Non-Verbal Communication Interfaces For Collaborative Virtual Environments

Fynn Braren[1], **Wladimir Nabok**[1], **and Kemal Amet**[1]

[1]HAW Hamburg, DMI-Media Systems, Hamburg, 22081, Germany

ABSTRACT

This article explores several methods for interpersonal communication in collaborative virtual environments. A focus lies on the visualization of common interaction mechanisms to make them comprehensible for remote users. Furthermore we will explore several methods that enable collaboration in asynchronous ways, with on- and off-screen scenarios. Accompanying this article, we present a prototype, containing implementations of our methods. As a conclusion, we will analyse our implementations in there comprehensibility and there applicability in collaborative virtual environments.

Introduction

Only in the recent past, the Virtual Reality market has expended to multi-user applications, an area housing for example social-VR, co-operative work or educational applications and multi-player games. Accompanying the development, collaborative virtual environments (CVE) are emerging as a key research focus, at the convergence of human–computer interfaces (HCI), information and communication technologies (ICT), and computer supported cooperative work approaches (CSCW)[1]. We present several methods for communication and multi-user interaction in collaborative virtual environments. The aim of this work is to adjust common single-user interaction mechanisms for virtual environments to make them applicable and comprehensible in multi-user applications and to allow users to collaborate in VR on long term projects. Considering that virtual environments can vary heavily in scale and setting, we approached the development of our mechanisms, with on- and off-screen scenarios in mind. We want to enable users to share informations in asynchronous ways, i.e. spatially or chronologically separated. To ensure our communication interfaces are relevant and useful, we renounced the use of a voice interface. The user has to rely on the non-verbal communication mechanisms, when testing our prototype. As a setting we chose the historic environment 'Digitale Speicherstadt', wich includes a detailed reproduction of the hamburger Speicherstadt, 100 years ago. The richly detailed environment, offers many explorable items. An obvious use-case would be a historic educational tour guided remotely by a professional.

Related Work

There are many applications for CVE's. Mostly it is used for educational purposes. Maurice Liebregt tried to figure out which advantages it might have to use a CVE in education and how to effectivley use it.[2] We are trying to create an environment in which one player can show another player something or highlight it with a marker. Another usecase for CVE's is the analysis of scientific data. Students from France made an evironment where two or three people were able to transform a plane in order to make a Cross-section of scientific data.[3]

Methods

In this section we will present the implementations of our methods and state our design decisions for each feature.

Technology

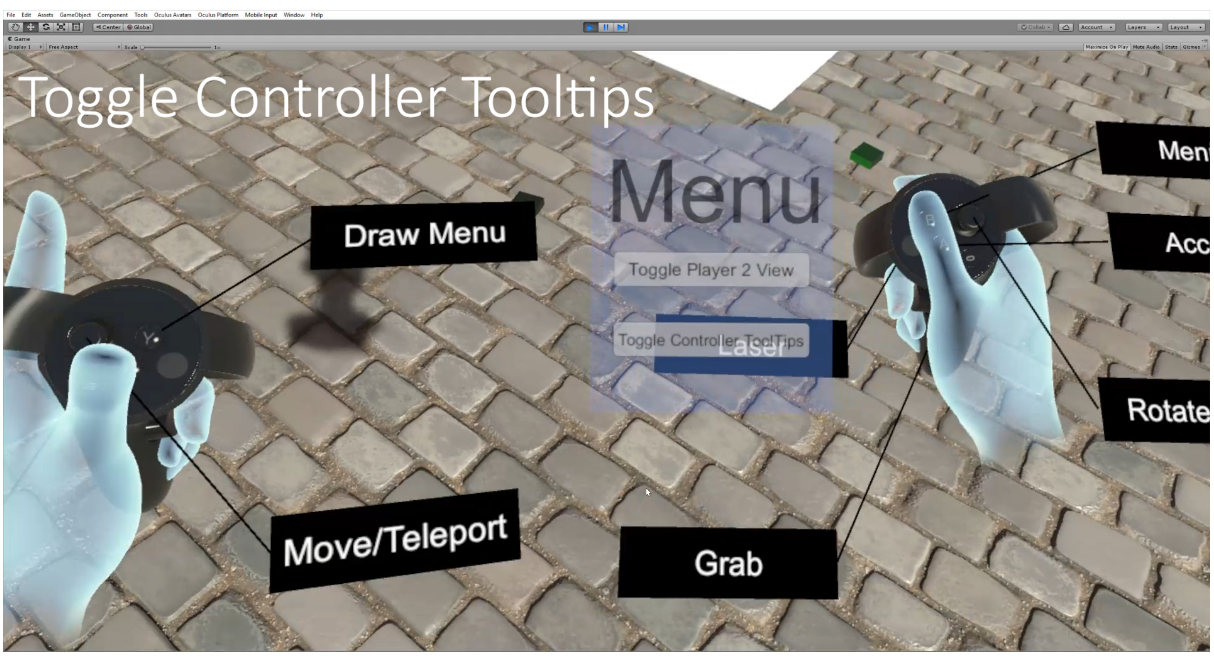
To realize our prototype, we used the game engine Unity (version 2017.2.0f3)[4]. Unity is one of the most used game engines today and has, as such, a huge development community, enlarging the core functionality. To implement the networking part we used the Photon Unity Networking (PUN) package by Exit Games. PUN re-implements and enhances the features of Unity's built-in networking and includes a huge part of the required functionality. Furthermore we used the Virtual Reality Toolkit (VRTK), a collection of useful tools and interaction mechanisms, aimed at single-user applications. Our prototype is built for the Oculus Rift device. Oculus offers a the OVR unity integration, wich includes the SDK for the device and a set of core functionality.

