment).

Fig. 2. Example step in the creation of a new study design: choosing the virtual environment

In the first part, the researcher must choose between a dyadic or group interaction, select the virtual environment and the robot for the study, select suitable measurement methods for their own study design, and choose whether the

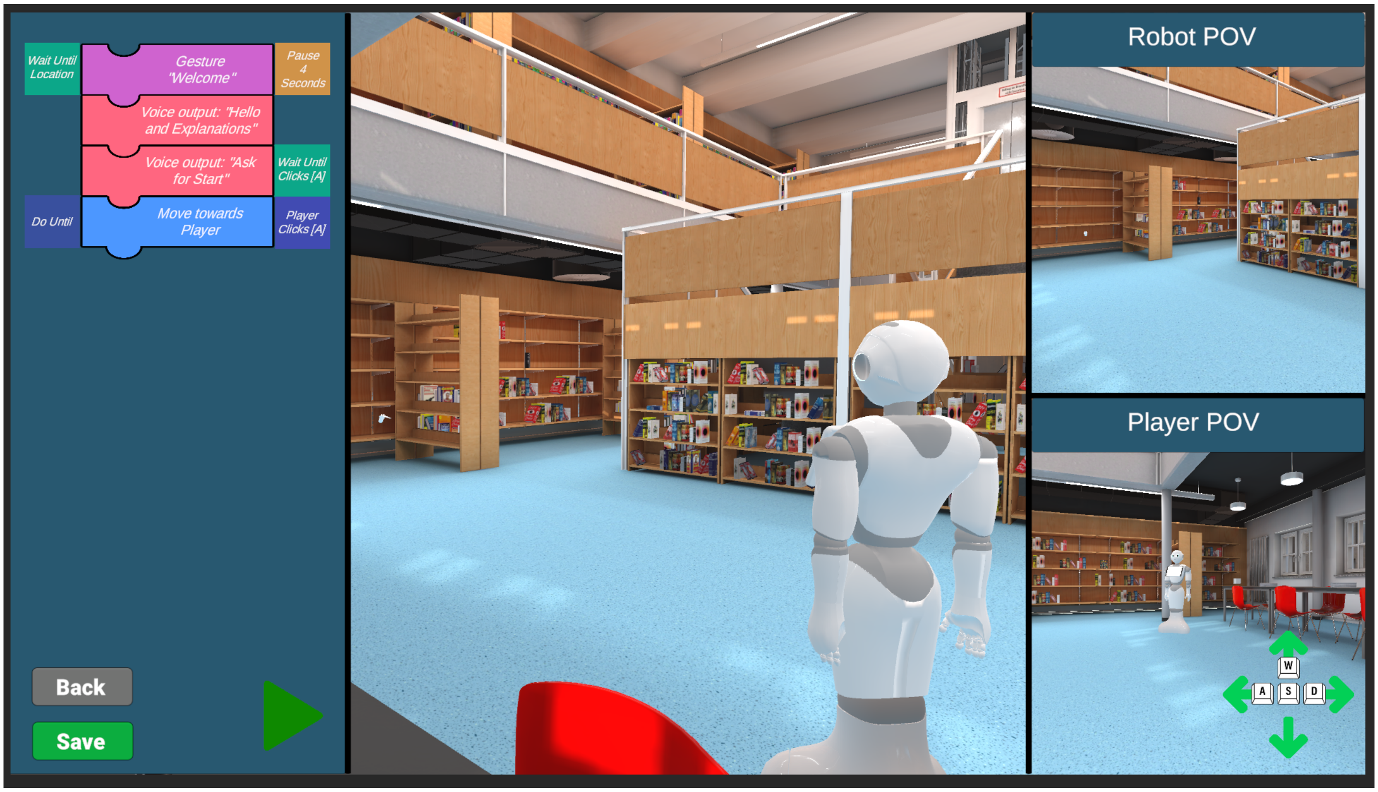

Fig. 3. Example interaction created with the visual programming interface

interaction of the robot should be teleoperated via a Wizard of Oz (WoZ) during the study or determined autonomously in advance via a visual programming interface. These screens are all implemented on a canvas and the respective panels and the flow logic was implemented by the internal *OnClick()* method in Unity. All inputs are also saved in the internal data management system so that they can be loaded for the study executions, which will be detailed later.

