THE MANY WAYS TO STUDENT ACTIVATION: LESSONS LEARNED FROM 25 YEARS OF TEACHING UNDERGRADUATE PROGRAMMING CLASSES

M. Meyer

Westfälische Hochschule - University of Applied Sciences (GERMANY)

Abstract

This paper reveals various approaches undertaken over more than two decades of teaching undergraduate programming classes at different Higher Education Institutions, in order to improve student activation and participation in class and consequently teaching and learning effectiveness.

While new technologies and the ubiquity of smartphones and internet access has brought new tools to the classroom and opened new didactic approaches, lessons learned from this personal long-term study show that neither technology itself nor any single new and often hyped didactic approach ensured sustained improvement of student activation. Rather it needs an integrated yet open approach towards a participative learning space supported but not created by new tools, technology and innovative teaching methods.

Keywords: Student Activation, Interactive Voting Systems, Flipped Classroom, Peer Instruction, Social Learning, Peer Assessment.

1 INTRODUCTION

It is by common-sense but also supported by empirical research [1] that focus, activation and engagement are among the most important success factors for students' performance [2].

Having started the author's academic career in times when a 'lecture theatre' had very much in common with a 'theatre' in the classic sense where something happened on stage while the auditorium mostly passively followed the class with almost no interaction – maybe except for Biggs' stereotypical Robert [3] just asking whether the topic is relevant for the exam – we soon recognized that we need to activate students to get out of their comfort zone and take an active part in their learning process.

Therefore, in the following years we introduced and studied various approaches to increase student activation and engagement ranging from the use of digital tools and platforms over applying innovative teaching methods up to letting students challenge their peers and assess their performance. Hence, the focus moved constantly from the teaching and teacher's perspective to the perspective of the learners and their role in the learning process which we want them to take ownership of ultimately.

2 THE MANY WAYS TO STUDENT ACTIVATION – A LONG JOURNEY

Since the very first days in my teaching career, motivating students to actively take part in the lectures has been a central issue. Over the past 25 years, this has led to exploring various means inside and later outside the classroom.

2.1 Introducing Interactive Voting Systems ('Clickers')

Long before every student had their own smartphone in class, we introduced so-called Interactive Voting Systems, also known as 'clickers' (see Fig. 1), in class to run re-cap questions and various types of quizzes, e.g. from the popular TV show "Who wants to be a Millionaire?" [4].

Figure 1. Interactive Voting System® - Handheld Devices and Receiver for the Instructor's Computer

They also proved very useful to collect truly anonymous feedback at the end of the class, as students randomly take a device from a box at the beginning of the class. This was quite important as in former times when using color cards, students often first waited for their neighbor's reaction or hesitated to take part in such polls at all.

While subsequently we moved from such hand-held devices to software applications running on the student's smartphone or computer, e.g. Mentimeter®, the latter became the only practical way in times of the pandemic when classes had to be held online. Even after returning to the classroom, we still stay with these software solutions as smartphones and internet access have become ubiquitous now.

Tools like Mentimeter® offer a modern form of 'clicker', with a wide range of options for different types of questions, including multiple choice, open-ended questions, and matrices. The students respond by logging on to a webpage using a code. Results can then be presented in various ways (see Fig. 2).

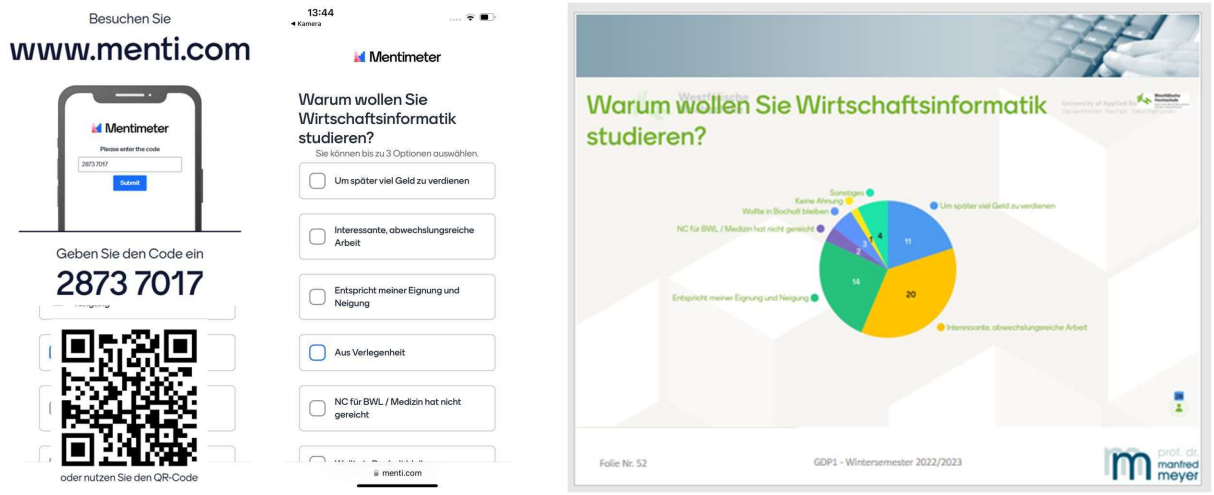

Figure 2. Using Mentimeter® for running polls on the students' smartphones and showing results

These are highly efficient tools, especially among today's technology-savvy students. In addition to creating engagement, they provide a way to involve the whole audience. The answers are anonymous, and the students can respond from their smartphone or PC. Thus, the only obstacles to responding are technological, as opposed to the courage it takes to raise your hand and speak in a large gathering.

