

RemoteLab: A VR Remote Study Toolkit

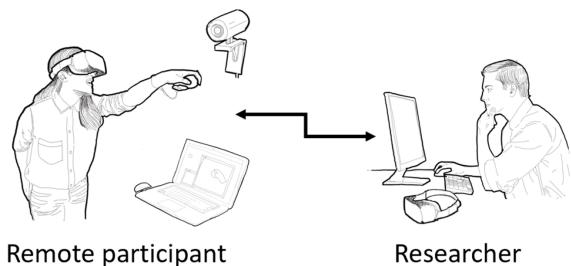
Jaewook Lee*
University of Washington
Seattle, Washington, United States of America
jaewook4@cs.washington.edu

Raahul Natarajan*
Vanderbilt University
Nashville, Tennessee, United States of America
raahul.natarajan@vanderbilt.edu

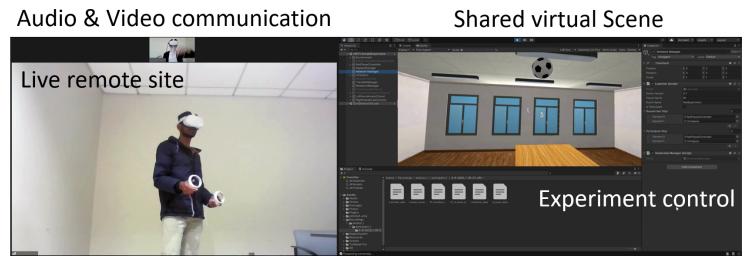
Sebastian S. Rodriguez
University of Illinois at Urbana-Champaign
Urbana, Illinois, United States of America
srodrri44@illinois.edu

Payod Panda
Microsoft Research
Cambridge, United Kingdom
payod.panda@microsoft.com

Eyal Ofek
Microsoft Research
Seattle, Washington, United States of America
eyalofek@microsoft.com



Remote participant



Researcher's overview

Figure 1: *RemoteLab* is a novel open-source Unity toolkit that enables synchronized play of a user study over multiple sites. The researcher can view the VR scene shared among participants, control the experimental conditions, and intervene if needed, such as guiding a participant through filling a questionnaire in VR (left).

ABSTRACT

User studies play a critical role in human subject research, including human-computer interaction. Virtual reality (VR) researchers tend to conduct user studies in-person at their laboratory, where participants experiment with novel equipment to complete tasks in a simulated environment, which is often new to many. However, due to social distancing requirements in recent years, VR research has been disrupted by preventing participants from attending in-person laboratory studies. On the other hand, affordable head-mounted displays are becoming common, enabling access to VR experiences and interactions outside traditional research settings. Recent research has shown that unsupervised remote user studies can yield reliable results, however, the setup of experiment software designed for remote studies can be technically complex and convoluted. We present a novel open-source Unity toolkit, *RemoteLab*, designed to

facilitate the preparation of remote experiments by providing a set of tools that synchronize experiment state across multiple computers, record and collect data from various multimedia sources, and replay the accumulated data for analysis. This toolkit facilitates VR researchers to conduct remote experiments when in-person experiments are not feasible or increase the sampling variety of a target population and reach participants that otherwise would not be able to attend in-person.

CCS CONCEPTS

- Human-centered computing → User studies; Laboratory experiments; User interface toolkits; Mobile devices; Handheld game consoles.

KEYWORDS

Virtual reality, user study, Unity, toolkit, remote user study

ACM Reference Format:

Jaewook Lee, Raahul Natarajan, Sebastian S. Rodriguez, Payod Panda, and Eyal Ofek. 2022. RemoteLab: A VR Remote Study Toolkit. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*, October 29–November 2, 2022, Bend, OR, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3526113.3545679>

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST '22, October 29–November 2, 2022, Bend, OR, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9320-1/22/10...\$15.00

<https://doi.org/10.1145/3526113.3545679>

1 INTRODUCTION

User studies are a critical part of human-computer interaction (HCI) research – they empower researchers to uncover insights from participant interactions. Often, user studies involve carefully planned scenarios where a diverse group of participants is introduced to new technologies or tasks that are designed to measure people's capabilities and behaviors. When conducting user studies, it is crucial for researchers to design the test environment such that it is identical across all participants, varying only the parameters that are of research interest. To achieve this, researchers utilize controlled laboratory environments, where participants are invited to partake in in-person studies under close supervision.

However, in-person user studies have some limitations. Notably, the need to bring users into a laboratory during working hours substantially restricts the quantity and variety of participants volunteering for an in-person study [8]. This effort can quickly grow in complexity if a study requires more than one participant to be present at the same time. Additionally, special consideration is needed if the population of interest may have difficulties accessing the laboratory due to physical limitations or the inability to leave their current location. Furthermore, in the past few years, it has become challenging to conduct in-person user studies due to the increasing need for social distancing brought on by the COVID-19 pandemic. To circumvent these roadblocks, researchers opted to conduct user studies remotely using various approaches, including delivering the needed software and hardware to participants' locations, observing participants through video conferencing software (e.g., Microsoft Teams, Zoom), and demonstrating the desired interactions to participants through video recordings [16].