Once the initial process with the basic settings of the HRI research process is completed, depending on the choice, either the WoZ environment will be loaded (which can be customized), or the automated visual programming interface for robot interaction will be loaded. The WoZ was demonstrated in a previous iteration [19]. Accordingly, the configurations from the first part (interaction form, environment, robot, etc.) are loaded into a new scene. Researchers thus have an immediate overview of what the study setting will look like visually. This is supported by three different cameras, as shown in Figure 4.

The main camera provides a top-down view of the scene and covers both the robot and the virtual hands of the participant. The Robot POV camera offers an additional perspective from the robot's viewpoint, while the Human POV view displays the perspective of the VR headset. This setup aims to enhance spatial perception of the virtual space. The main view can be navigated, as depicted in Figure 4 with icons, using the "WASD" keys, and can also be rotated with the mouse. This allows researchers to freely move the camera within the virtual environment, providing greater flexibility in creating interactions between human and robot. It is also possible to move and rotate the robot. When the mouse hovers over the robot, UI elements explaining these actions are displayed.

In the visual programming interface view, there are several configuration options (see Figure 4). It is possible to change the visual appearance of the robot, such as its color (G), configure the movement and rotation of the virtual character representing the participant (H), create a custom interaction (I) (which will be explained in the next subsection), reconfigure measurements (J), and set up data export for further analysis steps (K) (specifying the data format for export).

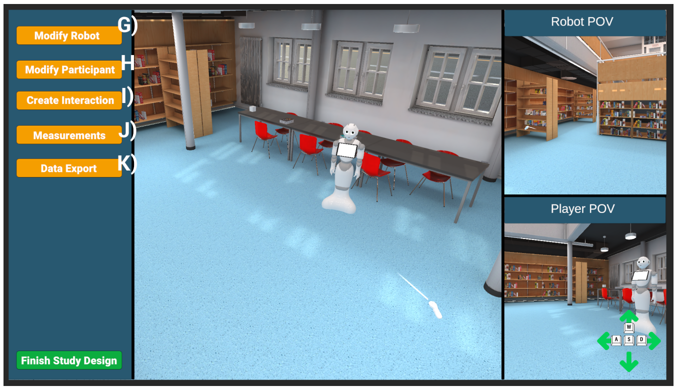

Fig. 4. Overview of the selected study design and menu for visual programming interface

B. Visual Programming Interface for Robot Interaction

We implemented a block-based visual programming language inspired by other commonly tools like Scratch [20] and Blockly [21]. When (I) is selected, the visual programming interface is loaded (see Figure 3). Using this view, it is possible to create the robot's behavior using predefined interaction blocks and program logic such as navigation tasks and voice output. The activity diagram for this process can be seen in Figure 5. Researchers have the option of gradually defining interactions that are carried out sequentially by the robot (see Figure 3 for an example interaction between robot and user). For each interaction block, researchers can choose between four forms of interaction: Movement, gestures, tablet content and verbal communication from the robot.

1) Robot Motion: Here it is possible to have the robot move to predefined points in the environment, which should represent a low hurdle for creating the interaction. The robot can always move to the user, a specific object in the environment (e.g., a book in the library) and a table. A behavioral handler, connected to a dedicated navigation agent system, enables the robot's autonomous navigation by incorporating static objects into a predefined navigation mesh. In addition to the predefined targets, it is also possible to use dynamically allocated targets in virtual space by researchers. If this option is selected, a controller deactivates the UI so that the researcher can click on a game object in the virtual environment with a mouse click, which is then saved as a target using raycast. The "move" and "rotate" functions allow the robot to move and rotate freely in any direction without having a fixed target. These functions do not use the navigation agent system, but a vector which the robots move to the position and a direction variable which allows the robot to rotate.

2) Robot Gesture: For the animated gestures, the Unity package "Final IK" was utilized [22], which is a system for inverse kinematics. This allows the end-effector (e.g., the robot's hand) to be moved to a specific target. This decision was made because it provides the robot with the necessary flexibility in its movements and includes pre-installed tools for interactions such as grasping and holding objects. Using this Unity package, predefined gestures were created, such as a greeting where the robot can wave, as well as an idle animation that plays by default to animate the robot's resting state.

