An experience report using Scrimba: An interactive and cooperative web development tool in a blended learning setting

Rolando Gonzalez
Westerdals – Oslo School of Art, Communication and Technology
Faculty of Technology

Per Lauvås
Westerdals – Oslo School of Art, Communication and Technology
Faculty of Technology

Abstract

In a blended learning environment, face-to-face classroom practices are combined with computer-mediated activities. New technical possibilities may deeply influence the way we organize our educational setting, but may also lead to smaller adjustments within a more static learning environment featuring lectures and traditional lab sessions.

In this study we look at an online, freely accessible code editor under development: Scrimba. Scrimba offers new possibilities for collaborative and interactive learning among a teacher and her students both inside and outside the classroom. Students and teachers may easily live stream coding performed in a browser, and may at any time branch the code into separate projects. These coding session may also be recorded with sound and reproduced as a video. As the tool was not initially exclusively developed with education in mind, we define different use cases for use of the tool in a learning environment.

These use cases were tested in a course delivery. The subject topic was introductory web programming with HTML, CSS and JavaScript. An effort was made to see if available functionality for collaboration and sharing of code in Scrimba within a classroom setting could have a positive effect on a course delivery, both for the students and the teacher. Data was collected using observation in the classroom, a questionnaire for the students and an interview with the teacher.

Early results from this paper suggest that the use of a tool that facilitates interaction and cooperation in an introductory web programming course is well accepted both by the teacher and the students. Easily jumping into live student code in class is especially promising for creating discussion and code modification as a class activity. Coding errors can easily be addressed and corrected live, and students can comment on different solutions to a problem.

This paper was presented at the NIK-2017 conference; see http://www.nik.no/.

1 Introduction

This paper explores the usage of an online tool for coding and collaboration, Scrimba[1], in the setting of blended and active learning. The term blended learning and technology as a tool to promote collaboration and engagement has been discussed at least since the early 2000s [7]. However, although technology has developed and improved the last 10-15 years there still seems to be challenges in making use of the full potential that lies in introducing blended learning in higher education. The literature still tells of "failed" implementations of blended learning, and maybe misunderstandings of what exactly must be done to obtain a good learning environment through blended learning. Part of the reason of the failures, can among other be attributed to the lack of understanding of how to combine face-to-face with online delivery of a course, and how technology can be used to actually achieve such things as collaboration, discussion, engagement and other activities which make up for active learning [9]. Blended learning is a way of achieving active learning and is the situation where the student isn't just a passive listener but rather works actively in a classroom setting.

Through this paper we give an introductory exploration of the online tool Scrimba made for promoting collaboration, sharing of codes and videos, forking of videos etc. and how it promotes blended learning and active learning.

2 Literature review

Even though technology has improved dramatically and become sophisticated and available, the dominant way of education is still, as mentioned by Bøe [2] in reference to a 2011 study, face-to-face lectures, tutorials, mentoring, and project work. There is a possible gap between the students as "digital natives", i.e. who are "comfortable using online sources to meet their information needs", and the university teachers.

Even though there seems to be resistance from teachers to online learning, blended learning in education has increased. The increased use of blended learning especially intensified after a study in 2010 which showed that students who took courses with blended learning performed better than both those that did online courses and those who only took face-to-face courses; when done right. However, there seems to be a failure in realizing the full potential of blended learning (Jeffrey et al. [9], Cavanagh [4]). Although it must be mentioned that already in the early 2000s it was found that technology could promote active and collaborative learning (Laird & Kuh [10]) and how active learning would even result in better grades as shown by Freeman, et al. [8].

Blended learning is in its simplest application giving a course face-to-face, in a classroom setting, in combination with online delivery. There are different definitions in the literature that also include the goals of blended learning, and some definitions do not mention online resources at all, but rather just "mix of media" [12].

Dziuban, et al. [7, p.3] describes mixed-mode or blended learning as a modality that "combines the effectiveness and socialization opportunities of the classroom with the self-directed and active learning opportunities that the online environment offers". In this paper we will focus on blended learning as the combination of face-to-face combined with online delivery in education to achieve an active learning environment.

[1] scrimba.com

One of the goals of blended learning, when used to increase student activity, is to move away from the traditional classroom situation where the student is relatively passive listening to his teacher, to an environment where interactive, active, motivational learning and collaboration takes place. In an environment where the student is more active, one expected result is getting the students attention, maintaining engagement and optionally re-engaging those students who drift away or fail to engage. In addition students in blended learning courses may have experiences that improve learning that are not found elsewhere [9].

