



OPEN ACCESS

EDITED BY

Doug A. Bowman,
Virginia Tech, United States

REVIEWED BY

Ferran Argelaguet,
Institut National de Recherche en
Informatique et en Automatique (INRIA),
France
Blair MacIntyre,
Georgia Institute of Technology,
United States

*CORRESPONDENCE

Anthony Steed,
A.Steed@ucl.ac.uk

SPECIALTY SECTION

This article was submitted to Virtual
Reality and Human Behaviour,
a section of the journal
Frontiers in Virtual Reality

RECEIVED 03 April 2022

ACCEPTED 25 August 2022

PUBLISHED 14 October 2022

CITATION

Steed A, Izzouzi L, Brandstätter K,
Friston S, Congdon B, Olkkonen O,
Giunchi D, Numan N and Swapp D
(2022), Ubiq-exp: A toolkit to build and
run remote and distributed mixed
reality experiments.
Front. Virtual Real. 3:912078.
doi: 10.3389/frvir.2022.912078

COPYRIGHT

© 2022 Steed, Izzouzi, Brandstätter,
Friston, Congdon, Olkkonen, Giunchi,
Numan and Swapp. This is an open-
access article distributed under the
terms of the [Creative Commons
Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other
forums is permitted, provided the
original author(s) and the copyright
owner(s) are credited and that the
original publication in this journal is
cited, in accordance with accepted
academic practice. No use, distribution
or reproduction is permitted which does
not comply with these terms.

Ubiq-exp: A toolkit to build and run remote and distributed mixed reality experiments

Anthony Steed*, Lisa Izzouzi, Klara Brandstätter,
Sebastian Friston, Ben Congdon, Otto Olkkonen,
Daniele Giunchi, Nels Numan and David Swapp

Department of Computer Science, University College London, London, United Kingdom

Developing mixed-reality (MR) experiments is a challenge as there is a wide variety of functionality to support. This challenge is exacerbated if the MR experiment is multi-user or if the experiment needs to be run out of the lab. We present Ubiq-Exp - a set of tools that provide a variety of functionality to facilitate distributed and remote MR experiments. We motivate our design and tools from recent practice in the field and a desire to build experiments that are easier to reproduce. Key features are the ability to support supervised and unsupervised experiments, and a variety of tools for the experimenter to facilitate operation and documentation of the experimental sessions. We illustrate the potential of the tools through three small-scale pilot experiments. Our tools and pilot experiments are released under a permissive open-source license to enable developers to appropriate and develop them further for their own needs.

KEYWORDS

virtual reality, 3D user interfaces, experiment design, distributed experiments, remote experiments

1 Introduction

Mixed-reality (MR) systems are a very active area of commercial exploitation. However, despite research on these systems beginning over 30 years ago [Ellis \(1991\)](#), there is still a huge design space to consider. For example, the recent rapid expansion of consumer virtual reality (VR) has led to many innovations in 3D user interface design [Steed et al. \(2021\)](#), and the increasing interest in social uses has led to many commercial systems exploring the telecollaboration space (e.g. [Singhal and Zyda \(1999\)](#); [Steed and Oliveira \(2009\)](#)). While these systems have interesting designs and features, there are still a lot of challenges to improve the overall usability and accessibility of MR.

Despite the availability of good toolkits to build MR content, building experimental systems to explore the design space is time-consuming. While building an interactive demonstration of an interesting design or interaction pattern might not be that time-consuming, evolving the system and tooling to be stable enough to run controlled experiments is significantly more difficult. There are several challenges with the system, such as the overall interaction design, implementing metrics and logging, data gathering

such as questionnaires, and providing standardised experimental protocols. There are also challenges in the deployment of experiments. For example, even within a laboratory, making sure that the experimenter has good visibility of the actions of the participant requires extra support in the system, such as rendering a third-person view. More challenging is the recent push, both as COVID-19 mitigation and also to diversify participant pools, to run experiments ‘in the wild’ and thus out of the direct control of the experimenter.

In this paper, we describe Ubiq-Exp - a set of tools, demonstrations, and conventions that we have developed to facilitate distributed experiments. At its core is a distributed system based on Ubiq, which is a toolkit for building distributed MR systems [Friston et al. \(2021\)](#). Ubiq comprises a Unity package and associated services. It is key to the utility of our system because it is open-source under a permissive license. Experimenters can set up their own system to have total control over data flows and easily configure the system to meet national legislation or local policy about data protection. To facilitate distributed experiments across multiple sites, we have added functionality for distributed logging, management of participants, and recording and replay of sessions while online. In addition, we have added features that are useful for experiment design, such as a flexible questionnaire system that includes prototypes of several popular MR questionnaires, tools to utilise the Microsoft Rocketbox avatars, and tools to analyse log files in external systems such as MATLAB and Excel. Although the demonstrations in this paper are all virtual reality examples, Ubiq supports MR systems, and one of the experiments we report later has been ported to a mixed AR/VR setup.

We demonstrate Ubiq-Exp through three pilot experiments. The first is a single-user experiment that exploits distributed logging functionality. This is a common case of moving an experiment out of the laboratory. To the experiment designer, who has access to a running Ubiq server, the relevant tooling to keep distributed logs is a one-line command on the server and a small change to any existing console-style logging. The second experiment is a two-person experiment that involves collaborative and competitive puzzle solving. Since the hypothesis of the study is unconscious use of body positioning, a key aspect was the ability of the experimenter to act as an invisible third participant, while recording the session so that the two experimental participants could then re-observe their actions and talk aloud about them. The third experiment is a three-person experiment. This needs the experimenter to manage more activities, so this experiment exercises features to manage inter-personal awareness and stage-manage the activities of participants.

Ubiq-Exp is not prescriptive about how experiments should be implemented in terms of allocation of conditions, participants identifiers, automation of phases and trials, etc. Our starting point with Ubiq-Exp is that the experimenters should be

primarily concerned with how to implement their experiment based on their own practice, conventions and perhaps their own code. However they can use Ubiq-Exp to facilitate the remote and distributed experiment modes. Having said this, our three pilot studies give examples of how we managed these issues in different ways. In particular, Experiment 1 is an autonomous experiment with three factors (eight conditions), with several scenes, two of which contain questionnaires, one a results table. This Unity project can act as a template for future developers.

Ubiq-Exp is released under a permissive open-source license (Apache). It can be re-used as required, even for commercial purposes. As far as possible, all three of the experiments presented are also released as open-source. Licensed assets cannot be shared, but the documentation includes instructions for how to purchase and include any required assets (e.g. FinalIK for avatar animation, but also some graphics and sound assets). As such, we hope these tools help other researchers realise their experimental systems more easily.

The contributions of our paper and Ubiq-Exp are as follows:

- We provide a breakdown of the tools required to do distributed and remote experiments.
- We described specific implementation of certain features that support a broad range of experiments. We explain how these features are hard to support on commercial platforms.
- Three demonstrations of the toolkits that illustrate the types of experiment that are possible.
- The source code to these three experiments serves both to reproduce our work, and to serve as templates for future experiments.

2 Related work

2.1 Experimental VR platforms

Early VR platforms (e.g., Reality Built For Two [Blanchard et al. \(1990\)](#)) were supplied with monolithic software systems that provided a deep stack of functionalities, from device management through graphics rendering to multi-user support. Thus, experimental systems were tightly bound to the hardware and software platforms provided by vendors. While open-source systems such as VRJuggler [Bierbaum et al. \(2001\)](#) were successful in providing a platform to port software to different hardware configurations, they were difficult to access for non-programmers.

While we do not know of a definitive survey, we are confident in predicting that in recent years, the majority of development of MR systems for research has moved on to game-development systems such as Unity or Unreal. These systems have broad support for different platforms and graphics architectures. Further, in recent years they have accumulated drivers and

supporting software for a wide variety of 3D interaction devices, including modern MR systems.

A relatively new class of web-focused engines can support WebGL and sometimes WebXR contexts. For example, Babylon.js¹ is an open source web rendering engine with support for WebXR. PlayCanvas is a game engine that targets HTML5². It also supports WebXR. While one could build experiments for these or similar platforms, their web deployment context does imply some restrictions, such as having to work with the JavaScript sandbox in the browser, and the page-based security model meaning that some functionality such as interfacing to peripherals is harder. A recent survey indicates a preference for Unity amongst researchers Radiah et al. (2021). Our own tools have used Unity for reasons we discuss later. In the Conclusion, we return to discuss the potential for developing WebXR support in our Ubiq-Exp framework.

There are emerging tools that support particular classes of experiments, such as behavioural experiments based on psychology Brookes et al. (2020) or neuroscience Juvrud et al. (2018) protocols. Ubiq-Exp focuses more on the support of more consumer content-like experiences, such as social VR or experiments on 3D user interfaces. However, there is nothing preventing its use with other types of experiments, so long as Unity supports them.

Over the past few years, a number of tools have appeared to facilitate experiments in MR. One class of tool provides questionnaires within the experience Bebko and Troje (2020); Feick et al. (2020). These are usable and effective Alexandrovsky et al. (2020), partly because they do not disrupt the participant's continued experience in the MR by causing a break in presence Putze et al. (2020); Slater and Steed (2000). Ubiq-Exp has a tool for generating in-world questionnaires, see Section 4.2. Another type of tool that several research teams have developed is a record and replay tool, described in Section 2.3.

2.2 Social MR and toolkits

The origins of networked and social VR systems can be found in game and simulation technologies Singhal and Zyda (1999); Steed and Oliveira (2009). While early immersive systems were often set up with multi-user facilities (e.g. Reality Built For Two Blanchard et al. (1990)), numerous early studies focused on desktop VR systems Damer (1997); Churchill and Snowdon (1998). Early social VR systems were often purpose-built with a focus on the network technology that would be required to

allow such systems to be deployed outside of the academic networks of the times (e.g., DIVE Carlsson and Hagsand (1993) or MASSIVE Greenhalgh and Benford (1995)). Various researchers have continued the technical exploration of distributed systems Latoschik and Tramberend (2011); Steed and Oliveira (2009), but much of the user experience work has migrated to commercial platforms.

