Biomedical Engineering Education (© 2021)
https://doi.org/10.1007/s43683-020-00044-7

BMES BIOMEDICAL ENGINEERING SOCIETY

Teaching Tips - Special Issue (COVID)

Developing Effective Screencast Modules for Teaching Computational Techniques in Remote Modalities

DEBANJAN MUKHERJEE

Department of Mechanical Engineering, University of Colorado Boulder, Boulder, CO, USA

(Received 30 June 2020; accepted 25 November 2020)

Abstract—Here we describe a systematic approach towards creating effective screencast based modules for teaching computational techniques in remote and online modalities. We adopted a multi-stage approach to create screencast videos that replaced in-person demos and active learning content in a finite element analysis based class. The stages include systematic preparation of video data and script; production stage, for recording and editing of captured video and audio; and post-production stage, for uploading generated media files into our learning management system. Modules were paired with assignments, thereby enhancing student learning and enabling assessment of module content efficacy. Our approach and technology received highly positive reception from students. Students also successfully navigated all associated assignments and final course project, which builds upon the content addressed in the modules. We identified several avenues for improvement in continued future offerings of such modules. We have outlined our design experience and student reception of screencast based modules for creating engaging learning content in remote teaching modalities. The description has been presented in form of teaching tips for other educators to adopt for their teaching needs.

Keywords—Screencast, Hands-on demo, Active learning, Remote teaching, Student engagement.

CHALLENGE STATEMENT

The ongoing Covid-19 pandemic has presented unprecedented challenges for educators worldwide in terms of delivering quality education to students. Remote and hybrid teaching modalities have gained a newfound prominence amidst these challenges. However, delivering all forms of content effectively in remote teaching continues to pose difficulties, particularly considering that remote teaching has entirely replaced any in-person education during the pandemic. Active learning, hands-on education, and project-based learning have increasingly gained eminence as valuable avenues to enhance student learning in engineering education.[8] Yet, conducting active and hands-on teaching in an entirely remote manner can be a complex task for educators, especially since in-person interactions are substantially impacted in a virtual or remote classroom.[3,6]

In the spring semester of 2020, amidst the Covid-19 shutdown and subsequent transition to remote learning, we faced similar challenges in teaching a technical elective course comprising senior undergraduate and entry-level graduate students of engineering at the University of Colorado Boulder. The course teaches computational techniques and analysis using the finite element method—a commonly used technique in biomechanics, medical device design, image-processing. The course is designed based on multiple active learning components, which involves in-person hands-on demos and activities that link mathematical aspects of the finite element method with computational implementation and simulation of a given problem. Course assignments are linked to these hands-on components to scaffold student learning and evaluate student progress. However, with a remote-only mode, we faced the challenge of rapidly converting all active hands-on modules into an equally effective online format. The modules required re-designing to factor in the lack of face-time and in-person interactions with the instructor and peers. Running these modules live over the internet was not an option as many students

Address correspondence to Debanjan Mukherjee, Department of Mechanical Engineering, University of Colorado Boulder, Boulder, CO, USA. Electronic mail: debanjan@colorado.edu

Published online: 04 January 2021

reported having unstable internet connection, and preferred downloading and reviewing recorded modules. Here, we describe our approach towards addressing these challenges by developing alternative online hands-on modules based on screencast technology.

NOVEL INITIATIVE

We begin describing our approach by providing further details regarding the course for appropriate context. Our course content is structured around three main aspects: (a) understanding fundamental concepts in computational analysis (*e.g.*, discretization, meshing, error, stability, convergence, boundary conditions); (b) learning the finite element method (*e.g.* weak formulations, matrix equations, h-p refinement, stabilization); and (c) building simulation workflows (*e.g.* pre-processing, simulation, post-processing, visualization, verification). The course is designed based entirely on open-source software libraries and applications, thereby avoiding significant licensing costs and hardware or system requirements for students. Students use the Python programming language for the computational work. In addition to lecture materials, the course includes multiple hands-on in-class active learning demos, and flipped classroom modules to enhance students' understanding of key underlying concepts. These include walking through meshing operations, demos of pre-processing and post-processing operations, programming examples, and demos of finite element method implementation for a given problem. Students are required to complete two assignments on mathematical techniques, five assignments on modeling and simulation work, one midterm examination, and a final group project culminating in a conference session style minisymposium open to faculty and students from outside the class. The five simulation assignments and the project (around 65% of total course grades) are directly based on concepts and techniques disseminated using the hands-on modules.

