Teaching Visualization for Large and Diverse Classes on Campus and Online

Johanna Beyer *
Harvard SEAS

Hendrik Strobelt
Harvard SEAS

Michael Oppermann
University of Vienna

Louis Deslauriers
Harvard FAS

Hanspeter Pfister
Harvard SEAS

Abstract

We present how principles and methods from active learning can be integrated and applied to a large undergraduate visualization course for students with a wide range of different educational backgrounds. The course covers the main principles of visualization (theory), showcases examples of visualizations and the main visualization types (visualization literacy), and covers visual design and critical discussions of visualization designs (design). In a final project, students have to design and develop their own web-based interactive visualizations using Javascript and D3 (programming). We describe how we have adjusted our previous course to include state-of-the art principles of active learning, with the goal to (a) increase student engagement, and (b) increase knowledge transfer. Quantitative and qualitative student feedback, as well as Q scores show successful engagement of students. The results of the midterm exam as well as the final projects indicate that we were successful at transferring knowledge about visualization theory, visualization literacy, design and programming.

1 Introduction

The concept of active learning has gained a lot of attention in recent years for teaching university-level courses [10]. The goal of active learning is to encourage students to not just listen to a lecture, but to also be actively engaged in higher order thinking tasks, such as analysis, synthesis, and evaluation [2]. In this paper we demonstrate how to apply concepts from active learning to Harvard's visualization course. Course enrollment is typically 200-250 students, approximately half of which are distant learning (i.e., online) students. The course is designed for undergraduate students and attracts students from a wide variety of different fields and majors. These factors greatly influence the curriculum and how we, the teaching stuff, have structured the overall course. The experiences we have gained in restructuring and teaching the active learning version of this course (as well as the pitfalls we have encountered) are summarized in this paper, in the hope to make it easier for other teaching staff to adapt their course to the concepts of active learning.

The visualization course at Harvard has several attributes that sets it apart from other data visualization courses:

- **Large class size.** In the last years, enrollment numbers have been consistently between 200 and 250 students per semester. This makes our course one of the largest undergraduate courses about visualization worldwide.
- **Undergraduate course.** Our course is designed for undergraduate students. This implies that students have significantly less background in programming, computer graphics, HCI, and other related fields. Therefore, we have to start at a slower pace and gradually bring students up to speed (e.g., with general programming), before we can focus on visualization topics that require more background knowledge.
- **Online and on-campus students.** Our course is taught to distant learners (online) and on-campus students simultaneously. These two student groups have very different backgrounds, expectations, and needs. Principles of active learning are particularly difficult to realize for online students, and often require special software and an extra set of instructions.
- **Students with a diverse set of majors.** Our visualization course aims to reach not only computer science majors but also students from fields such as engineering, natural sciences, journalism, architecture, or international relations. The diversity of our student body can be a challenge, but it also allows for group projects where students with different backgrounds can learn from each other.

We use Harvard's visualization course as an example of how a course concept and curriculum of a large and diverse undergraduate course can be updated to include principles from active learning [9]. Based on initial quantitative and qualitative feedback, our changes have increased student engagement (class attendance, participation, completed quizzes) and have improved the student's reception of the course (weekly online feedback and Q scores). At the same time, we have improved knowledge transfer (midterm exam) and succeeded in teaching technical skills in Javascript, D3, and visualization design (midterm project, final project).

The course material of one week (including lecture, lab, homework, pre-class reading assignments, quizzes, 1-minute papers, and the studio) is available online[1]. Further information about the course is available on the related webpage[2].

2 Course Design

In the initial phase of our course re-design a teaching expert guided us through the essential principles and pitfalls of active learning. To re-design the course, we first outlined our general teaching goals before creating the actual teaching content. We have also adjusted the roles of our teaching staff.

2.1 Teaching Goals

Implementing active learning as teaching method requires lecturers to revise the traditional course material. Lectures must be shortened to make room for in-class activities and discussions. Students should be "primed" by short reading assignments prior to lecture so that basic terms are already known at lecture time, which in turn frees up time to do higher-level analysis and thinking tasks. Therefore, it is vital to start by identifying the core principles that should be learned in a module and then structure lectures and assignments accordingly.