Avatar transmission

A representation of the users body in first person experiences is a common way to enhance the immersion of digital worlds. Non-verbal communication or bodily communication "takes place whenever one person influences another by means of facial expression, tone of voice, or any other channels (except linguistic)"[5] (involving body language, hand gesture, gaze and facial expressions, or any combination thereof).[6] As in VR's, a direct transmission of actual body movement and virtual counterpart is possible, full-body avatar embodiment becomes even more immersive in terms of physical presence. To translate this immersion to the remote user, we implemented the remote avatar contained in the OVR integration for Unity, by Oculus. In order to transmit the movement over the network, we needed to write our own Photon Avatar component. OVR includes methods for packet recording and those can be serialized and send to the remote user over the network.

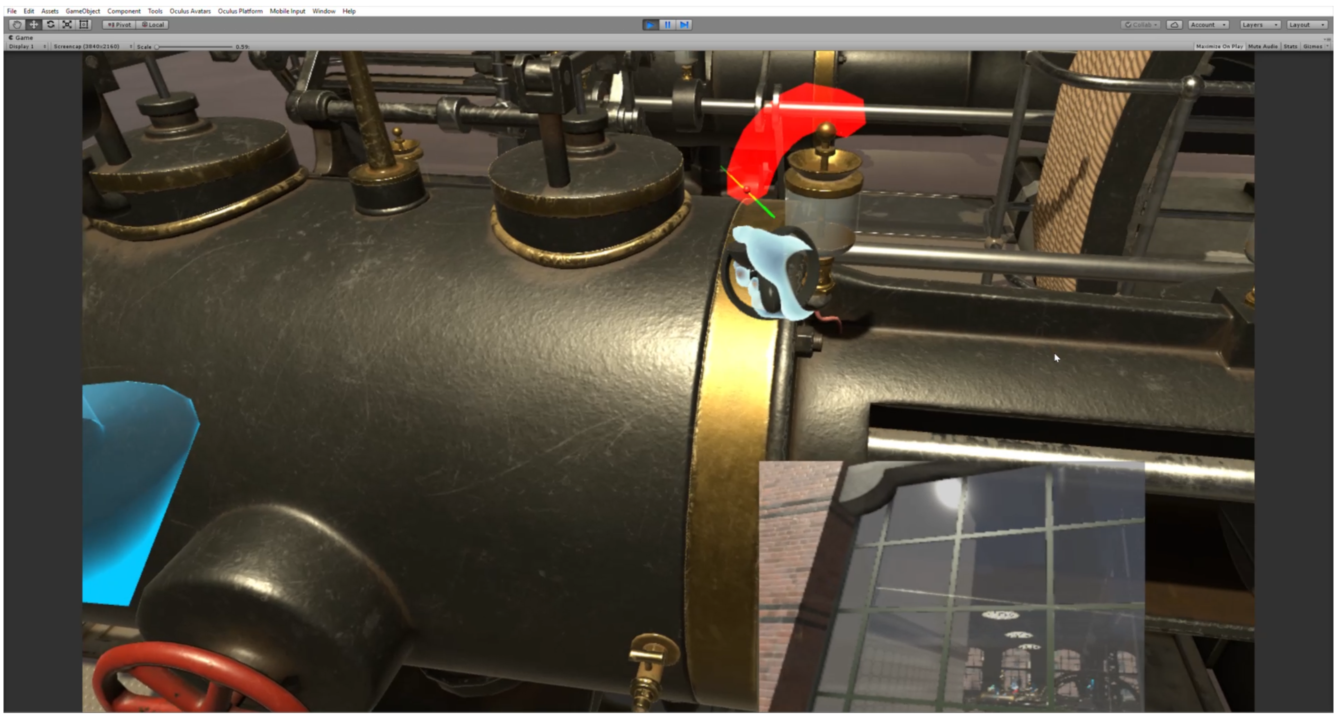
User Interface

To make the user Interface efficient and intuitive, we decided to follow old c onventions. We designed the user menu, with the WIMP (windows, icons, menus, pointer) interface in mind. The menu can be accessed by button press and consists of a simple panel with several buttons. The menu is following the users world transform and thus stays in front of the camera. To enhance the usability we implemented controller tooltips. This tooltip-toggle function can be found in the menu. The tooltip functionality is provided in the VRTK plug-in and consists of textlabels displayed live in the viewport.



POV Video/ Streaming

