"All the world's a stage" – the Open Broadcaster Software (OBS) as enabling technology to overcome restrictions in online teaching

Gerhard Kristandl
University of Greenwich, UK

Abstract

The Open Broadcaster Software (OBS) Studio is free software that enables tutors to overcome the restrictions typical of video-conferencing software like MS Teams or Zoom. This paper develops from a real-life situation where the pandemic-induced move to online teaching has put severe restrictions on otherwise engaging teaching and student activities – specifically role-plays and use of props – hitherto run in the classroom. The goal is to demonstrate how OBS Studio was set up and employed to overcome this situation. In doing so, the paper offers, against this background, a simple set-up tutorial for two scenes to show that using OBS Studio is straightforward and intuitive.

Introduction

When William Shakespeare had melancholy Jaques exclaim in ***As You Like It*** that "*All the world's a stage, and all the men and women merely players*", he likened human beings to actors on the stage of life. As we 'act out' our lives (to stay in that perspective) and assume many roles, so do we learn, not just by sitting and listening, but by doing, as "*action begets learning*" (Alkin and Christie, 2002, p.210; Cavaliere and Sgroi, 1992, p.7). It may be a long way from Shakespeare's past to our current present and future, but as COVID-19 removed us from the classroom stage and put us firmly in front of a computer screen, most of our 'stage of life' moved online in turn. More often with dread and anxiety than optimism and hope (Sycamore, 2020), many of us sat in front of that ever-watching webcam and MS Teams/Zoom platform, trying to figure out how this 'online teaching stuff' *actually* works. Or, to pursue the bard's own metaphor, how can we make this new 'online stage' work for our teaching and our students' learning? How can we overcome the restrictions imposed by the technology that seemingly limits or even obstructs what we would typically do in the classroom to engage our students?

The purpose of this case study is to present the Open Broadcaster Studio (OBS) software as an enabling technology that can help address the questions above. In the remainder of this paper, I shall show how OBS can help to overcome the tight technological restrictions of typical video-conferencing software in terms of screen layout and design. I shall demonstrate this against the backdrop of a specific situation I was facing in terms of moving a student-engaging classroom activity – role-playing – to online.

The situation

Technology might be either restrictive or enabling, but a mere screenshare functionality and a face below it in video-conferencing software like MS Teams or Zoom felt restrictive in terms of what we could show to our students and do with them (remember that it took a considerably long time for the much desired breakout-room functionality to finally become a feature in MS Teams). ‘Natural’ did not come to mind – a tight technological corset with little control over the instruments and aesthetics. This was especially true for many of my classes where, under regular circumstances, we would perform a lot of role-playing, often supported by the use of such props as poker chips to mimic products or cash, teddy bears and flags to convey the concepts behind business process capabilities or Lego bricks to demonstrate the difference between data and information. Over the course of my long career as a teacher, I learned to hold such role-playing activities in high esteem owing to their potential to have a lasting impact on student learning (Rao and Stupans, 2012). I define ‘role-playing’ in this context as assuming/acting out various positions and perspectives in pre-defined scenarios to encourage learning both interactively (from application) and passively (from observation of others, as Rao and Stupans, 2012, Lauber, 2007, Baruch, 2006 and Eitington, 2001 likewise believe). In this definition – inspired by Baruch’s (2006) inclusion of drama in role-play teaching – I include the stage props I’ve referred to already as a dramatic element to add realism and tangibility to my scenarios.

It is straightforward to imagine such role-plays being acted out in an actual classroom in front of and with students. But what about the digital ‘stage’ of a video-conferencing platform? Of course, we can try to do as many of these things as possible in front of one webcam – that may as well be the one built into that laptop the online teacher is cramped over during sessions (both live and recorded). We can hold up props to the camera, or – if the tutor is tech-equipped enough – move an external webcam around. However, this seems not very practical and rather distracting in the moment. Certainly, to me at the outset, it appeared that the alternative was a dim one: having to give up my popular role-playing intermissions and props – the very things that tended to stick in students’ heads long after they graduated.

Abandoning these in-class role-playing games in favour of ‘traditional’ teaching from the front (and now … to a camera!) was a big ask and seemed a few pedagogical steps backwards. I needed an alternative – an instrument enabling me to keep my role-playing, even in a digitally adapted form. That would entail a change of perspective: from the ‘view from the front’ to one more suitable to my purposes. In particular, I wanted to create a view similar to that of a visualizer in a regular classroom, where an overhead view of the person presenting would show the desktop as a stage for demonstrations or games and as visual base for role-playing online in the absence of a physical in-class presence. At the same time, I wanted the students still to see me ‘performing’ – reactions, facial expressions and gestures included – to mimic the in-class experience as closely as possible.

Use of such a perspective is not unheard of and can be observed in many videos distributed by online content creators on platforms like YouTube (a video distribution platform) and Twitch (a live-gaming channel platform) where video-in-video and overhead shots are commonplace. These views often show presenters ‘doing’ something with their hands (e.g., repairs, unboxing of new smartphones, drawing) and provide an ‘over-the-shoulder’ angle for the audience to feel physically present. The software that enables these content creators to achieve this effect is the **Open Broadcaster Software (OBS) Studio,** where the change in perspective could be employed to evoke in students that feeling of being physically involved in the role play, even though they are not because it’s online.

Implementation

‘Tangibility’ of the learning experience is key to running successful role plays and games in a classroom. For the online context, this means evoking tangibility by means of props (toy figurines) that can be moved around to help develop the concept of *internal controls* and *segregation of duties,* following a scenario from a paper by Groff (1989). In my case, the students who participate in this role play are Year 2 (level 5) Accounting Information Systems (AIS) students studying on an Accounting and Finance (BAAF) or Accounting and Financial Information Systems (AFIS) programme. I run the game half-way into term 1, at a stage when students have had an introduction to various types of employee and management fraud and segregation of duties, an internal control concept that can be difficult to grasp but is pivotal in preventing a wide range of fraud schemes that may threaten the business and its processes. In brief, the *segregation of duties* ensures that no single employee should be able to perform several steps in a process that entails physical custody of an asset, keep records for the asset and have authorisation to approve transactions and decisions regarding the asset (Romney *et al*., 2021). If these duties are only partially or not at all separated and assigned to different people, roles, departments (or similar), then there is a risk that individuals could engage in fraudulent activities (e.g., theft, larceny, lapping, embezzlement, etc.). It is a crucial learning outcome that students not only understand the concept per se, but understand how the absence of internal controls, even partially, may pose a considerable financial risk to an organisation. However, as students at this stage of their studies do not have much exposure to business process thinking or risk to business processes by fraudulent activities, they tend to struggle with this, a concept that has relevance to and is of significance in most other AIS topics. When I realised that in-class teaching would no longer be possible, I had therefore to exert the utmost effort to ensure that this crucial learning outcome was not compromised.