Experiences from introducing such tools, no matter whether hard- or software-based, showed that students like the innovation as such (something different) as well as the tangible device (control) in their hands, which especially applies to the handheld versions, but this effect quickly drops – unless these tools are used to support innovative teaching methods [5].

2.2 Peer Instruction

Besides for running quizzes and collecting feedback, interactive voting systems became also very useful, when we started applying the concept of 'peer instruction' [6t] to our classes: Now, technology was not only used for technology's sake, but helped implementing an entirely new teaching method. We still ran polls asking the students to answer some questions related to what has been discussed in the last class. But this time, depending on the result (if the ratio of correct answers was not too high and not too low), we included a short peer instruction session in our class: students were asked to discuss their answers within small groups of peers before the poll was continued, usually showing considerably better results as students were forced to discuss and explain their answers convincing their peers – or getting convinced by them [7].

Although, this was only a small change to our classes, it contributed much to the activation and engagement of our students. Suddenly, they were no longer just passively following the lecture but also forced to discuss and actively share insights with their peers.

2.3 Gamification and the PrimeGame Competitions

Another approach often used to increase student activation and engagement comes with the application of typical elements of game playing (e.g. point scoring, competition with others) to teaching, also known as 'gamification' [8]. While with the use of 'clickers' we already experimented with first steps towards

gamification by running quizzes in groups, counting points and giving small prizes to the winning group, we then applied this technique to the lecture content itself. Thus, we invited the students to develop a bot playing a simple two-player board game that we called 'PrimeGame' [9]. Students simply had to submit a function implemented in Java which selects a number from a given set of remaining numbers on the board. This number will be removed, its value added to the score of the player while all remaining factors will also be removed and their values added to the opponent's score. In order to continuously use the PrimeGame approach every year, it has been necessary to slightly modify the rules from time to time such that player implementations from previous competitions may not perform as well as last time - if they perform correctly at all.

During the final class of the year, an annual 'PrimeGame Competition' is being held since 2008, to which we soon also invited programming classes from nearby secondary schools to participate. Fig. 3 shows a live ranking shown in the classroom during the competition, here currently led by a bot from one of the nearby schools. Once the competition is completed, i.e. every bot has played against each other bot, the winner gets announced. Finally, the best students as well as the best team get awarded certificates and prizes sponsored by local companies.

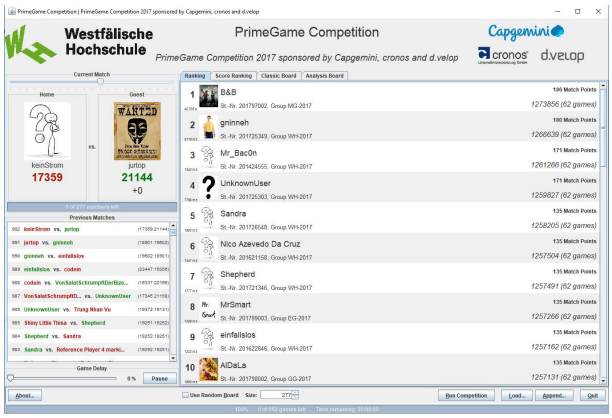

Figure 3. Live Ranking during the Annual PrimeGame Competition 2017

Having used the PrimeGame at different Higher Education Institutions, overall experiences show that when in practical classes it comes to introducing the PrimeGame and starting a two weeks period for player design, implementation and testing, most of the students almost instantly become highly motivated and keen to compete with other classmates. As usual, however, some students still choose not to participate as the PrimeGame is not a mandatory assessment due to formal reasons. Compared to other practical exercises, this number has been observed to be much smaller. Thus, the PrimeGame in general works very well for improving student's involvement and participation except for one case when using the PrimeGame in a programming class at a Namibian University where the observed impact on students' motivation was much smaller due to various reasons as discussed in [10].

Besides for the annual competitions, the PrimeGame approach has also been used to study various interesting features, e.g. inter-bot-communication models in an advanced class on agent programming [10], or different implementational aspects as part of a student's thesis, e.g. a cloud-based implementation for 'Social Gaming & Coding' (SoGaCo [11 as shown in Fig. 4]) or as a Bi-Directional Layered Notional Machine [12].