3) Robot Tablet: If the tablet function of the robot is to be used, researchers have two options: Firstly, they can load a static image from their computer into the tool to display it on the robot's tablet, or they can use the tablet as an interactive tool with the user. This allows users to input data on the tablet by clicking on embedded UI buttons. In that case, a panel with corresponding buttons is added to the interface and, depending on the selection, these elements are activated and assigned a function (such as a confirmation button for the user).

4) Robot Voice: For the verbal communication capabilities of the robots, the OpenAI API was integrated to utilize text-to-speech functionality [23]. The associated class employs an asynchronous method that sends a request to the OpenAI API and returns the audio data as a byte array. It constructs the JSON payload, sends the HTTP POST request, and processes the response. Additionally, it offers the internal OpenAI options for voice selection, playback speed, and pitch of the generated speech. Given that the speech output occurs through the robot, it was essential to include a robotic voice option. This was achieved using audio effects to make the voice sound more mechanical, such as distortion, bitcrusher, and chorus.

5) Interaction Logic Blocks: Since the interactive blocks presented only a single action each and can model simple interaction schemes of the robot through sequential arrangement, it is necessary for many applications to use program logic. The current logic includes *wait*, *wait-until*, *do-until*, and *repeat*. These logic functions can be attached to any interactive block to create more efficient interaction schemes. They are also crucial for responding to user interactions. For example, in Figure 3, the execution of the gesture waits until the user enters a specific area in the virtual environment, followed by a four-second pause. Then, the robot plays two voice messages, with the last one requiring confirmation via a button press. Finally, the robot moves towards the user until the user presses a button.

C. Data Management in Study Design Process

Data management is used to store configuration decisions made during the creation process, including settings in the visual programming interface for the robot's interaction and their integration into the VR application. All decisions and configurations made in the authoring tool are saved in the internal data management system. To enhance maintainability and facilitate future expansion of the VR authoring tool, each configuration and decision that needs to be saved and loaded is assigned a fixed index within the data structure. This approach simplifies the verification of whether these indexes contain data and streamlines internal processing by other functions, such as executing interactions in the visual programming interface. Finalized design entries are stored in a JSON file.

When executing a study design in VR, configurations are loaded into the virtual scene via the JSON file, which must be loaded into the VR environment using a UI button. The responsible script ensures that all entries in the data management system are processed sequentially, exactly as they were created. This is achieved by using branching logic to search for uniquely identifiable entries in the JSON file and executing their respective functions.

IV. DISCUSSION

The VR authoring tool presented in this paper addresses a need in the HRI research community by simplifying the creation of VR-based study environments. Based on expert interviews, the design of the tool includes essential features identified as necessary for effective HRI research, such as a choice of different virtual environments and robots, different interaction modalities and a visual programming interface [8]. This approach aims to lower the technical barrier typically associated with VR development and makes the tool usable