The game itself focuses on a fraudulent manager whose intention is to steal products from her/his company, from the warehouse where the manager is located. The students are not instructed before the actual session except for the fraudster – played by a student – about a week before class. Just as a company may be unprepared for fraud occurring, so is the rest of the class – thrown into the role-play unawares. Figure 1 below shows the basic process:

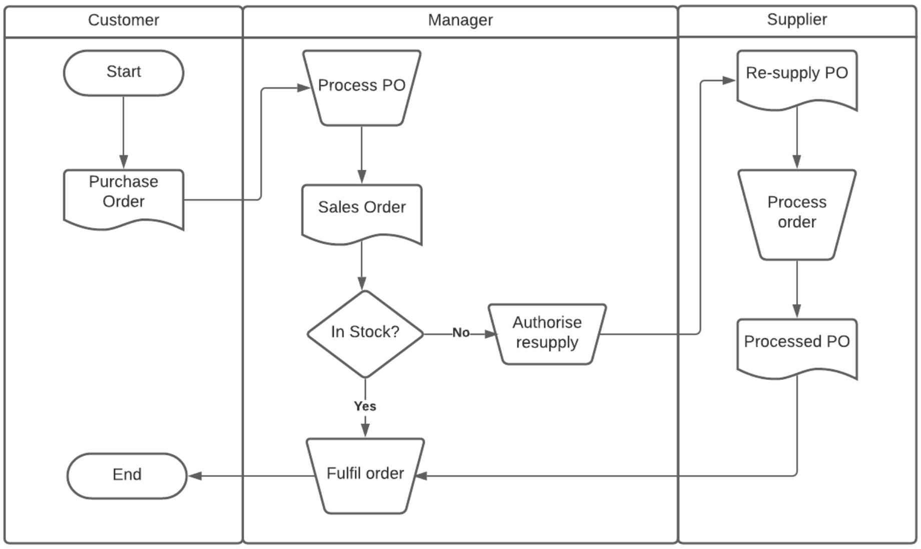


Figure 1. Basic process (Round 1)

Over three rounds, the manager attempts to steal goods and falsifies records to avoid detection.

Basic process: Three students are randomly named customers and given purchase orders (POs) with pre-defined quantity and price printed on them. Each customer approaches the manager, who checks the PO, processes it into a sales order (SO) and checks whether there is enough in stock. If the order can be fulfilled from current stock, the customer is handed the number of products ordered (represented by poker chips; see also figure 2). If the manager cannot fulfil the order, s/he authorises the re-supply order that the supplier (typically played by the tutor) fulfils in turn. The manager returns with the items and fulfils the customer's sales order and the process ends.

In each round, the manager attempts to steal a portion of the items procured from the supplier and allocates the total purchase cost to the items s/he officially records as purchased. The remaining students keep a tally of revenue and cost of goods sold, as well as inventory levels based on the transaction they observe (the 'true' figures). All figures (sales prices, quantities, purchase prices, etc.) are called out loud to enable students to calculate the true figures. A round ends when all three sales orders are fulfilled. At the end of each round, students report the figures they calculated. The manager writes their figures (profit, closing stock) on a piece of paper and hands it to the reporting official (not shown in the flowchart in figure 1). The reporting official (another student or the tutor) reads out the figures reported by the manager – unsurprisingly, these figures will differ (lower profit, lower closing stock) from the 'true' figures the remaining students observe. Each round will see changes to the duties that ought to be segregated:

Round 1: This corresponds to the basic process as described above. In this round, the manager holds all duties in her/his hand – physical custody of the asset, authorisation to purchase re-supply and record keeping. The manager is able to steal without detection. This scenario is devoid of any internal controls.

Round 2: In this round, a student will be randomly selected to be the accountant who takes over the record-keeping duty. Physical custody of the asset and authorisation to purchase re-supply remains with the fraudulent manager. As the accountant is not privy to the transaction itself, s/he can only rely on the figures reported by the manager, who can continue to steal items and allocate the costs to the remaining items. When compared, profit figures and closing stock again differ between figures reported by the accountant and figures the students calculate based on their observations. This corresponds to a partial implementation of internal controls.

Round 3: In the last round, a procurement manager is introduced, who authorises (or rejects) purchase requisitions issued by the fraudulent manager and turns them into purchase orders. Without this authorisation, the supplier cannot hand over the items required to fulfil the customers' orders. As such, it has become impossible for the fraudster to steal inventory or hide the cost.

Because of the tangible experience and direct involvement of the entire class in the role play, students experience first-hand the concept of internal controls and how their absence or presence can affect the degree of vulnerability of the system. As the fraud unfolds (for they can observe and catch the manager in the act), they start to develop an understanding of the fraud scheme and how internal controls act as preventive and detective countermeasures against it. In a way, everybody except the fraudster also acts as an auditing observer. They need to calculate the correct profit and closing stock and determine how the fraud happened. On stage, there unfolds not a *Whodunnit*, but a *Howdunnit*.

The OBS set-up (procedure described in appendix A) supports the delivery of the role-play to the students, using figurines as tangible stand-ins and schematics on a board (figure 2).

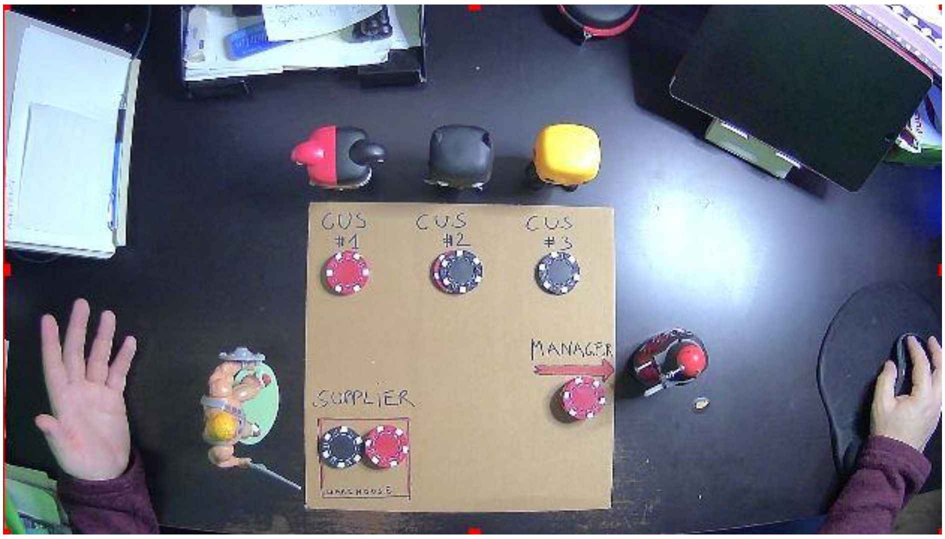


Figure 2. The set-up for the internal controls game via overhead cam

The scenes set up in OBS featured an overhead view, flipped so that the students were put in such a position as enabled them to look over my shoulder (the ‘watch somebody doing something’ view I’ve already referred to). To achieve this, I installed a second web cam and positioned it above the desk at an adequate distance. I happen to have the luxury of a shelf behind my computer screens, but without this advantage, a clamp and bendy holder for web cams may be attached to the computer screen).

