

Teaching Social Virtual Reality with Ubiq

Sebastian Friston

University College London

Ben Congdon

University College London

Anthony Steed

University College London

Abstract—We share our experiences of teaching virtual reality with Ubiq, an open-source system for building social VR. VR as a subject touches on many areas including perception, human computer interaction and psychology. In our VE module we consider all aspects of VR. In recent years networked VR, and in particular social VR, has become increasingly relevant, at the same time as demand for online and hybrid teaching has increased. Commercial Social Virtual Reality (SVR) systems have proliferated, but for a number of reasons this has not resulted in systems any more suitable for research and teaching. As a result we created Ubiq, a system for building social VR applications designed first for research and teaching. In this paper we describe how Ubiq came to be, and our experiences of using it in our Virtual Environments module over the last two years.

■ **VIRTUAL REALITY** touches on a diverse range of areas including software engineering, computer graphics, perception, hardware, human computer interaction and psychology. The goal of a VR interface is to synthesise virtual stimuli, such that users no longer interact with the interface but with the virtual world directly. This requires understanding how the human brain and body perceive the world, and the technology required to synthesise a virtual one.

We have run a module on Virtual Environments at the UCL Department of Computer Science for over 20 years. Our module attempts

to give an insight to the broad range of areas mentioned, with a focus on practical experience of systems. Our syllabus¹ includes ten 1-2 hour lectures on Presence, Displays, Mixed Reality, Social Virtual Reality, Avatars, Networking, Interaction, Tracking, Audio and Haptics. Additionally there is a lot of self-directed work (i.e. responses to open briefs or open essay titles), which allows students to focus on aspects that interest them most or fit their background. The module is offered to undergraduate computer scientists,

¹<https://www.ucl.ac.uk/module-catalogue/modules/virtual-environments-COMP0113>

and a variety of masters programmes, including programmes on graphics, HCI and robotics. We do support a range of students, including doctoral students, electing to take this module from other departments or faculties. A pre-requisite is that the topic is relevant to their area of study, and that they have some programming skills in any language. They do not necessarily need experience of Unity or C#, as we teach these as part of the course.

The main activity and assessment of the module is the construction of a practical VR scenario. The brief changes from year to year and has included developing experiments on interaction, designing immersive optical illusions, designing escape rooms and exploring mixed reality situations. We focus on interaction and the user experience, rather than asset creation. While we have long taught aspects of networking, avatars and social applications, these have gained more significance in the past few years. This trend, and the very direct need to run the module remotely during 2020-2021 because of Covid-19, gave us the impetus to make our practical component ‘networked-first’, with students implementing a networked experience as their primary assignment. This aligned with our growing need for a standardised platform for our research needs.

Despite renewed interest from industry however, in an internal review exercise in Q1-Q3 of 2020 we found that there were no existing commercial SVR platforms that met all the requirements for teaching and research. A teaching platform must not only be transparent enough to itself serve as teaching material, but flexible enough to allow students to demonstrate their skills, meet data protection and ethical requirements, and complement how the module is run. After evaluating various platforms, we came to the conclusion that the reasons these requirements remain unfulfilled is not a lack of interest, but because they directly conflict with various commercial goals. As a solution we created Ubiq: a system to build social virtual reality experiences. Ubiq is a software framework that allows users to quickly build social, networked experiences in a popular game engine (Unity). In this respect Ubiq is not unique. But Ubiq is distinct in that it is the only modern system designed first for research and teaching. In this article we discuss

our experiences of teaching VR and SVR. We describe the non-obvious challenges that led us to develop Ubiq, how it worked in our course, and where we intend to take it in the future.

SOCIAL VR IN RESEARCH AND TEACHING

Recent years have seen a renewed interest in social VR. A number of organisations such as Microsoft (AltSpace) [2], Mozilla (Hubs) [3] and Meta (Horizons) [4] have released social VR platforms. These creator-consumer platforms are proprietary applications which encompass the entire experience. There are also a number of networking libraries such as Photon [6] or UNET [7] that implement networking capabilities for common online game archetypes. These integrate with game engines, such as Unity. Tools such as Unity are now very popular with VR developers and students due to their accessibility. A number of academic efforts are exploring social VR for education (e.g. [8], [9], [10]). These typically focus on creating an effective platform for teaching *in* however, rather than an effective teaching *resource*. While there is considerable work on using VR in education ([11], [12], [13]), only a small subset of these address VR or SVR as the subject (e.g. [14], [15]).