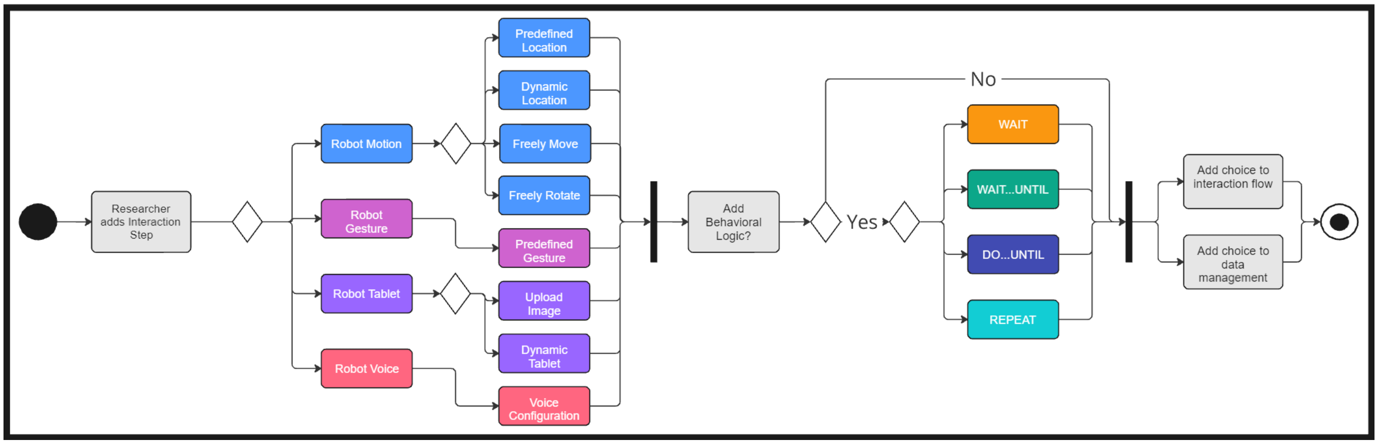

Fig. 5. Interaction flow of the creation of an autonomous interaction between the robot and the user

to researchers without extensive programming knowledge or financial capacities to buy robots. Despite the benefits, the current version of the tool has limitations that should be considered. The simplified nature of the “Beginner“ version presented here could lead to compromises in terms of depth of customization options and control compared to more complex solutions. Even though the “Expert“ version is intended for full control, it does not offer simplified structures to make the VR authoring tool accessible to all. It therefore remains a balancing act between the necessary complexity and the ease of use and application of the tool. In particular, the effectiveness of the visual programming interface in creating complex, nuanced behaviors remains to be evaluated. The intended modularity of the VR authoring tool also plays a crucial role here, raising the question of what range of different use cases can actually be addressed with the tool. Further technical enhancements are conceivable, such as a dedicated speech-to-text system so that the user can communicate verbally with the robot and the option of creating and saving gestures yourself via your own interface and using them in your own study design.

V. FUTURE WORK

This technical paper presents the first iteration of an authoring tool designed to generate study designs in VR for HRI research in a simplified and customizable manner. We have explained and illustrated the UI/UX design, along with the guided interaction flow of the application, which is intended to provide easy configuration of the study settings. Additionally, we introduced the visual programming interface, which represents many complex implementation processes, such as environment-sensitive navigation of a robot in virtual space, as well as voice output and the execution of animations like robot gestures, in a simplified, clickable block format. This approach is aimed at enabling researchers to utilize these functions in the virtual space, which would otherwise require significant implementation expertise and effort without the VR authoring tool. We also presented the data management system, which ensures that all individual settings are securely saved and can be further processed during program execution. In the future, this system will be evaluated through multiple user studies. Firstly, we plan to test the authoring tool with technically skilled researchers to identify barriers in applicability, workflow, and usability, which could lead to additional requirements for the next iteration of the tool. Additionally, we aim to conduct a user study with HRI researchers to assess how the tool supports actual study designs in HRI. Together, this approach aims to bridge the gap between technical challenges in the creation process of VR applications and the practical applicability of VR as a research method.

ACKNOWLEDGEMENTS

The presented work was supported by the RuhrBots (www.ruhrbots.de) competence center (16SV8693) funded by the Federal Ministry of Education and Research Germany. We would like to thank Veronica Schwarze for providing feedback on the manuscript.

REFERENCES

[1] R. Etzi, S. Huang, G. W. Scurati, S. Lyu, F. Ferrise, A. Gallace, A. Gaggioli, A. Chirico, M. Carulli, and M. Bordegoni, “Using virtual reality to test human-robot interaction during a collaborative task,” in *Volume 1: 39th Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2019.

[2] A. Gorini, C. S. Capideville, G. de Leo, F. Mantovani, and G. Riva, “The role of immersion and narrative in mediated presence: the virtual hospital experience,” *Cyberpsychology, behavior and social networking*, vol. 14, no. 3, pp. 99–105, 2011.

[3] Z. Campbell, K. K. Zakzanis, D. Jovanovski, S. Joordens, R. Mraz, and S. J. Graham, “Utilizing virtual reality to improve the ecological validity of clinical neuropsychology: an fmri case study elucidating the neural basis of planning by comparing the tower of london with a three-dimensional navigation task,” *Applied neuropsychology*, vol. 16, no. 4, pp. 295–306, 2009.

[4] J. Diemer, G. W. Alpers, H. M. Peperkorn, Y. Shiban, and A. Mühlberger, “The impact of perception and presence on emotional reactions: a review of research in virtual reality,” *Frontiers in psychology*, vol. 6, p. 26, 2015.