Research on the user experience of social VR systems takes several forms, from focused experiments through to longitudinal studies Schroeder (2010). A particular focus has been the experience of being together with other people, or social presence Biocca et al. (2003); Oh et al. (2018). Particular aspects of interest from user-interface and development points of view are how the representation of the user as avatar leads to natural non-verbal behaviour Fabri et al. (1999); Yee et al. (2007) and how the users react to different avatar representations Pan and Steed (2017); Latoschik et al. (2017); Moustafa and Steed (2018); Freeman et al. (2020); Dubosc et al. (2021).

In the past couple of years, many commercial social VR platforms have been developed. Schulz's blog lists over 160 at the time of writing Schulz (2021). These systems support a wide variety of avatar and interaction styles, see reviews in Kolesnichenko et al. (2019); Jonas et al. (2019); Tanenbaum et al. (2020); Liu and Steed (2021).

2.3 Logging and replay systems

The generation of user experiences in immersive MR systems requires tracking of the user, and this data can provide a useful source of information for future analysis. This might be recorded alongside voice, to provide data assets for creating novel experiences that involve replay of previous sessions Greenhalgh et al. (2002); Morozov et al. (2012). Alternatively, it might be recorded with other multi-modal data Friedman et al. (2006); Steptoe and Steed (2012) in documented formats so that it can be shared with multiple, analysts. For example, Murgia et al. Murgia et al. (2008), analysed and visualised patterns of eye-gaze in three-way social settings. Particularly promising areas of exploitation of log files are their use to enable after-action review Rajj and Lok (2008) or summarisation of experiences Ponto et al. (2012). Recently Wang et al. explored how participants could relive social VR experiences in different ways Wang et al. (2020). The goal was to facilitate a better joint understanding of the previous experience. In contrast, Büschel et al. visualise previous track logs from mixed-reality experiences in order to garner insights into log files as a whole Büschel et al. (2021). Ubiq-Exp includes functionality to log multi-user experiences and currently focuses on replaying in real time within a multi-user context, see Section 4.2. Visualisation of whole log files is part of future work.

¹ <https://www.babylonjs.com/>.

² <https://playcanvas.com/>.

2.4 Distributed XR experiments

Recently, some researchers have moved toward running studies out of the laboratory or ‘in the wild’ Steed et al. (2016); Mottelson and Hornbæk (2017). The COVID-19 pandemic has forced a lot of work to be done remotely Steed et al. (2020). There is a rapidly growing body of knowledge about the potential advantages and disadvantages of remote studies Steed et al. (2021); Ratcliffe et al. (2021); Zhao et al. (2021).

A recent paper by Saffo et al. examines the possibilities of running experiments on social VR platforms Saffo et al. (2021). However, the amount of precise control the experimenter has over the experiment is limited as social VR platforms do not typically provide for much end-user scripting. In particular, this limits the experimenter’s ability to export data. Further, the use of such platforms might not be allowable under local ethical rules or data protection laws. Williamson et al. modified the Mozilla Hubs (Mozilla Corporation, 2022) system to export data Williamson et al. (2021). Being open source, this is a potential platform on which to develop experiments, but it is a complex software stack with the client built around web standards. This means that the development process is not as straightforward as the current game-engine-based tools which include integrated development environments with support for key development activities. As noted in Section 2.1 other web technologies are gaining traction, but at the moment most VR content is distributed as native applications and thus is it natural to support native applications as a first stage. Ubiq-Exp aims to simplify the construction of social MR experiments by enabling key functionality without much additional coding.

3 Requirements for remote and distributed experiments

3.1 Styles of experiment

Our main aim with Ubiq-Exp is to support the running of distributed and remote experiments. Radiah et al. (2021) provide a useful framework for considering remote studies. They focus on the means by which participants are recruited, and then how the experimental software is supplied to the participants. These are key issues in remote XR experiments. It is usually expected that the participants have access to VR or AR devices, but software distribution is still a problem. Ubiq-Exp targets the development of full applications. This is by design: as discussed in Section 2.1 and Section 2.4, we consider that a large segment of the anticipated MR experiments cannot be developed and distributed on commercial platforms. The reasons for this include the fact that platforms do not allow the necessary customisations to the user experience, the client or server software do not allow export of data for security reasons, or the platform’s data processing policies might be opaque. Further,

reproducibility of experiments on commercial platforms is limited, whereas with Ubiq-Exp and similar tools, there is potentially the option of releasing the full source code. This is one motivation for making Ubiq-Exp open source. Thus, any application built with Ubiq-Exp would need to be distributed directly (e.g. downloadable zip files or APKs), or using a third-party distribution service (e.g. Steam, Google Play, or SideQuest). Given that the participant has access to the software, there are still many styles of experimental design. We identify four styles of experiment that we summarise in Table 1 and describe below.

If we consider single-participant, in-person experiments, we see that there is the need for both *supervised, single participant* and *unsupervised, single participant* modes. That is, whether an experimenter needs to be online or not at the same time. In a supervised mode, we might want the experimenter to have observation abilities, i.e., that they can remotely watch what the participant is doing through the collaboration software, as they might do during an in-lab experiment. Video observation of participants might be required. If needed, it can be supported by Microsoft Teams, Zoom or a similar tool. Our own local ethics practice is to not collect video of remote participants unless strictly necessary and thus is it not a primary requirement for Ubiq-Exp. It would probably require a second device to be included within the participant’s setup and this might be tricky to support within the same technical system. In a supervised mode, the experimenter can also have control over the experiment. For example, to set conditions and progress the experiment, or even to run a Wizard of Oz-style experiment where they have significant controls over behaviours of objects in the environment. This mirrors the way in which an experimenter might use a keyboard or other device to interact with the software for an in-person experiment. For example, in our own lab it is common to run experiments from within the Unity Editor directly provided it does not affect the speed and responsiveness of the application. Once running, the experimenter still has access to the editor, and optionally a view of the scene. Thus they might use a keyboard, in-view scene controls (e.g. on screen button), or even set values or use buttons directly in the Unity property editor sheets.

In unsupervised mode, the experiment must be self-contained and run without intervention from the experimenter. Such applications require more effort to build because they need to include full instructions, deal with failure cases, unexpected user behaviour, etc. For example, the software must be able to deal with participants being interrupted while doing the experiment and thus pausing the software.

If the experiment involves multiple users, we can similarly identify *supervised, multi participant* and *unsupervised, multi participant* experiments. For a supervised experiment, the experimenter might want the option of being embodied in the scene. We anticipate that the experimenter might need additional functionality such as the ability to make themselves invisible and manage the participants. We can also envisage unsupervised

TABLE 1 Key functionality for different experimental types in distributed and remote MR experiments. 1P refers to single participant, and 2 + P to multi-participant.

Style	Key Functionality
Super, 1P	Single participant can take part in experiment. Experimenter is able to supervise and optionally present to the user. Logs should be recorded locally.
UnSuper, 1P	Single participant runs experiment independently. No experimenter control, but log is captured.
Super, 2 + P	Multiple participants can be coordinated in a single session by the experimenter. Experimenter is able to control the session, observe and optionally present to the users. Logs should be recorded locally.
UnSuper, 2 + P	Multiple participants are automatically marshalled into sessions. No experimenter control, but logs are captured.

TABLE 2 Requirements for the four styles of experiment identified and the three pilot experiments.

	R1	R2	R3	R4	R5	R6	R7	R8	R9
	Collaborate	Rendezvous	Logging	Record&Replay	Manage	Lobby	Avatar	Metrics	Video
Style									
Super, 1P	?	✓		?	✓		?	?	?
UnSuper, 1P		✓	✓	?		?	?	?	?
Super, 2 + P	✓			?	✓		?	?	?
UnSuper, 2 + P	✓	?	✓	?		✓	?	?	?
Ubiq-Exp									
	✓	✓	✓	✓	✓	✓	✓	✓	X
Pilot Study									
Bar		✓	✓				✓	✓	
Jigsaw	✓	✓		✓					
Poster	✓	✓			✓		✓		

Note that features are necessary (✓) unless otherwise indicated. Features are categorised as being supported by Core Ubiq or Ubiq-Exp, with additional requirements for the pilot experiments described later. A '?' indicates that the features are only optional in the different styles of experiment. See Section 4.2.7 for discussion of cell marked 'X'.

multi participant experiments, which needs some mechanisms to manage participant connections automatically.

3.2 Requirements

In order to run a remote or distributed experiment we need certain features from the underlying platform. We assume that any technical platform chosen should support some basic features of building MR content, such as support for common XR devices and features to support networking.

For example, if the users are co-located physically but using independent MR systems and all measures are done outside the experience itself, then few features are needed other than basic social VR connectivity. However, many experiments have additional requirements, especially in the remote and distributed scenarios we are interested in. Thus we will

describe requirements for remote and distributed experiments and indicate whether these are fully supported in Ubiq already or require feature extensions to Ubiq. Examples of fully-implemented demonstrations are also very valuable, so we will also indicate whether we have demonstrations of this type of functionality. Table 2 summarises the requirements and matches these against the styles of experiment we identified in Section 3.1. It also summarises for reference back from Sections 5, 6, and 7, which of our demonstrations emphasises these requirements and the implied features.

3.2.1 R1: Collaborative environment

A key feature of supervised experiments is that users and experimenters need to communicate in real-time by controlling avatars, talking, and interacting with objects. We expect that any platform chosen would have these features built in, or provide functionality for adding these features.