Among many tools used for research, virtual reality (VR) has rapidly gained popularity across various fields, especially within HCI and interactive computing research [12]. VR research can be challenging in that it requires specialized hardware such as head-mounted displays (HMDs), motion-tracked handheld controllers, and haptic devices. Often, participants have little to no experience interacting with such technologies. Therefore, to ensure the safety of participants and fruitful findings from the experiment, researchers must perform several procedures, including setting up the experiment in a controlled environment, making sure that the equipment is worn properly, guiding participants throughout the study, and monitoring the surroundings of participants. Because of the complexity of this process, VR studies are predominantly conducted in-person [12].

However, the recent proliferation of consumer-level VR hardware [7] opens new opportunities to distribute user studies outside of a traditional laboratory setting. Prior research has shown that it is feasible to collect valid and meaningful data from remote VR user studies [17, 18]. To facilitate the process of conducting remote VR studies, there is a growing need for researchers to be able to manage one or more remote devices, control several different software from game engines to video recorders, interact with participants, administer questionnaires and interviews, among other approaches.

In this paper, we present *RemoteLab*, an open-source toolkit for the Unity game engine [4]. *RemoteLab* facilitates the development and execution of remote VR experiments by synchronizing the virtual environment across users and recording data from mixed

media sources, including participant behavior in the Unity scene, audio communication, and recordings of the participants' physical environment. It provides researchers with greater flexibility in conducting remote VR studies from changing experimental parameters on the go to joining the virtual environment themselves to assist participants. *RemoteLab* streamlines the experiment flow, enabling participants to complete various tasks such as answering questionnaires all within VR without having to remove their HMD and related equipment. The resulting data is recorded locally on the participants' machines, which can later be compressed and transferred to the researcher's machine upon completion of the experiment to lower bandwidth usage. A researcher can then replay the study session interactively, scrubbing through a slider in the Unity Editor to replay the virtual experiment, including participants' avatars and actions, as well as synchronized audio and video recordings.

We present the following contributions:

- (1) *RemoteLab*: A Unity toolkit that enables researchers to easily author, execute, and control an experiment across multiple remote locations.
- (2) The ability to synchronously record multi-modal data from multiple remote sites,
- (3) as well as a replay system that enables researchers to simultaneously view said information.
- (4) *RemoteLab* is open-sourced and architected with modularity in mind, allowing modifications as needed by the research community.

We hope that this toolkit will motivate researchers to employ remote VR user studies more often and obtain a diverse sample of participants that may have been otherwise unreachable in the past.

2 RELATED WORK

As VR hardware progressed to becoming more portable, consumer-level devices, researchers have attempted to move the experimentation from a laboratory towards participants in order to increase the variety of users sampled. For instance, Hodgson et al. used portable computers stored in commercial backpacks [15], while Steed et al. used commercial panoramic headsets such as Samsung Gear VR and Google Cardboard [23]. Biener et al. distributed standalone applications that require the use of HTC Vive headsets for experimentation and data collection, in which participants were asked to send researchers the outputted result files [9].

Radiyah et al. conducted a survey of VR HMD owners – showing great potential for out-of-lab VR experiments [21]. Their work has focused on the distribution methods towards the recruitment of a large number of participants, and the ease with which participants can join an experiment. They identify four primary paths to carry out a VR user study that utilizes different tools for conducting, distributing, and recruiting participants for research. The pathway that provides the most independence to the researcher is to distribute a standalone application to remote participants. This approach relies on users successfully running an application without the researcher's supervision. While this allows high flexibility for participants, researchers found that they had to invest more effort than in-person laboratory studies to ensure that their applications are robust against software issues and other unexpected events.

To overcome this problem, Radiah et al. investigated using online social virtual worlds (e.g., VRChat, Horizon Worlds) as platforms for experiments [21]. While such solutions enable easy setup for remote participants and some level of communication with the researchers, it suffers from service dependencies when implementing experiments. Furthermore, no information can be perceived in the participant's physical environment, since communication usually begins when the participant is already in VR and connected to the service. Any issues that would stem from the physical environment can then only be resolved by the participant. However, recent research has demonstrated the viability of using both social VR platforms [22] and remote experimentation [18] to collect valid research results.

While the above approaches are promising in their ability to reach a large population of participants for simple VR experimentation, we see a practical gap in this approach, as remote sites do not lend themselves to highly controlled experiments as an in-person laboratory experiment. Experiments that use specialized hardware, from haptic devices to motion-tracking cameras, may require the researcher to guide a participant on how to operate the equipment. Experiments that involve the movement of users in physical space should ensure participant safety as their connection to the physical world is altered by the HMD. Conducting an experiment with multiple concurrent participants may require fine access and timing controls to ensure valid data collection. To address the different needs of various VR researchers, a toolkit should allow for ease of integration and extendability.

RemoteLab shares some features with existing VR toolkits. For instance, the literature highlights the importance of a replay system [19] and questionnaires [10], both of which *RemoteLab* makes extensive use of. *RemoteLab* is motivated by prior work, yet brings several key novel features. While existing VR toolkits focus on a specific set of users (e.g., teachers [19]), *RemoteLab* is user-agnostic. External use cases may include user playtesting, interventional pedagogy, and remote debugging. *RemoteLab* also aims to address the gap in remote experimentation toolkits: it enables researchers to view, query, and visualize recorded data for observations and communicate with participants and intervene when necessary.

