OBS SHARE AND MULTIVIEW: TWO METHODS FOR SHARING STUDENT WORK IN DISTANT TEACHING

Jun Iio
Faculty of Global-Informatics, Chuo University
1-18 Ichigaya-Tamachi, Shinjuku-ku, Tokyo, 162-8478, Japan

ABSTRACT
The COVID-19 pandemic has forced us to conduct online lectures at universities. Regarding ordinal courses in which teachers give their lectures and students simply listen, migrating online took place easily in a relatively straightforward way with the help of online meeting tools, such as Zoom, Webex, Google Meet, and Microsoft Teams. However, we faced some difficulties in the online migration of practical training courses. Particularly, in software development courses, a teacher must be able to monitor student learning progress to help them correct errors, check student code, provide advice, etc. This paper presents two proposals for sharing student progress in distant teaching for software development courses.

KEYWORDS
Real-Time Sharing, Student Desktop, OBS Studio, Google Colaboratory

1. INTRODUCTION
Many university teachers worldwide have been forced to provide lectures through the Internet in 2020 because of the COVID-19 pandemic. There is no exception for teaching computer science classes (Brooks et al., 2021). Fortunately, online meeting tools, such as Zoom, Webex, Google Meet, and Microsoft Teams, have become widespread (Singh and Awasthi, 2020), and students have become accustomed to using them. Additionally, various tips shared among teachers (Sandars et al., 2020) have helped them conduct distant teaching. Overall, it is relatively easy to migrate lectures online for regular classes, where teachers give lectures and students simply listen to them.

However, classes in universities are not limited to lectures. There are several difficulties in providing practical training courses online (Liu, W. et al., 2020). Particularly, for computer science classes, teaching programming is accompanied by practical training, such as writing appropriate code and testing software (Liu, A., 2020). In many cases, writing codes is necessary. Therefore, transitioning classes from offline to online for these types of courses is relatively complicated.

To conduct online practical training for writing adequate code or testing software, a teacher must monitor students’ progress in their learning. This paper proposes two typical and straightforward methods that can aid in distant teaching, namely, sharing student desktops or notebooks, which are representative of student progress, among teachers and students.

2. PROPOSED METHODS
This section proposes two methods for real-time sharing of student working conditions. One method uses OBS Studio¹ and Zoom², and the other uses Google Colaboratory (hereinafter referred to as Colab)³.

¹ https://obsproject.com/
² https://zoom.us/
³ https://colab.research.google.com/
2.1 OBS Share: Sharing Student Desktops

It would be helpful to use multiple monitors for student desktops when teaching for software development education is conducted. Various systems for sharing desktops have been proposed (Ohshima et al., 2017), and there is a simple method to realize desktop sharing using OBS Studio. OBS Studio is an open-source broadcasting software package, and many teachers utilize it to conduct online classes (Rosenthal and Walker, 2020; Kristandl, 2020).

The combination of OBS Studio with online meeting tools allows classroom participants to share their desktops in real time (see Figure 1).

![Diagram of OBS Studio and Zoom integration](image)

**Figure 1.** Overview of using OBS Studio to share student desktops

On November 22, 2021, eight students and the author attempted to set up a distant teaching environment using OBS Studio and Zoom\(^4\). Installing OBS Studio seemed to be easy for the students and they did not encounter any significant troubles. Figure 2 shows the teacher’s desktop screen where the teacher monitors students desktop using Zoom’s gallery view mode.

![Teacher monitoring student desktops](image)

**Figure 2.** The teacher can monitor student desktops using Zoom’s gallery view mode in real time

---

\(^4\) Zoom can be replaced with other online meeting software.
The preparation steps can be summarized as follows:
1. Let every student install OBS Studio and start the program.
2. Let OBS capture each student’s desktop by selecting capture screen as the source and begin sending it using a virtual camera by selecting start virtual camera from the control section.
3. Start Zoom and connect to the meeting.
4. Change Zoom’s camera source setting to the virtual camera provided by OBS.

These steps allow students to share their desktop screen to Zoom, enabling the teacher to monitor what the students are doing in real time. During preparation, some students using Mac encountered minor issues, such as being unable to capture desktops or being unable to find OBS’s virtual camera in Zoom’s camera section.