For our course, we have identified four high-level teaching goals:

- **Visualization Principles.** Our course covers basic knowledge about the human perceptual system and cognitive biology,

*email address for all tutors: team@cs171.org

[1] http://www.cs171.org/2016/assets/material/paper-attachment.zip

[2] http://www.cs171.org

Course Element	Teaching Goal	Weekly Workload	Grading Weight
Lectures	Visualization Principles, Visualization Exploration, Design Skills	1.5h	5%
Labs	Programming and System Building Skills	1.5h	5%
Project	Team Work, Design, Programming, System Building	6h (weeks 8-14)	40%
Studios	Design Skills, Visualization Principles, Group Discussions	1h	5%
Homeworks	Design (and Design Critiques), Programming Skills	5h (weeks 1-7)	25%
Midterm	Design, Programming, System Building	1.5h (in-class exam) 5-10h (take-home project)	20%

Table 1: Course elements, their specific teaching goals, weekly workload, and grading weight.

leading up to works like Bertin's marks and channel [1] and McGills ranking of visual variables [4]. Discussions about visual design and design critiques are core elements to introduce a vocabulary for students to discuss visualizations. We also cover core concepts of interaction and introduce the process of how to conduct a visualization project using data and task abstraction, prototyping, and evaluation [11].

- **Visualization Exploration.** We showcase examples of visualizations that form the standards for visualization literacy - including bar charts, parallel coordinates [7], or treemaps [8]. Similar to museum excursions in the humanities, we highlight prominent examples of the design space that act as seeds for further exploration.
- **Design, Programming and System Building Skills** are essential to enable students to implement their design ideas into interactive prototypes. Our class teaches pen and paper prototyping for designing visualizations, interaction storyboards for designing interactive applications, and Javascript and D3.js [3] for all programming assignments. Since this course is designed for undergraduate students and non computer science majors, we aim to reduce and limit the amount of technical complexity for students. Therefore, we have decided to teach only a single visualization framework and programming language (i.e., D3 and Javascript).
- **Team Working Skills** are a fundamental goal in our teaching approach. We think that training students on team work equips them for real-world scenarios [6]. We actively encourage students to work with and learn from each other, which is especially beneficial in classes where students come from many different disciplines, each with their own visualization-related challenges and interests.

The teaching goals are mapped to specific learning objectives for each individual lesson and are directly reflected in the course structure. *Visualization Principles* and *Visualization Exploration*, as well as basic *Design Skills* are taught in active lectures. *Programming and System Building skills* are covered by in-class labs, and *Team Working Skills* are trained in the project phase. Table 1 provides an overview of the different pedagogical elements in our class, and their main teaching goals.

2.2 Course development

We started our course re-design with a training session of a teaching expert, who gave us an introduction to the principles of active learning, guidelines to follow, and pitfalls to avoid. Following these guidelines, we then prepared an example teaching unit (i.e., lecture and lab) and presented it to him. After another iteration with the teaching expert we went ahead and restructured the entire course.

To prepare a teaching unit, we start by defining the learning objectives, and subsequently design the lecture and lecture activities. After that, we design the programming lab and homework assignment, whenever possible directly relating them to lecture content and activities. Finally, we define the content of the studio, based on the current lecture, lab and homework.

2.3 Teaching staff

Transforming an existing course into an active-learning based course is time intensive. Four lecturers and teaching assistants started on the course re-design a semester before the start of the course.

During the actual course, we have a ratio of one teaching assistant per 15 students. This ensures that teaching assistants are present for lecture activities, can help in the lab, hold studios and office hours and complete weekly homework grading. Communications with teaching assistants is done at weekly meetings and over the messaging platform Slack [3].

Students are assigned to teaching assistants based on their studio. Each teaching assistant always grades the students from his/her studio, and is also the first point of contact for their students. This assignment is critical for online students, who can not just easily talk to the course staff after lecture.

3 Course Elements

Our course consists of several different *course elements*, each with their own specific teaching goals. Different course elements vary in setup and participant size. In this section we provide details about each course element.

3.1 Lecture

Lectures are held for 90 minutes once a week in a large lecture hall for the entire class. In the first half of the semester, lectures focus on general *Visualization Principles*, leading to a midterm exam. Each lecture is made up of five to six sub-units. A sub-unit consists of a short introduction directly followed by a three to ten minute long student activity. The goal of an activity is to *prime* students for a certain topic, which is then subsequently discussed and elaborated. In many cases, the activities are consecutively building up on each other.