Integrating remote functionality in software for experiments requires a significant amount of effort by research developers. Dealing with common needs such as state synchronization, data recording, two-way communication, access to non-VR instructions, and authoring questionnaires often take away time that could be used for enhancing the virtual environment or debugging functionality. We hope that *RemoteLab* will enable researchers to do so with little effort.

3 DESIGNING REMOTELAB

Our design choices were motivated by the following high-level goals:

- (1) Enabling a platform for online synchronized experimentation of VR user studies.
- (2) Giving researchers full, streamlined control in controlling and executing experiments.
- (3) Providing extensive recording and replay utilities with minimal data bandwidth usage.

- (4) Designing the toolkit to be easy to integrate and use.

3.1 Control

In VR user studies, participants with various levels of experience interacting with VR technology would need to learn how to navigate an unfamiliar, synthesized environment – all while following the researchers' directions. In an in-person study session, researchers can directly assist participants by reviewing their actions both in the physical and virtual environments. In contrast, researchers do not have this privilege in a remote setting. To this end, *RemoteLab* was implemented to give researchers as much feedback about both the real and virtual environment. Researchers can observe participants in several different ways: 1) participant's physical space through traditional video call software, 2) participant's virtual space using a movable virtual camera, and 3) participant's virtual space using a VR avatar. With these features, researchers can assist the participant in wearing the equipment, ensure that the participant remains in a safe physical space throughout the experiment, observe how interactions take place from the participant's point of view, and remotely manipulate the virtual environment by editing scene parameters, such as rearranging objects in the scene or instantiating a questionnaire.

3.2 Streamlining Experimental Flow

User studies are often comprised of several phases. For instance, participants may begin by first completing a questionnaire, then entering the VR environment to complete a task, completing another questionnaire, and so on. We aim to minimize transitions between phases that break experimental flow, such as removing the HMD to complete a questionnaire. Such transitions require heightened participant attention, lengthen the duration of the experiment, and increase the risk of unexpected complications at the remote site [20]. To streamline the experimental flow, we designed *RemoteLab* to facilitate the development of within-environment surveys, so that participants have no need to remove the HMD when changing through phases.

3.3 Ease of Use

Although our toolkit requires researchers to implement experiment-specific logic, we included various pre-built components to promote efficiency. For example, standard questionnaires – such as the NASA Task Load Index (NASA-TLX) [14] and the System Usability Scale (SUS) [11] – have already been implemented to speed up the development of virtual experiments.

We created custom editor user interfaces for each tool in *RemoteLab* to allow researchers to customize and control the features used in their application, from creating new questionnaires to controlling the amount and type of data collected, with little to no additional programming. Additionally, experimenters are able to expose any experiment parameters in the Unity editor and change them during the experiment. The system handles the synchronization of changes between the multiple sites.

Ultimately, researchers can expand the toolkit's functionality, as the codebase is open-sourced and modular.

3.4 Recording and Replaying

RemoteLab enables researchers to document a wide range of events spread out across one or more remote locations. Additionally, researchers have the option to synchronize the recording of media channels and generate a replay scene that synchronously plays all collected channels. Researchers can replay the study and interactively scrub through its progress using a simple slider to affect all recorded media.

3.5 Bandwidth

Retrieving the data collected from participant interactions with the environment, and reliably sending it to the researcher's machine is crucial. To minimize upload bandwidth during the study, all data is recorded locally at each remote site. A subset is shared with other participants and the researcher during the experiment for state synchronization, but upon completion, all remote sites send their full copy of recorded events to the researcher's machine.

4 SYSTEM IMPLEMENTATION

Figure 2 shows an overview of *RemoteLab*. The left side of the diagram describes the researcher's site. The researcher is running the experiment using the Unity Editor, where they can control any scene objects and experiment parameters. The researcher uses a video conference application to view feeds from participants and establish verbal communication with participants. *RemoteLab* uses Open Broadcasting Software (OBS) [2] to synchronously record both the Unity Editor and the video conference – which keeps track of all video feeds on the researcher's machine. Additional data accumulated on remote participants' machines is transferred to the researcher's machine at the end of the study session.

The right side of Figure 2 shows the participant's site. Each participant needs their own equipment (e.g., computer, HMD, or any specific hardware) to run the experiment. Participants are provided with an executable program compiled by Unity, which may be for a tethered HMD (e.g., Oculus Rift, Valve Index) or for an untethered HMD (e.g., Meta Quest 2). Participants additionally run a video-conferencing client for verbal communication and transmitting a

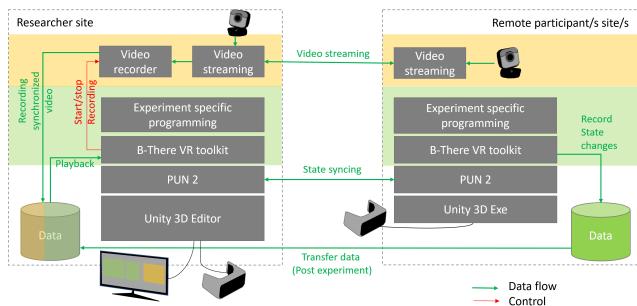


Figure 2: Data flow and component diagram of *RemoteLab*. The flow of the data (green areas) includes the experimental scene, the participants' gaze and hands, as well as questionnaire results. Additional multimedia sources (yellow) include remote sites and the researcher's videos.

video feed of their physical space, as mentioned in the researcher's side.

Below, we describe the system implementation and the use of the toolkit.