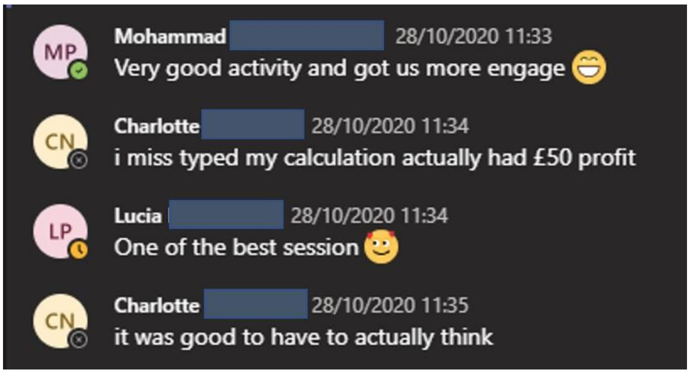
For visual cues, I drew a board that indicated what roles the figurines represented in the role-play – as figure 2 shows, there are the three customers, the fraudulent manager and a supplier. When first the accountant and then the procurement manager entered the stage, additional figurines were added to the view. As I went through each step in each round of the play, explaining what was happening in real time, students were able to follow the events on screen through this overhead view and communicated, either via microphone or chat, with one another and me. A second view (not shown here) featured the same overhead view as seen in figure 2, but with me as a ‘talking head’ in the bottom left corner, allowing me to come in between rounds to discuss with students the events of the preceding round.

Did it work? - Making in-class role-plays work as desk-cam plays

Using this set-up had clear advantages, as students, in lieu of ‘acting’ on the stage themselves in a classroom, could refer to the figurines as stand-ins and actors on their behalf (the ‘drama’, as Baruch, 2006, puts it). This set-up allowed them to observe the events being played out as a desk-play as well as the reactions of other students in the chat (learning passively from observation) and still actively participate by taking the perspective of customers, accountants, procurement managers, or suppliers (learning interactively). This observation is fully in line with Rao and Stupans (2012), Lauber (2007), Baruch (2006) and Eitington, 2001): in their assumed role as auditing observers especially, they assumed both passive and interactive positions across the three rounds. Dramatic elements and equal opportunities to learn passively and interactively created a tangible experience, despite them taking place online, as action indeed begot learning (Alkin and Christie, 2002; Cavaliere and Sgroi, 1992). This pulled students in and had a similar effect to that of the in-class role plays before.

Linked to the 'learning by doing' nature of role-plays (Alkin and Christie, 2002; Cavaliere and Sgroi, 1992), the students experienced a learning curve throughout the play – the first round with its basic set-up and small number of roles allowed students to learn the game play mechanics. By calculating basic accounting figures (revenues, cost of goods sold, profit, opening and closing stock), students could apply previous knowledge from their programmes. In each additional round, students gained more confidence in their understanding of the process, as complexities were gradually added. As such, there was no discernible difference in the online version from the same role-play performed in the classroom. If anything, the online delivery, with its focus on an auditing experience, enabled students to achieve the learning outcomes quicker than in class, where at times the distinctions between the roles were at times overlapping: for example, students who played the customers also needed to be auditors. This clearer distinction of the roles and which individuals are playing them is a great takeaway from the online performance.

From my perspective as a lecturer and host of the role-play, the online delivery enabled by OBS allowed a much better focus on a smooth delivery of the process. The need to focus even more on the roles and the process led to a highly satisfactory session. Student feedback during and after the session was extremely encouraging, as figure 3 shows:



MP Mohammad [redacted] 28/10/2020 11:33
Very good activity and got us more engage 😁

CN Charlotte [redacted] 28/10/2020 11:34
i miss typed my calculation actually had £50 profit

LP Lucia [redacted] 28/10/2020 11:34
One of the best session 🥰

CN Charlotte [redacted] 28/10/2020 11:35
it was good to have to actually think

Figure 3. Student feedback in teams during the session

It appealed to this age group, owing to their familiarity with this type of media (Headleand, 2019), but, most importantly, it was fun for all involved. Based on that experience, I consequently put all my in-class role plays and games online, using my overhead cam enabled by OBS.

Limitations

New technologies can be daunting at any point. The overwhelming wave of software and hardware considerations, alongside the plethora of training and how-to sessions after our teaching moved online, put an incredible amount of pressure on all of us. OBS Studio, used purely as virtual camera feed into video-conferencing software, as described above, does demand some basic system requirements (see above), but it can be reasonably run next to the MS Teams or Zoom if the computer is not too old or under-specified. When OBS Studio is used for direct streaming and/or recording, the system resources are much more taxed because of the video encoding requirements. For example, I had many issues when running OBS next to MS Teams with my four-year-old laptop with 8GB RAM, even with a quicker solid-state drive (SSD) installed. However, as it is a free software, tutors can quickly find out if their computer can, or cannot, run OBS Studio without issues. I recommend investment in a strong graphics card with a corresponding graphical processing unit (GPU) that enables hardware encoding, as well as good-quality microphones and additional webcams. I also suggest – especially for tutors new to the basics of video production and scene layout – adequate help and training in the use of software like OBS, which might otherwise be a daunting and even off-putting experience.