After the midterm, lectures focus on supporting students in their final projects. We do this by presenting examples of the visualization design space in *Visualization Exploration*. We devote one lecture to *Team Working Skills* and project planning. This includes writing a semi-structured team expectation agreement as "contract" within each team [12].

For each lecture, we have four teaching assistants present. They provide trays of material (paper, pens, scissors, post-its, printouts) for the activities of each lecture, observe activities, indicate completion, and make photos of student solutions. The photos are uploaded to a cloud service and directly displayed to the entire class in the subsequent discussion of an activity. One teaching assistant is responsible for keeping time, and notifying the lecturer if on-the fly time adjustments are necessary. The reason for this is that the length of activities should always be adjusted on-the-fly, depending on student engagement, or when interesting questions arise.

[3] www.slack.com

Online Students are able to watch the lecture in a video live stream or a recording. They have to complete activities on their own, or in self-organized student groups and submit the completed activities together with their homework assignments.

3.2 Lab

Labs focus on the teaching goal *Programming and System Building Skills*. They are designed as self-guided programming worksheets that teach HTML, CSS, Javascript, and D3.js. Labs are designed to take 90-120 minutes and have to be submitted along with the homework. Labs count towards the students' participation grade.

Each lab is held in a classroom setting for the entire class, with four teaching assistants available for answering immediate questions. We give introductory remarks at the beginning of each lab to provide practical advice for solving the tasks and teach general coding principles. On demand, we interrupt coding to indicate commonly made mistakes or give additional helpful tips.

Online Students also submit their completed labs with their homeworks. They do not have the benefit of direct feedback or help during the completion of their lab. Instead they can use the course's online forum or online office hours for further assistance.

3.3 Homework Assignment and Studios

Homework assignments weave together concepts from lectures and labs. They consist of a design part and a programming part. Examples of design parts include critiquing a given visualization, designing and sketching a new visualization, or sketching a storytelling website or interaction storyboard. Programming parts re-iterate the concepts taught in labs and tie in with the content covered in the lectures whenever possible.

Students have to complete homework assignments on their own, working in groups is not allowed. However, students can seek individual help during office hours. Individual feedback for homework assignments is given within a week or two of submission. A more general discussion of the homework, common mistakes, and general coding guidelines, is done during studios.

Online Students have (asynchronous) discussions in social media platforms (i.e., Google Hangout and Piazza) led by their assigned teaching assistant.

3.4 Pre-class Reading and Quizzes, 1-minute Papers

To be able to use class time as efficiently as possible, we assign a short pre-class reading assignment for every lecture and lab. Pre-readings provide an introduction to the upcoming material, introduce new vocabulary, and prime students towards the topics that will be covered during class time. Pre-readings are available a couple of days prior to class and students have to submit a simple quiz about the reading before class time.

To ensure that students read the assigned texts, two principles should be followed:

1. Pre-Readings should not take longer than 20 minutes to read [5]. Often, this requires to limit texts to some specific paragraphs/chapters. If reading assignments are longer, students are more likely to stop doing their reading assignments over the course of the semester.

2. A short pre-class quiz for every reading assignment should check the factual knowledge that was covered in the assigned text. It is important that inference or reasoning is not tested in this quiz, it is only meant to check simple facts that were covered in the assigned text.

Students are asked to submit a 1-minute paper at the end of each lecture or lab. This feedback is essential to allow lecturers to monitor the learning process, clarify remaining questions, and improve the material. Our set of core questions are:

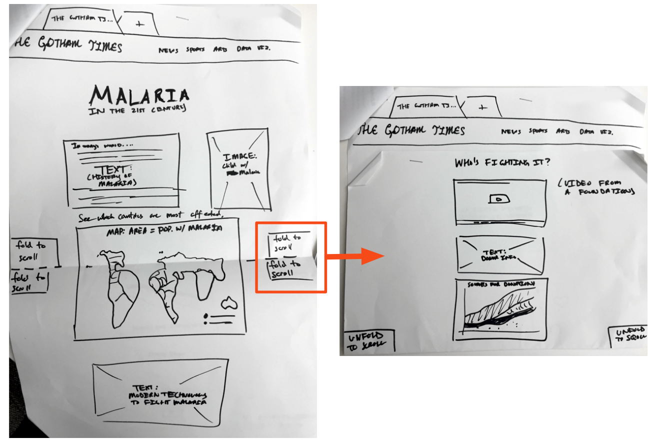

Figure 1: Example midterm exam. The student created a sketch of a visual storytelling webpage focusing on Malaria.