4.1 System Overview

We developed *RemoteLab* using the Unity game engine [4] (version 2020.3.16f1).

RemoteLab uses Photon Unity Networking 2 (PUN 2) [3], a popular networking solution for Unity available at the Unity Asset Store¹. To enable one or more participants to join an experiment session, we use PUN 2 to synchronize the virtual environment across the participants' multiple sites. PUN 2 is free to use for up to 20 concurrent users, suitable for most common user study scenarios. PUN 2 broadcasts any changes to selected objects' states (such as position changes) across connected clients, which is independent of the device each client uses. In addition to the *RemoteLab* toolkit, OBS and the obs-websocket² API are installed on the researcher's PC. Besides a video conferencing software, participants only require the executable for the experiment, which is distributed directly to them.

Participants that own a tethered HMD (e.g., Oculus Rift, Valve Index) can execute the application provided by the researcher. For a standalone system – such as the Meta Quest 2 – participants should be given instructions (either written or verbal) as to how to install the given executable, for example, by using SideQuest³.

For our experiments, we approached participants with a network connection of at least 50 Mbps download speed and 10 Mbps of upload speed.

4.2 Authoring Experiments

A typical VR experiment contains one or more scenes, and each may contain a unique environment, task, or questionnaire to be completed by the participants. *RemoteLab* provides a framework of C# scripts, components, and Unity prefabs that allows the researcher to easily record and replay experiments conducted in each of these scenes.

The main code of *RemoteLab* is centered in a few Unity components and *prefabs* that are added to scenes of the experiment. The *NetworkManager* and *TransferManager* prefabs handle communication and synchronization of scene states between clients. The *ReplayManager* component handles the recording of all marked GameObjects and sets up interactive playback of the experiment alongside other stored media, such as OBS recordings.

A *Recordable* component is added to all objects that will be recorded by *ReplayManager* and be a part of the shared environment between sites. Shared object data includes its position, rotation, scale, and active status in the scene. To share UI events in the Unity editor, such as buttons' click events or sliders' change value events, an *InteractableUI* component is added to the UI elements.

In order to synchronize the experiment across multiple participants, *RemoteLab* provides a release of PUN 2 and companion scripts for researchers to apply to their Unity scenes. To use PUN

¹<https://assetstore.unity.com/packages/tools/network/pun-2-free-119922>

²<https://github.com/obsproject/obs-websocket>

³<https://sidequestvr.com/>

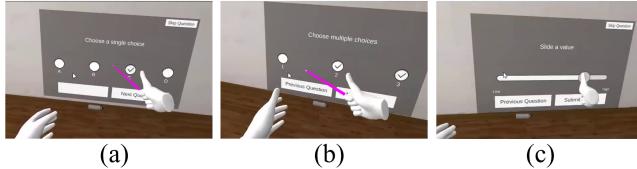


Figure 3: Supported types of questions in virtual reality questionnaires: (a) a single answer selection or a Likert scale, (b) multiple answers, and (c) a slider for entry of analog values.

2, researchers must create an account and application ID on Photon Engine’s website⁴. PUN 2 provides a robust implementation of data synchronization across devices. The researcher should add all objects that need to be shared between sites. In a similar fashion to recording, any object that will be shared among sites in the shared environment is set by adding a *Photon* component to it.

We included an example scene targeting the Meta Quest 2 which is one of the most popular commercial HMDs at the current time. Although not tested, researchers should be able to modify *RemoteLab* to use it with other platforms.

Once *RemoteLab* and PUN 2 are set up in the experiments, *RemoteLab* is ready to collect, transfer, and replay experiment data.

4.3 Questionnaires

Asking participants to fill questionnaires before and after executing tasks is a common procedure to collect subjective impressions from participants. There are a wide variety of standard questionnaires that researchers use to measure qualities such as workload, motion sickness, and system usability, among others. Beyond this, researchers may need to author new questionnaires when trying to measure novel constructs.

Putze et al. [20] found that transitioning from a virtual scene to the physical world in order to fill experience questionnaires can potentially lead to systematic biases, and administering questionnaires in VR (inVRQs) should be preferred. When running a remote experiment, administrating questionnaires in VR is crucial, as transitions involving changes in setups are potential causes for technical issues that will be lengthier to handle since the researcher is remote.

RemoteLab has built-in tools to enable researchers to generate either common or custom-made questionnaires in VR. In contrast to a questionnaire on paper, *RemoteLab* will display one question at a time as an interactive object in the virtual scene. Users can browse through the questions using the next and previous buttons. In doing so, the question will be changed but the question object itself stays in the same location. Thus, it enables the user to walk toward the interactive question widget and interact with it. To maintain flexibility akin to paper questionnaires, we desire to have users browse the questions, and return to past questions to edit if needed.