Within this course structure, we re-designed all the existing hands-on demos using screencast videos for transition to remote teaching. Screencasts have been identified as a valuable avenue for introducing active learning and flipped classroom content in a variety of settings.[5,7,11] Here, we adopted screencast technology for teaching core computational analysis topics to the students. Our screencast modules were developed using the software package Camtasia (TechSmith Corporation, USA).[1] While this is a paid software package, there are several equally useful free alternatives. Audio recording of instructor narration was handled using a small podcast microphone, assuring clean audio quality for the entire duration of the narration. The total expenses for licensing, microphone, and setup were around $200.00. All screencast production work was completed on a standard laptop by the instructor.

Each screencast development involved a preparation phase and a production phase. Various steps in this process are illustrated using an actual example of a screencast from the class in Fig. 1. The preparation phase involved three steps: (a) creating data and files for the video; (b) creating a script for the video; and (c) conducting a rehearsal dry-run. For step (a), materials associated with the planned video, including code samples, data files, steps and/or charts for the workflow, were prepared and clearly organized. Examples of this include: geometry handling and mesh generation operation workflow; scripts and codes for simulations; code templates to be filled in during the demo; and workflow for post-processing data. For step (b), a short script was generated for each video to guide and plan out on-screen actions and audio narration. The script included step-by-step decomposition of the task or demo, including file transitions, mouse actions, typing, and navigation between windows (see for example, Fig. 1). This was accompanied by points for narration, and integration of components from the lecture materials into the videos. Verbal cues were included for taking pauses, revisiting concepts from lecture, and reviewing actions to ensure all steps have been properly followed. For step (c), we conducted a rehearsal run-through of the entire script and video, to identify any missing aspects that would then be included, as well as to check overall demo run-time. For this class, the screencast videos had average run-time around 5 min, with a couple longer ones running to 25 min (with multiple pauses and review cues). After preparation, and in the production phase, the complete video along with narration was recorded in a single pass on Camtasia. Simultaneous audio recording and screen-capture were specifically chosen to avoid complications associated with overlaying and syncing post-recorded audio with the screen-capture. The software's built-in annotation tools were used to edit in any annotations on the video itself (Fig. 1, panel b). This included highlighting sections of a simulation or code; highlighting features of a geometry or mesh; or simply indicating points in the video when students need to pause and review. The planned incorporation of verbal cues, highlighting, and annotation in steps b and c, helped introduce signaling and segmenting in the videos for managing cognitive load.[5] Specifically, these helped enhance the intrinsic and germane cognitive load (per instruction design theory),[12,13] by highlighting key information and elucidating the structure and interconnection of this information in

