



# How to help digital-native students to successfully take control of their learning: A return of 8 years of experience on a computer science e-learning platform in higher education

Laurent Beaudoin<sup>1</sup> · Loïca Avanthey<sup>1</sup>

Received: 17 November 2021 / Accepted: 13 October 2022 / Published online: 4 November 2022  
© The Author(s) 2022

## Abstract

In this article, we present an educational reform we implemented a few years ago to respond to a marked drop in the success of students entering higher education in computer science. The main objective of our reform is to adapt the teaching methods of Generation X-Y to Generation Y-Z or more. To do this, we propose two approaches. First, to make learning more active, stimulating and empowering. Second, to make learning more individualized in a context of a large group of students with an optimization of the teacher's time. We first present an analysis of the probable reasons for the lower level of students and the specific issues encountered by both students and teachers. Then, we detail how we implemented these solutions in the form of an original e-learning platform based on two back-end tools able to manage large number of students: an efficient real-time auto-corrector of source codes and a very robust anti-plagiarism software based on computer distance information theory. Finally, we present the results obtained after an experimentation of 8 years. These results are put into perspective by an analysis of 15 years of qualitative and quantitative indicators carried out on hundreds of students each year. The analysis shows that the technical skills and the involvement of students are improved even in groups of several hundreds of students. Computer science teachers wishing to quickly and concretely develop the involvement of their students can rely on these tried and tested key levers and solutions presented.

**Keywords** Computer science education · E-learning platform · Automatic corrector · Automatic plagiarism detector · Auto-monitoring · Near real-time feedback

---

✉ Laurent Beaudoin  
laurent.beaudoin@epita.fr

Loïca Avanthey  
loica.avanthey@epita.fr

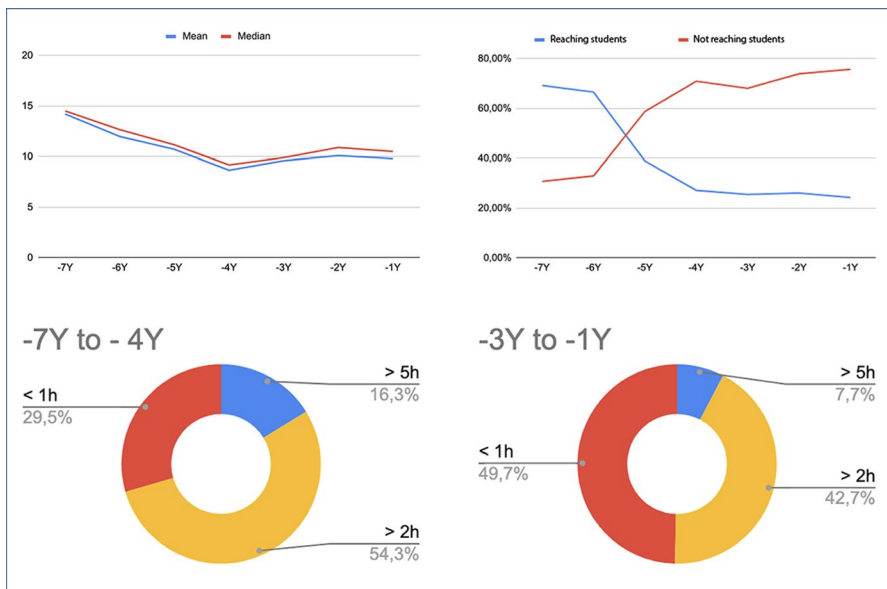
<sup>1</sup> SEAL (Sense, Explore, Analyse & Learn), EPITA Engineering school, 14-16 rue Voltaire, 94270 Le Kremlin-Bicêtre, France

## 1 Introduction

In the first decades of the 2000s, we observed a significant drop in the success of our French engineering school students in computer science on the first year course of algorithms design and programming. By success, we mean the ability of students to know how to apply their practical, theoretical and technical knowledge to new problems as well as the pleasure of working on the discipline. The quality indices we used to estimate this have deteriorated in a few years (see Fig. 1). The average and median of student grades over the first 7 years of the study show an average overall decrease of 22%. The percentage of students reaching the educational objectives has been divided by 2 over the period. And the average amount of weekly personal work reported by students, and therefore their personal engagement, also declined.

However, during this period, there was no significant modification at the institution level (same sequencing of courses, same team of teachers, same process of validation of prior learning, etc.), nor at the course unit (same hourly volume, unchanged theory / practice balance, etc.) which could explain this situation.

These problems of lack of success and involvement were not specific to our students but were also observed in other universities in France and abroad. At that time, digital natives (Sadiku et al., 2017) were entering higher education. They mark the transition from Generation Y to Z (McCrinkle, 2009). However, the learning methods and tools used were inherited from those created for Generation X with a small



**Fig. 1** Declining indicators of success over the first 7 years of the study. Top left: average and median of student grades. Top right: evolution of the percentage of students reaching the objectives. Below: average amount of weekly personal work reported by students over the 4 oldest years (left) and the 3 following years (right)

evolution to work with Generation Y. Taking into account the specifics of this Z generation in a educational strategy was the starting point of this study.