In order to improve the communication between remote sites, videoconferencing systems were developed to allow direct communication between collaborators[7] To ease the communication of spatially separated users, we developed a component, that implements Live-camera transmission of the other users view. The participants can share content without the need to be in the same room. [picture here] The component is realized through a virtual camera placed under the users rig. The camera output is rendered to a render target in the other users World Canvas. To enrich the camera information, it is placed as a first person view, and is thus transmitting hand movement and every interaction of the remote user. To reduce the network latency of the camera picture, the view transmission is only active, when both users are in the same scene. The network has no extra work, because transform and animation data of the remote user are already available locally. The user can activate or deactivate the camera stream through the menu interface.



Pointing and Pinging

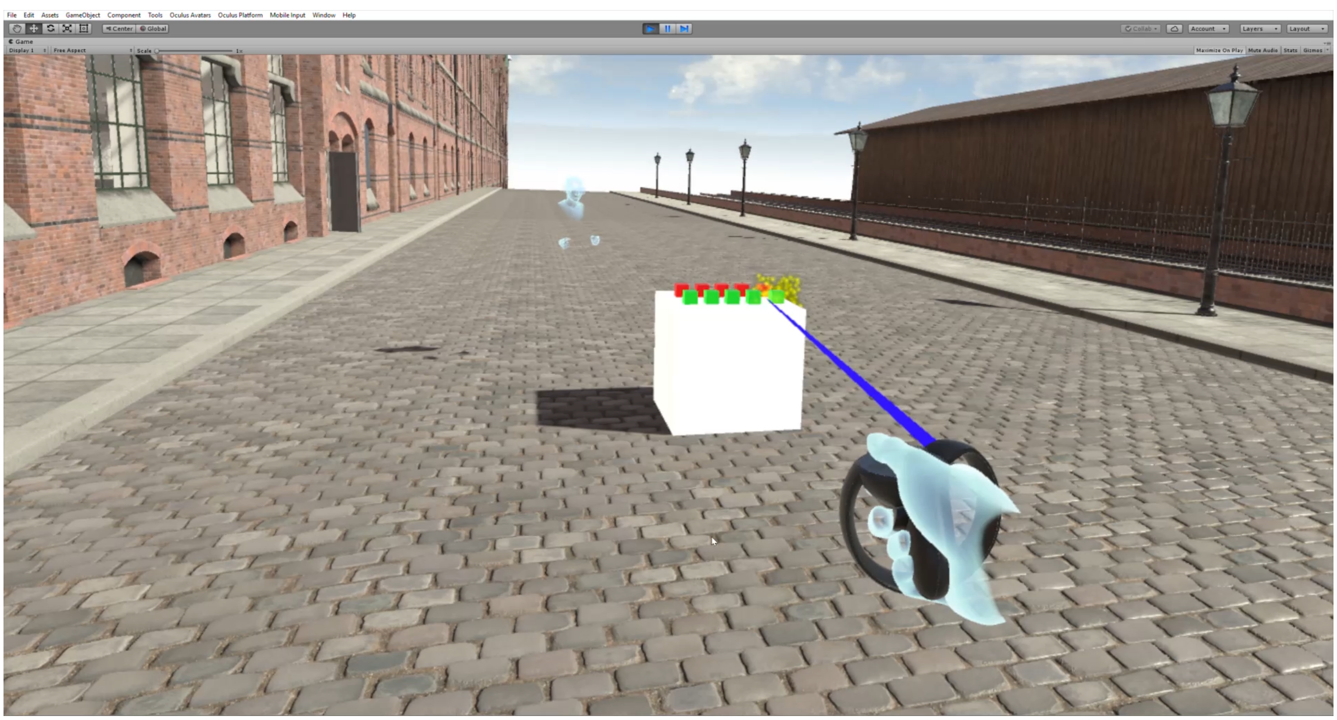
Pointing the index finger at an object is a gesture used frequently by presenters to lead the viewers attention to a specific object. This requires, that the viewer is supposed to look at the time of the pointing action at the presenter. In a Unity-VR-Environment we implement the action, with the help of a Line Renderer Component. PUN delivers the necessary Networking-Infrastructure, like Remote Procedure Calls to execute code on other clients. For the operation executed by the Oculus Controller, like not touching the trigger to point and pushing the thumb stick to ping, VRTK supplies useful methods to write these event listeners.