3.2.2 R2: Rendezvous

By Rendezvous, we refer to the process of discovering and negotiating access to other peers on a network. Any platform would need to have procedures for connecting to other users. Examples including hard-coding a rendezvous into the applications, supplying something such as a URL to users to connect, or giving clear instructions on how to reach the destination, such as giving a room name. Even the Super 1P and UnSuper 1P styles of experiment need to be able to connect to a logging service (see R3 below) or service that manages the experiments (see R5).

3.2.3 R3: Logging

In almost all user studies, some logging is necessary to validate the experiment protocol, capture appropriate metrics, and ensure participants engaged with the task. In a face to face experiment, some of these tasks can be performed by an experimenter observing the participant or manually saving data from different processes. In the distributed and remote situation, this should be facilitated on the network. There is an alternative which is to log data locally and have participants return devices or manually upload data. Manual upload adds additional risk to the experiment, partly because the instructions might not be straightforward, and it still requires a network service to be available. There are potential data protection issues, such as users resorting to insecure methods such as emailing the data. Therefore, we would like to enable remote data logging in asynchronous situations where the experimenter is not online. In the case of a synchronous experiment, then local logging from the experimenter's side might be sufficient, but the logging service should ideally be flexible enough to accommodate different logging strategies.

3.2.4 R4: Record and replay

As discussed in [Section 2.3](#), the ability to record and replay sessions has significant value in specific experimental settings. This goes beyond logging in that we want to recreate the behaviour of elements of the scene's evolution over time. We want to be able to record the movement and state change of parts of the environment including avatars, and optionally replay them. For example, our later pilot experiment in [Section 6](#) records data and then replays it to participants to solicit their reflections on their activities. Replays can also be useful for the experimenter to observe from different viewing positions, or even take the viewpoint of one of the participants.

3.2.5 R5: Scene management

By scene management we refer to the ability of a process or experimenter to manage the experiment by controlling aspects of its behaviour, including some properties of the representations of users. This can include managing where participants are located, activating behaviours or scripts, and muting or un-muting players.

3.2.6 R6: Lobby system

By Lobby System, we refer to the ability to automatically place users in certain sized groups (pairs, triples, etc.). While a Rendezvous mechanism connects users together, a lobby system would put users into smaller logical groups. In synchronous experiments, the experimenter could manage the peer groups directly. For the asynchronous situation, we add the requirements for a lobby system that operates much like similar systems in games: as participants join the system, they are batched up into groups based on a criteria, usually just the number of participants, and split off into a separate group.

3.2.7 R7: Avatar options

In social MR experiments, it might be very important to control the types of avatar that each user has available and to provide options for this.

3.2.8 R8: Questionnaires

The platform should support the distributed and remote use of simple measures such as questionnaires. While this seems relatively straightforward if the platform allows authoring of new content, there are specific issues, such as supporting independent use of multiple users and remote logging.

3.2.9 R9: Video

In some remote and distributed experiments it would be advantageous to have live video of the participant to enable direct help in synchronous experiments or for subsequent analysis. In some situations it might be useful to have video from the cameras mounted on a HMD, in which case it is possible that the same client software that the participant uses to run the experiment might stream the software. However, often it would be more useful to have a camera that observes the user from a distance. This would probably involve a second device being used, such as the participant's smartphone or laptop. Use of such features needs careful consideration given privacy concerns around video especially in remote situations where the participant might be in their home environment.

3.3 Platform analysis

Before describing how Ubiq and Ubiq-Exp fulfil some of the requirements identified, we give an overview of how such requirements might be implemented on some existing platforms. A longer discussion can be found in [Table 3](#).

First we can identify a large group of commercial social VR platforms, as discussed in [Section 2.2](#). For the experimenter designer, a key feature of such platforms is that they have large user bases already. This could be a pro or a con depending on the population being studied: if the study is concerned with practical use of such systems, this may be a

TABLE 3 Analysis high-level and low-level platforms against general requirements of building experiment content and the specific requirements of distributed and remote experiments.

General Social and Development Features									
	Editing/Flexibility	User Accounts	Audience	Architecture	Devices	Data Control			
High Level Platforms									
Rec Room	L	Y	Y	Fixed	M	N			
VRChat	L	Y	Y	Fixed	M	N			
AltSpaceVR	L	Y	Y	Fixed	M	N			
Spatial	L	Y	Y	Fixed	M+	N			
Mozilla Hubs	M	N	U	Semi-Flexible	M	Y			
Ubiq-Exp	H	N	N	Flexible	H	Y			
Low Level Platforms									
Unity	H	N	N	Flexible	H	B			
Unreal	H	N	N	Semi-Flexible	H	B			
WebXR	M	N	U	Flexible	M	B			
Experiment Requirements									
	R1	R2	R3	R4	R5	R6	R7	R8	R9
	Collaborate	Rendezvous	Logging	Record&Replay	Manage	Lobby	Avatar	Metrics	Video
High Level Platforms									
Rec Room	Y	Y	N	N	N	N	M	N	N
VRChat	Y	Y	N	N	N	N	H	N	N
AltSpaceVR	Y	Y	N	N	N	N	M	N	N
Spatial	Y	Y	N	N	N	N	M	N	N
Mozilla Hubs	Y	Y	B	B	B	B+	H	B	B
Ubiq-Exp	Y	Y	Y	Y	Y	Y	M+	B+	B
Low Level Platforms									
Unity	B	B	B	B	B	B	B	B	B
Unreal	Y	B	B	B	B	B	B	B	B
WebXR	B	B	N	B	B	B	B	B	B

By high-level we refer to platforms targeted at generating social MR experiences. By low-level we refer to more general toolkits for building MR systems. Aside from 'Y'/'N' indicating support for this feature, 'B' implies that such a feature could be built, 'L', 'M' and 'H' indicate low, medium or high level of support for this feature and 'U' indicates that although a permanent audience does not exist, the platform can be accessed directly by a URL. For the architecture column, "Semi-Flexible" indicates that the platforms allows for reconfiguration, but enforces certain configuration, where as 'Flexible' indicates that the experimenter has several choices over potential architecture. A qualifier '+' means that the platform has enhanced support for this requirement compared to other platforms.

pro but we should expect users recruited through the platform to be biased positively towards that platform. If the study needs a more general population, then the current users of any specific platform might not be representative. These commercial platforms tend to support some editing of worlds, but these are not comprehensive in the same way that game toolkits or editors are. The specific assets that can be used might be restricted and the opportunities to write scripts and control behaviour of objects might be limited. These restrictions and limitations are partly to maintain the security of the platform and ensure a good experience for all users. For commercial platforms key issues are that the architecture of the system is closed, so certain features, such as logging of data or control

over lobby system are not possible. In [Table 3](#) we highlight four popular systems, Rec Room (Rec Room, 2022), VRChat (VRChat Inc., 2022), AltSpaceVR (Microsoft, 2022) and Spatial (Spatial Systems Inc. 2022). There is not space in this article to cover their full features and potential for supporting experiments. However, we note that of these, VRChat does support a range of avatar customisation that could support R8. Spatial supports web browsers and certain AR devices directly, so has broader support than the other platforms. We refer the reader to other surveys that could identify specific features that are important to particular experiments [Kolesnichenko et al. \(2019\)](#); [Jonas et al. \(2019\)](#); [Tanenbaum et al. \(2020\)](#); [Liu and Steed \(2021\)](#).

Next we identify WebXR and Mozilla Hubs as technologies that support distribution via web browsers. WebXR is a relatively low-level technology, so in [Table 3](#) we indicate that certain features can be developed and that the platform has the necessary APIs. We noted in [Section 2.1](#) that toolkits are emerging that support more complex social demonstrations. Mozilla Hubs is a complete social VR system, with front-end client and back-end server support. While scenes can be built and run on existing services provided by Mozilla and others, because the front-end and back-end are both open source they can be deployed by an experimenter in a way that ensures that data capture can be secure and compliant with data protection requirements. While WebXR itself does not directly provide any of the features we identify for remote and distributed experimenters, it should be clear that they could be implemented, as Mozilla Hubs demonstrates several of them. The downside of using Mozilla Hubs itself is that it is relatively complex and while there is a scene editor, the platform does not support the general flexibility of scene creation that popular game engines do. Thus together, WebXR and Mozilla Hubs indicate that there is the potential to build a platform that supports our requirements, but that this has not emerged so far. For work that needs novel devices (e.g. external biosensors), needs access to more control over rendering (e.g. for foveated rendering) or needs access to multiple processors, a platform such as Unity or Unreal is an advantage.

Finally, we identify Unity and Unreal as platforms that could be used to develop systems fulfilling all the requirements. The key difference is that Unreal has a networking model built in, while Unity offers a few options for building networking, including commercial solutions such as Photon. The paper introducing Ubiq goes into a deeper analysis of such platforms [Friston et al. \(2021\)](#).

As regards Ubiq-Exp we note that being based on Unity, it inherits the basic editing capabilities of that platform. Ubiq does impose specific choices on how the networking works, but does not constrain how servers and peers are connected and deployed.

4 Ubiq-exp design and implementation

4.1 Ubiq system

We use the Ubiq system which is an open-source toolkit for building social mixed-reality systems [Friston et al. \(2021\)](#). It is designed to fulfil requirements that are often ignored in commercial platforms, either because they do not align with, or in some cases actively conflict with, the goals of these platforms. These include characteristics relevant to distributed experiments. For example, Ubiq is fully open source. Users can easily create their own deployments, and have full control over the data flow for data protection considerations or to ensure a

certain level of quality of network service. Ubiq is designed to work primarily with Unity, making available all of Unity's functionality for building experiments.