To obtain certain goals with blended learning there has to be a conscious and strategic integration, from the teachers side, of the two delivery forms; i.e. face-to-face and online delivery. In addition, the online delivery may be of many types ranging from purely text based online courses to quizzes, forums, chats, collaborative tools, webinars and the combination of the mentioned or other components; each component with different possibilities [12], [9], [3].

As Okaz [12] study shows there is great variation in what achievements are done when implementing blended learning. Technology provides many possibilities but it is up to the teacher to design its use in a course delivery setting [2], [9].

Some of the positive results for the students of using blended learning to promote active learning are: more accessible information, easier to work independent at own time and pace, facilitation of collaborative learning experiences, easier to communicate and encourages information exchange also for introverted students. It is also found that both the social and cognitive skills of students may be improved when combining class debates with computed mediated learning [12].

In this paper we would like to add to the literature on ways to use online technology to achieve blended learning to create a more active and interactive environment for the students.

3 Method

Scrimba, a start-up company, approached Westerdals Oslo ACT in December 2016 with a request to demonstrate a new tool they were developing. The web application was described as an interactive "video linking format" tool with several advantages over traditional video. They suggested the tool could be a valuable addition in an IT educational setting, and wanted an academic opinion on this.

The tool makes it possible to record a coding session with HTML, CSS and JavaScript. Such a recording is described as a "cast". Audio may be recorded, and the development of the code is stored in text with the possibility of being reconstructed in a video-like manner. By doing so, the actual storage of the recording requires less storage than using a video format.

Another advantage over video is the possibility of creating forks or branches of a cast. Anywhere within a cast a viewer may create her own branch, and by doing so further develop the code from the branching point of the already existing cast. This opens up opportunities of combining observation of code creation with instant collaboration.

Although not a dedicated supported feature in the early process, coding activity could also be viewed live.

Defining use cases

Several meetings were held in order to investigate if the tool could be a useful addition in an educational setting where HTML, CSS and JavaScript is taught. In those meetings, different kind of scenarios where the tool could come in use were discussed. Developers from Scrimba provided technical possibilities and limitations within the application. Two lecturers from Westerdals (the authors of this paper) provided input on the educational setting where the tool could be tested in a natural context.

As a result of this process, three use cases where defined as candidates to be tested in a learning environment.

Live coding

A teacher may develop code using Scrimba, and students may follow the coding live in their own laptop browser. In a classroom setting, this could replace the display of the code from a central "big screen" to multiple local student screens. Or the code could be displayed both om the big screen and multiple laptops.

This provides the students with options: Follow the creation of code through the actions of the teacher, or write your own code by following the steps of the teacher. And between those two options, there will be an easily available opportunity for a transition. You can decide to start watching the teacher code, and when you feel ready, you can branch that code and start developing your own code.

Live coding can also open up opportunities for following a coding session although the student is not physically attending the session.

Collaborative assignments

As a student can follow a live coding session and create her own branches at any time, new opportunities for collaboration arise. A teacher may start coding, let's say an example HTML page. The students may be asked to expand or complete the example and use the existing code as the starting point when branching. When some students have completed the assignment, the teacher can jump into a students branch and easily display the solution for the entire class. And if further development of the student code can bring additional learning opportunities, the teacher may live edit the student work on display. The teacher may give small assignments in class where students are to collaborate with each other. This could for example be one student producing HTML code, and letting another student style the page with CSS (and vice versa). Or it could be a student creating an HTML page and introducing errors for another student to find and correct.

Recording outside class

Casts may be used when preparing for class. When student preparation is an important part of a course delivery (e.g. in a Flipped Classroom setting), casts produced prior to the classroom activities may be examples of such class preparation resources.

When questions arise during class, the teacher may find it reasonable to, at a later stage, create casts in order to explain what was found difficult during a class.

And, as stated before, in contrast to the video format the students may more easily work with the code that is explained in these casts.

Putting use cases to the test

A small scale learning environment was picked for testing the tool and the defined use cases. The course included 26 digital marketing students and the dedicated module within the course was introductory web programming with HTML, CSS and JavaScript. The teacher of the module have multiple years of teaching experience in the specific topic. The chosen editor for earlier deliveries of introductory web programming modules is called Brackets[2]. It was decided to use Brackets in the first day of the module when introducing HTML. In day 2 and three, Scrimba was to be used. By doing so, we could identify how a specific student group reacted to the difference between a classical editor and Scrimba with its new possibilities.