- What are the two [three, four, five] most significant [central, useful, meaningful, surprising, disturbing] things you have learned during this session?
- What question(s) remain uppermost in your mind?
- Is there anything you did not understand?
- Do you have any other comments about today's class?

We also regularly asked students on feedback about the perceived difficulty and length of labs and homework assignments.

For Online Students we used more flexible deadlines for the quizzes and 1-minute papers. To enable online students to complete most of their work on the weekends, their deadlines for all assignments was always Monday night.

3.5 Midterm

The goals of our midterm exam and midterm project are to (a) evaluate student progress; (b) prepare students for the final project; and (c) create an incentive for students to attend lectures regularly.

The midterm consists of two main parts: The first part is a 90 minute in-class exam, where *Visualization Principles* and basic design and sketching skills are tested. Students are given a real dataset and a corresponding visualization problem and are asked to come up with an appropriate visualization design (see Fig. 1). All theory questions are based on the given dataset and visualization problem.

The second part is a take-home midterm project where they have to implement an interactive website including several D3 visualizations for the given dataset. They can either decide to implement their own sketches from the midterm exam, or they can implement sketches we have defined for them.

We review the midterm project in class and also show some selected midterm projects during class time. This exposes students to the work of others and often serves as an eye-opening experience to them in what is already possible with their current skill set.

Online Students fill out their midterm exam in an online system that automatically limits their time to 90 minutes.

3.6 Final Project

For the final project, students have to combine all their acquired skills to create an interactive, innovative, effective visualization which tells a story about a data set of their choice. Teams of three students work on their final project over the course of five weeks.

In the initial phase, students write a project plan including a description of their data, domain questions, and initial sketches. These project plans are evaluated by teaching assistants for feasibility. In a second stage, students adjust and extend their plans to fulfil the course requirements of including multiple coordinated views and at least one innovative visual encoding.

During the project phase, lectures and labs focus on project execution and implementation. Teaching assistants are available for individual meetings with project teams. In the semi-final week, students present their preliminary projects to each other and use the feedback to refine their projects prior to their final submission. This presentation exposes students to the work of other groups, and we found this to increase the overall project quality.

All visualization projects are displayed at a school-wide design fair, where the best project and most popular project are awarded with a price. This increases visibility of our course and motivates students to work hard on their final projects (see hall of fame[4]).

Online Students are encouraged to build project teams with other online students, preferably in roughly the same time zone. Additionally, we gave them a short guideline on how to collaborate and (asynchronously) communicate with remote project partners with busy schedules.

4 Grading

Our grading scheme reflects our teaching goals. We want students to learn a mix of visualization theory, visualization design, and the skills to implement interactive web-based visualizations.

Our detailed grading scheme is shown in Table 1. We place most weight onto the final project and the midterm. Homework assignments are meant to let students deepen and practice their technical skills from the labs. Therefore, we emphasize overall understanding of the concepts of D3 over minute implementation details and grade each homework only on a scale from 1-4. Pre-quizzes for lectures and labs, as well as completion of the 1-minute papers all count towards a *particpation grade*. This encourages students to attend class, without us having to enforce a mandatory attendance.

Online Students are graded to the same standards as on-campus students. The only difference is in different deadlines for some assignments (i.e., pre-reading quizzes, 1-minute papers).

5 Software Tools

Managing a large class, and a combination of on-campus and online students can cause a significant administrative overhead. We have experimented with several different software-based solutions and have found the following tools to be very useful:

- **A messaging application for the course staff.** For large courses, e-mails can quickly get overwhelming. Therefore, we use Slack for structuring communication within the course staff.
- **A learning management system (LMS).** Depending on the specific platform a LMS allows teaching staff to administrate a course, track grades, updated the syllabus, and to deliver electronic content. For our course we use Canvas[5] to publish our course content, schedule studios, and publish and grade quizzes.
- **A discussion forum**. We encourage students to use our online discussion forum to post questions, discussion, or interesting links. This not only limits the amount of emails we get from students, but also usually results in faster answers for students. We currently use Piazza[6], but are looking for other open-source forums we could use.