Ubiq is built around a messaging system and a core set of services, and comes with a small number of complete examples. Ubiq includes support for session and room management, avatars, voice chat, and object spawning. It includes a fully functional social VR application example. Ubiq follows Unity's Component-based programming model, making it familiar to Unity users who may not have networking experience. Users can build their own networked objects by implementing an interface that allows them to exchange messages between instances of a Unity Component class. All Ubiq services are implemented separately. Users can begin to build experiments by modifying the main sample scene. They can add and remove support for different services by adding and removing Prefabs and Components from the scene.

Communication is based on a logical multi-cast model: when an object transmits a message, it is received by every other object in the network that shares the object's identity. This happens at the messaging level, so objects do not need to know explicitly about other objects. For rendezvous and relay over the network, a lightweight server is included. This server includes the concept of rooms allowing different groups of peers to join their own sessions. A room is effectively a list of peers that can interact. When peers are in a group together, the server will forward all messages between them. Where connections will only ever be uni-cast, for example voice channels, direct peer-to-peer connections are made.

Another feature of Ubiq that is particularly useful for developers is that the system is fully operational in a desktop environment. This means that all the functionalities of Ubiq or Ubiq-Exp (e.g. questionnaires, see [Section 4.2.6](#)) are available in the editor or a desktop client. Ubiq supports a broad range of consumer VR hardware. Recently we added support for various AR devices, including HoloLens and Magic Leap.

Ubiq itself meets requirements R1 and R2.

4.2 Features

Ubiq-Exp extends Ubiq APIs, adds new example scenes, and implements a new network service. The following features are included in Ubiq-Exp: distributed logging; record and replay; scene and user management; lobby system; avatars; and questionnaires. These satisfy requirements R3 through R8.

4.2.1 Distributed logging

Ubiq-Exp includes an event logging subsystem. To log events, users create objects (emitters) that have familiar methods, e.g. `Log()`, to record events. Emitters are lightweight and can be created per-Component, avoiding the need to maintain references between objects. One peer in the group

has a LogCollector object. This receives all events, from all emitters, across all peers, and is responsible for writing them to disk or third-party service. The logging system automatically serialises arbitrary arguments and outputs structured logs (JSON), allowing easy ingestion by platforms such as Python/Pandas or MATLAB. Logs can be tagged with a type to distinguish between, e.g. diagnostics, status reports or experimental data. Records with different tags are written to different files.

One use case is for an experimenter to have a special build of an application with a LogCollector, or add the LogCollector directly to a running Unity Editor instance, allowing them to receive all the logs for an experiment they are monitoring as it happens. We call this local logging. Alternatively, Ubiq-Exp also includes a LogCollector-as-a-Service sample to support remote logging. This is a standalone service implemented in Node.js that connects to a Ubiq server. It automatically collects all the logs for a specific room. This is useful for the asynchronous experiment modes.

The distributed logging has been integrated into the core Ubiq system so there are no additional installation steps beyond including the Ubiq.Logging namespace. Users should add the LogCollector Component to at least one scene that will be a member of the experiment room. The social sample scene contains a LogCollector which can be activated via a button in the Editor, or programmatically. So long as one Peer contains an activated LogCollector users can create and use LogEmitter instances anywhere in their code. Subclasses of LogEmitter that write different types of logs are provided as a convenience.

4.2.2 Record and replay

The record and replay feature intercepts and replays incoming and outgoing Ubiq network messages. This means everything that is networked can also be recorded.

Messages are captured and buffered in the NetworkScene, the interface in Ubiq between Components and underlying Peer connections. Components register with a NetworkScene and receive a NetworkContext object which holds an Object and Component Id. When messages are sent over the network, the NetworkScene makes sure that they are received by the correct Component [Friston et al. \(2021\)](#).

A message is a byte array that contains an Object and Component Id followed by the original networked message. The Recorder gathers all messages that are received in one frame in a MessagePack which stores additional information such as pack size and individual message sizes and writes this data to a binary file.

To replay the recording, the information from the metadata file is loaded to create the recorded objects from a list of available prefabs using Ubiq's NetworkSpawner, which persistently spawns and destroys objects across all clients. Object Ids in recorded messages are manipulated to match the newly created target's Object Id. This ensures Components from the

newly created objects receive the recorded messages. Then, all the previously recorded messages can be sent to the new object's designated Components. Thus, from a networking perspective, a replayed object behaves no differently from any other networked object in the scene.

Users can record in a Room and immediately replay the previous recording using the Unity editor or while in VR using an in-world record and replay UI. A replay can also be selected from a list of previous replays that are automatically saved to Unity's persistent data directory in a 'Recordings' folder. Replays can be paused and resumed, and it is possible to jump to different frames in the replay. To distinguish replayed objects from normal objects in the scene, the replayed objects can be rendered with coloured outlines.

The feature also supports record and replay of WebRTC audio. Whenever a peer joins a new Room, they establish a peer-to-peer connection and the recorder intercepts and saves the sent audio information to a separate binary file. During replay, the audio data is loaded into a Unity AudioSource Component that is added to each replayed avatar and plays back the audio.

The record and replay feature is not part of the core Ubiq system but has been developed in a fork of the main Ubiq repository. It is available as a code sample. The easiest way to use the record and replay feature in custom projects is to build on the existing code sample as the feature extends some of Ubiq's core components. This is to allow the recording and replaying of various edge cases: Messages to and from the RoomClient are intercepted to make it possible to record and replay users joining and leaving rooms. It is also important to provide consistent replays over the network for users that join a room with an ongoing replay. Ubiq's NetworkSpawner has been extended to not only spawn but also un-spawn networked objects to simulate joining and leaving of recorded avatars during replays.

4.2.3 Scene management

Ubiq-Exp provides tools to manage scenes and users. The experimenter can use these tools to ease the management of multi-participant studies. They are accessible in the Unity Editor window. Customizable interfaces can be implemented with Unity custom editors and can use OnSceneGUI widgets to control the features. In the third demonstration experiment (see [Section 7](#)), a scene management feature is used to load different scenes (intro, task, and outro scenes) and to quit the experiment. The user management features are used to control the audibility and visibility of all connected users. The audibility feature looks for each AudioSource object in the active scene and changes its volume to 0 or 1 to mute/unmute users. The visibility feature looks for each Avatar object in the active scene and changes both its and its children objects' layers to hide/unhide users. These tools are easily extendable to provide other control and management functionality, as we expose hints about which participants have different capabilities or roles.

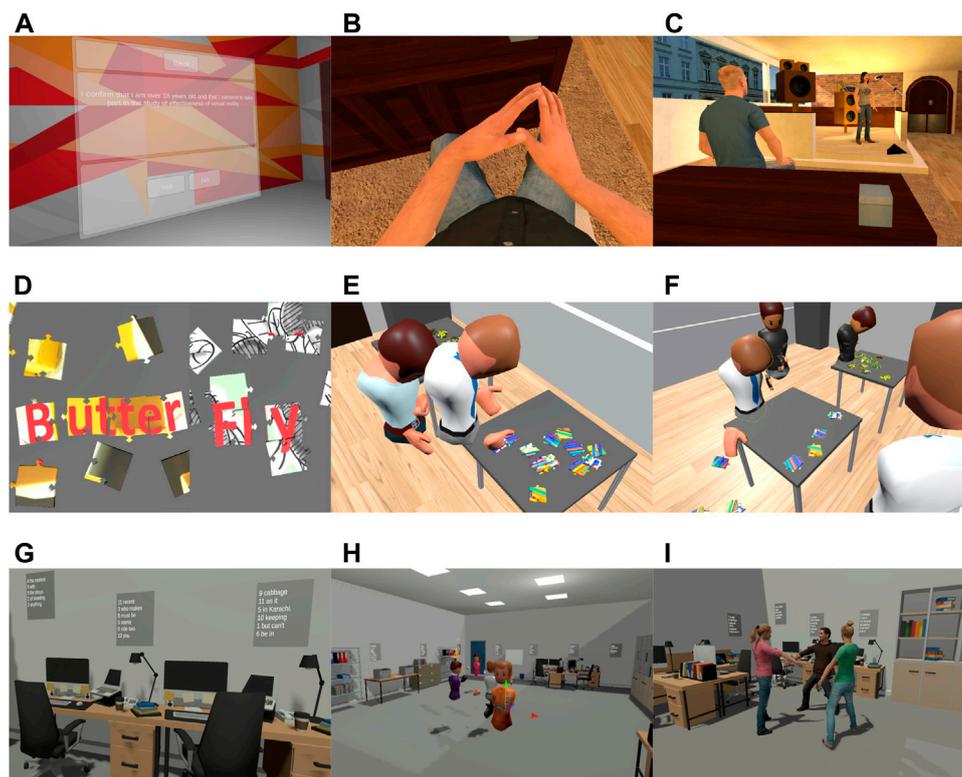


FIGURE 1

Images from the three experiments and environments. **(A)** One of the questions in the pre-questionnaire for the singer in the bar experiment. **(B)** View of the male first-person sat avatar with tracked hands. **(C)** View of the singer. **(D)** Two example jigsaw puzzles that reveal the compound word 'butterfly'. **(E)** One participant is using his body to shield the puzzle. **(F)** Two participants are watching their replays (avatars with yellow outlines) in VR. **(G)** Example of the word-puzzle posters. **(H)** View of the participants' and experimenter's avatars. **(I)** A view of the same scenario, but using the RocketBox avatars.

This feature is available as an example script that can be adapted to fit the experimental protocol and control mechanisms that are desired.

4.2.4 Lobby system

In Ubiq, peers join others in rooms through shared secrets. These are user friendly join codes that are dynamically created or static GUIDs that are specified at design time. Usually, a player creates a room and communicates the new room join code out of band. However, for multi-participant asynchronous experiments, with participants who do not already know each other, it is necessary for the system to place participants together. This can be done by Ubiq-Exp's lobby system. An experimenter can use the lobby system to support asynchronous experiments where they themselves do not attend in person. For example, they could schedule an experiment slot and invite any number of participants to join, where participants would be grouped up and moved to the new room as soon as a group is complete. We bundle an example of this with Ubiq-Exp as a code sample. The design of Ubiq allows the lobby system to be implemented

without server changes. Users need to add the Lobby Service script to their scene. When enough users are in a room, the users are invited to move to a new room together, emptying the old room and freeing space for new users to join.