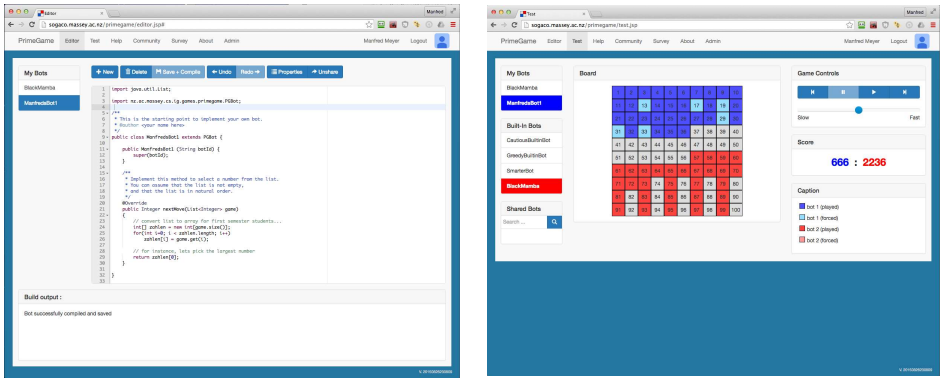

Figure 4. Development and Live Test of a PrimeGame Bot using the SoGaCo Platform.

2.4 Social Learning and Just-in-Time Teaching

Motivated through a keynote at EDULEARN 2018 by Eric Mazur and a follow-up visit to his group at Harvard University, we decided to introduce the social learning platform Perusall [13] in our first- and second-year programming classes. Perusall basically is a collaborative annotation tool that aims at turning solitary reading assignments into engaging collective learning activities. It allows instructors to digitally assign readings to students, who then collaboratively engage with texts through annotation and commentary. Perusall is based on extensive behavioral research at Harvard University and is used by a growing number of faculties and students at various universities worldwide, however at that time not yet in German-speaking countries. By using Perusall, we expected to turn the often-skipped solitary reading assignments into engaging collective activities students don't want to miss. Students collectively annotate each reading by asking questions, responding to other peers' questions, or sharing other perspectives or knowledge. Perusall's data analytics automatically grade these annotations to ensure that students complete the reading. Thus, in a perfect world, instructors get a classroom of fully prepared students every time.

Using Perusall, students share questions, answers, perspectives and external knowledge in threads deriving from annotations to text, images, or even program code. Fig. 5 captures a discussion thread (right column) that evolved around some part of the reading material, i.e. the annotation with pink background colour.

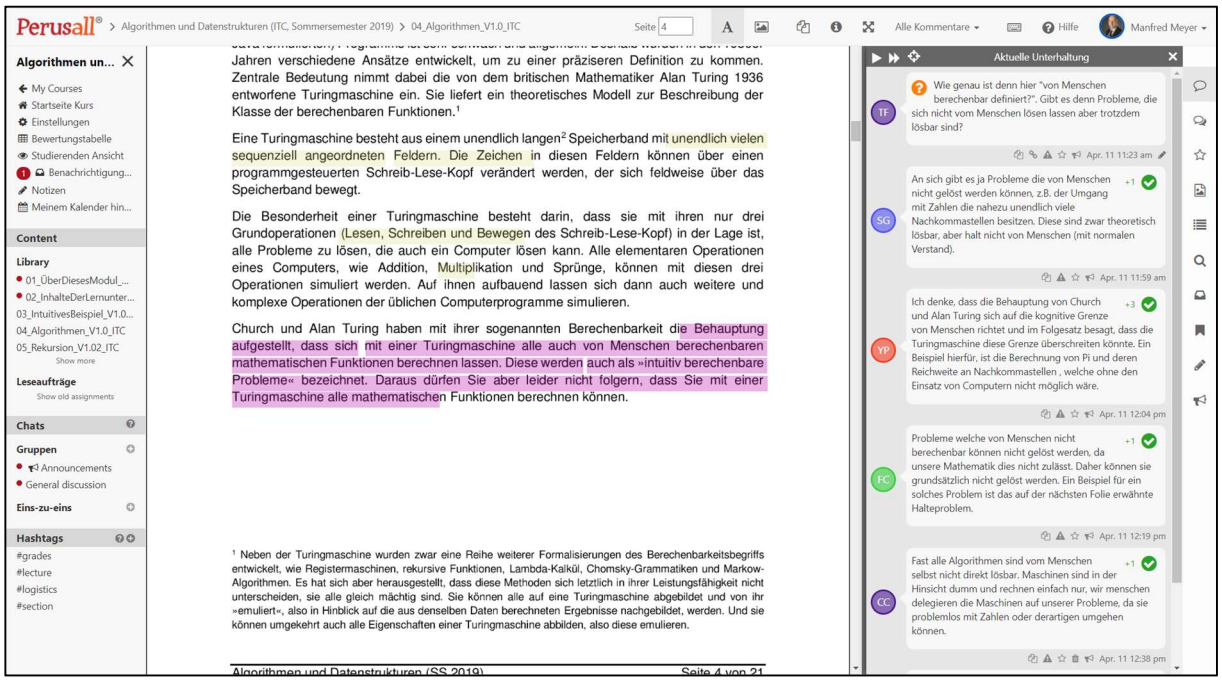

Figure 5. Example of an annotated and commented reading assignment

The so called "confusion report" provided by Perusall after the completion of an assignment generates a valuable summary based on intense discussions or open questions left by students and summarizing areas the students misunderstood, disagreed about, or were most engaged with. This allows us to tailor our classes around the questions that students asked between classes instead of reviewing the entire reading assignment or asking questions on students' progress prior to starting the class, which again saves time.