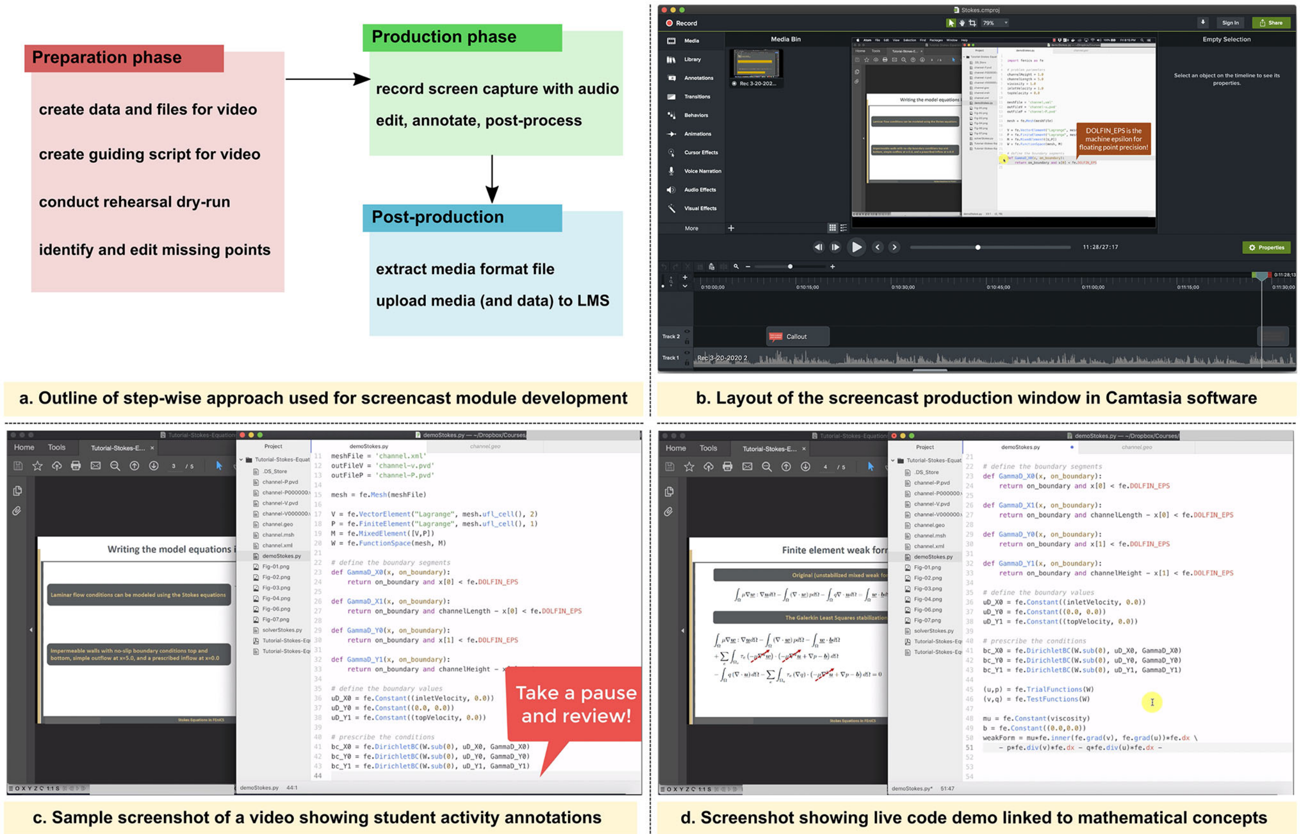

FIGURE 1. An overview illustration of screencast modules as described in this work. Panel a presents a schematic outline of the steps involved in developing each screencast module. Panel b presents an illustration of the interface used for recording and editing the video and audio. Panel c presents a sample screenshot from one of the modules, showing how multiple windows are utilized, and how student activity annotations are included. Panel d presents another screenshot from the same module, illustrating a live code demo, and showing how mathematical concepts discussed in course material are integrated within the demo (mouse highlight shown in yellow). Panels b–d are from an actual demo prepared for the course. All course identifying information has been edited out.

the videos. The final step in the production phase was to export the screencast as a media file. The mp4 media format was chosen to ensure portability across operating systems. The media file, along with associated data files, was then uploaded onto the course learning management system (LMS).

Assignments were designed such that the underlying learning outcomes were addressed by one (or more) such screencast or active learning modules. Assessment of assignments, as well as the project, was based on rubrics devised in accordance with these learning outcomes. This was communicated clearly to the students, and learning outcomes remained unchanged from prior (pre-pandemic) editions of the course. We share two specific examples here. In our first example, students were provided a screencast module on conducting low Reynolds number flow simulations (a common application in many biofluids problems). The screencast walked through one set of inputs, geometry, and boundary conditions; and in an associated assignment students had to conduct simulations on another set of inputs, geometry, and boundary conditions with some code modifications. In our second example, a set of screencast modules was provided outlining various meshing operations and mesh quality analysis. In an associated assignment, students were required to conduct a combination of these operations using a new set of geometry data. This paired assessment design motivated students to review and complete the screencast modules. Subsequent class performance in assignments as well as the course project, directly based on these computational tasks, served as one potential indicator for the success of our approach.

REFLECTION

Owing to the rapid transition to remote teaching mid-semester, we were unable to set up targeted evidence collection mechanisms for evaluating our approach. However, in the university wide course questionnaire administered at the end of the semester, there was a specific question addressing this particular aspect. The question asked students to rate whether the course ''Effectively used available technology to enhance learning'' on a scale of 1–5 (1 indicating ''Hardly ever'' and 5 indicating ''Almost Always''). We received an average rating of 4.87 (standard deviation of 0.34) based on a 60 percent response rate, indicating clearly that students overall had a highly positive perception of our technology based screencast and other active learning modules. Course questionnaire ratings were accompanied by anonymized comments from students, which further supported student enthusiasm and positive perception regarding our approach. These observations correlated well with instructor assessment of student performance in the associated assignments and final course project. Despite the various impacts of the pandemic and lockdown measures, and transition to remote teaching, all students completed all of these assignments in full, and presented their final projects on the scheduled presentation date. All projects involved finite element simulations on applications significantly different and substantially broader in scope when compared to assignments. Specifically, between a previous edition (without screencast) and spring 2020 edition (with screencast), course structure and learning outcomes remained same, and assignments targeted the same learning outcomes for assessment. Hence, monitoring general class performance on assignments and the final project, and comparing performance trends between prior and current offerings provided an indication of whether the screencast approaches effectively substituted hands-on modules in in-person instruction modes.