Conclusions

What can other tutors take away from this? This paper aimed to establish OBS Studio as an enabling technology to overcome restrictions otherwise imposed by video-conferencing software used in teaching. The COVID-19 pandemic has accelerated the move to hybrid and full online teaching and has caught out tutors who have found themselves no longer able to apply such engaging classroom activities as role-playing, a learning activity that fosters interactive and passive acquisition of knowledge (Rao and Stupans, 2012). As the case study demonstrates, such activities can still be made to work and achieve their purpose by using the right enabling technology. I found OBS Studio a very useful tool that enabled me to take full control over what I can show my students on screen, down to a very granular level if so required. This case study highlights the set-up of a visualizer-styled overhead camera (the ‘desk-cam’) that can easily be switched to and from other scenes. Student feedback generally supported the impression that the possibilities of this free software helped generate a variety of ‘stages’ on which the tutor can break free from the restrictions of full-frontal cam teaching, especially useful in an activity like a role-play, where observation and interactivity are key. OBS is easy to set up initially and a basic layout (such as shown in appendix A) with a second webcam does not constitute a prohibitive barrier.

Of course, running OBS Studio may be demanding on the system requirements that the individual tutor might not be able to meet at home. While using OBS Studio as a virtual camera does not place an overly heavy burden on the computer’s CPU, GPU and memory, the workload does increase considerably should the tutor wish to expand using OBS towards direct live-streaming and/or direct recording. Added investment in peripheries, such as high-quality microphones and webcams, might also be required but, given their added value, these may find financial support from the institution.

Future research could focus on a wider range of different learning situations both online and in the classroom to evaluate its full impact yet further, but also possible limitations to be factored into future applications.

Reference list

Alkin, M.C. and Christie, C.A. (2002) ‘The Use of Role-Play in Teaching Evaluation.’ *American Journal of Evaluation*. 23(2), 209-218. Available at: https://doi.org/10.1177%2F109821400202300210 (Accessed: 25 January 2021).

Baruch, Y. (2006) ‘Role-play Teaching: Acting in the Classroom.’ *Management Learning*, 37(1), 43-61. Available at: https://doi.org/10.1177%2F1350507606060980 (Accessed: 25 January 2021).

Cavaliere, L. and Sgroi, A. (1992) ‘Active learning: Perspectives on learning that leads to personal development and social change.’ *New Direction for Adult and Continuing Education*, 53, 5-10. Available at: https://www.academia.edu/download/47016690/ace.36719925303201607041526-1rttkpx.pdf (Accessed: 18 March 2021).

Eitington, J.E. (2001) *The winning trainer winning ways to involve people in learning*. 4th ed. Boston, MA: Butterworth Heinemann. ISBN: 978-0750674232

Groff, J.E. (1989) ‘A Game To Introduce Accounting Information Systems Students To Certain Internal Control Concepts.’ *Developments in Business Simulation & Experiential Exercises*, 16, 165-168. Available at: https://journals.tdl.org/absel/index.php/absel/article/view/1887 (Accessed: 18 March 2021).

Headleand, C. (2019) ‘Streaming with Students.’ *Manual*. University of Lincoln. Available at: https://eprints.lincoln.ac.uk/id/eprint/44688/ (Accessed: 15 June 2021).

Lauber, L. (2007) ‘Role-play: Principles to increase effectiveness.’ In: Silberman, M.L. (ed.) *The Handbook of Experiential Learning*. San Francisco, CA: Wiley, 185-201. ISBN: 978-0470117392

Rao, D. and Stupans, I. (2012) ‘Exploring the potential of role play in higher education: development of a typology and teacher guidelines.’ *Innovations in Education and Teaching International*, 49(4), 427-436. Available at: https://doi.org/10.1080/14703297.2012.728879 (Accessed: 15 February 2021).

Romney, M.B., Steinbart, P.J., Summers, S.L. and Wood, D.A. (2021) *Accounting Information Systems.* 15th ed. Harlow: Pearson. ISBN: 978-0135573006.

Sande, S. (2020) ‘Video Production with OBS: What it is and why you should use it.’ [Blog] *Rocket Yard*. Available at: https://eshop.macsales.com/blog/62755-video-production-with-obs-what-it-is-and-why-you-should-use-it/ (Accessed: 15 March 2020).

Sycamore, R. (2020) ‘Anxiety during COVID-19.’ *Journal of Paramedic Practice*, 12(6), 247-247. Available at https://doi.org/10.12968/jpar.2020.12.6.247 (Accessed: 15 Feb 2020).

Appendix A

OBS is a free open-source software that enables live video production (https://obsproject.com). It is used by online content creators for both live-streaming and recorded videos and is available for Windows, Mac, and Linux operating systems. It allows for the creation and management of a variety of screen layouts (so-called 'scenes') where the presenter has full control over source/s, size, and location of its elements on screen. To start understanding how OBS can be used in conjunction with video-conferencing software like Microsoft Teams or Zoom, or recording software like Panopto, I will demonstrate how to set up two simple scenes, a *shared-screen* one both with me as a *talking head* in a bottom left corner and an *overhead-cam* one.

1. Technical requirements
2. Download and set-up OBS
3. Screen components – overview
4. Scenes and Sources
5. Controls: start streaming, start recording, start virtual camera, studio mode, settings, exit. Use virtual camera to use for streaming or recording in MS Teams and/or Panopto

1. Technical requirements

For OBS to run smoothly, there are some basic requirements (https://obsproject.com/wiki/System-Requirements) that will enable it to start without errors:

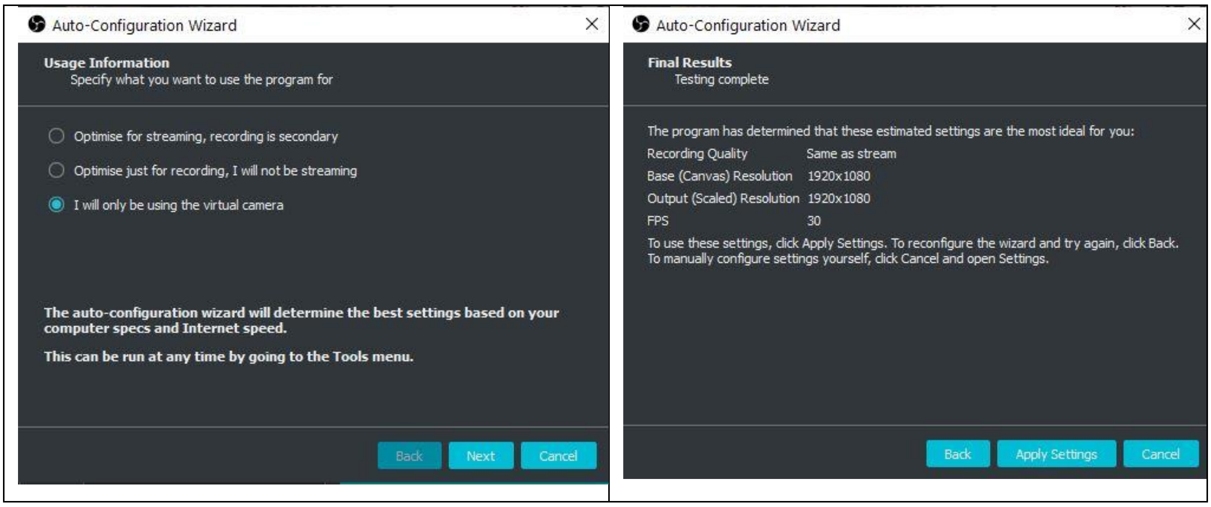
Windows	macOS	Linux/Unix
DirectX 10.1 compatible GPU Windows 8 / 8.1 / 10	Intel CPU (PPC is not supported) OpenGL 3.3 compatible GPU macOS High Sierra (10.13) or later	OpenGL 3.3 compatible GPU X window system