Thus, the confusion report helps to identify which topics should be discussed in class in depth while other aspects that did not cause any questions or discussions can be skipped completely leaving more time to use other didactic approaches and classroom activities for the hard issues. It specifically allows us to apply 'Just in Time Teaching (JiTT)' [14], a teaching and learning strategy designed to promote the use of class time for more active learning.

In order to ensure that students continuously submit comments, raise questions or answer those of their peers, this active participation on the social learning platform needs to be honoured. Thus, for each reading assignment, each student's participation is being evaluated and graded automatically by the platform according to the course settings defined by the instructor for the respective course. These

course settings include, for example, the weight assigned to the content of a student's comment itself, equal distribution of comments throughout the paper, upvoting comments from other students or getting comments upvoted by peers etc. Thus, for each assignment, all submitted annotations are graded. Based on these grades for each assignment and the cumulative grade table at the end of the course (see Fig. 6), credits can be given or bonus points be awarded that can later be forwarded towards the final exam.

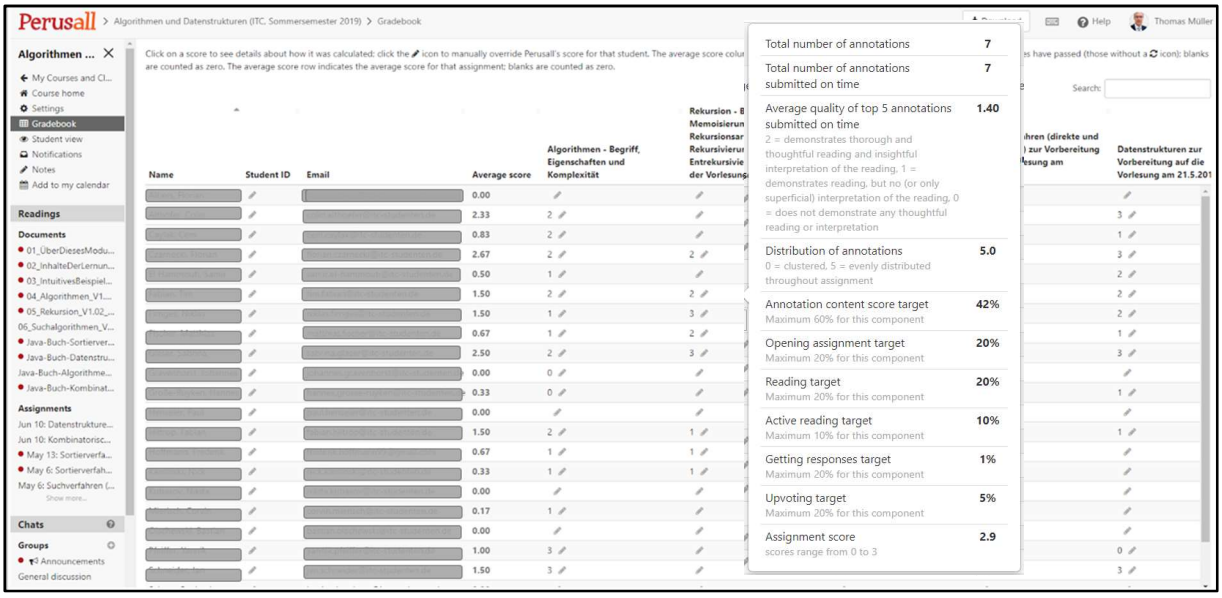

Figure 6. Perusall's Gradebook shows the earned bonus points of every single student

Having applied this approach to the "Algorithms and Data Structures" class at our institution, we were able to obtain some observations: Introducing a social learning platform like Perusall can definitely help encourage students to work on the subject matter beforehand and get well-prepared for class. Especially the social aspect of the learning platform and its various types of interaction with peers or even lecturers showed to be a substantial improvement to the reading assignment approach ("two-week-cycle" [15]) that was used before. However, implementing this approach still requires incentives like bonus points or credits awarded for pre-class activities (extrinsic motivation) which from a more formal point-of-view may not be desirable as they damage the uniqueness of the SOLO taxonomy mapping used in constructive alignment theory [16].

2.5 Peer Challenges and Peer Assessment

Having recognised that offering a 24/7 social learning platform contributes already well to increased student activation and engagement by enabling interaction and collaboration between peers, we then recently further extended our ongoing process towards introducing more and more peer-oriented activities.