Observation

The use of Scrimba in the introductory web programming delivery was observed over two days for a total of 8 hours. The observer was presented to the class and the motivation for the observation was explained. The observer was situated in the back of a Harvard style lecturing room. From that position he cold move behind the back row in the room to observe all student laptop screens present in the room. As the room was limited in size, it was easy to pick up conversations between students.

Survey

After the last lecture in the course, a questionnaire was delivered to the students through an online survey provider. The questionnaire consisted of 8 questions with multiple sub-questions. The questions evolved around student satisfaction on using Scrimba when being introduced to web programming. The use cases used in the course delivery were specifically targeted one by one. By doing so, we could find out if certain use cases worked better than others. Of a total of 26 students, 14 answered the survey. The survey was anonymous, but the students could leave an optional contact information if they could be interested in a short interview i addition to the survey. Three students left their contact information, but it turned out to be impossible to schedule any meetings with them before the summer break started. Interviews with the students were therefore omitted from the study.

Interview

A week after the last session in the course, the teacher was interviewed. The semi-structured interview lasted for approximately 50 minutes. A voice recorder was used, and the interview was later transcribed to a total of 6000 words. The interview evolved around the teachers experience of using Scrimba in the course delivery. What worked well, and what did not? And how did the delivery differ from a delivery without such a tool? As the interview was performed by the observer in the classroom, some observations were discussed in the interview.

4 Findings

For simplicity, we have decided to categorize the results in regard to the use cases that were tested.

[2] brackets.io

Recording outside class

The lecturer created and distributed a recorded cast to the students between the second and third day of the module. The cast was an introduction to JavaScript - the topic for the following session. The video was published in the Learning Management System (LMS) and a message was sent to the students asking them to watch the video prior to the next lecture. The motivation for the cast was to prepare the students for the next class. And as opposed to a traditional video where the viewer can watch only, the students had the opportunity to branch the code at any time and play with the code themselves.

When asking the students whether or not they had watched the cast, only a few students gave a positive response. That was the perception by the lecturer, and that was what was observed by the observer. The questionnaire told the same story.

Prior to the lecture at May 29., Rolando released a video describing loops in JavaScript. Have you watched the video? If so, did you continue working with the code, or did you use it as video only?

Of the 13 answers to the question, only two students had watched the cast, and none of these two had done any further work with the code. Some students gave a reason for not watching the video, and it was typical answers like "I was on vacation" or "I know some of this stuff from before".

But another question reveals that they think manipulation code in a video can benefit their learning:

In Scrimba you may copy/clone solutions others have made and do further work on them. This may contribute positively to your learning.

Totally disagree	Somewhat disagree	Somewhere in between	Somewhat agree	Totally agree
0%	7%	21%	36%	36%

Live coding

The students were positive to the use of Scrimba for live coding. A large portion of the students valued the possibility of following live coding in their own browser when evaluating the following statement:

With Scrimba you may follow the lecturers live coding in your browser. This may contribute in a positive way for your learning.

Totally disagree	Somewhat disagree	Somewhere in between	Somewhat agree	Totally agree
0%	0%	23%	38,5%	38,5%

However, in the live sessions during the two days of testing, the lecturer explicitly advised the students to live code *together* with the lecturer as opposed to following the code live in the browser (as Scrimba supports a live coding stream) or on a big screen in the classroom. It was observed in the classroom that the students followed this advice. Of a total of 26 students in the class, just a few students (2 or 3) were observed to not code for themselves. These students were busy doing something

else (checking Facebook etc). Once in a while, a few students experienced technical issues with Scrimba, and therefore had to watch in the browser or on the big screen for a while. This observation was supported by another question in the questionnaire:

Rolando (the lecturer) has used Scrimba for live coding. As a student, you could watch the coding live on the big screen in the classroom, or you could code in your own browser. How did you experience this? What did you choose?

The majority of the students replied that they did write their own code. And by doing so, the live stream supported by the tool in use was irrelevant. They could watch the big screen and see the lecturers code, no matter what the tool was, and write in some tool on their own laptops.

In the interview with the lecturer, the lecturer confirmed that for pedagogical reasons he strongly suggested that the students wrote their own code in stead of just watching the lecturer do the coding. And when the students wrote code *with* the lecturer, the live streaming possibility was less important. But other scenarios were discussed. In a larger setting (large auditorium with 200+ students), students arriving late could catch up with the current example, as opposed to having to wait for the next coding opportunity. There could also be settings where students are not able to be present, and therefore could have use for watching live code in their browser.