To author a new questionnaire, a researcher can use a blank questionnaire object and add to it a series of question objects. We implemented three types of common questions (see Figure 3). The

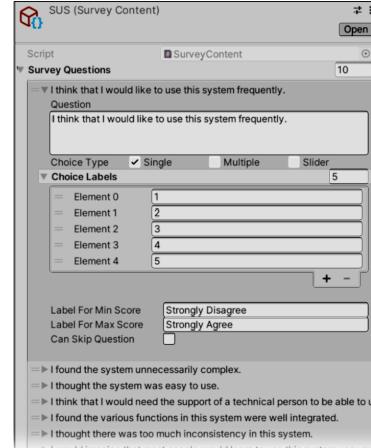


Figure 4: Authoring a question in a questionnaire example.

first is a multiple choice question with a single answer. The set question is displayed to the participant and asks them to select one of the options set by the researcher. A common example of this type is a Likert scale, where the options are in many cases, but not limited to, an ascending numerical scale. The second type is similar to the former, yet enables the user to flag multiple answers as valid, and finally, a slider-type question enables the user to select a value out of a continuous range by sliding a knob to the desired value. The authoring researcher has full control over the number of answers and their values. He can give the participant the ability to skip questions and return to past questions and review them. We currently did not include a free response question; enabling text entry without limiting participants is planned as future work.

We have included several standard questionnaires in the toolkit such as the NASA Task Load Index (NASA-TLX) and System Usability Scale (SUS). Figure 4 shows in detail a single question out of an authored SUS questionnaire. We hope that researchers that will use the open-source toolkit will contribute new questionnaires to the repository, encouraging the creation and reuse of different scales across research fields.

4.4 Researcher Overview of Remote Sites

It is imperative to provide the researcher with an overview of the remote site during a user study, such as a view of the participant’s VR workspace and their equipment setup to overcome possible technical difficulties during the study.

The researcher can guide the participants to place a camera using video conferencing software such that the experiment area is in full view. Seeing the video feed in addition to the shared virtual scene provides the researcher with a more comprehensive view of the participant’s physical and virtual space, enabling them to guide the participant in a task in case a need arises. This view, along with the participants’ audio, is streamed to the researcher using a video-conferencing application.

While running the experiment, the researcher is provided an interface with the current Unity scene, the participants’ physical

⁴<https://www.photonengine.com/pun>

environment, and the participants' point of view. Figure 5 shows a view from a user study, combining multiple virtual views.

The researcher can join the shared virtual scene to instruct, interact with, or guide participants as an avatar. Figure 6 shows the researcher remotely guiding the participant on how to answer a questionnaire's question using a raycast from the avatar's hand. In such cases, it is recommended to have a second researcher watching over the participant's physical environment feed.

In addition to the above displays, the researcher has full control of all parameters of the experiment via the Unity Editor, which will affect the remote sites, and (if needed) can manually control the video recording, which for most cases is automatic.

4.5 Recording

RemoteLab allows recording of the experiment session, including the virtual scene, questionnaire results, and variables of interest into open, time-marked CSV files, as well as additional media such as the remote environment's video and audio communication using the same timestamps.

All transformation changes or status changes of objects and UI elements are marked for recording using the *Recordable* and *InteractableUI* components, and are stored with a timestamp in text data files. Additionally, the researcher can add any custom variables to the logging using *RemoteLab* through simple, one-line, scripting. The researcher and participants each have their own version of the recorded data. To minimize storage size, *RemoteLab* records only changes to the parameters' values, with a periodical full recording of all parameters (*I-frames*).

Alongside the object recording system, *RemoteLab* uses OBS to record additional video sources, including the participants' videos (as streamed by a video conference software) and potentially any other display on the researcher's machine. *RemoteLab* controls OBS and its clips of separate experiment stages can be saved separately, specified by the researcher.

Once the experiment session is finished, *RemoteLab* transfers locally saved data from all remote sites. *RemoteLab* uses Google's protocol buffers⁵ and C#'s DEFLATE algorithm implementation⁶ to compress the text files prior to transfer.

⁵<https://developers.google.com/protocol-buffers>

⁶<https://docs.microsoft.com/en-us/dotnet/api/system.io.compression.deflategroup?view=net-6.0>

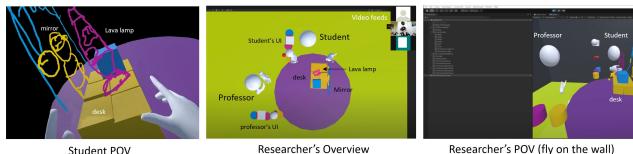


Figure 5: (left): The participant's view shown to the researcher. (middle): The researcher's overview of the activity in the scene from an isometric top-down view. (right): The researcher's virtual camera shows the scene from the side, while she can control the experiment via the Unity-based dashboard.

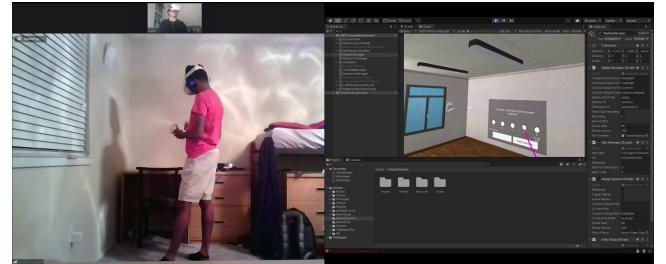


Figure 6: When needed, the researcher can join the shared scene and interact with the participants, such as to show them how to interact with a questionnaire.