Within a pragmatic approach for stepwise introduction of peer assessment elements, students are now invited to challenge their peers with their own programming exercises to be submitted through our Learn Management System (LMS) and evaluated by other students according to a predefined rubric and supervised by teaching assistants [17].

Therefore, in our programming classes we introduced weekly 'challenges': programming exercises submitted by students to be solved by their peers. They are currently used as continuous formative assessment throughout the lecturing period by which students can collect 'bonus points' (for both submitting and solving challenges) which can get credited up to 20% towards the summative assessment, thus reducing stress at the final exam.

As our programming courses have been designed as entirely competency-oriented and constructively aligned classes and exams (exam notebooks provided with IDE and state-of-the-art programming tools), continuously solving challenges provides a perfect training for the final exam. Moreover, by designing and submitting 'good' challenges, students demonstrate deeper understanding of the topic while balancing its complexity: a 'good' challenge is challenging enough to get solved by many but probably not all students.

The current implementation is integrated into the Moodle LMS where students can submit challenges through a form basically asking for the exercise statement (text) plus Java code template (optional) and test cases (input and expected output). Challenges are then checked by teaching assistants (TAs) for validity and understandability, and selected challenges get presented to all students on Moodle. With the use of the CodeRunner plugin, the provided test cases help students check the functionality of their solution while design aspects etc. are being assessed manually – by the authors of the respective challenge currently still being supervised by TAs for evaluation and ensuring assessment quality and fairness (see Fig. 7).

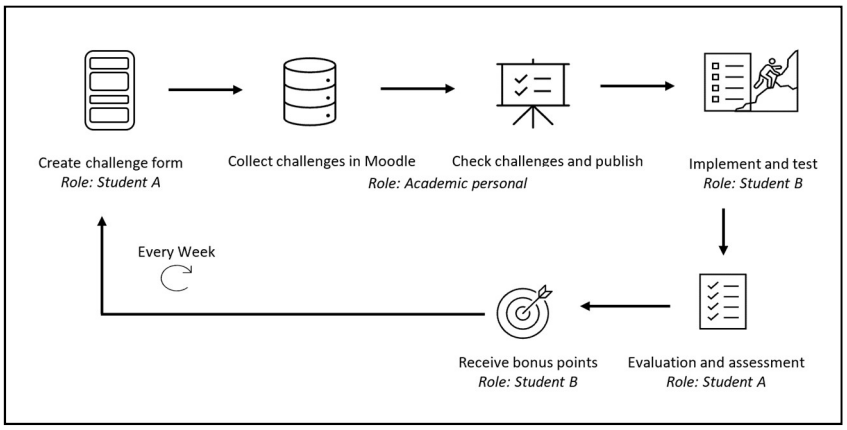

Figure 7. The Peer Challenges Process

Students are invited to submit so-called “Challenges” in Moodle through a predefined form using the Test feature of Moodle. Fig. 8 shows the student’s view on the process of submitting a challenge.

