Social Quizzes with Scuiz

Massimo Di Pierro and Peter Hastings

School of Computing, DePaul University, 243 S Wabash Ave, Chicago IL 60604

Abstract. Scuiz is a platform for social quizzes. Students, not only the teacher, create quiz questions which are then randomly assigned to other students. Questions are organized in feeds similar to feeds found in common social networks. The system allows students to "challenge", or dispute a question's premise or answer, and discuss them. A mechanism of incentives ensures that students engage with the system early and substantively. The system keeps statistical information about students' questions, and automatically flags some of them as "too easy" or "too difficult". The teacher can review these challenges and accept/reject the submitted questions, as well as participate in the discussion and change the status of questions. This paper describes Scuiz, which was implemented by the first author, some issues arising from the use of it in classes, and future research directions concerning its effectiveness.

1 Introduction

Today technology plays an important role in teaching, not only as a subject of study, but also as a teaching tool. The most common teaching tool is the Learning Management System (for example Canvas[1], Blackboard[2], Desire2Learn[3], and Sakai[4]) and most LMSs include a quiz component. Most existing tools for quizzing students share a similar design: the quiz questions are created by the teacher or selected from a pool of existing questions, and students answer those questions. With Scuiz we took a radically different approach. Scuiz looks to students very much like Facebook, the social network, looks to friends or social connections. They "follow" classes that they are enrolled in. Their feed consists of questions created by their peers. Every student can create a question by posting one on the feed. The student also has to provide the correct answer, examples of incorrect answers, and an explanation about the correct answer. Once a question is posted, it randomly appears on the feeds of their classmates. Students who provide the correct answer earn one point. After a student has answered a question, correctly or incorrectly, the student gets access to a discussion board about the question, very much like discussions on Facebook posts. As part of the discussion, students can "challenge" a question (because for example it is ambiguous, or because the student disagrees with the official answer provided

[1] `https://www.canvaslms.com/`
[2] `http://www.blackboard.com/`
[3] `https://www.d2l.com/`
[4] `https://sakai.luc.edu/`

by the author of the question). Once a question is challenged, it is assigned to the feed of the teacher of the class and flagged for attention. The teacher then decides whether the question is appropriate and should be accepted or rejected. If the question is rejected, points are adjusted accordingly (extra credit for the student that challenged it, and potentially a penalty for the author of the wrong question). The system also enforces a set of rules to push students to create more and clearer questions:

- Students must post questions to have access to other students' questions with a ratio of 1:10. For each question they post, they get access to 10 more questions created by their peers.
- Students get points for answering a question correctly, for challenging a question that is later rejected by the professor, and for creating questions of the right difficulty level.
- Students lose points for posting a question that is contested and then rejected by the teacher, or for contesting a question that is found to have a correct answer.
- The system automatically assesses the difficulty level of each question by keeping track of how many students answer correctly. If the percentage is higher than a certain threshold, the question is flagged as "too easy" and removed from the pool. If the percentage is lower than another threshold, the question is flagged as "too difficult" and also removed from the pool.
- The teacher has access to all questions and can override their status of accepted/rejected/contested/too easy/too difficult.
- Once a student answers a question, the student can see immediately if his/her answer was the correct one. At this point the student can also read the explanation about the correct answer provided by the author, enter in discussion mode, and eventually challenge the official answer.
- The system always shows each student the total number of points and the rank with respect to the other students in the class, in anonymized form.
- Questions are anonymized to students. The teacher has the option to see the questions anonymized (to demo the system in class for example) or not, and can visualize the name of the authors of questions and comments.
- The teacher can also impersonate students and monitor what a particular student sees in the feed.

This system of rules and incentives/disincentives forces students to engage with the system early and regularly. In fact, by default, the system has no deadlines and it is technically possible for a student to skip every interaction with the system until late in the term. This has two negative effects: all the easy questions have been removed for being too easy, so the student will only see the remaining more difficult questions. For a student to access questions, the student must have created questions. Postponing the creation of questions make it harder to create good questions, since other students expect questions of increasing difficulty or they will be flagged as "too easy". It becomes increasingly hard to catch up to the number of points of students who engaged with the system early.

From the point of view of the teacher, the system is very easy to use. In fact in principle the system does not require any interaction other than logging in, creating a "class container" and adding the emails of the students to the container, effectively subscribing the students to the class feed. These steps could be automated if the system were to be integrated with an existing LMS. The teacher can optionally create questions and/or seed the system with pre-existing questions from previous courses or a database of questions. This would allow the teacher to provide examples and set the tone for the kind of questions he/she expects. The teacher is also optionally required to periodically monitor his/her own feed and accept/reject questions that have been challenged by students. The teacher is encouraged to enter into discussions about challenged questions and explain the correct answer. By monitoring his/her feed, the teacher has an immediate picture of types of questions posed by students. This gives an indication of what students understand the most about the class material. From the challenged questions the student gets a picture of what students are most confused about and what material may need clarification in class. From the real time ranking of students, the teacher can see which students are not engaged with the class material and which students have the most difficulty.