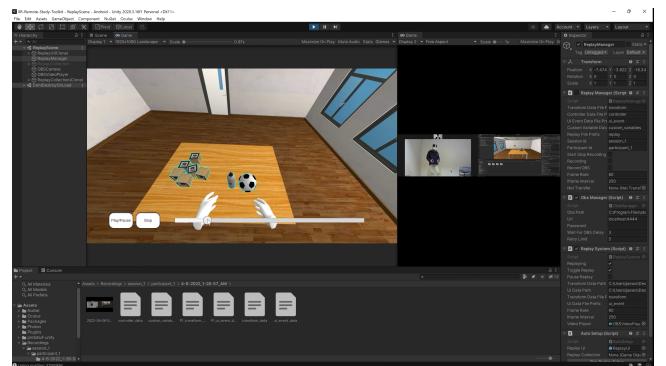


Figure 7: *RemoteLab*'s replay system. On the left of the screen, there is a display of the playback of the 3D scene from the point of view of the participant, along with play control and an interactive time slider. On the right are two synchronized recorded videos: the remote room (with accompanied recorded audio) and another overview of the scene.

To estimate the upper limit of required storage space, we benchmarked 37 recorded 3D objects in a Unity scene, of which 18 objects were moved constantly and with high vigor by a participant for 1 minute. The participant then completed the SUS questionnaire. An average of 5 benchmarks resulted in a text file with an average length of 14,250 lines of stored events, approximately 3 megabytes (MB) in size.

4.6 Playback

Many user studies record videos of study sessions, yet, in many cases, these recordings are underutilized due to difficulty in accessing the information stored. *RemoteLab* enables researchers to easily browse and replay the stored data. With a simple time slider interface, researchers can play, pause, and interactively scrub over a timeline, to watch both a 3D replay of the recorded experiment and additionally of recorded videos (see Figure 7). Moreover, the scene replay can be navigated by the researcher in the Unity Editor and opens new opportunities for analysis such as examining a specific part of the space, or set triggers to enable quick access to events in the scene and their corresponding video moments.

4.7 Conducting a Remote Experiment

Figure 8 shows the steps taken by a researcher and remote participants when preparing and starting a remote experiment. The current system supports one or more participants in a single experiment session, and the researcher and participants can join from remote locations of their choice.

First, the researcher contacts the participants using a video conference tool, and guides them in setting a web camera in position that provides a good view of the experiment area. Together, they can make sure that the play area is free of obstacles, and set any physical objects that need to be used for the experiment.

After signing consent forms and filling out any initial questionnaires, the participants run their executable of the experiment. The researcher guides them on how to don the equipment, and the participants enter the virtual experiment environment. In the environment, they will see the first scene of the experiment, a welcome text, or text instructions relating to the task at hand. Now, the participants listen to the researcher describing the experiment and the task to be completed. The researcher sets OBS to capture any video sources to be recorded. The actual recording is automatically triggered by *RemoteLab*.

During the experiment, the researcher can guide participants using the video conferencing software and interact with the scene itself through the Unity Editor. When needed, they can enter the shared environment in VR to interact with or guide participants.

After the researcher finishes onboarding the participants, the experiment starts. The experiment may contain multiple stages with different tasks, questionnaires, and or interviews by the researcher. When the experiment ends, the participants are instructed to take off their equipment, and the researcher stops the recording application and triggers the data transfer from the remote sites. During this time, participants can fill out any final questionnaires, and the researchers can conduct interviews with participants. When the data transfer is completed, the researcher notifies the participants that they can stop the experiment executable on their devices.

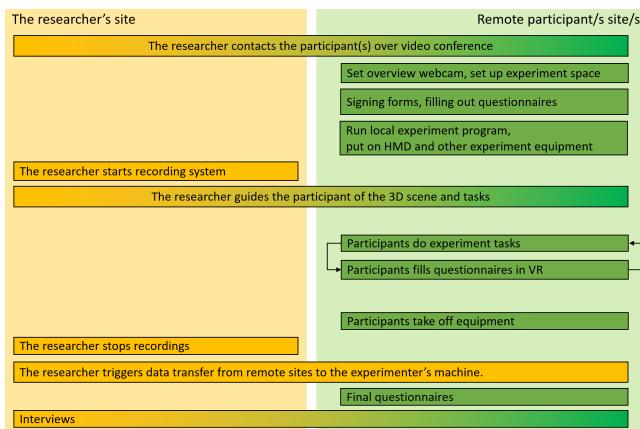


Figure 8: Steps to conduct a remote experiment. Orange steps are completed by the researcher and green steps are completed by participants.

PUN 2 is used to share data during the experiment also to transfer the data at the end of the experiment. This is not a task it is designed for, and as a result, sometimes, the data transfer process at the end of the experiment may take longer than the average we reported. Additionally, the data is also stored on the participants' devices so that the researcher can ask participants to send it over manually in the case of data loss. We plan to use a separate transmission mechanism in future work.

5 INFORMAL FEEDBACK

We elicited informal feedback about our toolkit from researchers with experience running VR user studies, both in industrial and academic institutes. We showed a video of the toolkit in use to 5 researchers. Generally, they felt that our toolkit would make running remote VR studies easier for them, and expressed that they would like to use the toolkit when conducting future studies.

Researchers felt that using *RemoteLab* would be simple and practical for several reasons. First, our toolkit can easily be integrated into researchers' current workflow due to the *prevalence of Unity as a development platform*. Second, our *drag-and-drop modular approach* makes our toolkit user friendly. Researchers especially appreciated how objects can be logged and shared across multiple sites simply using drag-and-drop components. Third, the simple integration of our features with the *addition of one line of code* contributes to the simplicity of our tool. Fourth, the inclusion of *authoring support tools* such as pre-built questionnaires and an interface that empowers the creation of new ones instilled confidence in even non-technical researchers. Fifth, researchers valued the replay functionality, in particular the ability to *synchronously replay the experiment with video and audio recordings*, as it allows them to analyze parts of user study sessions in greater detail.