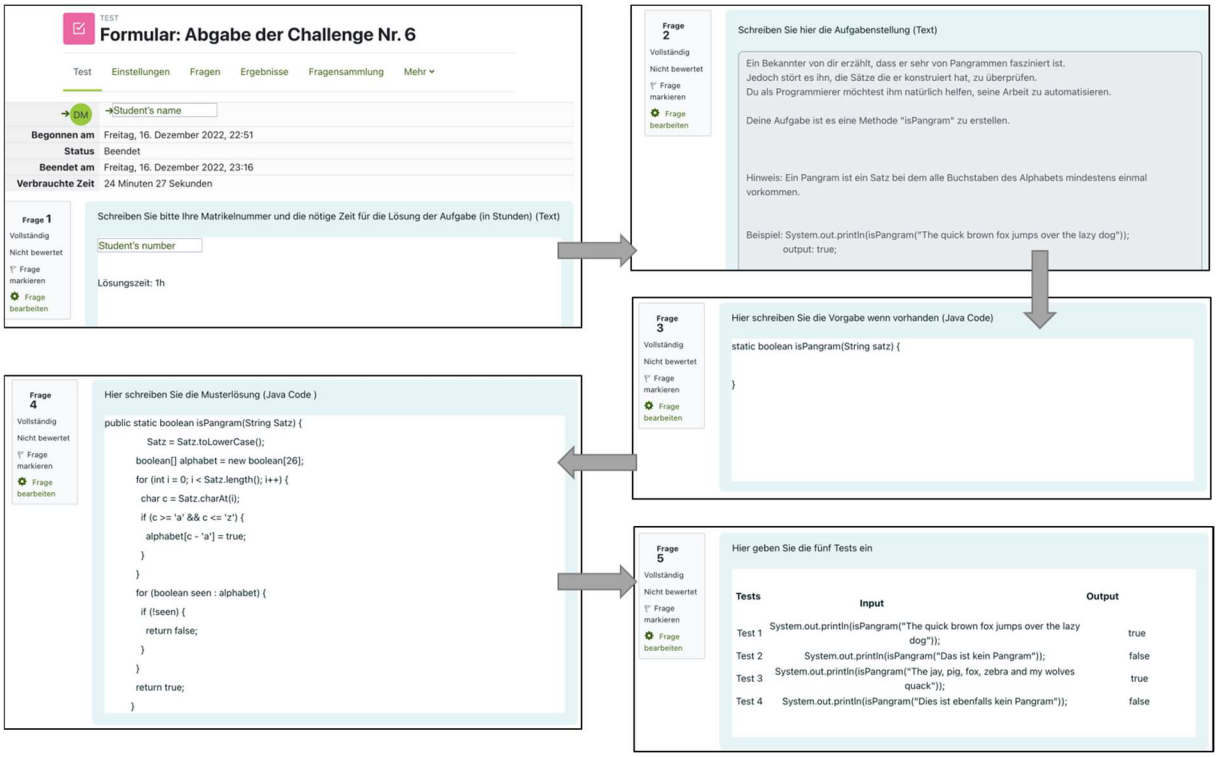

Figure 8. Submitting a new Challenge (student’s view)

It comes with five questions (“Frage”) asking for the student’s ID and the estimated time needed for solving the challenge (Frage 1) followed by the detailed introduction to the given exercise and explanation what to do (as plain text, “Frage 2”), a Java code template (“Frage 3”) , a sample solution (Java code, “Frage 4”) and concluded by a list of test cases, each consisting of a Java statement and its expected output (“Frage 5”).

Once a challenge is submitted, it gets checked by TAs for correctness and completeness, understandability, appropriate test cases, difficulty and time estimate to solve it. If a challenge passes these checks it gets published on Moodle, otherwise the student who submitted the challenge gets detailed feedback why it didn't get published and how to improve for resubmission. Additionally, authors of challenges being published receive up to three bonus points towards the final programming exam.

All accepted challenges - plus some ready-prepared challenges in case there are not enough submissions from students – get published on Moodle on a weekly basis giving students one week to work on them. While students can use any external IDE to develop a solution (Java program), they can also directly use the built-in editor that comes with the CodeRunner plugin for Moodle where students can iteratively work on the code until all visible test cases are being passed and the solution can be formally submitted as shown in Fig. 9. There is no time limit imposed and students may also work on challenges in teams, however solutions can only be submitted by individual students.

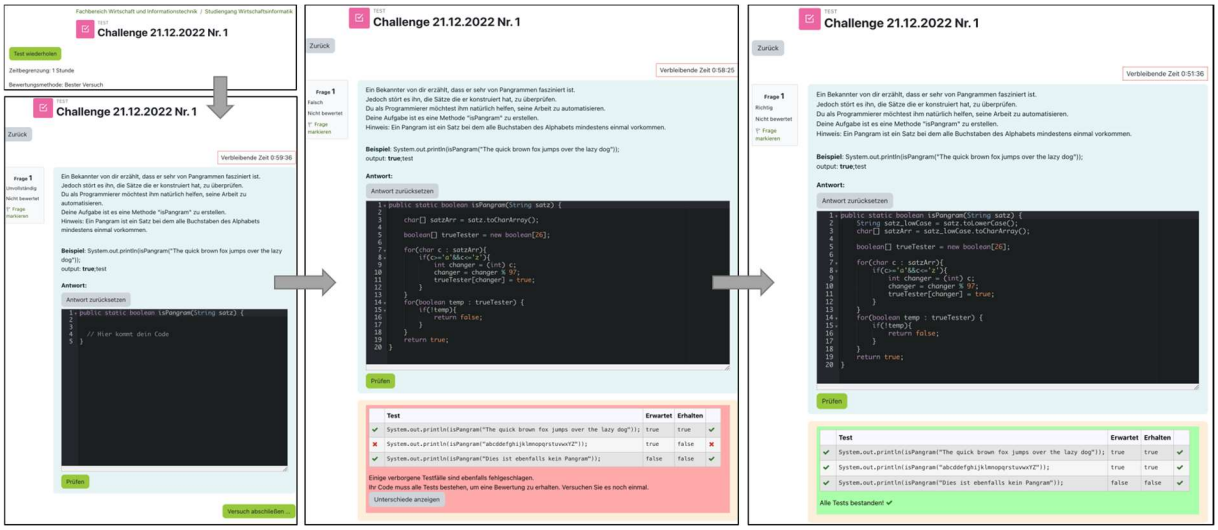

Figure 9. Solving a Challenge (student's view)

In the third phase, all submitted solutions to each challenge need to get evaluated. This is done also by students, normally by those who submitted the challenge (author). However, in rare cases students submitted a challenge but did not feel comfortable with also doing the assessment of their peers. In those rare cases, other students are invited to assess.

In order to ensure a fair assessment, a detailed rubric has been developed and all students doing assessments have been instructed beforehand how to apply this rubric to a given student's submission. However, especially in the beginning until students have developed some routine in using the rubric, they get supported by TAs.