Before creating Ubiq we evaluated the suitability of a number of platforms for our module. Many platforms were easy to use and effective at SVR, but all fell short in at least one way for research or teaching. We cover all of the requirements for Ubiq in [1]. For teaching though, we highlight the following requirements:

Remote Code Execution To properly assess students, they need enough freedom in the design space, which includes not just the inhabited virtual space but the interactions with it, and its behaviour. Most commercial platforms such as RecRoom [5] or AltSpace allow users to re-skin or re-combine objects with given behaviours, but not create new ones.

Data Protection Experiments are subject to strict ethical constraints around data protection, and there are similar issues around teaching as well. Our local rules about software provision require us to ensure any third party services are GDPR compliant. Therefore, any platform would not be able to require any hardware or service

that requires a specific login.

Transparency & Accessibility It is important VR students understand how SVRs are built. Commercial systems make decisions that might hide much of this behaviour.

Open Source Our students own the code they develop, so we wanted this to be useful to them for their portfolio or to use in further work. We wanted to be able to allow students to open source their code and make it openly available.

UBIQ

Many of the above requirements are not just unimportant for commercial uses, but directly conflict with their business goals. Consequently, we became confident that no suitable commercial system would be forthcoming, and so we created Ubiq. Ubiq is a framework for building social VR, for the purposes of research and teaching. What makes Ubiq novel is not its SVR capabilities, but rather how these are implemented to complement teaching and adhere to the above requirements.

We refer the reader to our paper on Ubiq [1] for more details, but emphasise some key points below. Ubiq is integrated into users' projects as source code. It implements a number of common SVR features, such as rooms for users to find each other on the public internet, tracked avatars, and voice chat. It also implements uncommon functions specific to research, such as a distributed logging system, and VR-specific quality-of-life features, such as a PlayerController that supports both desktop and VR. Ubiq's API allows users to easily implement their own networked object behaviours. The API is slightly lower-level than similar libraries such as Photon or UNET, a deliberate pedagogical decision. Support for cross-platform sessions is another feature, and especially important for our remote module as we cannot provide every student with a VR headset.

The Unity side of the Ubiq project consists of a number of core Unity Components, and a small set of very complete samples. We designed the samples with the expectation that they would be used as a starting point for students to modify. The *Hello World* sample (Figure 1) implements a fully functioning SVR application, with a menu system for joining a room, all of the common services such as audio chat, and an example item that spawns networked objects (fireworks).

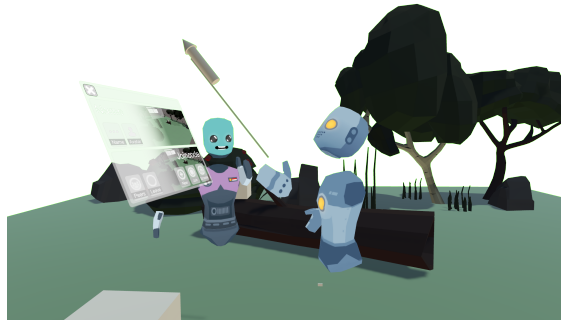


Figure 1. Two avatars in Ubiq's *Hello World* Sample

All the assets in the Ubiq samples are CC0² (in the public domain) or similarly redistributable. Users can use the *Hello World* or other scenes as a starting point, and confidently do what they like with the resulting application, subject only to the constraints of what they themselves add.

TEACHING WITH UBIQ

Ubiq's role is to support the networking and social VR aspects of the module. Ubiq is designed to support teaching in two ways. First, to provide enough functionality and be easy enough to use, that students can create a fully-functional SVR experience within the time constraints of the module. Second, to be easy enough to understand, but low level enough, that students learn the principles of networked VR systems.