However, researchers also expressed some concerns. One of the researchers commented that questionnaires in VR may not be necessary since participants often fill them online outside VR. However, research indicates that there is value in including questionnaires as part of the VR experience to reduce break-in-presence, and increase the ecological validity of the participant's evaluation of a VR task [6, 20]. Additionally, several researchers were worried that setting up PUN 2 may be challenging to some. To assist those researchers, we included an example scene in our toolkit that showcases how to set up PUN 2.

Researchers expressed that the following features may be beneficial additions to *RemoteLab*:

- Integration of Google surveys.
- A panel that displays all objects being tracked by the toolkit and by PUN 2.
- Toggling researcher visibility while in VR (to enable fly-on-the-wall observation).

6 DISCUSSION, LIMITATIONS, AND FUTURE WORK

In the past two years, social distancing measures forced researchers to switch to conducting user studies remotely. Indeed, going forward, a remote lifestyle is likely to take hold more widely, making the ability to conduct remote user studies increasingly important. However, VR experiments are particularly difficult to run remotely

due to a few reasons: (1) The need for specialized hardware, (2) the difficulty of controlling and troubleshooting an experiment setup in real-time while the experiment is running, and (3) the subsequent logistical hurdles like the types and quality of data that can be captured, and the transfer of collected data to the researcher.

With recent cost reductions and wide availability of consumer VR headsets, the number of people that own VR devices has grown. It is estimated that by the end of 2021, Meta sold 10 million Quest 2 units [13]. This wide distribution eases the first hurdle of hardware requirement, and opens the possibility for a software-distribution only approach to VR experiments. Additionally, this allows access to a much wider audience than is available today to researchers.

In this work, we present a tool that eases the difficulty of setting up, conducting, and collecting good quality data from a VR user study remotely, particularly for supervised remote user study scenarios where researchers are involved in every session. While such scenarios limit the number of experimental sessions, and may not fully utilize the full population of HMD owners, our goal was to provide flexibility to researchers and enable them to implement an experimental scenario with the capability to synchronize and collect data from remote participants. Even with the need to contact and distribute software (and sometimes hardware) to every participant, *RemoteLab* can enable researchers to access a previously-inaccessible participant pool, facilitating participant recruitment.

While this toolkit is designed for remote sites, it is entirely possible to use the modular features for in-person experiments too. One caveat for multiple participants in an in-person approach is that the participants need to calibrate their position in the real world with the virtual scene (so as to not mismatch and potentially cause safety issues), which we have yet to implement.

Our current implementation uses simple compression techniques to reduce storage usage and bandwidth transmissions during and after the experiment. While it makes the toolkit functional, we want to improve the compression and serialization techniques used in the system so data upload is not an obstacle for researchers and participants with low bandwidth.

We want to further extend the platform to include more media streaming and recording; for instance, depth sensors are becoming commonplace in HMDs, as well as stand-alone depth cameras (e.g., Microsoft's Kinect [1] or StereoLabs's Zed stereo camera [5]) for improved sensing of the physical environment. The current toolkit was designed and tested for Unity environments designed for VR HMDs, but we can envision its use for other displays, from AR HMDs to projection system, that uses content designed in the Unity game engine.

It is unfeasible to account for every need and request that may be raised by researchers in the future. We have designed *RemoteLab* with a modular architecture, so it could be freely modified for the needs of the research community – its codebase open-sourced for everyone to use. An example is that the tool includes a small set of questionnaires used for user studies, but anyone could author and contribute new questionnaires that could be used in the future. We hope that with contributions from the research community, this toolkit will be expanded and include more features to help researchers author and share virtual environments and features.

7 CONCLUSION

In this paper, we present *RemoteLab*, an open-source Unity toolkit for authoring, conducting, and replaying remote synchronized experiments in virtual reality. This toolkit facilitates the process of conducting remote VR experiments when in-person experiments are not feasible or when researchers desire a more diverse set of participants – reaching participants that otherwise would not be able to attend in-person. It is designed with modularity and simplicity in mind, allowing researchers to easily integrate new features needed to continue their research.