Thus, students can collect bonus points both by submitting challenges that get selected for publication on Moodle as well as by solving a given challenge and submitting their solution also through Moodle. However, the overall number of bonus points that can be credited towards the final programming exam is limited to 20% in total by our university's examination regulations.

The peer assessment approach presented in this paper has been applied to first-year undergraduate programming classes (Introduction to Programming 1 and 2) within the Business Informatics program at our Bocholt campus. At the beginning, we first introduced challenges developed by TAs as additional training exercises. This helped testing the implementation in Moodle using the CodeRunner plugin and also activating students and getting them familiar with the process of submitting solutions through Moodle. The awarded bonus points also helped to motivate students to take part in this experiment. Around mid-term we then started inviting students to also submit their own challenges once a sufficient number of students worked on the given challenges quite regularly.

As expected, students showed to be much more hesitant when it came to developing and designing their own challenges, so it took some weeks and also support from TAs until we received challenges quite regularly such that most of the challenges presented to the students came from their peers. But still student submissions got evaluated and bonus points assigned by the TAs.

It appeared that students didn't feel very comfortable in being involved in the evaluation process and thus it needed some time and clarification of the evaluation criteria and standards. Only towards the end of the term, we were able to complete our peer assessment approach by also having the students themselves doing the evaluation of their peers' submissions and giving constructive feedback.

However, after having overcome this restraint, we now observe a small but constant stream of challenges being submitted. Students have now accepted our approach as an additional means for training for the exams while also earning some bonus points already. As there are up to three times more bonus points awarded to submitting challenges and doing the assessment afterwards than for just solving a given challenge, more and more students do also participate in this core part of our approach focusing not only on the programming training and learning aspect but also on developing analytical and communication competencies when it comes to give constructive feedback to their peers.

Summarizing, the informally gathered feedback from the students is now quite positive and our challenges approach is being accepted by most of the students while some students still hesitate to participate. However, a more formal validation of this recent approach to student activation still needs to be done after the end of the first year. Based on the results we will then derive conclusions on how to improve the process and its implementation in the future.

While our approach extends usual peer assessment by also asking students to design the assessment ('challenge') itself, its introduction in first programming classes also revealed several lessons learned: from student's participation (acceptance), the need for clear rubrics and prior assessment training to technical issues with Moodle integration.

3 LESSONS LEARNED

When the author started his teaching career more than 25 years ago, classes were managed mostly based on information transfer whereas students became passive listeners. Unfortunately, in many higher education institutions worldwide, this is still the case. However, passive learning comes with serious disadvantages mainly because students do not engage in the lesson. Therefore, we introduced various tools and techniques in order to activate and get students engaged in classes. These approaches need not be technology-based [18], e.g. when we used different color cards in times before introducing Interactive Voting Systems [4]. Also, active learning methods like Peer Instruction [7] could also be applied without using 'clickers' or alike beforehand. However, technology-enabled tools help using their full potential. On the contrary, the simple use of such tools does not really contribute to increase student engagement, at least not in the long run. As our experiences clearly show, to ensure long-term engagement it needs new engaging didactic concepts like Flipped Classroom, Peer Instructions, Gamification (Competitions), offering a Social Learning platform or even running Peer Challenges and Peer Assessment which then, of course, make use of state-of-the-art technologies (smartphone apps and web platforms available 24/7). Thus, with the use of the various approaches being presented and discussed in the previous section (and a few more), we have been able to turn the classroom over the years from a mostly unidirectional 'theatre' like event into a cooperative and participative learning venue.

However, when it comes to get the students engaged in out-of-class activities, e.g. using our social learning platform, we experienced some serious challenges beyond the control of an individual lecturer. Aiming at intrinsically motivated students that understand the pre-class reading and discussion of the subject matter as an efficient and effective way to achieve the best learning results, will require nothing less than a culture change towards an entirely different yet continuous learning process. As long as classical forms of teaching predominate (in which the subject matter is completely covered in the classroom course) it is difficult to prevent students from decaying into a consumer attitude.

Finally, as our Peer Assessment approach ('challenges') ultimately aims at completely substituting the final exam by continuous assessment throughout the lecturing period, this requires substantial changes to the examination regulations. However, preliminary feedback from students shows that the majority (but not all) like this approach both for continuous training for the exam and also for improving their marks by bonus points collected through challenges.

4 CONCLUSIONS

Activating students and keeping them engaged in the lessons has since long been identified as being crucial for long-term teaching and learning performance. With the methods and tools discussed in this

paper, we have been able to achieve significant improvements. However, it also has to be noted that further approaches aiming at continuous students' engagement even outside the classroom, e.g. through our social learning platform, or including not only the lecturing but also assessment aspects, need further support and a cultural change within the institution.