Several significant issues encountered in configuring the described environment and critical points for troubleshooting are outlined below:
- Selecting the option of “use as a virtual camera” in the configuration of OBS on its first use.
- Providing permission to access screen capture and other devices in the security configuration of the control panel.
- Starting Zoom after completing OBS configuration. Zoom will not find the camera provided by OBS if Zoom is started before OBS configuration is completed.
- Turning off Zoom’s virtual background.
- After all configuration steps are completed, the windows of OBS and Zoom should be minimized.

2.2 MultiView: Sharing Student Notebooks

The second option is to share student notebooks using software such as Jupyter Notebook\(^5\) and Colab. However, this option has limitations in terms of language restrictions. Specifically, the notebook environment can host only Python and several other programming languages.

For example, the case using Colab is illustrated here. Colab is a cloud service, and it runs on the cloud computing environment. Therefore, it should be suitable if the client machine does not have significant computational power. Although the setup of Colab also requires several steps, it is easier than that of OBS and Zoom.

\[^5\]https://jupyter.org/
The essential component of this option is to share student Colab notebooks with the teacher. This allows the teacher to check student progress at any time. Additionally, the teacher can communicate with students by commenting on or modifying their code.

MultiView was developed to capture a complete view of all student progress, rather than using the gallery view in Zoom with OBS. This is a simple program implemented using PHP and JavaScript. The source code for MultiView is available at https://github.com/iiojun/MultiView.

Figure 3 presents a screenshot of the MultiView setup page. The MultiView system provides two setup options: one-by-one configuration and semi-automatic setup by uploading a configuration file.

2.2.1 One-by-one Configuration

The first option is the one-by-one configuration method. Initially, the teacher enters the student number and clicks the submit button on MultiView’s configuration page (see Section A. in Figure 3). Then, a page that contains a small section with the entered student number appears (see Figure 4 (A)). Each page is embedded as an inline frame, which is shrunk to half size.

Each block contains three buttons: load, reset, and open. In the initial stage, only the load button is enabled. Each page also has text fields to fill in the URL and name. After specifying the URL and name, a half-size webpage appears on the page section by clicking the load button.

Figure 4 (B) shows that the teacher has configured two pages: Joe’s page as Student 1 and Bob’s page as Student 2. These pages are embedded in the main page. These pages are synchronous with student pages. Although there is a subtle time lag, the teacher can confirm the actual status of student progress.

Specifying a student’s name is not mandatory. If the name field is empty, only the placeholder “Student #” is used. When the page is embedded, the load button is disabled, while the reset and open buttons are enabled. Pressing the reset button removes the embedded page and reverts it to the initial state. The open button opens the page in a new window in the browser. If a shrunked page is not suitable for teaching, the teacher can access a regular independent page by clicking this button.
2.2.2 Bulk Configuration

To configure pages semi-automatically, MultiView has another option for embedding student Colab pages. If you select this option, you only need to prepare a CSV file that includes student names and the URLs of their notebooks in pairs on one line for each student, which are separated by commas, and upload it to the server.

By specifying a CSV file using the file chooser on MultiView’s configuration page and uploading it (see Section B. in Figure 3), all web pages corresponding to the listed names will appear embedded on the main page.

Note that a standard web browser cannot accommodate MultiView functions for security reasons (Huang et al., 2012). The user must implement the “Ignore X-Frame headers” addon6 for Google Chrome to enable MultiView’s functionality.

3. CONCLUSIONS AND FUTURE WORK

In this paper, two methods were proposed to share multiple student desktops or notebooks simultaneously. OBS sharing, which is realized using OBS Studio and Zoom, is independent of the choice of programming language. Therefore, it would be helpful not only for computer science courses but also for other fields. MultiView, which shares student notebooks from Colab on a web page, is suitable for programming education. It is highly dependent on the synchronization ability of the Colab page sharing function.

This paper discussed how to implement the two methods mentioned above. These methods should be evaluated and compared in actual distant teaching scenarios. Experiments and evaluations for this purpose remain for future work. In addition, there are some other services to share the remote desktop. Comparison to such services should be considered.

REFERENCES


Kristandl, G., 2020. “All the world’s a stage”- the Open Broadcaster Software (OBS) assembling technology to overcome restrictions in online teaching. Compass: The Journal of Learning and Teaching at the University of Greenwich, Vol. 14, No. 2, pp.1–16.


6 https://chrome.google.com/webstore/detail/ignore-x-frame-headers/gleekbfjekinecknkbafmnkhkpodhe