Collaborative assignments

In the interview with the lecturer, the lecturer stated that the most promising find when testing Scrimba, was the use of collaborative assignments in class. The first example of this was asking students if they had any trouble writing code for a particular exercise handed to the students. Some students raised their hand, and the lecturer could display the solution on the big screen. It created a more personal atmosphere. As the teacher put it:

> And that makes things a bit more personal. You can see what the others have coded. You see as a student that: Ah, this person has made such an example, with these and those pictures. So funny. And, it was a little humorous too. She used some funny pictures and a little funny theme. Again, it will suddenly be much more personal. You share something within the community.

And this code sharing made it easier for the lecturer to find problems that might not have been discovered. As a professional with years of experience, it might be easy to forget those initial problems that may occur. These problems may come in different shapes and forms.

> Problems that many students face are addressed. The problems you usually do not get to discover if you do not share like that. Because if we had not shared that way, I might not have known that: Oh, yes, it's very hard to use this and that code. Or it's very hard to see the type of errors students do. So this opened up to getting a little closer to the student's head. See things from the student side.

But it was also valuable to see different well functioning solutions to the same assignment.

> You can see that the same code can lead to very different types of results. And in addition, this student might have done a couple of things that the others had not mastered. I think I remember that I heard "Oh, did you manage that? Such a hovering effect on that thing..." It could be inspiring for the others and maybe open up to: "Oh yes, there are different possibilities with this code."

An enthusiasm for collaborative assignments were also observed in the classroom. These exercises resulted in high activity among the students. A large majority of the students seemed to enjoy watching each others solutions. One female student was asked to share her solution to an open assignment involving creating a sample web page. Her particular solution involved a sparrow and a waffle. When everyone saw her solution on the big screen, she told the rest of the students how she had created the solution. In the following break, a fellow student approached her and gave her credit for creating the solution. Apparently, exposing own work like that is not something they are too accustomed to as the student replied: "Oh my god. That was so embarrassing!".

The results from the questionnaire support the perception of the lecturer and the observation in the classroom.

On Wednesday, Rolando presented student solutions on the canvas ("Waffle to sparrow" and "Snorkmaiden"), and further developed these solutions. How did you experience this?

Within the 12 answers, the general opinion was that it was a positive experience. They enjoyed examples from fellow students, and it was helpful to see typical beginner mistakes and how they may be corrected. And it was nice that it was easy to continue working on others code without having to spend time sending code back and forth.

But there were some students who reported that using collaborative exercises and solving them together in class could be unnecessary for those who already understood the topic of attention.

In Wednesdays lecture, Rolando introduced solutions that did not work (words that were added to a list + print of numbers from 0-99). The students were to find the mistake. How did you think this worked?

When answering the question, the majority of the students were positive, but there were also replies like this:

It works well for those who have not got it, but get a little bit like gilding the lily for those who have already done it.

and

Okay, but not too engaging. When you are a little ahead of the others, such tasks can quickly become a little too simple.

5 Discussion

The use of video in education is nothing new [13]. In recent years, video has been widely used in different learning settings like online studies [5], flipped (inverted) classroom [6] or as a classroom activity [1]. When video is in use when explaining something regarding coding, code on display may be accessed and further modified by the viewer. But normally, this requires additional actions performed both by the creator of the video and the viewer. As an example, the code in the video may be placed in an open code repository like GitHub or BitBucket and referred to by video metadata. With Scrimba, what looks like a stored video is actually a video simulation based on stored text and audio.

Although our study is in a preliminary stage, there are early indications that Scrimba may contribute to active learning when teaching an introductory web development course. Scrimba may contribute to new possibilities within already existing activities. Within a classroom, in a lecture style setting, Scrimba may facilitate activities both inside and outside a classroom setting. It may promote active and collaborative learning, as blended learning should do, as described in Laird & Kuh [10].

Live coding in a classroom setting has also been used for many years. But with Scrimba, students may watch live coding on their own laptops, and at any time use the current status of the coding project as a starting point for manipulating the code by themselves. In our initial investigation, the students did not see a big reason to do so. As the lecturer strongly advised the students to do all coding in parallel with the lecturer, there was no good reason to watch a live coding stream. A somewhat strange contradiction was found in the collected data as the students were of the impression that following the live coding stream could be beneficial to their learning. At the same time, they all took the lecturers advice to code for themselves. And by doing so, they found the live stream to be irrelevant. Perhaps the students saw other possible benefits with live coding than the one they had tried initially. Some of these possibilities were proposed in the interview with the lecturer, and others may be invented. There could exist a setting where it is a pedagogical reason for the students to first pay attention to live coding, and then start manipulating the code at a specific time. But this has not been tested in this study.