Two tasks are left completely to the instructor: providing instructions about the type of questions expected by the students, and establishing a grading policy. The system ranks students and assigns a total score to each student which corresponds to the sum of the points from correct answers (corrected for challenge points), but the system does not provide an absolute A-F grade.

The benefit of using quizzes as a learning tool is a subject of great debate. Some argue that they only provide a short term benefit [1], but there is agreement that more frequent testing is better than limited scheduled testing [2]. There is also evidence that when quizzes are accompanied by discussion, student engagement, or other creative activity, the benefit persists on cumulative semester and end-of-year exams. [3–5]. Scuiz provides a platform that allows frequent quizzing of the students in an engaging and collaborative manner. Students have to create questions thus offering an opportunity for thinking about the problems and expressing creativity. Although this paper focuses on the technological innovation of Scuiz and not on its pedagogical effects, our experience gives reason to believe Scuiz to be a valuable teaching aid to teachers and a valuable learning tool for students compared with traditional systems.

2 Architecture and Markmin

The system is built as a single page web2py[5] application, as shown in Fig. 1. Web2py provides basic services such as a database abstraction layer which dynamically converts API calls to SQL queries for any of the supported databases (although in our deployment we use PostgreSQL), authentication (login, logout, edit profile, change password, reset password, verify email), access controls

[5] `http://web2py.com`

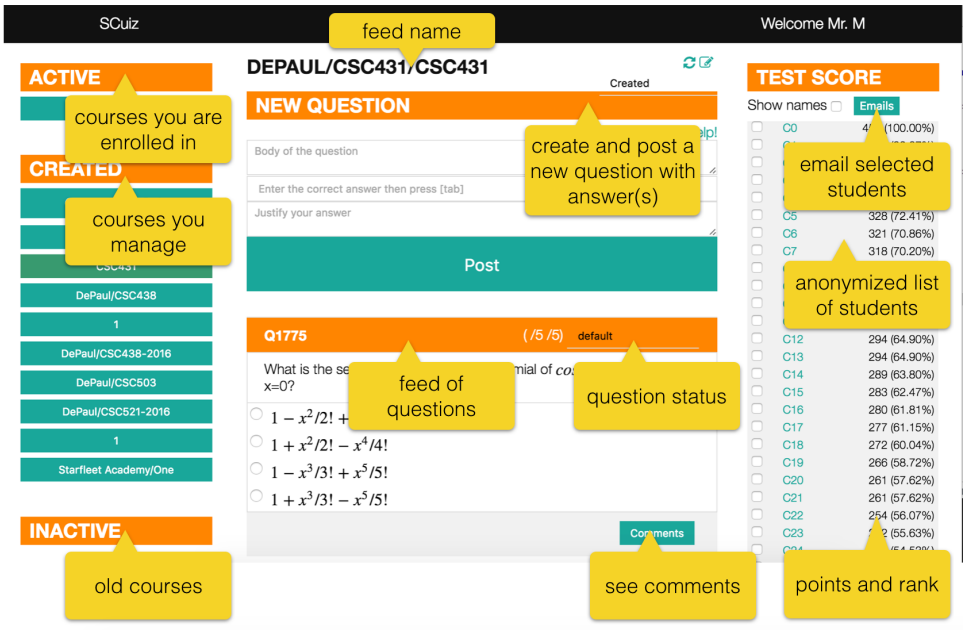

Fig. 1. The main Scuiz teacher interface with anonymized students.