Neither of our multi-user demonstrations in this paper used the lobby server, as the experiments were pilots and the respective experimenters needed to be online to manage the sessions. However, as noted in [Section 5](#), the lobby system would have been useful even in the single person experiment, as two individuals did run the experiment in over-lapping time periods and thus the logging would have been more straightforward if they had joined different rooms.

4.2.5 Avatars

Ubiq already provides cartoony avatars, see [Figures 1E,F,H](#). Ubiq-Exp provides examples on how to use the Microsoft RocketBox avatars [Gonzalez-Franco et al. \(2020\)](#). These have been rigged using the commercial FinalIK plugin. This plugin is thus not included within our repository. Online, we provide instructions on how to install the plugin and run the examples.

Once the plugin is installed, Ubiq-Exp provides a Unity prefab that can be imported into a scene. From a developer point of view, this is a simple variant of the built-in Ubiq avatars. At run time the prefab is spawned for each user. The head and hand positions are automatically networked. The animation with FinalIK is calculated on the receiving side, but the computational cost is negligible.

The first example of use of this feature is in our first demonstration experiment, see [Section 5](#). The participant should be seated for this experiment. On scene load, the user is centred over the avatar's root bone. FinalIK is used to animate the upper body, bending the spine to match the current head position, and arms to match hand trackers positions. The legs are static. This example also demonstrates how to animate RocketBox avatars with motion capture data.

The second example of this feature is in the environment of the third demonstration experiment, see [Section 7](#). The pilot experiment described in [Section 7](#) only uses the cartoony avatars from Ubiq, but the experiment implementation was extended to support the RocketBox avatars to enable a future experiment, see [Figure 11](#). To enable the Rocketbox avatars we use VRIK, the full body solver from the plugin FinalIK. The camera and hand controllers are calibrated to fit the avatar head and hands. The plugin has a procedural locomotion technique to animate the feet. We extended the built-in Ubiq avatar sharing mechanism to ensure the procedural animation is deterministic at each site.

4.2.6 Questionnaires

Ubiq-Exp provides a demonstration functionality to create immersive questionnaires. It provides the facility to read a questionnaire specification from an XML file and present multiple-choice questions and Likert scales. We include functionality to dynamically select whether to show questions based on previous question answers or experiment conditions. We include optional functionality to allow navigation backwards to change answers. We include functionality to trigger audio files on specific questions. All answers are automatically logged, and if distributed logging is enabled, these can be captured from multiple sites by a log collector. See [Figure 1A](#) for a screenshot of the questionnaire inside the first of our three pilot studies. We provide examples of some common questionnaires, including the SUS presence questionnaire [Slater et al. \(1994\)](#), a questionnaire about co-presence [Steed et al. \(1999\)](#), and an embodiment questionnaire [Peck and Gonzalez-Franco \(2021\)](#).

We provide a Unity prefab that embeds the questionnaire code. This can be instanced as required in the scene. The developer would need to separately edit the XML file specifying the questions, and add that and any audio assets to the Unity project.

4.2.7 Video

Ubiq-Exp does not support video streaming. The main reason for this is that our own ethics processes do not sanction the capture of video of participants remotely unless it is strictly necessary. If it were necessary then using a standard tool such as Zoom or Teams would probably be preferred to ensure security. It would be relatively straightforward to build a Ubiq client that did support video streaming using WebRTC standards, as we already demonstrate how to use audio streaming with WebRTC.

5 Experiment 1: Singer in the bar

The first example experiment is an update of a study previously developed for Samsung Gear VR and Google Cardboard [Steed et al. \(2016\)](#). That demonstration only supported three degrees of freedom (3DOF) head tracking, so menu interaction was done by fixating on buttons. The re-implementation supports 6DOF head and hand tracking. It is an asynchronous single-person study that uses Ubiq-Exp's distributed logging facility to capture log files on a server.

5.1 Scenario

The scenario is the same as described by the study of [Steed et al. \(2016\)](#). The participant watches a singer performing in a bar, see [Figure 1C](#). They are seated behind a table on which there is a box. On the other side of the table there is a male spectator. This spectator shuffles his chair repeatedly, and on the third shuffle, the box falls onto the virtual knee of the participant, or where their knee would be for conditions where there is not an avatar. The bar scene lasts 170 s.

There are three binary factors that produce eight conditions: whether or not the participant had a self-avatar (Body); whether or not the singer faced the participant or sang towards an empty part of the room (LookAt); and whether or not the singer asked the participant to tap along to the beat (Induction).

5.2 Implementation

The application was developed for Oculus Quest 1 and 2. It was implemented as a sequence of five scenes, one loading screen, one informed consent and pre-questionnaire scene, the bar scene itself, a post-questionnaire scene and a results scene. The Ubiq network scene is enabled initially in the second scene if the user allows data collection to ensure that the Ubiq server can be contacted. If it cannot then data collection is disabled. The application connects to a static Ubiq Room, but does not

implement the multi-user functionality (these are logically separate in Ubiq, see [Friston et al. \(2021\)](#)).

In the earlier version we had used the RocketBox avatars under a commercial license. Since these have now been made open source, we replaced the four avatars in the scenes with the open source versions. Two of the avatars in the scene are driven by animation sequences. For the self-avatar, we used one male and one female version of the rigged avatars in the Ubiq-Exp assets see [Section 4.2.5](#). We disabled any leg animation and posed the avatars in a sitting position. We then added an inverse-kinematics chain to move the spine. This example would be useful for anyone wanting to support sitting avatars.

We also used the Ubiq logging system to capture data. We set up a logging service on the Ubiq server machine to record all logged events. Once data collection is enabled, the application logs participant head and hand movements at 1 Hz. It also logs the questionnaire answers and condition flags, see below.

Most of the assets for this experiment are available open source. Please contact the authors or see our website for instructions on how to install or license three assets that are commercial (FinalIK, some geometry assets for the bar scene, and a license for the song recorded).

5.3 Measures and hypotheses

The participant completes a short pre-questionnaire before they visit the bar scene. They then complete a longer post-questionnaire after the bar scene. The same pre- and post-questionnaires as [Steed et al. \(2016\)](#) are used. The pre-questionnaire concerns game playing and VR experience. The post-questionnaire has a selection of questions about presence (Q1-Q4), reaction to the box falling (Q5-Q6), feeling as if the hand disappeared (Q7) and embodiment (Q9-Q10). The other questions are control questions. We hypothesise that: the Body factor will lead to higher ratings of body ownership; the Induction factor will lead to higher ratings of body ownership; the LookAt factor will lead to higher ratings of self-reported presence. These questionnaires are implemented using the system described in [Section 4.2.6](#).

5.4 Methods

A short advert was posted on various social media and also on [xrdrn.org](#). These participants were directed to a web page with participant information and instructions to install the application via the SideQuest tool. Participants were thus unsupervised. Additionally, we invited students at UCL to take part in study when they visited the laboratory. Such participants were shown the web page with the participant information. No personally identifying information was kept for in-person visitors. Participants gave consent for data collection in the application

(see [Figure 1A](#)). No information was collected from participants that refused data collection, but they could still experience the scenario. Participants were not compensated.

5.5 Discussion and reflection

This study was a pilot trial that ran for 1 week. Of the 37 persons who started the application, 15 provided complete sets of data, meaning that they completed the final questionnaire. No data sets were excluded because the answers to the questionnaires were not considered accurate (i.e. no participant answered the same to all questions, nor completed the questionnaire in < 20 seconds). The results are discussed further in [Supplemental Material](#) in Section A.1.

The pilot trial was a success from a point of view of Ubiq-Exp. The application was already well tested from the point of view of the self-avatar kinematics. The logging worked as expected with one flaw that has since been rectified. The logging capture system would open a file when a user connected to a specific Room and log all events for the Room in one file. We had one instance where two client applications connected at the same time, causing one file to merge two participants' data. The logging system can now create one log per participant rather than per room.

6 Experiment 2: Jigsaw puzzle task

The second example is based on a study by Pan et al. for HTC Vive where dyads had to solve jigsaw puzzles collaboratively and competitively inside VR with and without self-avatars to measure their impact on trust and collaboration [Pan and Steed \(2017\)](#). The study was run on Oculus Quest 1 and explores previous observations made by Pan et al. during the jigsaw puzzle tasks. It uses Ubiq-Exp's record and replay feature to support qualitative data collection.

6.1 Scenario

The scenario is similar to the one in the previous study [Pan and Steed \(2017\)](#). In groups of two, participants have to solve two jigsaw puzzles that spawn on two tables next to each other in an otherwise plain virtual room. Each puzzle consists of 12 (4×3) puzzle pieces and shows an image and a word that corresponds to each other. The words on both puzzles form a compound word that has to be found by the two participants. An example of two such puzzles can be seen in [Figure 1D](#). Every round starts with the participants standing in front of a table each. Each group has to find four compound words (eight puzzles) in four rounds. In the first two rounds, they have to work together to find the word. In the last two rounds, they are competing and have to figure out the word before the other participant does. They do not need to

finish the puzzles to guess the word. A compound word is considered as found when one of the participants says it out loud. Participants can walk freely in the area defined by the guardian bounding box created by their Quest 1 or can teleport using the controllers.