This component viusally represents the line, rendered from the origin of the index finger, with the direction forward and it ends at the first point of colission. Rendering and unrendering this component on the network, when the touching state of the trigger button changes, is achieved by RPC rendering and unrendering functions.

In order to reduce confusions among the object that is pointed at on can extend the functionality with a ”pinging” object and a highlighting coroutine.

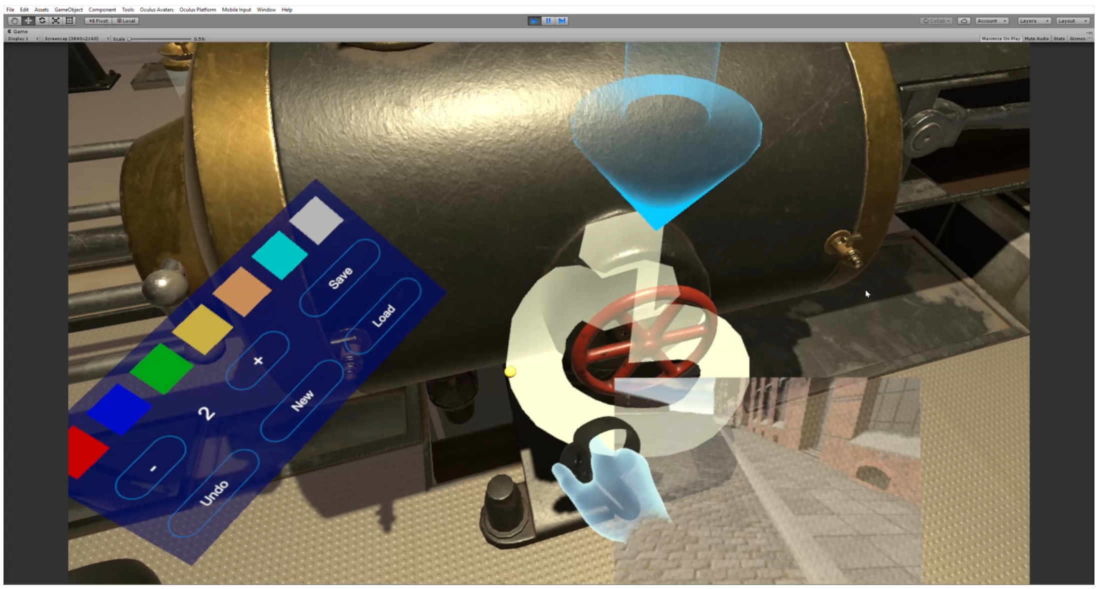
The pinging object is realized with a particle system as a visual key that gives the receiver of the ping a feedback, at which point the sender wants the attention to be concentrated at. It is accompanied with a sound effect, to grab the attention, that a ping action has been executed.