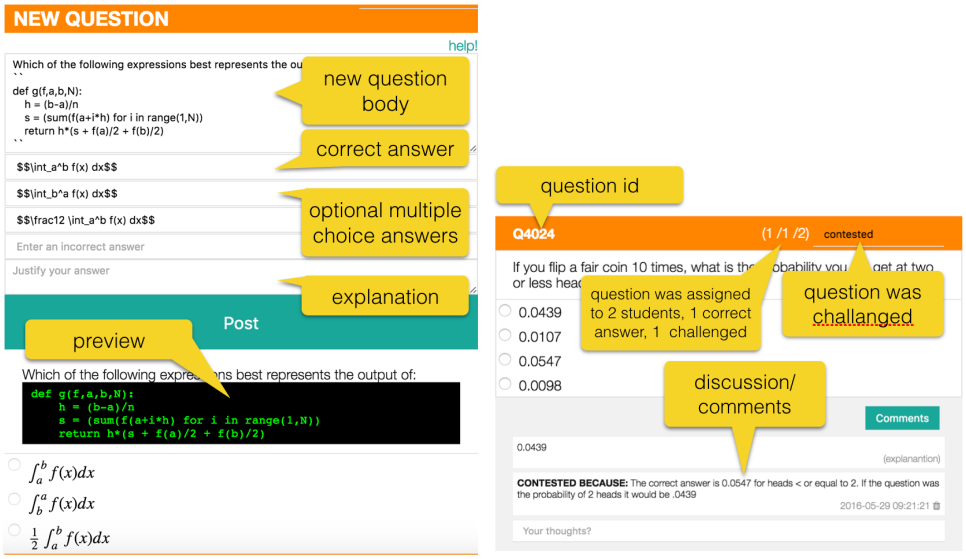

Fig. 2. The interface for creating questions with an example of the markmin syntax (left) and the interface for reviewing challenged/contested questions and comments (right).

(users, groups, membership, permissions), server-side sessions, internationalization service (although we currently only support English), error logging and an administrative dashboard.

The client is built mostly on top of ractive.js[6] a library for dynamic DOM. It alters the HTML dynamically based on the data returned by the API without the need to reload the page or refresh the entire content. Only those parts of the HTML that depend on data that has changed are re-drawn in real time. The client also utilizes a number of custom made plugins that allow cut and paste of links to images, videos and other media files. We utilize MathJax[7] to render formulas using LaTeX[8] typesetting.

Questions are created using a new markup language called "markmin2", which is derived from the web2py markmin markup language[9], itself derived by the popular and ubiquitous markdown[10]. This is shown in Fig. 2. The rules of markmin2 are designed to be simple and allow students to insert structured content with minimal training. Bold text is inserted as `**text**`, italic text as `''text''`, code text as ``` ``text`` ```, and LaTeXexpressions as `$$formula$$`. Ordered lists and unordered lists are created automatically when multiline text is preceded by a number or a dash respectively. URLs in text are automatically converted to links. If a link is preceded by `image:http://...` the link is interpreted as an image and the image is embedded. The same for `video:http://...` and a for `audio:http://...`. Markmin2 supports the oembed protocol[11] so it can embed widgets using `embed:http://...` syntax. The choice of using a markup language instead of an inline HTML editor guarantees that all questions are formatted similarly, even if written by different students. It allows enough flexibility to insert bold, italic, code, formulas, and media files, but not so much freedom as to allow choosing arbitrary font sizes and colors.

3 Issues

Different development versions of this system have been utilized for teaching different courses at DePaul University. All courses were taught by the first author. These courses include Introduction to Python Development (undergraduate), Web Frameworks (CSC438, graduate and undergraduates), Monte Carlo Simulations (CSC431, graduate) and Scientific Computing (CSC431, graduate). In all the courses Scuiz was one of the tools used for grading but not the only tool and it was complemented by a final exam and other forms of electronic assignments.

In general students had no difficulty using the system. One problem was that a few students decided to sign up with a different email address than the official one registered with the University, therefore they were not automatically added

[6] `http://www.ractivejs.org/`
[7] `https://www.mathjax.org/`
[8] `http://latexproject.org`
[9] `https://github.com/mdipierro/markmin.js`
[10] `https://daringfireball.net/projects/markdown/`
[11] `http://oembed.com/`

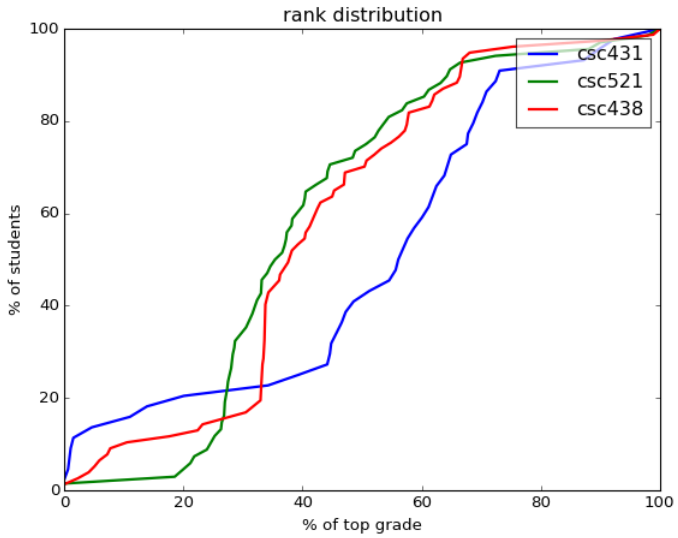

Fig. 3. The plot show the students distribution relative to the top performing student for three difference courses. CSC431 was taught before including a disincentive for challenging questions which resulted in a shortage of questions. The CSC521 and CSC438 courses were taught after the introduction of the penalty. This allowed more students to catch up. (The data was obtained de-identified from the system).

to the class feed. The professor was able to correct this once notified. Initially students did not make any effort at utilizing the proper markup until they were told that their questions would be rejected if the markup was not done properly. In particular, code snippets in questions had to be properly marked as such (``code``), otherwise the indentation would appear broken. Also there was a tendency for students to write easy questions just to get access to more questions from their peers, apparently not caring about the rating of the questions they generated. This resulted in many similar questions that had to be rejected manually.