6.2 Implementation

The previous study by Pan et al. [Pan and Steed \(2017\)](#) used Unity UNET for networking, whereas our re-implementation uses Ubiq and the record and replay code sample. The jigsaw puzzles from the previous study were created by a commercial tool from the Unity Asset store [Muhammad Umair Ehtesham \(2018\)](#) and did not have a 3D shape. For the re-implementation an open-source script for Maya is used to create the 3D puzzle pieces [Puzzle Maker \(2013\)](#). This example uses Ubiq's floating body avatars (see [Figure 1E](#)). The scene consists of an empty room with two tables on which the puzzle pieces are spawned for each participant. Ubiq's floating body avatars are included in recordings and replays by default as they are already in Ubiq's prefab catalogue from which the NetworkSpawner spawns them during replays. Because we want to record the participants while solving the jigsaw puzzles, the puzzle piece prefabs have to be added to the prefab catalogue too. The puzzle pieces are implemented as networked objects and as such their messages are automatically recorded by the record and replay feature. Later, they are replayed by spawning them using the stored prefabs in the catalogue. The experimenter is present during the tasks with an invisible avatar on the desktop to monitor the participants in the Unity editor. A custom Unity inspector is used to spawn and un-spawn puzzles, to record the participants during each trial and show them their replays afterwards.

All of the assets for this experiment are available open source from the Ubiq website.

6.3 Measures and hypotheses

In their study, Pan et al. [Pan and Steed \(2017\)](#) observed post-hoc that participants had been hiding information on puzzle pieces from each other by using their avatar bodies to obstruct the view. Based on this observation, we hypothesise that in our re-implementation, people will also use their body as a shield to hide information and we ask whether it is a behaviour that people are aware of. In our experiment, after the four puzzle rounds, participants watch a replay of their collaborative and competitive puzzle tasks together in VR. While watching, they are encouraged to comment on their replayed actions, behaviours and potential strategies. Following the watching of the replay, each group is interviewed about the replay experience. They are asked the following two questions: "How did it feel seeing yourself?" and "Did you recognise yourself in your behaviours?".

The sharing and reliving of recorded experiences in VR has been shown to be more engaging and informative than other methods [Wang et al. \(2020\)](#) as it allows participants to move around freely and discover new things they might have overlooked previously. Motivated by this, the participants' movements throughout the study were recorded using Ubiq's record and replay feature to gather behavioural data and to replay it in front of the participants in VR to encourage discussion.

6.4 Methods

The participants were using an Oculus Quest 1 to perform the puzzle tasks in VR. They were co-located and could talk to each other freely. Once both in the virtual room, the participants were shown how to teleport around and interact with the puzzle pieces by showing them two example puzzles. The collaborative task was then explained to them. The experimenter started the recording of the collaborative rounds and spawned the first two puzzles. When a participant said the correct compound word out loud, the puzzles were unspawned and the next set of puzzles spawned. This was repeated for a total of two collaborative rounds. The same procedure was followed for the two competitive rounds. After all four rounds, participants were shown the replays of each round and prompted to discuss the outcome. Questions were on each participant's actions, thoughts, and any other observations they had made about themselves or the other participant. Afterwards, the participants could remove their headsets, and short interviews were conducted to question the participants about their experience with the record and replay feature.

Participants took between 30 s and 2 minutes to find a compound word. The total time with each group was between 15 and 20 min.

Throughout the puzzle rounds, the experimenter was present in the virtual room via the Unity editor with an invisible avatar to observe the tasks. The results are based on the behavioural data from the record and replay and the short post-study interviews. As a backup, the whole study was video recorded from the experimenter's view in the Unity editor. For the interview, audio was recorded via a PS3 Eye microphone and a smartphone.

This was a pilot study running for 1 week at UCL with 20 participants. Emails were sent out to students at UCL to participate in this study. Participants were added to a lottery to win an Amazon voucher with a value of £20.

6.5 Discussion and reflection

The results are discussed further in [Supplemental Material](#) in Section B.1.

The pilot study was used to test the functionality and practicability of the record and replay feature for user studies.

As a backup, the whole study was video recorded from the experimenter's view in the Unity editor. However, the backup was not necessary as the recording worked flawlessly. When the study was run, audio recording and replay was still under development; instead audio was recorded via a PS3 Eye microphone and a smartphone. Replaying the puzzle tasks to the participants also worked as expected. We believe that the record and replay feature is a practical tool that encourages participants to talk in more detail about their own and their partner's actions, to point out specific actions to participants, and to gain a better understanding of why participants are acting in certain ways. A follow-up study on the effect of watching one's own replay with regards to reflections about previous actions would be of interest.

7 Experiment 3: Poster task

The third example is based on a study by [Steed et al. \(1999\)](#) that investigated what happens when a small group of participants meet to carry out a joint poster task in a virtual environment. The original study involved three participants simultaneously: two participants used a desktop system while one participant used an immersive system employing a two tracker Polhemus Fastrak, Virtual Research VR4 helmet and a 3D mouse with five buttons. In the re-implementation one participant uses a desktop system with a 1280 × 1024 screen, a 2D mouse with three buttons and five buttons from a keyboard. Two participants use an Oculus Quest 2.

7.1 Scenario

The scenario is similar to the initial study [Steed et al. \(1999\)](#). In groups of three, participants collaborate on solving a word-puzzle task in a virtual environment filled with office furniture from a commercial asset [Office Environment \(2017\)](#). The virtual environment has 15 posters arranged around the room. On each of the posters there is a set of words, prefixed by a number. The goal for the participants is to find and rearrange all the words with the same number to form the completed sentences. There are 11 sentences to solve. Participants are given 15 min to complete as many sentences as they can. A sentence is considered as found when one of the participants says it out loud correctly.

In the initial study a group of strangers was recruited and asked to perform a joint poster word-puzzle task twice: first, in the virtual environment, and then in the real environment from which the virtual one had been modelled. In the re-implementation we recruited groups of three acquaintances. The environment is big enough, and the sentences are complicated enough for the participants to need to collaborate to solve the task efficiently. Indeed, it would be difficult for one participant alone to remember all the words in one sentence. An example of the word puzzle posters can be seen in [Figure 1G](#).

7.2 Implementation

The application was developed for Oculus Quest 2 and desktop use. It was implemented as a sequence of two scenes. In the first scene, participants enter a waiting room where instructions are displayed. At that point they cannot see or hear each other. The experimenter is in the same environment, can see and hear all participants, and monitor the experiment from the Unity Editor window. The experimenter uses scene management an user feature management control with the OnSceneGUI widgets. When it is confirmed that all participants have connected to the Ubiq room and that they understand the instructions, the experimenter loads the second scene and enables the visibility and audibility parameters for all participants. In the second scene participants collaborate on solving the poster word-puzzle task.

The OnSceneGui widgets allow the experimenter to Mute and Hide all participants (chosen by default when the experiment starts). They also allow the experimenter to load the study scene, automatically unmuting and un hiding all participants. Finally the widgets allow the experimenter to end the application for all participants.

We also use the Ubiq logging system to capture data. We collect the distance travelled by each participant and the distance between participants by logging participant head positions. We also log which participant gives the answer to the word-puzzle task. Participants are asked to shout out the answers by giving the number of the sentence and completing it with the words they found. When one answer is given, the experimenter logs which participant answered, based on the color of the avatar. For this, the experimenter uses the OnSceneGUI widgets.

This example uses Ubiq's floating body avatars. The participants all embody white-skinned avatars, with clothes coloured red, orange and purple (see [Figure 1H](#)). The experimenter is present during the tasks with an invisible white-skinned avatar with white clothes. In a later version of this example we use Rocketbox avatars. Rocketbox avatars use FinalIK commercial asset for full body animation.

Most of the assets for this experiment are available open source from the Ubiq website. Please contact the author or see our provide URL for instructions on how to install or license the assets that are commercial (some office furniture assets, FinalIK for the use of Rocketbox avatar).

7.3 Measures and hypotheses

The main measure is the responses to a post-study questionnaire, which is reproduced in [Supplemental Material](#). Participants were asked to self-report presence, co-presence, embodiment, and enjoyment of the task. They were also asked to assess their own degree of talkativeness and whether they considered themselves as the leader. We recorded which

participant gave the answer to the puzzle task and the task performance of each group.

In the original study it was suggested that the immersed participant tended to emerge as a leader in the virtual group Steed et al. (1999). Therefore, based on the initial study, we hypothesise that immersed participants will be the leaders of the task. One would be considered as leader of the task if self-rating of talkativeness and leadership were higher than other group members.

As this is a pilot study, we are also interested in observations of group behaviour around this task.

7.4 Methods

On arrival at the experimental lab, the three participants were given a consent form, information sheet, task, and navigation instructions. They completed a pre-study questionnaire with demographic and social anxiety questions. These questionnaires are reproduced in [Supplemental Material](#). When the participants entered the virtual environment, they were given instructions again in the virtual environment and started the task if they had no other questions. Participants were asked to collaborate on solving as many word-puzzles as they could, in the order of their choice.

Immersed participants could walk freely in the area defined by the guardian bounding box created by the Oculus Quest 2, by using the joystick or the teleportation button of the controllers. Non-immersed participants could navigate the environment using W-A-S-D keys and the mouse on desktop. After completing the task, participants completed a post-study questionnaire, reproduced in [Supplemental Material](#). Participants were co-located and could talk freely to each other. Participants were given 15 min to do the experiment. On average groups of participants solved seven to eight word-puzzles out of 11.

This pilot study ran for 1 week at CYENS Centre of Excellence, Nicosia, Cyprus. Emails were sent to employees of the research center to participate in the study. A total of nine groups of three participants were recruited. Participants were not compensated.