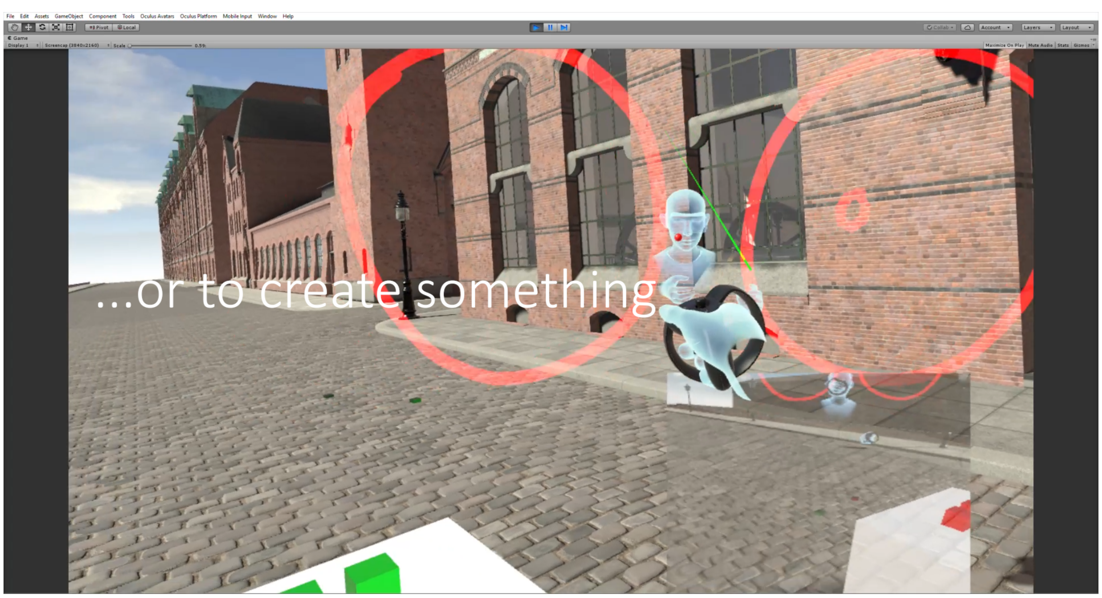
Furthermore the highlighting coroutine reinforces the attention grabing mechanism to the object as a whole, the pointer has created the ping at. The whole object flickers in the color of the pointer for two-three seconds. Also different colors can give information about who has pinged, when the clients have their pointers in different colors. So we implemented an extention of an index finger pointing mechanism with additional attention grabbing keys for a smooth presentation experience.



Marking/ Painter

To enable the users to work asynchronous, meaning spatially or chronologically separated, we implemented a brush component. In case a user wants to mark an object, draw a path on through the scene or even write comments, the brush can be accessed through a button-press. Strokes are instantiated on local and remote client and remain in the scene for the duration of the session. We used another unity package, VR Brush[8] for the core painting functionality and extended it by a networking component. VR Brush brings the possibility of changing colors and line-width on runtime through a world canvas attatched to the controller. Interaction with the canvas is possible through the VRTK plug-in. When painting VR Brush instantiates a Line-Init object containing a position array, wich is filled with Vector3 elements. Each stroke spawns another object. By adding a Photon View component to the Line-Init Prefab we were able to send the positions and color values to the remote client and instantiate a remote Line-Init object. Locally and remotely Line-Init objects are added into a List structure, so strokes can be undone by the user