Our approach to design these learning modules had two core objectives. First, we ensured that the screencast is not simply a reproduction or alternative presentation of some course content, but is an engaging and interactive experience enabling students to actively work with the material.[5] Second, by pairing the assignments with these active learning modules, we ensured that students are able to use the modules as scaffolds to boost their own learning experience.[9] This latter aspect is essential in enabling students to transcend from simply learning and reproducing aspects of the method, to applying these aspects to solve actual problems—that is, from lower to higher cognitive learning levels.[4,10] We consider these two factors as key determinants for the successful outcomes of our effort so far. Additionally, we note that the design principles and approach outlined here are not limited to any particular computational technique or software, and can easily be adopted for teaching other techniques (*e.g.* finite difference/volume methods, molecular dynamics, data analytics) or using other software tools (*e.g.*, SimScale, OpenFOAM, Matlab) relevant for biomedical engineering curriculum. Furthermore, we note that the screencast modules did not replace lectures during remote instruction. Questions and discussions during lecture and office hours (conducted online), as well as dedicated online discussion forums monitored by the instructor, provided multiple avenues for student-instructor interactions pertaining the screencast modules—where students communicated their questions, clarifications, experience, and feedback. Active student engagement with the material beyond regular lectures, followed by student-instructor interactions, thus rendered a flipped-classroom flavor to these modules, even though the entire class was not designed as a flipped-classroom. Finally, the alternate avenue to engage with the material using multimedia content enhanced course content accessibility course content especially for remote teaching.

Through these experiences and student feedback, we have identified a few specific areas for improvement in future implementations. The first aspect is the production quality of the screencasts. The transition to remote teaching gave us limited lead time to polish our production quality. However, with a first iteration of all the material now prepared, we plan on specifically improving production video and audio quality, background noise removal and audio post-processing, and including closed captioning to make the content more accessible. A second area of improvement concerns collecting further longitudinal evidence on student experience and success. While we tracked the number of times a video had been watched (using the LMS), we could not track video analytics for each student. In future implementations we plan on tracking individual student video analytics, including self-reported usage statistics through a survey form, and correlating the responses with student grades and success in the associated assignments as well as the final course project. Finally, student experience and feedback revealed that lack of in-person interactions with peers was viewed as a major deficiency in remote learning environments. To address this, we plan on integrating aspects of collaborative student work[2] by enabling collaborative editing of codes and scripts through web-based platforms like JupyterHub and PythonAnywhere.

In conclusion, here we have described an approach for systematically using screencast based modules as a means to present interactive hands-on demos of computational analysis techniques in remote teaching modalities. These modules received positive student response and enabled student success, can be used broadly for a variety of course content (by other educators), and will continue to be offered in future editions of the course.

AUTHOR CONTRIBUTIONS

DM was the instructor responsible for creating the screencast content, and alsodeveloping the manuscript

DATA AVAILABILITY

Supplementary material included: sample screenshot for producing a screencast module referred to in the manuscript.

REFERENCES

[1] Barkley EF, Cross KP, Major CH. Collaborative learning techniques: a handbook for college faculty. New York: Wiley; 2014.

[2] Bernard RM, Abrami PC, Borokhovski E, Wade CA, Tamim RM, Surkes MA, Bethel EC. A meta-analysis of three types of interaction treatments in distance education. Rev Educ Res. 2009;79(3):1243–89.

[3] Bloom B, Engelhart M, Furst E, Hill W, Krathwohl D. Taxonomy of educational objectives: the classification of educational goals. Handbook 1: cognitive domain. New York: Longman; 1956.

[4] Brame CJ. Effective educational videos: principles and guidelines for maximizing student learning from video content. CBE—Life Sci Educ. 2016;15(4):es6.

[5] Camtasia by TechSmith. https://www.techsmith.com/. 2020.

[6] Darling-Hammond L, Flook L, Cook-Harvey C, Barron B, Osher D. Implications for educational practice of the science of learning and development. Appl Dev Sci. 2020;24(2):97–140.

[7] De Grazia JL, Falconer JL, Nicodemus G, Medlin W. Incorporating screencasts into chemical engineering courses. In: 2012 ASEE Annual Conference & Exposition; 2012. pp. 25–762.

[8] Freeman S, Eddy SL, McDonough M, Smith MK, Okoroafor N, Jordt H, Wenderoth MP. Active learning increases student performance in science, engineering, and mathematics. Proc Natl Acad Sci. 2014;111(23):8410–5.

[9] Kerr B. The flipped classroom in engineering education: a survey of the research. In: 2015 International Conference on Interactive Collaborative Learning (ICL), p. 815–818. IEEE; 2015.

[10] Krathwohl DR. A revision of bloom's taxonomy: an overview. Theory Pract. 2002;41(4):212–8.

[11] Morris C, Chikwa G. Screencasts: how effective are they and how do students engage with them? Active Learn Higher Educ. 2014;15(1):25–37.

[12] Sweller J, Van Merrienboer JJ, Paas FG. Cognitive architecture and instructional design. Educ Psychol Rev. 1998;10(3):251–96.

[13] Sweller J, van Merriënboer J.J, Paas F. Cognitive architecture and instructional design: 20 years later. Educ Psychol Rev. 2019;1–32.