7.5 Discussion and reflection

The pilot study was used to reproduce a past study and to implement Ubiq-Exp features to run and manage multi-user experiments. We used the scene management features, logging system and as a backup video recorded the study in the Unity editor from a top view of the environment. It was challenging to schedule the study with groups of three participants and to manage it in the lab. The experimenter first creates a Ubiq room from a control computer. Next they set up each

participant, by setting the correct avatar and by joining the created Ubiq room from the participants' devices. Then the experimenter can check the control computer and make sure all are connected through Ubiq. Participants were each put in different parts of the environment where they could not see each other, and the audio connections were muted. Then the participants were unmuted and their position transformed into a common room. This worked well in that participants were not witness to the process of setting up. The experimenter did have a lot of work to do as they had to manually choose scene, participants' avatars and enter participants' answers in the Unity editor window on the control computer. Future work should look more at automating these tasks. For example, performing training tasks, confirming participants are ready, procedurally setting the scenes with the right parameters and delegation of experiment control to timers where possible. We would also suggest that in-app control panels for the experimenter could be extended to visualise and evaluate participants' live data, such as using heat maps for location and gaze and to get real-time transcripts of speech.

The results are discussed further in [Supplemental Material](#) in Section C.1.

8 Conclusion

We presented Ubiq-Exp, a set of tools and examples that extends the Ubiq system to support distributed and remote MR experiments. Ubiq-Exp features distributed logging, record and replay, scene and user management, lobby system, avatars, and questionnaires. We demonstrated Ubiq-Exp with three pilot experiments that reproduce prior studies. The role of these experiments is mostly proof of concept of Ubiq-Exp's features and its ability to support remote and distributed experiments. The experiments have mixed scientific results, but each experiment has helped us refine the tools to enable future larger scale experiments.

The Ubiq-Exp features that are specific for asynchronous experiments are now robust: as noted asynchronous experiments are harder to develop as the applications need to be self-contained as the experimenter cannot intervene or shepherd the participant's actions. This paper does not explore an asynchronous multi-participant experiment. Running an experiment in this mode is a target for the near-term future work. For synchronous experiments, although we demonstrate some tools to control participants, we feel that this is an area where there is much more to do in order to enable experimenters to behave within the virtual environment as they might in a face-to-face experiment. For example, instructing participants differently, controlling their interfaces indirectly to provide assistance or guidance, and stage-managing the scene in more complex ways.

For all experiment types, while logging, and record and replay are now enabled, future work will develop more in-scene tools to generate useful statistics (e.g. interpreting eye or head gaze), and also external tools to analyse the recorded data. As noted we do provide basic Python/Pandas and MATLAB scripts to interpret logfiles. We believe that in the future WebXR will be an important method for delivering remote user studies as it avoids the need for a participant to install or manually download applications. Ubiq-Exp currently has a WebXR export demo with all core features (messaging, VOIP, avatars). The sample is available publicly but WebXR support is not yet core. There are various requirements for WebXR development that go against best practice in other environments (e.g., avoiding multi-threading) so we must carefully consider how best to integrate this into the main branch.

Finally, Ubiq-Exp will be available as open source at <https://ubiq.online/publications/ubiq-exp/>. All non-commercial assets for the three experiments will be made available. For those assets that are licensed (e.g. FinalIK for avatar animation, but also some graphics and sound assets), instructions to acquire the relevant assets are online so that the experiments can be reproduced in their entirety. This will enable experimenters to directly reproduce the three studies, and we hope that this provides an interesting basis to develop and share novel experiments.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Ethics statement

The studies involving human participants were reviewed and approved by the UCL Research Ethics Committee. The patients/participants provided their written informed consent to participate in this study.

Author contributions

AS coordinated the project effort, the paper writing and experiment 1. KB was responsible for experiment 2. LI was responsible for experiment 3. BC and SF contributed to software development. DS assisted with experiments. NN assisted with experiment 2. All authors contributed to the writing.

Funding

This work was partly funded by United Kingdom EPSRC project Graphics Pipelines for Next Generation Mixed Reality Systems (grant reference EP/T01346X/1), EU H2020 project RISE (grant number 739578) and EU H2020 project CLIPE (grant number 860768).

Conflict of interest

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

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frvir.2022.912078/full#supplementary-material>

SUPPLEMENTARY FIGURE S1

Interaction effect on Q4.

SUPPLEMENTARY FIGURE S2

Interaction effect on Q5.

SUPPLEMENTARY TABLE S1

Observed collaborative and competitive strategies per group.

SUPPLEMENTARY TABLE S2

Observed behaviours per group during relay.

SUPPLEMENTARY TABLE S3

Numbers of answers per group and per participants.

SUPPLEMENTARY TABLE S4

Questionnaire response (scaled out of 10) based on level of immersion.

SUPPLEMENTARY TABLE S5

Pre-questionnaire.

SUPPLEMENTARY TABLE S6

Full post-questionnaire. Subjects without a body only answered Q1 to Q8.