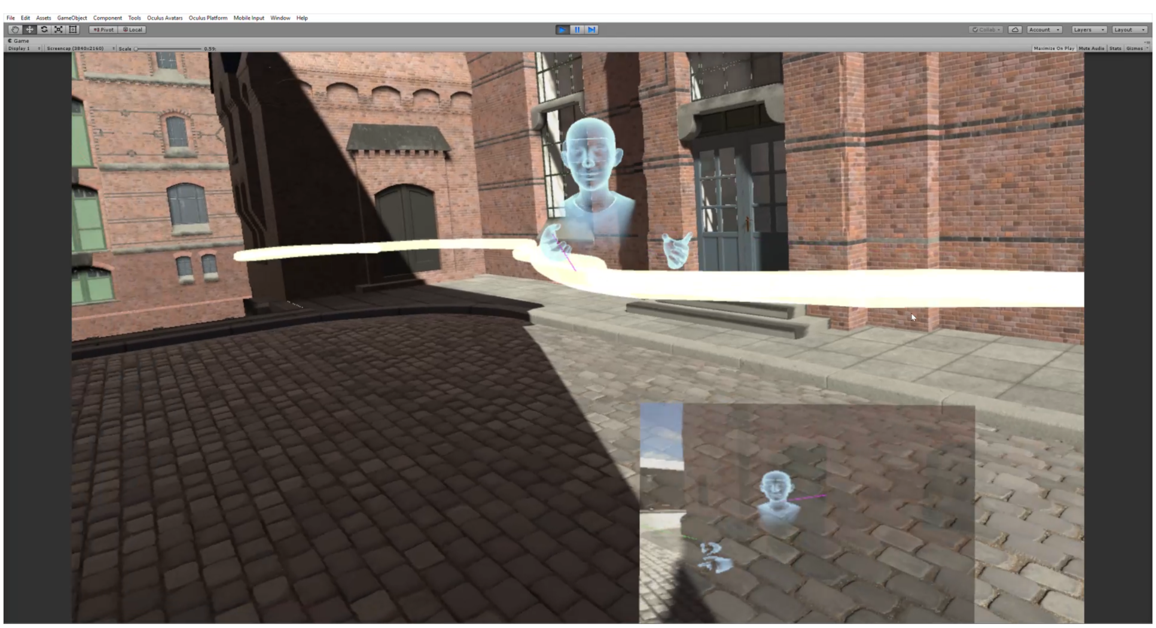


Object Interaction

A very common player-world interaction in VR is the grab/release functionality. VRTK includes advanced methods for grabbing and holding interactable Objects. While they do not communicate a message between users, they are elements in the scene, that can be communicated about.
Interactable objects shall visualize functions of objects for exmaple a mechanism of a two-way lever on a historical machine. By making the object interactable a presenter can indicate the interactability of an objects point at these objects and motivate others to interact with these, visualizing their functions, thus creating a smoother learning experience.
Therefore an interactable object can be an interesting tool for a collaborative virtual environment.

Conclusion

We presented several methods, that extend single-user virtual reality mechanisms by networking components and allow users to communicate and collaborate in a CVE. In this section we want to review these methods in there applicability and state our thoughts on future implementations. Virtual avatars allow the user a feeling of embodiment in the virtual world. While this is true for single-user applications, it is also true for collaborative environments. In our prototype we were able to show that the virtual body allows a much more detailed communication. Especially in non-verbal communication scenarios it is indispensable for communication quality, to allow communication by natural gestures, like pointing, waving, nodding etc. The be aware of the other users position and body pose, gives an idea of what the user is doing. In collaborative work environment, this can greatly enhance productivity. Another method controlled by gesture is our pointing implementation. It allows users to steer attention and point out important features in the scene. For an even better attention steering in a future project, we would advice to enhance the pinging functionality by a 3D audio component, so that objects of interest can be located faster. To allow users to collaborate in asynchronous scenarios, for instance being seperated by sight or orientation, we implemented the point-of-view streaming. Being able to see a live view of the other users camera, greatly enriches information flow and productivity in CVE's. For future projects, we would extend this feature by a record option. By capturing video or screenshots or even 360 degree content, users will be able to collaborate on the same projects asynchronous in time and space. Messages could be attached to objects of interest and be played back at any other moment by other users. Audio as a communication interface is obviously not replaceable and would greatly improve communication, because it enables the users the communication about abstract, fictional and non-related content.



References

1. M., S. J. . A. Haptic interpersonal communication: improvement of actions coordination in collaborative virtual environments. *Virtual Real.* **16**, 173–186 (2012).
2. Liebregt, M. Collaborative virtual environments in education. *Univ. Twente, Fac. EEMC* (2005).
3. Steed, C. F. T. D. V. G. A. Evaluation of remote collaborative manipulation for scientific data analysis. *IRISA, UMR CNRS 6074,UEB, Rennes, France* (2012).
4. Technologies, . U. Unity2017.20f3. https://unity3d.com/de (2017).
5. M, A. Bodily communication. *Methuen, Lond.* (1988).
6. Economou D. Doumanis I. Argyriou, L. e. a. User experience evaluation of human representation in collaborative virtual environments. *Pers. Ubiquitous Comput.* **21**, 989–1001 (2017).
7. Bly SA, I. S., Harrison SR. Media spaces: bringing people together in a video, audio, and computing environment. *Commun ACM* **36(1)**, 28–47 (1993).
8. Gao, J. Vr brush. https://assetstore.unity.com/packages/tools/integration/vr-brush-101781 (2017).