Table A1. OBS system requirements

Further technical requirements are very much dependent on what exactly a tutor wants to do and show and would require a closer look at hardware encoders. For the three scenes we are about to set up, a basic modern laptop should be sufficient, but this may vary individually. Most issues with OBS streaming and recording quality stem from Central processing CPUs, GPUs, and RAM that are too weak to process the video quality. As an in-depth discussion of the software's technical requirements is outside the scope of this paper, please consult the link above to learn more about hardware encoding.

2. Download OBS and use the auto-configuration wizard

Downloading OBS from https://obsproject.com/download is straightforward and available for the most-used operating systems Windows, macOS, and Linux/Unix. Straight after installation, an auto-configuration wizard window will pop up and ask what the program's intended use is (figure A1).



Usage Information
Specify what you want to use the program for

○ Optimise for streaming, recording is secondary
○ Optimise just for recording, I will not be streaming
● I will only be using the virtual camera

The auto-configuration wizard will determine the best settings based on your computer specs and Internet speed.

This can be run at any time by going to the Tools menu.

Final Results
Testing complete

The program has determined that these estimated settings are the most ideal for you:

Recording Quality	Same as stream
Base (Canvas) Resolution	1920x1080
Output (Scaled) Resolution	1920x1080
FPS	30

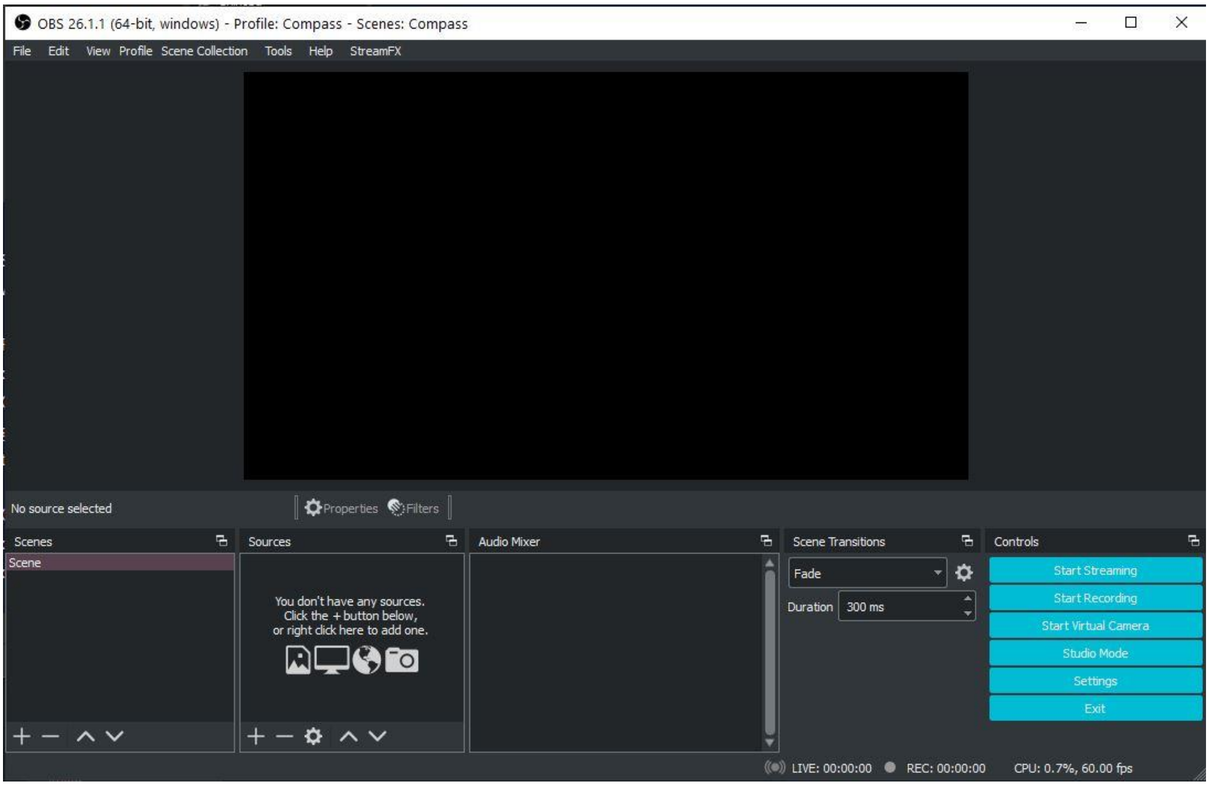
To use these settings, click Apply Settings. To reconfigure the wizard and try again, click Back. To manually configure settings yourself, click Cancel and open Settings.

Figure A1: OBS Auto-Configuration Wizard – virtual camera selection and suggested settings

In the context of this article, we will focus on using it as a virtual camera, as we want to feed this into another application (MS Teams, Zoom, or Panopto). Selecting 'Next' will show the final results of the OBS system test which may differ between computers. You can change these settings later (see later in this article), but as you get acquainted with OBS, there will not be a need to at this stage. The suggested resolutions come from the monitor/s in use, and the FPS (frames-per-second) are more than sufficient for the virtual camera. Selecting 'Apply Settings' will complete the wizard.

3. Screen components – overview

Next, a base OBS screen will open. Figure A2 and table A2 below highlight and explain these initial components:



No source selected

You don't have any sources.
Click the + button below,
or right click here to add one.