[5] H. E. Yaremych and S. Persky, “Tracing physical behavior in virtual reality: A narrative review of applications to social psychology,” *Journal of experimental social psychology*, vol. 85, 2019.

[6] T. D. Parsons, “Virtual reality for enhanced ecological validity and experimental control in the clinical, affective and social neurosciences,” *Frontiers in human neuroscience*, vol. 9, no. 660, 2015.

[7] M. Slater and M. V. Sanchez-Vives, “Enhancing our lives with immersive virtual reality,” *Frontiers in Robotics and AI*, vol. 3, 2016.

[8] A. Helgert, C. Straßmann, and S. C. Eimler, "Conceptualizing and designing a virtual reality authoring tool for human-robot interaction studies - learnings and guidance from expert interviews," *Mensch Und Computer*, vol. 24, p. 331–341, 2024.

[9] S. B. i. Badia, P. A. Silva, D. Branco, A. Pinto, C. Carvalho, P. Menezes, J. Almeida, and A. Pilacinski, "Virtual reality for safe testing and development in collaborative robotics: Challenges and perspectives," *Electronics*, vol. 11, no. 11, p. 1726, 2022.

[10] C. Straßmann, C. Eudenbach, A. Arntz, and S. C. Eimler, "don't judge a book by its cover: Exploring discriminatory behavior in multi-user-robot interaction," in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, D. Grollman, E. Broadbent, W. Ju, H. Soh, and T. Williams, Eds. New York, NY, USA: ACM, 2024, pp. 1023–1027.

[11] L. Meng, J. Liu, W. Chai, J. Wang, and M. Q.-H. Meng, "Virtual reality based robot teleoperation via human-scene interaction," *Procedia Computer Science*, vol. 226, pp. 141–148, 2023.

[12] S. T. Mubarrat, A. Fernandes, and S. K. Chowdhury, "A physics-based virtual reality haptic system design and evaluation by simulating human-robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 54, no. 4, pp. 375–384, 2024.

[13] I. A. Chicchi Giglioli, G. Pravettoni, D. L. Sutil Martín, E. Parra, and M. A. Raya, "A novel integrating virtual reality approach for the assessment of the attachment behavioral system," *Frontiers in Psychology*, vol. 8, p. Article 959, 2017.

[14] 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*, vol. 52, no. 2, pp. 455–463, 2020.

[15] J. Grübel, R. Weibel, M. H. Jiang, C. Hölscher, D. A. Hackman, and V. R. Schinazi, "Eve: A framework for experiments in virtual environments," in *Spatial Cognition X*, ser. Lecture Notes in Computer Science, T. Barkowsky, H. Burte, C. Hölscher, and H. Schultheis, Eds. Cham: Springer International Publishing, 2017, vol. 10523, pp. 159–176.

[16] M. Vasser, M. Kängsepp, M. Magomedkerimov, K. Kilvits, V. Stafinjak, T. Kivisik, R. Vicente, and J. Aru, "Vrex: an open-source toolbox for creating 3d virtual reality experiments," *BMC psychology*, vol. 5, no. 1, p. Article 4, 2017.

[17] H. Younesy, "Unity-robotics-hub," https://github.com/Unity-Technologies/Unity-Robotics-Hub/tree/main, 2020.

[18] C. Bartneck, T. Belpaeme, F. Eyssel, T. Kanda, M. Keijsers, and S. Šabanović, *Human-Robot Interaction: An Introduction*. Cambridge University Press, Feb. 2020. [Online]. Available: http://dx.doi.org/10.1017/9781108676649

[19] A. Helgert, C. Straßmann, and S. C. Eimler, "Unlocking potentials of virtual reality as a research tool in human-robot interaction: A wizard-of-oz approach," in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 535–539. [Online]. Available: https://doi.org/10.1145/3610978.3640741

[20] ScratchFoundation, "Scratch: Programming for all." [Online]. Available: https://scratch.mit.edu/

[21] GoogleDevelopers, "Blockly: A visual programming language." [Online]. Available: https://developers.google.com/blockly

[22] RootMotion, "Final ik." [Online]. Available: https://assetstore.unity.com/packages/tools/animation/final-ik-14290

[23] OpenAI, "Openai api." [Online]. Available: https://openai.com/api/