In our VE module, Ubiq is used in the practical coursework component, which comprises 50% of the module's marks. Students are introduced to Ubiq through in-person lab practicals, which interleave 30-60 minute lecture-style presentations with practical exercises. During these led sessions we also introduce the other tools, such as Unity. After 2-3 sessions, students are instructed to form groups (of 3-5) in which to complete the assignment. The module also includes a two-hour lecture specifically on networked VR. This does not reference Ubiq however, but instead covers networked VR principles from a more general perspective, and discusses lower level networking technologies³.

In the past two academic years, students have

²<https://creativecommons.org/share-your-work/public-domain/cc0/>

³Our teaching materials are available at <https://ubiq.online/teaching/>

been assigned the task of building a social virtual reality experience. The task is deliberately broad. Students are permitted to build any type of application they like (naturally, most choose to build games). The guidance provided is that the highest marks are achieved by implementing some type of networked behaviour beyond the standard capabilities we provide with Ubiq. For example, students could create an application that makes use of a new social channel, or objects that use networked physics. It is also acceptable to create tasks that rely on a shared state, so long as the task requires two or more people to complete. For example, a number of teams implemented collaborative cooking games (similar to *Overcooked*).

In 20/21 students were taught remotely, and Ubiq was presented with the expectation that students could actually use it to meet while developing their coursework. In 21/22 we adopted a hybrid model with 90% of the class attending in-person lectures and labs, and 10% fully remote. Though the majority of taught material was attended in person, most of the lab support was done remotely. Students worked on this coursework part-time (20%) for two months.

For support, the module employed 5 teaching assistants who were familiar with Ubiq and networked VR who would meet remotely with students one on one on request. There was also a Moodle forum dedicated to technical support. Students also had the option of attending the remaining lab sessions in person to receive support on demand.

STUDENT ACHIEVEMENTS

Students returned projects with varying levels of quality. Of the 28 projects across two years (115 students in total), four are described here to provide examples of what students were able to realise. All four of these demonstrations have been made available open source by the students. Students are encouraged to look at previous year's projects. The teaching team maintains a fork of each project, keeping them up to date with any Ubiq changes.

Two-Scale Maze is a combination of a sliding puzzle and a labyrinth. Players must arrange tiles on a table to assemble a maze that can be traversed. However, some tiles are covered, so the

only way to see how they should fit together is for at least one of the players to shrink to a fraction of their size and traverse the incomplete maze while communicating to the full-size players what needs to be changed.

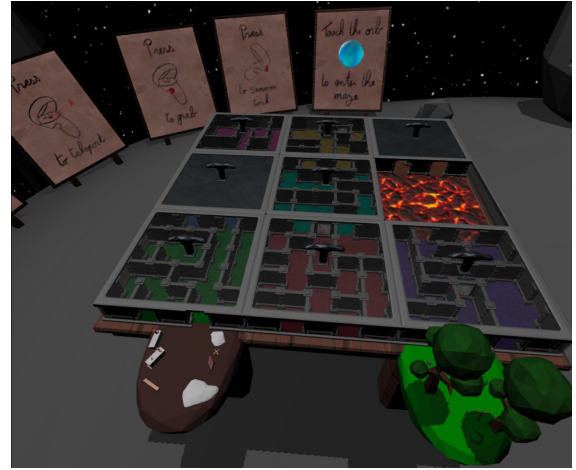


Figure 2. The perspective of a full-size player in Two-Scale Maze

Transballer is a collaborative take on the *Fantastic Contraption*-style game. Players begin in an empty room with a spawn point and a goal. The challenge is to use assorted items in their inventory to create a scaffolding to carry balls across the room. The challenge can be increased with the addition of obstacles and multiple spawn points.