Fade

Duration 300 ms

LIVE: 00:00:00 REC: 00:00:00 CPU: 0.7%, 60.00 fps

Figure A2. Initial screen of OBS Studio without scenes or sources set

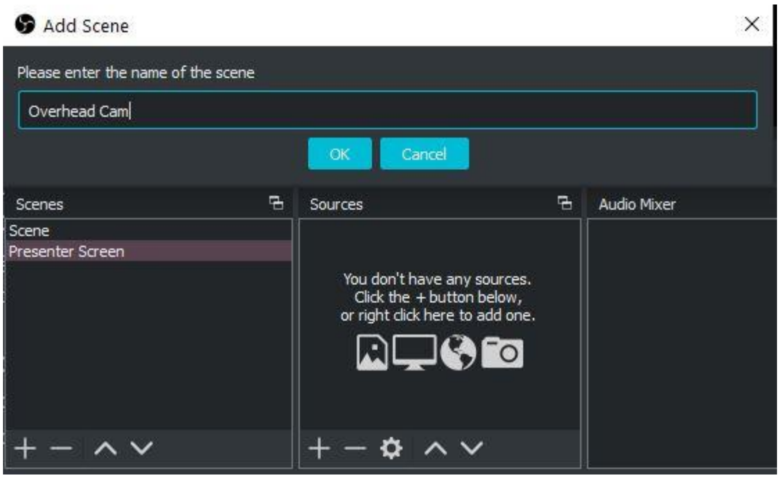
Screen component	**Purpose**
Screen preview/canvas	The preview window where the screen layout can be arranged as required.
Scenes	Contain specific layouts and enable switching between scenes to transition between layouts.
Sources	Define what is to be captured inside a given scene. This can be anything from web cams, images, text, active windows, displays, microphones, etc.
Audio mixer	Offers control over the audio captured via microphones and/or desktop.
Scene transitions	Transitional effects when switching between scenes (e.g., fade, cut, swipe, slide).
Controls	Buttons to start streaming, recording, virtual camera, studio mode, settings, or exit OBS.

Table A2: OBS screen components and their purpose

As the beginning OBS user will get more acquainted with these components, s/he will experience the wide range of options to change their appearances and uses. Free add-on programs and plug-ins add extra-components to OBS – examples are closed captioning (using Google AI), special effects for sources, remote-controls for scene transitions and many more. However, for the purpose of this paper, we will be able to work with what came in the initial installation package.

4. Scenes and sources

Scenes and sources are the bread and butter of OBS and will help to set up an overhead visualiser view on the presenter. Selecting the '+' sign on the bottom left adds a new scene. Figure A3 shows the addition of the 'Overhead Cam' scene.



Add Scene

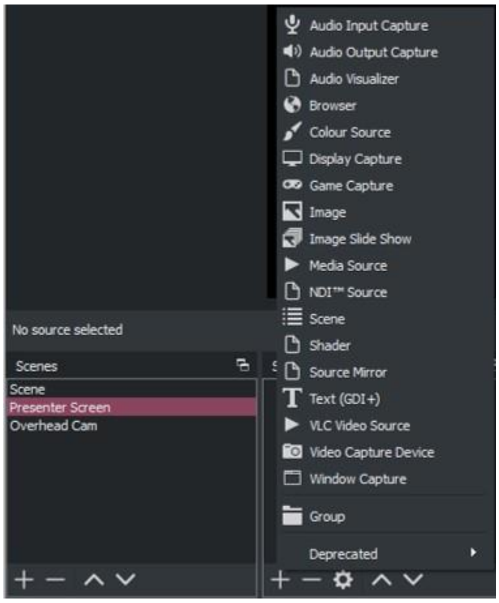
Please enter the name of the scene

Overhead Cam

You don't have any sources.
Click the + button below,
or right click here to add one.

Figure A3. Adding a scene in OBS

Selecting the scene titled 'Presenter Screen' allows the addition of various types of sources by selecting '+' in the Sources window, as figure A3 shows. The variety of sources allows for the addition of audio, web browsers, displays, images or specific windows currently open on the computer (figure A4).

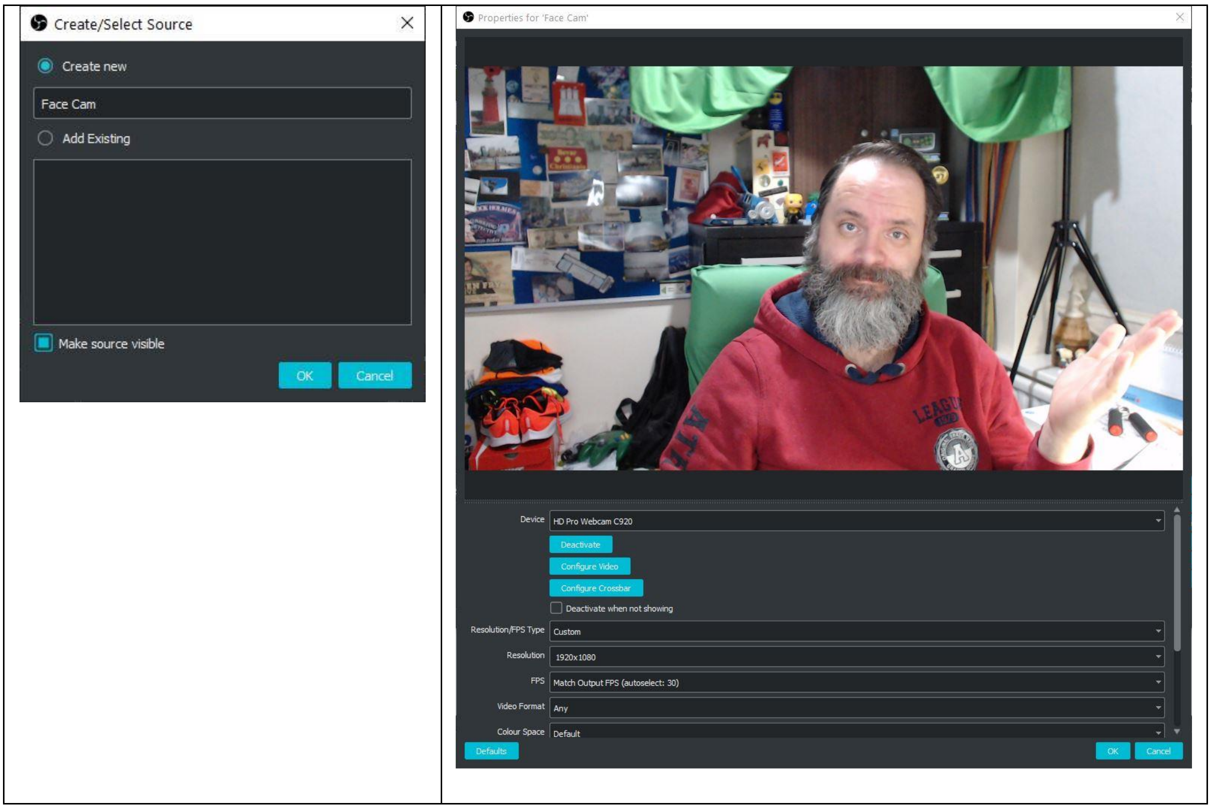


No source selected

Figure A4. Selection of potential sources for the *Presenter Screen* scene

Adding a *Display Capture* source allows the presenter to use her/his main screen (often the only screen) to become the ‘Shared Screen’ to show e.g., slides, documents, images – anything that can be shown on the main monitor. After selecting ‘OK’, the corresponding display can be selected and confirmed.