References

- Alexandrovsky, D., Putze, S., Bonfert, M., Höffner, S., Michelmann, P., Wenig, D., et al. (2020). "Examining design choices of questionnaires in VR user studies," in CHI '20: Proceedings of the 2020 CHI conference on human factors in computing systems (New York, NY, USA: Association for Computing Machinery), 1–21. doi:10.1145/3313831.3376260
- Bebko, A. O., and Troje, N. F. (2020). bmlTUX: Design and control of experiments in virtual reality and beyond. *i-Perception* 11, 2041669520938400. doi:10.1177/2041669520938400
- Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., and Cruz-Neira, C. (2001). "VR juggler: A virtual platform for virtual reality application development," in Proceedings IEEE virtual reality (IEEE), 89–96. doi:10.1109/VR.2001.913774
- Biocca, F., Harms, C., and Burgoon, J. K. (2003). Toward a more robust theory and measure of social presence: Review and suggested criteria. *Presence. (Camb)* 12, 456–480. doi:10.1162/105474603322761270
- Blanchard, C., Burgess, S., Harvill, Y., Lanier, J., Lasko, A., Oberman, M., et al. (1990). "Reality built for two: A virtual reality tool," in I3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics (New York, NY, USA: Association for Computing Machinery), 35–36.
- Brookes, J., Warburton, M., Alghadier, M., Mon-Williams, M., and Mushtaq, F. (2020). Studying human behavior with virtual reality: The Unity Experiment Framework. *Behav. Res. Methods* 52, 455–463. doi:10.3758/s13428-019-01242-0
- Büschel, W., Lehmann, A., and Dachselt, R. (2021). "Miria: A mixed reality toolkit for the *in-situ* visualization and analysis of spatio-temporal interaction data," in Proceedings of the 2021 CHI conference on human factors in computing systems (New York, NY, USA: Association for Computing Machinery).
- Carlsson, C., and Hagsand, O. (1993). "DIVE A multi-user virtual reality system," in VRAIS '93: Proceedings of IEEE virtual reality annual international symposium (IEEE), 394–400. doi:10.1109/VRAIS.1993.380753
- Churchill, E. F., and Snowdon, D. (1998). Collaborative virtual environments: An introductory review of issues and systems. *Virtual Real.* 3, 3–15. doi:10.1007/bf01409793
- Damer, B. (1997). *Avatars! exploring and building virtual worlds on the internet*. Berkeley: Peachpit Press.
- Dubosc, C., Gorisse, G., Christmann, O., Fleury, S., Poinot, K., and Richir, S. (2021). "Impact of avatar anthropomorphism and task type on social presence in immersive collaborative virtual environments," in 2021 IEEE conference on virtual reality and 3D user interfaces abstracts and workshops (VRW) (IEEE). San Francisco, CA: Peachpit.438–439. doi:10.1109/VRW52623.2021.00101
- Ellis, S. R. (1991). Nature and origins of virtual environments: A bibliographical essay. *Comput. Syst. Eng.* 2, 321–347. doi:10.1016/0956-0521(91)90001-L
- Fabri, M., Moore, D. J., and Hobbs, D. J. (1999). "The emotional avatar: Non-verbal communication between inhabitants of collaborative virtual environments," in *Gesture-based communication in human-computer interaction*. Editors A. Braffort, R. Gherbi, S. Gibet, D. Teil, and J. Richardson (Springer Berlin Heidelberg), Lecture Notes in Computer Science), 269–273.
- Feick, M., Kleer, N., Tang, A., and Krüger, A. (2020). "The virtual reality questionnaire toolkit," in UIST '20: Adjunct publication of the 33rd annual ACM symposium on user interface software and technology (New York, NY, USA: Association for Computing Machinery), 68–69. doi:10.1145/3379350.3416188
- Freeman, G., Zamanifard, S., Maloney, D., and Adkins, A. (2020). "My body, my avatar: How people perceive their avatars in social virtual reality," in CHI EA '20: Extended abstracts of the 2020 CHI conference on human factors in computing systems (Honolulu, HI, USA: Event-place), 1–8.
- Friedman, D., Brogni, A., Guger, C., Antley, A., Steed, A., and Slater, M. (2006). Sharing and analyzing data from presence experiments. *Presence. (Camb)* 15, 599–610. doi:10.1162/pres.15.5.599
- Friston, S. J., Congdon, B. J., Swapp, D., Izzouzi, L., Brandstätter, K., Archer, D., et al. (2021). "Ubiq: A system to build flexible social virtual reality experiences," in VRST '21: Proceedings of the 27th ACM symposium on virtual reality software and technology (New York, NY, USA: Association for Computing Machinery). doi:10.1145/3489849.3489871
- Gonzalez-Franco, M., Ofek, E., Pan, Y., Antley, A., Steed, A., Spanlang, B., et al. (2020). The Rocketbox library and the utility of freely available rigged avatars. *Front. Virtual Real.* 1. doi:10.3389/frvir.2020.561558
- Greenhalgh, C., and Benford, S. (1995). Massive: A collaborative virtual environment for teleconferencing. *ACM Trans. Comput. Hum. Interact.* 2, 239–261. doi:10.1145/210079.210088
- Greenhalgh, C., Flintham, M., Purbrick, J., and Benford, S. (2002). "Applications of temporal links: Recording and replaying virtual environments," in Proceedings IEEE virtual reality (IEEE), 101–108. doi:10.1109/VR.2002.996512
- Jonas, M., Said, S., Yu, D., Aiello, C., Furlo, N., and Zytka, D. (2019). "Towards a taxonomy of social VR application design," in Extended abstracts of the annual symposium on computer-human interaction in play companion extended abstracts (New York, NY, USA: Association for Computing Machinery), CHI PLAY '19 Extended Abstracts), 437–444.
- Juvrud, J., Gredebäck, G., Åhs, F., Lerin, N., Nyström, P., Kastrati, G., et al. (2018). The immersive virtual reality lab: Possibilities for remote experimental manipulations of autonomic activity on a large scale. *Front. Neurosci.* 12, 305. doi:10.3389/frns.2018.00305
- Kolesnichenko, A., McVeigh-Schultz, J., and Isbister, K. (2019). "Understanding emerging design practices for avatar systems in the commercial social VR ecology," in DIS '19: Proceedings of the 2019 on designing interactive systems conference (New York, NY, USA: Association for Computing Machinery), 241–252.
- Latoschik, M. E., Roth, D., Gall, D., Achenbach, J., Waltemate, T., and Botsch, M. (2017). "The effect of avatar realism in immersive social virtual realities," in VRST '17: Proceedings of the 23rd ACM symposium on virtual reality software and technology (New York, NY, USA: Association for Computing Machinery), 1–10.
- Latoschik, M. E., and Tramberend, H. (2011). "Simulator X: A scalable and concurrent architecture for intelligent realtime interactive systems," in 2011 IEEE virtual reality conference (IEEE), 171–174. doi:10.1109/VR.2011.5759457
- Liu, Q., and Steed, A. (2021). Social virtual reality platform comparison and evaluation using a guided group walkthrough method. *Front. Virtual Real.* 2, 52. doi:10.3389/frvir.2021.668181
- Morozov, M., Gerasimov, A., and Fominykh, M. (2012). "vAcademia – educational virtual world with 3D recording," in 2012 international conference on cyberworlds (IEEE), 199–206. doi:10.1109/CW.2012.35
- Mottelson, A., and Hornbæk, K. (2017). "Virtual reality studies outside the laboratory," in VRST '17: Proceedings of the 23rd ACM symposium on virtual reality software and technology (New York, NY, USA: Association for Computing Machinery), 1–10. doi:10.1145/3139131.3139141
- Moustafa, F., and Steed, A. (2018). "A longitudinal study of small group interaction in social virtual reality," in VRST '18: Proceedings of the 24th ACM symposium on virtual reality software and technology (New York, NY, USA: Event-placeTokyo, Japan: Association for Computing Machinery).
- [Dataset] Muhammad Umair Ehtesham (2018). Puzzle maker. Available at: <https://assetstore.unity.com/packages/tools/game-toolkits/puzzle-maker-19810>.
- Murgia, A., Wolff, R., Steptoe, W., Sharkey, P., Roberts, D., Guimaraes, E., et al. (2008). "A tool for replay and analysis of gaze-enhanced multiparty sessions captured in immersive collaborative environments," in 2008 12th IEEE/ACM international symposium on distributed simulation and real-time applications (IEEE), 252–258. doi:10.1109/DS-RT.2008.25
- [Dataset] Office Environment (2017). 3D models office interiors pack. Available at: <https://assetstore.unity.com/packages/3d/props/interior/3d-models-office-interiors-pack-99853>.
- Oh, C. S., Bailenson, J. N., and Welch, G. F. (2018). A systematic review of social presence: Definition, antecedents, and implications. *Front. Robot. AI* 5, 114. doi:10.3389/frbot.2018.00114
- Pan, Y., and Steed, A. (2017). The impact of self-avatars on trust and collaboration in shared virtual environments. *PLOS ONE* 12, e0189078. doi:10.1371/journal.pone.0189078
- Peck, T. C., and Gonzalez-Franco, M. (2021). Avatar embodiment. A standardized questionnaire. *Front. Virtual Real.* 1. doi:10.3389/frvir.2020.575943
- Ponto, K., Kohlmann, J., and Gleicher, M. (2012). Effective replays and summarization of virtual experiences. *IEEE Trans. Vis. Comput. Graph.* 18, 607–616. doi:10.1109/TVCG.2012.41
- Putze, S., Alexandrovsky, D., Putze, F., Höffner, S., Smeddinck, J. D., and Malaka, R. (2020). "Breaking the experience: Effects of questionnaires in VR user studies," in CHI '20: Proceedings of the 2020 CHI conference on human factors in computing systems (New York, NY, USA: Association for Computing Machinery), 1–15. doi:10.1145/3313831.3376144
- [Dataset] Puzzle Maker (2013). Jigsaw puzzle maker 1.0.0 for Maya. Available at: <https://www.highend3d.com/maya/script/jigsaw-puzzle-maker-for-maya>.
- Radiha, R., Mäkelä, V., Prange, S., Rodriguez, S. D., Piening, R., Zhou, Y., et al. (2021). Remote VR studies: A framework for running virtual reality studies remotely via participant-owned HMDs. *ACM Trans. Comput. Hum. Interact.* 28, 1–46. doi:10.1145/3472617
- Raij, A. B., and Lok, B. C. (2008). "IPSViz: An after-action review tool for human-virtual human experiences," in 2008 IEEE virtual reality conference (IEEE), 91–98. doi:10.1109/VR.2008.4480756

- Ratcliffe, J., Soave, F., Bryan-Kinns, N., Tokarchuk, L., and Farkhatdinov, I. (2021). "Extended reality (XR) remote research: A survey of drawbacks and opportunities," in Proceedings of the 2021 CHI conference on human factors in computing systems (New York, NY, USA: Association for Computing Machinery), 1–13.
- Saffo, D., Di Bartolomeo, S., Yildirim, C., and Dunne, C. (2021). "Remote and collaborative virtual reality experiments via social VR platforms," in Proceedings of the 2021 CHI conference on human factors in computing systems (New York, NY, USA: Association for Computing Machinery), 1–15.
- Schroeder, R. (2010). *Being there together: Social interaction in shared virtual environments*. Oxford University Press.
- [Dataset] Schulz, R. (2021). Comprehensive list of social VR platforms and virtual worlds. Available at: <https://ryanschultz.com/list-of-social-vr-virtual-worlds/>.
- Singhal, S., and Zyda, M. (1999). *Networked virtual environments: Design and implementation*. Reading, MA: Addison-Wesley.
- Slater, M., and Steed, A. (2000). A virtual presence counter. *Presence. (Camb)*. 9, 413–434. doi:10.1162/105474600566925
- Slater, M., Usoh, M., and Steed, A. (1994). Depth of presence in virtual environments. *Presence. (Camb)*. 3, 130–144. doi:10.1162/pres.1994.3.2.130
- [Dataset] Steed, A., Archer, D., Congdon, B., Friston, S., Swapp, D., and Thiel, F. J. (2021). Some lessons learned running virtual reality experiments out of the laboratory. Available at: <https://arxiv.org/abs/2104.05359>.
- Steed, A., Friston, S., Lopez, M. M., Drummond, J., Pan, Y., and Swapp, D. (2016). An 'in the wild' experiment on presence and embodiment using consumer virtual reality equipment. *IEEE Trans. Vis. Comput. Graph.* 22, 1406–1414. doi:10.1109/TVCG.2016.2518135
- Steed, A., and Oliveira, M. F. (2009). *Networked graphics: Building networked games and virtual environments*. Elsevier.
- Steed, A., Ortega, F. R., Williams, A. S., Kruijff, E., Stuerzlinger, W., Batmaz, A. U., et al. (2020). Evaluating immersive experiences during Covid-19 and beyond. *Interactions* 27, 62–67. doi:10.1145/3406098
- Steed, A., Slater, M., Sadagic, A., Bullock, A., and Tromp, J. (1999). "Leadership and collaboration in shared virtual environments," in Proceedings IEEE virtual reality (cat. No. 99CB36316) (IEEE), 112–115.
- Stephoe, W., and Steed, A. (2012). Multimodal data capture and analysis of interaction in immersive collaborative virtual environments. *Presence Teleoperators Virtual Environ.* 21, 388–405. doi:10.1162/PRES_a_00123
- Tanenbaum, T. J., Hartoonian, N., and Bryan, J. (2020). "How do I make this thing smile?": An inventory of expressive nonverbal communication in commercial social virtual reality platforms," in CHI '20: Proceedings of the 2020 CHI conference on human factors in computing systems (New York, NY, USA: Association for Computing Machinery), 1–13.
- Wang, C. Y., Sakashita, M., Ehsan, U., Li, J., and Won, A. S. (2020). "Again, together: Socially reliving virtual reality experiences when separated," in Proceedings of the 2020 CHI conference on human factors in computing systems (Honolulu HI USA: ACM), 1–12. doi:10.1145/3313831.3376642
- Williamson, J., Li, J., Vinayagamoorthy, V., Shamma, D. A., and Cesar, P. (2021). "Proxemics and social interactions in an instrumented virtual reality workshop," in Proceedings of the 2021 CHI conference on human factors in computing systems (New York, NY, USA: Association for Computing Machinery), 1–13.
- Yee, N., Bailenson, J. N., Urbanek, M., Chang, F., and Merget, D. (2007). The unbearable likeness of being digital: The persistence of nonverbal social norms in online virtual environments. *Cyberpsychol. Behav.* 10, 115–121. doi:10.1089/cpb.2006.9984
- Zhao, J., Simpson, M., Sajjadi, P., Wallgrén, J. O., Li, P., Bagher, M. M., et al. (2021). "CrowdXR - pitfalls and potentials of experiments with remote participants," in 2021 IEEE international symposium on mixed and augmented reality (ISMAR) (Bari, Italy: IEEE). doi:10.1109/ISMAR52148.2021.00062