REFERENCES

- [1] 2022. Microsoft Azure Kinect. <https://azure.microsoft.com/en-us/services/kinect-dk/#overview>
- [2] 2022. OBS Studio. <https://obsproject.com/>
- [3] 2022. Photon engine. <https://www.photonengine.com/en-US/PUN>
- [4] 2022. Unity. <https://unity.com/>
- [5] 2022. Zed Stereo Camera. <https://www.stereolabs.com/zed/?msclkid=e0fa30cb2ab11ec9a2be99fe5c72fc1>
- [6] D. Alexandrovsky, S. Putze, M. Bonfert, S. Höffner, P. Michelman, D. Wenig, R. Malaka, and J.D. Smeddinck. 2020. Examining Design Choices of Questionnaires in VR User Studies. *Human Factors in Computing Systems* (Apr 2020). <https://doi.org/10.1145/3313831.3376260>
- [7] Thomas Alsop. 2021. Consumer Virtual Reality (VR) market size worldwide 2023. <https://www.statista.com/statistics/528779/virtual-reality-market-size-worldwide/>
- [8] L. Barkhuus and J.A. Rode. 2007. From Mice to Men - 24 Years of Evaluation in CHI. In *CHI '07 (CHI '07)*. ACM. <https://doi.org/10.1145/1240624.2180963>
- [9] V. Biener, D. Schneider, T. Gesslein, A. Otte, B. Kuth, P. Kristensson, E. Ofek, M. Pahud, and J. Grubert. 2020. Breaking the Screen: Interaction Across Touchscreen Boundaries in Virtual Reality for Mobile Knowledge Workers. *IEEE Transactions on Visualization and Computer Graphics* 26, 12 (dec 2020), 3490–3502. <https://doi.org/10.1109/TVCG.2020.3023567>
- [10] R. Bovo, D. Giunchi, A. Steed, and T. Heinis. 2022. MR-RIEW: An MR Toolkit for Designing Remote Immersive Experiment Workflows. In *IEEE Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 766–767. <https://doi.org/10.1109/VRW55335.2022.00024>
- [11] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
- [12] P. Cipresso, I.A.C. Giglioli, M.A. Raya, and G. Riva. 2018. The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature. *Frontiers in Psychology* 9 (2018). <https://doi.org/10.3389/fpsyg.2018.02086>
- [13] C. Gartenberg. 2021. Meta's Oculus Quest 2 has shipped 10 million units, according to Qualcomm. <https://www.theverge.com/2021/11/16/22785469/meta-oculus-quest-2-10-million-units-sold-qualcomm-msclkid=411e0153aab911ecbbdd4ee744273573>
- [14] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9)
- [15] E. Hodgson, E.R. Bachmann, D. Waller, A. B, and A. Oberlin. 2012. Virtual reality in the wild: A self-contained and wearable simulation system. *IEEE Virtual Reality Workshops (VRW)* (2012), 157–158.
- [16] N. Marquardt, N. Henry Riche, C. Holz, H. Romat, M. Pahud, F. Brudy, D. Ledo, C. Park, M.J. Nicholas, T. Seyed, E. Ofek, B. Lee, B. Buxton, and K. Hinckley. 2021. AirConstellations: In-Air Device Formations for Cross-Device Interaction via Multiple Spatially-Aware Armatures. In *UIST'21*. ACM, ACM, 1252–1268. <https://www.microsoft.com/en-us/research/publication/airconstellations-in-air-device-formations-for-cross-device-interaction-via-multiple-spatially-aware-armatures/>
- [17] A. Mottelson and K. Hornbæk. 2017. Virtual Reality Studies Outside the Laboratory. In *ACM Symposium on Virtual Reality Software and Technology (Gothenburg, Sweden) (VRST '17)*. ACM, Article 9, 10 pages. <https://doi.org/10.1145/3139131.3139141>
- [18] A. Mottelson, G.B. Petersen, K. Lilija, and G. Makransky. 2021. Conducting Unsupervised Virtual Reality User Studies Online. *Frontiers in Virtual Reality* 2 (2021). <https://doi.org/10.3389/fvr.2021.681482>
- [19] M. Nebeling, S. Rajaram, L. Wu, Y. Cheng, and J. Herskovitz. 2021. XRStudio: A Virtual Production and Live Streaming System for Immersive Instructional Experiences. In *CHI '21 (Yokohama, Japan) (CHI '21)*. Article 107, 12 pages. <https://doi.org/10.1145/3411764.3445323>

- [20] S. Putze, D. Alexandrovsky, F. Putze, S. Höffner, J.D. Smeddinck, and R. Malaka. 2020. Breaking the Experience: Effects of Questionnaires in VR User Studies. *Human Factors in Computing Systems* (apr 2020). <https://doi.org/10.1145/3313831.3376144>
- [21] R. Radiah, S. Prange, S. Delgado Rodriguez, University Munich ROBIN PIENING, Bundeswehr, K. Köhle Y. Zhou, L. Munich KEN PFEUFFER, Y. Abdelrahman, University Munich MA. H., Bundeswehr, A. Schmidt, L. Munich FLORIAN ALT, V. Mäkelä, R. Piening, K. Pfeuffer, M. Hoppe, F. Alt, D. Rodriguez, K. Pfeuffer, Y. Abdelrahman, F. Alt, R. Piening, Y. Zhou, K. Köhle, M. Hoppe, and A. Schmidt. 2021. Remote VR Studies: A Framework for Running Virtual Reality Studies Remotely Via Participant-Owned HMDs. *ACM Transactions on Computer-Human Interaction* 28, 6 (2021). <https://doi.org/10.1145/3472617>
- [22] D. Saffo, S.D. Bartolomeo, C. Yildirim, and C. Dunne. 2021. Remote and collaborative virtual reality experiments via social VR platforms. *Conference on Human Factors in Computing Systems - Proceedings* (May 2021). <https://doi.org/10.1145/3411764.3445426>
- [23] A. Steed, S. Frilston, M.M. Lopez, J. Drummond, Y. Pan, and D. Swapp. 2016. An 'In the Wild' Experiment on Presence and Embodiment using Consumer Virtual Reality Equipment. *IEEE TVCG* 22, 4 (2016), 1406–1414. <https://doi.org/10.1109/TVCG.2016.2518135>