From the perspective of the teacher, the system is also very easy to use, and saved a lot of time in writing and grading questions since this is automatically done by the system. The two main issues from the teacher's perspective were the following:

It is hard to establishing a grading policy before knowing how many questions would be created by the students on average. As is probably expected, there is a huge disparity between the best students and the worst ones. In a typical class of 80 students, the top students earned twice as many points as the average. Also, students want to know in advance (in the syllabus) what the minimum number of points required to earn a guaranteed A in the class (assuming perfect

score on the final exam and other assignments). While the system is designed to put students in competition with one another, justifiably students need to budget their time, in particular students who are employed. When students were informed that they need to obtain, "at least 200 points", the majority of the students stopped using the system as soon as they reached 200 points. As a good rule of thumb, asking students to create 2 questions per week, and to answer about 20 questions per week, gets them to the target of 200 in 10 weeks, assuming they answer all questions correctly.

The other issue was the proliferation of challenged questions. In a class of 80 students, if each student creates 2 questions/week, there are 160 questions/week and 1600 questions every quarter (assuming 10 weeks/quarter). These numbers are typical, although in some of the classes the number exceeded 2000 questions. In early versions of the code, there was no penalty for contesting questions and thus students tended to challenge almost every question. It is clearly impossible to review thousands of questions for each course. In the later versions of the code, we introduced a penalty for challenging a question and being incorrect in the challenge. This means that points can go down if they challenge and the teacher finds the original question/answer to be instead correct. This added enough deterrent to reduce the number of questions needing review to about 10% of the total. That means the teacher had to review about 20 questions every week, which is a manageable number. Pedagogically, it gives students an incentive to "check their work" to ensure they are right before challenging a question. Notably, the system can run pretty much in autopilot mode in the sense that if the teacher does not perform his or her duty to review challenged questions, the system still continues to work, except that the challenged questions are not assigned to others. The only downside is that students may complain if a challenged issue has not been settled. An example of the challenge interface is shown in Fig. 2. The effect of this change in policy is shown in Fig. 3.

The system allows the teacher to access challenged questions in anonymized format in class. This provides an opportunity for explanation and discussion that is engaging to the students.

4 Conclusions

Scuiz has been a valuable tool for the teacher. It was fun to build and to use. It saved time from grading and the teacher could focus more on developing teaching material and providing feedback to students based on their Scuiz answers. The main issue encountered is that students needed more guidance than originally expected about the level of questions they could and should submit to the system. Providing them with examples helped them more than simply stating "create questions based on the lecture content this week". The system provided a challenge to the students who are not used to creating questions, but are instead used to answering them. It forced them to think about the most important ideas from class lectures. Some loved it and some didn't. The Scuiz system generated a lot of discussion which had a positive effect on class engagement. The ques-

tions submitted to Scuiz remain in the database, which allows for the creation of a growing knowledge base of questions tagged by course, topic, and difficulty. This is a feature of scuiz that is theoretically very important, although we have not yet implemented the ability to export and import questions across different courses.

Scuiz is currently available as a service to selected whitelisted teachers, but it will soon become available as a service for everyone, and we plan to integrate it with the most common LMSs. Scuiz is avalable in demo mode at:

`http://scuiz.org`

Acknowledgements

Dr. Di Pierro would like to thank his students and Dr. Luca De Alfaro for the many interesting discussions while developing Scuiz.

References

1. Stanlee, Lloyd S., and W. James Popham. "Quizzes' contribution to learning." Journal of Educational Psychology 51.6 (1960): 322.
2. Connor-Greene, Patricia A. "Assessing and promoting student learning: Blurring the line between teaching and testing." Teaching of Psychology 27.2 (2000): 84-88.
3. McDaniel, Mark A., et al. "Test-enhanced learning in a middle school science classroom: The effects of quiz frequency and placement." Journal of Educational Psychology 103.2 (2011): 399.
4. Rao, Sumangala P., and Stephen E. DiCarlo. "Peer instruction improves performance on quizzes." Advances in Physiology Education 24.1 (2000): 51-55.
5. Daniel, David B., and John Broida. "Using web-based quizzing to improve exam performance: Lessons learned." Teaching of Psychology 31.3 (2004): 207-208.