The face cam can now be inserted as a Video Capture Device source to add the talking-head feature often seen in tutorial videos. Selecting the main cam (provided there is a second one for the overhead cam later) will insert the Face Cam (figure A5).



Create/Select Source

● Create new

Face Cam

○ Add Existing

☑ Make source visible

Properties for 'Face Cam'

Device: HD Pro Webcam C920

□ Deactivate when not showing

Resolution/FPS Type: Custom

Resolution: 1920x1080

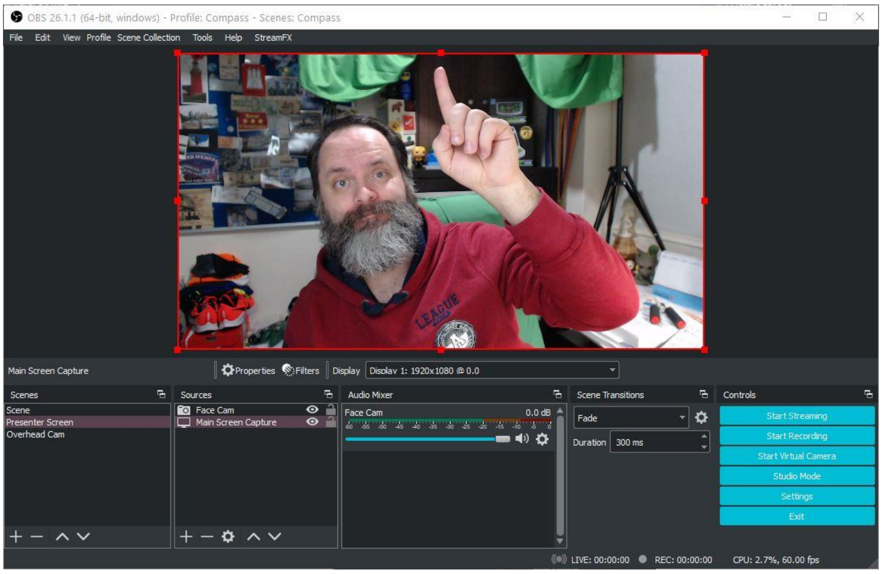
FPS: Match Output FPS (autoselect: 30)

Video Format: Any

Colour Space: Default

Figure A5. Adding the *Face Cam* as a source and its suggested properties

The rearranging of the two sources (face cam and main screen capture) can be achieved by



OBS 26.1.1 (64-bit, windows) - Profile: Compass - Scenes: Compass

Main Screen Capture

Display: Disolav 1: 1920x1080 @ 0.0

Scenes: Scene, Presenter Screen, Overhead Cam

Sources: Face Cam, Main Screen Capture

Audio Mixer: Face Cam 0.0 dB

Scene Transitions: Fade

Duration: 300 ms

LIVE: 00:00:00 REC: 00:00:00 CPU: 2.7%, 60.00 fps

Figure A6. Adapting the scene using handlers

To crop out parts of the face cam view, it is necessary to hold down the Alt key while using the red handlers accordingly. A bit of trial and error at this stage may lead to a layout like the one shown in figure A7. It needs to be noted that I have added a so-called Chroma Key filter to my face cam, which, together with a green screen in my home office, allows me to blend in with the background. This is not a necessity, however, and is outside the scope of this article.

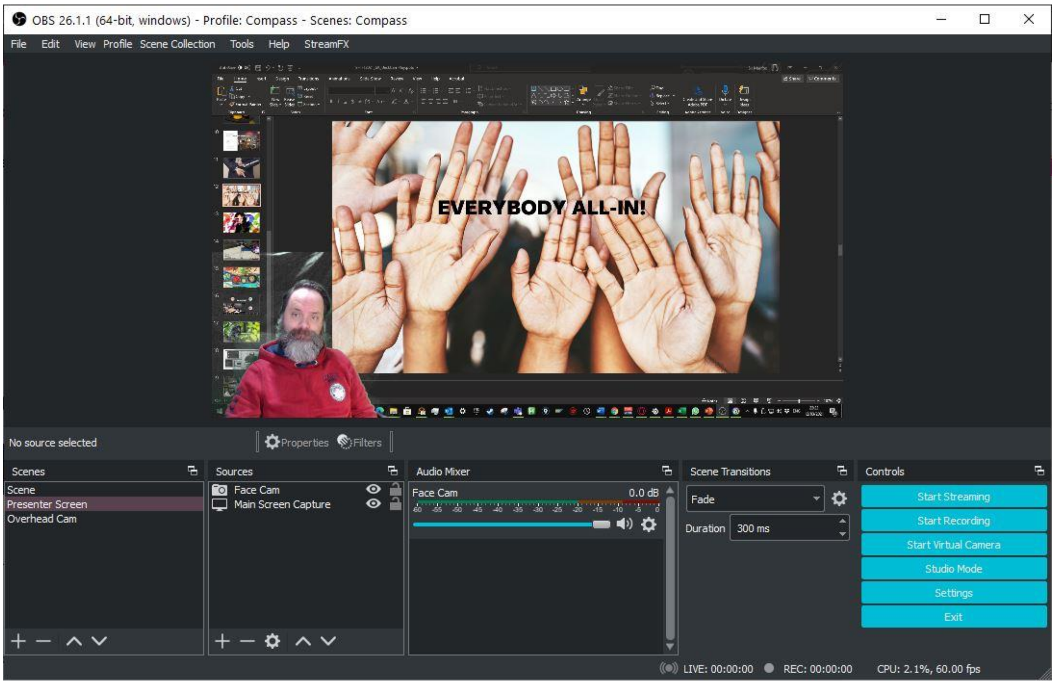


Figure A7. Talking head feature over main screen

The layout of the Overhead Cam scene can be designed in a similar fashion and requires another Video Capture Device source. A second webcam mounted above the desk (in my case) or above any other suitable surface will achieve that effect. However, as the first part of figure A8 below shows, this looks off in my case owing to the way the overhead cam is mounted on a shelf atop my desk. To achieve that 'over-the-shoulder' view rather than the 'from-the-front-and-top' one, the view has to be modified ('transformed') by flipping the image vertically and horizontally.