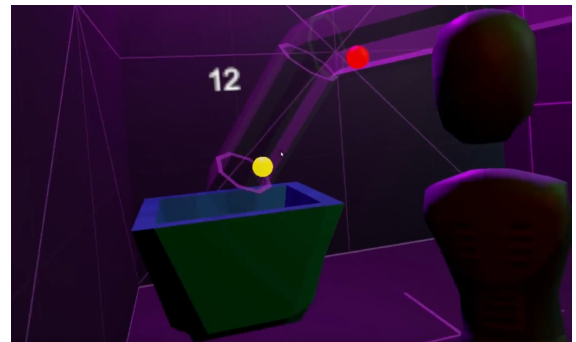


Figure 3. An avatar watches balls flow through a collaboratively created pipe network in Transballer

CoRails is a collaborative track-building game. Users begin in an environment with a locomotive that moves at a constant speed. Players must harvest resources from the environment and build a track in front of the train to keep it going.

The speed of the train sets the challenge of the game as players must collaborate to make best use of their time to keep the resource rate high enough to prevent the train going off the rails.

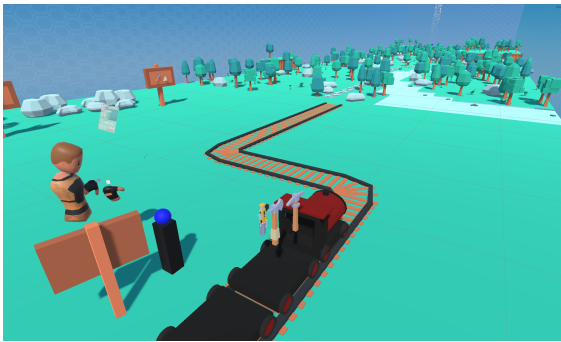


Figure 4. A user collects tools to build a train-track through a forest in CoRails

Blockism puts players in a factory in which they must assemble items on an assembly line, however the players are given asymmetric roles with specific capabilities, preventing one person from completing the assembly alone.

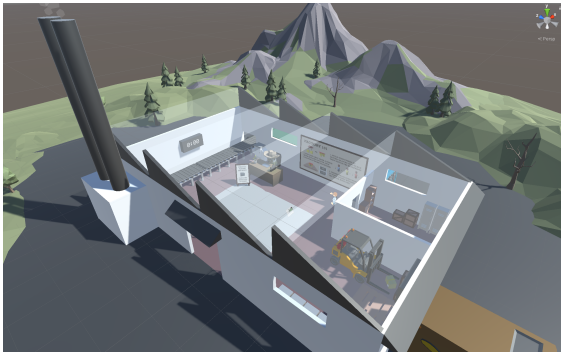


Figure 5. A user waits to be assigned a role which decides which part of the task they can contribute to in Blockism

DISCUSSION

Based on the outcomes of the coursework, we consider Ubiq a successful component of the transition to a hybrid-teaching networked-first coursework. Students achieved the learning outcomes as evidenced by their effective VEs. These were necessarily collaborative, so students have learnt principles of social VR.

Whether Ubiq was the optimal framework depends on a number of factors. The first is

proficiency. As Ubiq prioritises transparency and flexibility, students may have been more proficient in an application or existing library. Though none of the projects used Ubiq's more advanced research features, hardly any of them would have been possible in current commercial applications as they all shared at least a custom state. Some (e.g. Transballer) went further and implemented their own synchronisation mechanisms. This validates our decision to use a library rather than a platform.

This and many more projects could have been implemented in a library such as Photon however. So, the second consideration is whether students learnt the principles of how networked VR systems operate from Ubiq's lower-level API. The risk of a lower-level API is that students input more effort into using it at a cost of creativity in the networked behaviours. All student projects implemented some sort of synchronisation, and so would have written code to exchange messages. It is harder to evaluate how much they learnt about the facilitation of message exchange compared to using a framework which handled it transparently however, as there is no assignment that directly measures this. However, existing mid-level libraries are designed for general PC gaming, not VR, and do not include the fully integrated social services (such as spawning avatars with voice chat), that Ubiq provides. Some teams (e.g. Two-Scale Maze) also used mechanisms such as asymmetric communication between scenes which are unusual, if even supported, in networking libraries, validating at least Ubiq's flexibility.

As Ubiq is open source, the sharing of code between years is feasible and not something that we could rely on with any commercial platform.