[4]http://www.cs171.org/2016/fame/
[5]https://www.canvaslms.com/
[6]https://www.piazza.com

- **LMS for programming assignments**. We use a system that allows online submission, execution, and grading of programming assignments. We use Vocareum[7] which also offers a *homework gallery*, where students can look at and run each others visualizations and provide peer feedback.

The above tools are what we are currently using in our class, but we want to emphasize that any software packages with the same features could easily be used in a course like ours.

6 Lessons Learned and Conclusion

While re-designing the course, we have learned valuable lessons which are worth sharing:

- **Student feedback** provides a direct measure for monitoring the learning progress and for which teaching strategies work well. The direct and frequent feedback of the 1-minute papers at the end of each lecture and lab caused only a small overhead for students, but allowed us to quickly address open questions from the preceding week and to adjust our teaching accordingly. This resonated well with students, but to keep student engagement up, the course staff needs to explicitly address student feedback regularly.
- **Online students** participate in the course indirectly and asynchronously. Both factors contradict the idea of active learning. In this paper, we have highlighted some initial approaches we have implemented. However, it is not straight-forward to extend an active-learning based course to an online setting. An online course induces a significant overhead and involves allocation of extra resources for material preparation, video recording, and software tool support.
- **Pre-class reading assignments** that adhere to the time constraints described earlier are hard to find for the topics we have covered in our class. To provide sufficient literature, we have recommended paragraphs of book chapters or created annotated versions of scientific papers. Creating a consistent body of reading material that does not exceed our 20 minute time limit is left for future work.
- **Project team building** has been extensively discussed in preparation of our course. While the literature is advocating for teams being assigned by instructors, we have had more success in letting students form teams on their own. This incorporates the risk of unbalanced skills within a team but we have found the motivation of students in their self-assigned teams to be higher. The reason for this might be that the self-assignment allows them to join a team where the topic of the final project really interests them, instead of being forced to work on a project they only have mediocre interest in.

In comparison to former versions of our course, we have experienced an increased motivation in students as well as teaching staff after implementing the active learning principles.

In our re-design we have changed more than a single aspect of our course at a time. This makes it harder to identify which change might have improved the course most significantly, but indicators like the increased quality of final projects or increased Q scores show that our changes had the desired effect. Starting from this new basis, we want to apply smaller changes in the future to measure their effects on learning outcome more systematically.

Acknowledgements

The authors wish to thank the organizers and participants of the "Teaching in Visualization" panel at IEEE VIS 2015. The discussion was great inspiration for changes in our course.

[7]https://www.vocareum.com

REFERENCES

[1] J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983.

[2] C. C. Bonwell, J. A. Eison, and E. C. on Higher Education. *Active learning : Creating excitement in the classroom*. George Washington University, ERIC Clearinghouse on Higher Education Washington, DC, 1991.

[3] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[4] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):pp. 531–554, 1984.

[5] L. Deslauriers, E. Schelew, and C. Wieman. Improved learning in a large-enrollment physics class. *Science*, 332(6031):862–864, 2011.

[6] T. B. Hilburn and W. S. Humphrey. Teaching teamwork. *IEEE Softw.*, 19(5):72–77, Sept. 2002.

[7] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st Conference on Visualization '90*, VIS '90, pages 361–378, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[8] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2Nd Conference on Visualization '91*, VIS '91, pages 284–291, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[9] P. M. Does active learning work? a review of the research. *Journal of Engineering Education*, 93:223–231, 2004.

[10] R. Motschnig, M. Sedlmair, S. Schröder, and T. Möller. A team-approach to putting learner-centered principles to practice in a large course on human- computer interaction. In *Frontiers in Education 2016*. IEEE Computer Society, October 2016.

[11] T. Munzner and E. Maguire. *Visualization analysis and design*. AK Peters visualization series. CRC Press, Boca Raton, FL, 2015.

[12] B. Oakley, R. Brent, R. M. Felder, and I. Elhajj. Turning student groups into effective teams. *Journal of Student Centered Learning*, pages 9–34, 2004.