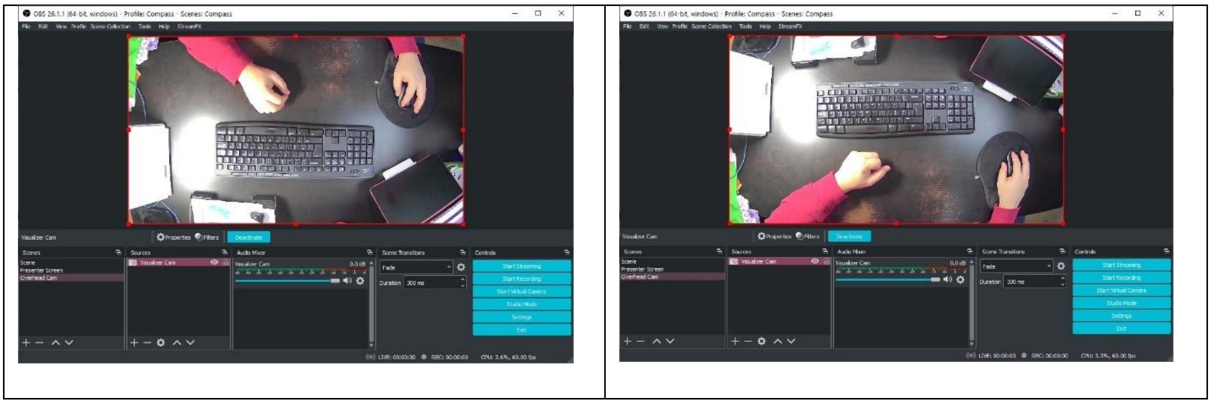


Figure A8. Initial overhead view and flipped view for the over-the-shoulder effect

Now we have two scenes ready to go! We can switch between a shared-screen with the presenter as a talking-head in the bottom corner to an overhead view on the desk to perform role-playing and games and using props, with the scenes seamlessly transitioning back and forth between them. It would be a simple task now to put another talking head into the overhead scene, resize it, move it to a different place on the screen – with OBS, this is a quick task once the respective screen layout has been decided on.

Table A3 summarises the scenes, intended layout and the sources to be added for the purposes of enabling a shared screen and an on-desk display via overhead cam:

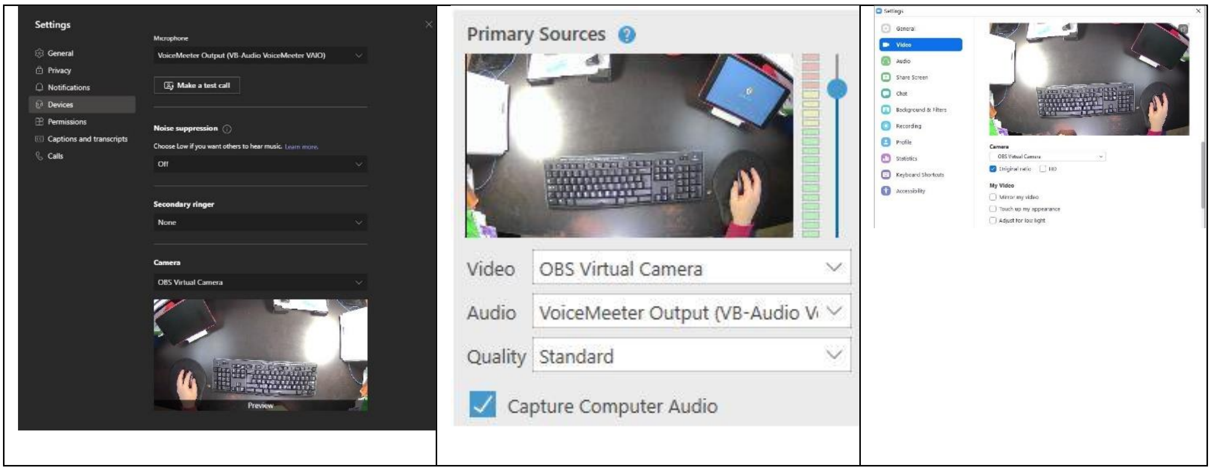
Title	**Intended design/layout**	**Sources**
Presenter Screen	Shared screen with a talking head in the bottom left corner	Face cam (video capture device) Main Screen Capture (
Overhead Cam	Visualizer	Overhead Cam (video capture device)

Table A3. Summary of scenes and sources created

For this simple set-up, it is not necessary to define audio capture devices (like a microphone) because of its intended use as virtual cam feed into MS Teams, Zoom, or Panopto. Any audio would be captured by the video-conferencing software directly. Therefore, the Audio Mixer component that also visible in OBS, used to set the volume of individual audio sources, can be safely ignored at this point and for the purposes of this paper.

5. Controls – virtual camera

The Controls component enables various pathways to output the scenes created. They can be either directly live-streamed to platforms like YouTube or Twitch (given the correct streaming settings), recorded and saved to the computer or output as virtual cam. It is also possible to live-stream, record and output as virtual cam at the same time, or any combination thereof. However, this would put added strain on the CPU. To output our scenes to MS Teams, Zoom, or Panopto, we will select ‘Start Virtual Camera’. The feed from OBS is now usable in any video-conferencing software that allows use of a webcam in their corresponding settings (figure A9).



Microphone: VoiceMeeter Output (VB-Audio VoiceMeeter VAIO)

Noise suppression: Off

Secondary ringer: None

Camera: OBS Virtual Camera

Video: OBS Virtual Camera

Audio: VoiceMeeter Output (VB-Audio V

Quality: Standard

☑ Capture Computer Audio

Camera: OBS Virtual Camera

☑ Original ratio ☐ HD

My Video

☐ Mirror my video

☐ Touch up my appearance

☐ Adjust for low light

Figure A9. Using the OBS virtual camera in MS Teams, Panopto and Zoom

Switching between scenes can either be done in OBS itself by selecting the desired scene, via a keyboard hotkey shortcut that can be defined by the user (this is possible for any task or command in OBS) or via a remote device. These devices can be physical ones (for example, the Elgato StreamDeck (https://www.elgato.com/en/stream-deck) or, if those are prohibitively expensive, via apps such as TouchPortal (https://www.touch-portal.com/) or Deckboard (https://deckboard.app/).

The possible applications of OBS as a virtual cam are not limited to MS Teams, Zoom, or Panopto. Any application that allows use of a video feed can access the OBS Virtual Camera instead of an in-built one or external one.