Anecdotally, over the past few years students have required less and less technical support. Tools such as Unity have become more accessible, and usually at least one student group member has experience before starting the module. The majority of Ubiq support questions fielded by the teaching team were concerned with bugs or missing features (e.g. desktop controls for one-button mice or trackpads), or instructions for downloading and setting up the project. We do not consider it acceptable for Ubiq to have bugs or require workarounds. However, the fact most students are able to implement their network

behaviours without staff support suggests the API is accessible, and validates the decisions behind the messaging model and imitation of Unity's Component based system. It also emphasises the importance to students of having the desktop interface. While students learnt the API, we cannot say whether they did so more or less easily than they would a library like Photon or UNET, which would require a controlled evaluation.

We must also consider students' transferable skills. While Ubiq is commercial-friendly, students should learn to use networking libraries in general, rather than our networking library. To this end we try to keep Ubiq concerned with networking, and only add additional functionality such as the PlayerController, where there is no in-built Unity alternative. Ubiq is designed to be a networking package, rather than a 'layer', and so we also encourage students to use third-party packages for other aspects of their projects, and take time in the taught sessions to emphasise what Ubiq shares with other networking libraries.

While the coursework is networked-first, it is intended to evaluate students broad understanding of VR, including interaction, audio, presence and other topics. Ubiq's focus is social VR; it is not intended to directly support these other topics and so we do not consider Ubiq's contribution to these teaching outcomes. However they do form part of the mark scheme and students are evaluated for them in the coursework.

FUTURE WORK

Ubiq is constantly being revised based on our observations in teaching, and the research requirements of our group and collaborators. Changes we are making that directly flow from the student experience are expanded support for devices and input systems, and streamlined delivery of the project. We are also intending to refactor how object addressing works to make the API simpler, but without making the messaging itself higher level. We will also introduce a set of Components that facilitate automatic synchronisation of objects. This should bring Ubiq's ease-of-use up to par with Photon, UNET, etc for students who wish to use the higher-level API.

A number of students go on to use Ubiq in individual projects, and these have resulted in new features and demonstrations. We expect this

year to be no different, with a group of students now working on new features such as networking optimisation and avatars.

It is difficult to perform controlled evaluations within our module, but we hope that Ubiq's broad use-case means that summer schools, internships and masters projects could provide an opportunity to gather objective comparisons between Ubiq and alternative libraries.

CONCLUSION

In this article we share our experiences of teaching using an in-house system for building social VR experiences. We created this system out of necessity, after evaluating a number of commercial platforms. We concluded that the reason they were missing important features was not due to lack of interest, but because some requirements directly conflict with business goals. The first version of Ubiq was created over four months in the summer before our 2020/2021 module commenced and has undergone continuous improvements since. Ubiq has been used to support our networked-first VE module coursework for two years, during which time the module has been delivered online and in hybrid form.

Requiring students to learn an in-house system carries risk; that students would be more productive in a more mature system, that their skills are not transferable, and that the development time would be better spent elsewhere. For the last two iterations of our module we see Ubiq validated in a number of ways. Students implemented mechanisms that would not be possible in commercial systems. Students shared support across years. Students made their projects open-source, which would not be possible with legal limits on redistribution or third-party backends. As not all students had access to VR headsets, it was important that student projects easily work in VR and on a PC, which Ubiq handles seamlessly.

Anecdotally, the quality of the student projects matches or exceeds those of previous years which did not require a networking component. The lack of queries about the messaging system, combined with the successful projects suggests that the API is easy to learn and does not inhibit progress. We are constantly revising Ubiq based on our observations of students interacting with it. In the future we intend to explore more about

the student experience and learning outcomes with more focused investigation. Despite the non-trivial investment from the group so far however, we consider Ubiq an important component in our recent, successful networked-first coursework.

ACKNOWLEDGMENTS

We would like to acknowledge all those who have assisted in teaching with Ubiq over the past two years: David Swapp, Lisa Izzouzi, Klara Brandstätter, Felix Thiel, Nels Numan and Dan Archer. We would also like to acknowledge the student teams who have allowed their work to be presented here. This work was supported by EPSRC grants EP/P004040/1 and EP/P004016/1.