ACKNOWLEDGEMENTS

Some part of the research presented in this paper has been carried out as part of the project "Social Learning and Just-in-Time Teaching in Computer Science Education" funded through a personal grant ("Fellowship for Innovations in Digital Teaching and Learning") by the Ministry for Culture and Science of the State of North-Rhine Westphalia (MKW-NRW).

The recent work on Peer Challenges presented here is part of a Fellowship on Digital Examinations which the author got awarded by the initiative "Prüfung hoch 3" funded by the Stifterverband through the Friedrich-Alexander-University Erlangen-Nürnberg.

REFERENCES

[1] E. Kahu, and K. Nelson, "Student engagement in the educational interface: understanding the mechanisms of student success", *Higher Education Research & Development*, vol. 37, no. 4, pp. 1-14, 2017.

[2] K. Schnitzler, D. Holzberger, and T. Seidel, "All better than being disengaged: Student engagement patterns and their relations to academic self-concept and achievement", *European Journal of Psychology of Education*, vol. 36, pp. 627–652, Springer, 2021.

[3] J. Biggs, "What the Student Does: teaching for enhanced learning", *Higher Education Research & Development*, vol. 18, no. 1, pp. 57-75, 1999.

[4] M. Meyer, "Who wants to be a Millionaire? Or: How Interactive Voting Systems help Activating Undergraduate Students and Improving Teaching Performance", *Proceedings of the 3rd International Conference on Education and New Learning Technologies (EDULEARN11)*, Barcelona, pp. 6912-6921, IATED 2011.

[5] M. Meyer, T. Müller, and A. Niemann, "Serious Lecture vs. Entertaining Game Show – Why we need a Combination for Improving Teaching Performance and how Technology can help". *Proceedings of the 10th International Conference on Education and New Learning Technologies (EDULEARN 18)*, Palma de Mallorca, pp. 7030-7039, IATED 2018.

[6] P. Zhang, L. Ding, and E. Mazur, "Peer Instruction in introductory physics: A method to bring about positive changes in Students' Attitudes and Beliefs", Physial Review Physics Education Research, vol. 113, pp. 010104-1–9, 2017.

[7] E. Mazur, "*Peer Instruction: A User's Manual*", Prentice Hall, 1997.

[8] M. Olsson, and P. Mozelius, "*Game-Based Learning and Game Construction as an E-Learning-Strategy in Programming Education*", XI International GUIDE Conference and IX International EdTech Ikasnabar Congress - E-Learning 2016: New Strategies and Trends, Madrid, 2016

[9] J. Fendler, and M. Meyer: "The PrimeGame: Combining Skills in Undergraduate Computer Science Programmes", *Proceedings of the 4th International Technology, Education and Development Conference (INTED2010)*, Valencia, pp. 5454-5465, IATED 2010.

[10] M. Meyer, "The PrimeGame Reloaded: Finding the right Balance between Cooperation and Competition in Undergraduate Computer Science Classes", *Proceedings of the 3rd International Conference on Education and New Learning Technologies (EDULEARN11)*, Barcelona, pp. 5323-5330, IATED 2011.

[11] J. Dietrich, J. Tandler, L. Sui, and M. Meyer, "The PrimeGame Revolutions: A cloud-based collaborative environment for teaching introductory programming", *Proceedings of the 24th Australasian Software Engineering Conference (ASWEC 2015)*, Adelaide, pp. 8-12, ACM 2015.

[12] L. Sui, J. Dietrich, E. Heinrich, and M. Meyer, "A Web-Based Environment for Introductory Programming based on a Bi-Directional Layered Notional Machine", *Proceedings of the 21st Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2016)*, Arequipa, ACM 2016

[13] K. Miller, B. Lukoff, G. King, and E. Mazur, “Use of a Social Annotation Platform for Pre-Class Reading Assignments in a Flipped Introductory Physics Class”, *Frontiers in Education,* vol. 3, 2018.

[14] M. Camp, J. Middendorf, and C. Sullivan, “Using Just-in-Time Teaching to Motivate Student Learning”, in: S. Simkins, and M. Maier (ed.), *Just-in-Time Teaching: Across the Disciplines, Across the Academy*, Stylus Publishing, Sterling, 2010.

[15] M. Meyer, and T. Müller: If it were that easy: First Experiences of Introducing a Social Learning Platform in an Undergraduate CS Course. *Proceedings of the 12th International Conference of Education, Research and Innovation (ICERI 2019)*, Seville, pp. 10688-10697, IATED 2019.

[16] J. Biggs, and C. Tang, “*Teaching for Quality Learning at University: What the Student Does*”, 4th ed., Open University Press, Maidenhead, 2010.

[17] M. Meyer, “Challenges – Towards Continuous Peer Assessment in Undergraduate Programming Classes”, in: Mafalda Carmo (Ed.): *Education and New Developments Vol. 1*, World Institute for Advanced Research and Science (WIARS), 2023

[18] Y. Ghilay, and R. Ghilay, “TBAL: Technology-Based Active Learning in Higher Education”, *Journal of Education and Learning*, vol. 4, no. 4, 2015.