The lecturer realized, through the interview, that teaching web programming could be too much about the lecturer displaying how to do something perfectly, and not focusing on problems that might occur. It is not always easy to remember what was originally difficult to understand when being introduced to something new. By facilitating sharing of student code, lecturers may more easily notice problems occurring among the students. And students find it motivating to easily be able to see other students work. It creates a level of transparency in the classroom that normally is hard to achieve. When errors occur, they can be highlighted and discussed. This leads us into debugging, and how important it is for students to be able to develop debugging skills [11].

6 Conclusion

Using the online tool Scrimba as an enabler, we explored blended learning with the goal to promote active learning. We used different use cases as a starting point for the exploration, and the first results seem to be positive in regards to promoting

student engagement, collaboration and discussion. We found it interesting to observe how students collaborated with small group assignments, facilitated with the ease of sharing in the tool, and how the students then discussed these amongst them, creating a combination of a professional and social setting. In addition, we saw new possibilities of improving the learning experience in regards to the teacher sharing students' codes on display for all to see which contained errors and discussing these in the class; this as a way to getting closer to the challenges the students face as they are learning new coding languages. We hope this paper can be used as inspiration for further research on alternative ways of using technology to promote learning.

7 Limitations

The main limitation is a small sample size of respondents on our questionnaire, as the tool was tested on a small group of students. We would have liked to perform interviews with the students, but it was not possible due to the time of the semester. However, this paper is meant to be an exploratory introduction to the online tool Scrimba, and a first identification of its potential in an educational setting.

In this paper we discuss educational possibilities within one new coding application. Further investigation could be performed in order to compare Scrimba with other alternatives.

There are other possible use cases to explore within the tool. Some are already identified but not tested yet. These include *Recording during class, Individual assignments* and *Deliveries*. These are planned to be investigated and tested in a later study.

References

[1] Berk, R. a. Multimedia teaching with video clips : TV , movies , YouTube , and mtvU in the college classroom. *International Journal of Technology in Teaching and Learning 5*, 1 (2009), 1–21.

[2] Boe, T. Adoption of Technology in Higher Education :. *Nokobit 22*, 1 (2014).

[3] Carrington, D., Kim, S. K., and Strooper, P. An experience report on using collaboration technologies for distance and on-campus learning. *Conferences in Research and Practice in Information Technology Series 103*, Ace (2010), 45–51.

[4] Cavanagh, T. B. The Postmodality Era: How "Online Learning" Is Becoming "Learning". In *Game changers: Education and information technology*, D. G. Oblinger, Ed. Educause, 2012, ch. 16, pp. 215–228.

[5] Choi, H. J., and Johnson, S. D. The effect of context-based video instruction on learning and motivation in online courses. *American Journal of Distance Education 19*, 4 (2005), 215–227.

[6] Dazo, S. L., Stepanek, N. R., Fulkerson, R., and Dorn, B. An Empirical Analysis of Video Viewing Behaviors in Flipped CS1 Courses. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '16* (2016), 106–111.

[7] Dziuban, C. D., Hartman, J. L., and Moskal, P. D. Blended Learning. *Research Bulletins 2004*, 7 (2004), 1–44.

[8] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. Active learning increases student performance in science, engineering, and mathematics. *Pnas 111*, 23 (2014), 8410–8415.

[9] Jeffrey, L. M., Milne, J., Suddaby, G., and Higgins, A. Blended Learning : How Teachers Balance the Blend of Online and Classroom Components. *Journal of Information Technology Education 13* (2014), 121–140.

[10] Laird, T. F. N., and Kuh, G. D. Student experiences with information technology and their relationship to other aspects of student engagement. *Research in Higher Education 46*, 2 (Mar 2005), 211–233.

[11] McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., and Zander, C. Debugging: a review of the literature from an educational perspective. *Computer Science Education 18*, 2 (2008), 67–92.

[12] Okaz, A. A. Integrating Blended Learning in Higher Education. *Procedia - Social and Behavioral Sciences 186* (2015), 600–603.

[13] Smith, T., Ruocco, A., and Jansen, B. Digital video in education. *ACM SIGCSE Bulletin 31*, 1 (1999), 122–126.