■ REFERENCES

1. Sebastian Friston, Ben Congdon, David Swapp, Lisa Izzouzi, Klara Brandstätter, Daniel Archer, Otto Olkkonen, Felix Johannes Thiel, and Anthony Steed. 2021. Ubiq: A System to Build Flexible Social Virtual Reality Experiences. In *27th ACM Symposium on Virtual Reality Software and Technology (VRST '21)*, December 08–10, 2021, Osaka, Japan. ACM, New York, NY, USA 11 Pages. <https://doi.org/10.1145/3489849.3489871>
2. Microsoft, What is AltSpace VR. [online]. Available: <https://docs.microsoft.com/en-us/windows/mixed-reality/altspace-vr/overview> (URL)
3. Mozilla Foundation, Hubs. [online]. Available: <https://hubs.mozilla.com/> (URL)
4. Meta, Horizon Worlds. [online]. Available: <https://www.oculus.com/horizon-worlds/> (URL)
5. RecRoom, RecRoom. [online]. Available: <https://recroom.com/> (URL)
6. Photon, PhotonEngine. [online]. Available: <https://www.photonengine.com/pun> (URL)
7. UNET, Unity3D. [online]. Available: <https://docs.unity3d.com/Manual/UNET.html> (URL)
8. Gary Burnett. 2021. Bringing the metaverse to life: how I built a virtual reality for my students – and what I've learnt along the way. [online]. Available: <https://theconversation.com/bringing-the-metaverse-to-life-how-i-built-a-virtual-reality-for-my-students-and-what-ive-learnt-along-the-way-171760> (URL)
9. Sai Shreya Nuguri, Prasad Calyam, Roland Oruche, Aniket Gulhane, Samaikya Valluripally, Janine Stichter, and Zhihai He. 2021. VSocial: a cloud-based system for social virtual reality learning environment applications in special education. *Multimedia Tools Appl.* 80, 11 (May 2021), 16827–16856. <https://doi.org/10.1007/s11042-020-09051-w>
10. C. Zizza, A. Starr, D. Hudson, S. S. Nuguri, P. Calyam and Z. He. 2018. Towards a social virtual reality learning environment in high fidelity. 15th IEEE Annual Consumer Communications & Networking Conference (CCNC). <https://doi.org/10.1109/CCNC.2018.8319187>.
11. Rojas-Sánchez, M.A., Palos-Sánchez, P.R. & Folgado-Fernández, J.A. 2022. Systematic literature review and bibliometric analysis on virtual reality and education. *Educ Inf Technol* (2022). <https://doi.org/10.1007/s10639-022-11167-5>
12. Mystakidis, Stylianos, Eleni Berki, and Juri-Petri Valtanen. 2021. Deep and Meaningful E-Learning with Social Virtual Reality Environments in Higher Education: A Systematic Literature Review. *Applied Sciences* 11, no. 5: 2412. <https://doi.org/10.3390/app11052412>
13. Jaziar Radianti, Tim A. Majchrzak, Jennifer Fromm, Isabell Wohlgenannt. 2020. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*. Volume 147. <https://doi.org/10.1016/j.compedu.2019.103778>.
14. Takala, Tuukka M., Malmi, Lauri, Pugliese, Roberto and Takala, Tapio. 2016. Empowering Students to Create Better Virtual Reality Applications: A Longitudinal Study of a VR Capstone Course. *Informatics Education*. 15, no. 2 (2016): 287-317.
15. V. T. Nguyen, R. Hite and T. Dang. 2018. Web-Based Virtual Reality Development in Classroom: From Learner's Perspectives. 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), 2018, pp. 11-18, doi: 10.1109/AIVR.2018.00010.

Sebastian Friston is a Research Associate with the Virtual Environments and Computer Graphics Group, Department of Computer Science, University College London. Contact him at sebastian.friston@ucl.ac.uk.

Ben Congdon is a Research Associate with the Virtual Environments and Computer Graphics Group, Department of Computer Science, University College London. Contact him at ben.congdon.11@ucl.ac.uk.

Anthony Steed is a Full Professor, Head of the Virtual Environments and Computer Graphics Group, and Deputy Head of Department in the Department of Computer Science, University College London. Contact him at a.steed@ucl.ac.uk.