So we have therefore favored a global and generational approach to a purely local approach specific to our establishment. The objective was for our students to regain the expected technical level while enjoying learning and feeling more involved and therefore more responsible.

The results of this study can be used by computer teachers wishing to quickly and concretely develop the involvement of their students. They can rely on the key levers and the solutions presented which have been successfully tested over a period of time of several years.

In Section 2, we review the observed issues, from the point of view of both students from this generation and teachers. We present in Sections 3 and 4 the solutions we have created and deployed to solve these problems. In Section 5, we present and analyze the results obtained on thousands of students over 8 years of experience and conclude this article in Section 6.

## 2 Observations on the state of things

Digital natives do not present the same difficulties or facilities as their predecessors because they do not have the same prerequisites (Oblinger, 2003; Jones & Shao, 2011; Gu et al., 2013; Smith et al., 2020). With the democratization of digital technology in their daily lives (Tapscott, 1997; Howe & Strauss, 2000; Dede, 2005), they have been facing an over-demand of their attention from a very young age. They therefore have more difficulty staying focused and trusting to invest time (Duse & Duse, 2016; Nicholas, 2020).

They have developed a capacity to quickly switch from one task to another to the detriment of in-depth processing of each task (McCrinkle, 2009; Rosegard and Wilson, 2013; Bradbury, 2016). This could explain their difficulties with the teaching methods designed for the previous generation (Eastwood et al., 2012).

From these observations, we have established a questionnaire in order to be able to make a more detailed diagnosis of the problem encountered by our students and to highlight specific blocking points. These qualitative indicators complement the quantitative indicators presented in the previous section (grade, achievement of educational objectives, personal working time).

So that the proposed solution does not remain only at the level of the students, we decided to include in this study the problems encountered by the teachers to better respond to the operational constraints.

The methodology used to carry out this study is as follows. The indicators were calculated over 7 years before the implementation of the reform and 7 years after its implementation. This made it possible to monitor the evolution of the situation and objectively measure the impact of the reform by comparing the before and the after. The anonymous questionnaires were carried out through systematic annual campaigns as part of a quality approach. The proposed questions are either in the form of multiple choices or in the form of an open field. A questionnaire contains about 150 questions. In addition to these targeted questions, a questionnaire include

fields for free expression on each subject covered. Response trends were compared over the years for each subject. The size of the population surveyed is approximately 100 to 200 students per year, of which more than 80% respond to the survey. These systematic surveys were supplemented by oral interviews.

In the continuation of this section, we review the issues we have identified, from the perspective of both students (Section 2.1) and teachers (Section 2.2).

## 2.1 Observations on the issues for students

Table 1 shows some of the questions posed to students in the questionnaires. Trends are deduced from the percentages obtained in multiple-choice questions and are refined using open-ended questions and free expression fields. In the first analysis, we find on our scale the characteristics specific to Generation Z.

For example, in relation to the over-demanding of their attention, student responses to our surveys show that if there is no quick feedback on their work (ideally an interactive feedback), students quickly lose interest and are caught up in other things. However, they point out that this frequent individual feedback is difficult to obtain from the teacher in a collective teaching context.

Students declared that they need to be convinced that what they are currently doing makes sense and has a real and immediate benefit. These collected elements are consistent with our observations in the classroom. Consequently, we observe through the evolution of the student results during the year and their investment in class that this amplifies their discouragement during long processes of acquiring skills or bringing complex projects to fruition. This is quite paradoxical because at the same time they declare in our surveys and during our interviews that they are particularly attracted by difficult subjects. This attraction to challenge is a

**Table 1** Excerpts to show examples of the questions asked in the survey

<p><b>Subject: Corrections to exercises</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> I don't mind getting a correction soon after I finish an exercise</li> <li><input type="checkbox"/> I prefer to have a correction quickly after I finish an exercise</li> <li><input type="checkbox"/> If I don't have the corrections quickly, it blocks me from moving forward</li> <li><input type="checkbox"/> If the correction comes too quickly, I don't have time to complete the exercise by myself</li> <li><input type="checkbox"/> ...</li> <li>• Free field of expression on the correction of exercises: _____</li> </ul>
<p><b>Subject: Waiting time for an answer to a question in class</b></p> <ul style="list-style-type: none"> <li>• The waiting time for an answer to a question in class: <ul style="list-style-type: none"> <li><input type="checkbox"/> suits me</li> <li><input type="checkbox"/> is quite fast</li> <li><input type="checkbox"/> is too long</li> </ul> </li> <li>• What is the maximum tolerable waiting time according to you to have an answer to a question in class: _____</li> <li><input type="checkbox"/> ...</li> <li>• Free field of expression on the waiting time: _____</li> </ul>

particularity of this generation (Gibson et al., 2009). However, they seem rather powerless to autonomously set up a strategy to give themselves the means of their ambition.

Although our teaching is aimed at higher education students, the answers of the students to the surveys and the analysis of the corrections of the returned exercises show a certain naivety when it comes to the acquisition of skills. Thus, a growing number of students do not finish their work at the end of the supervised session if they are not coerced. They are satisfied with the collective correction, considering it sufficient. This lack of mastery then leads to an accumulation of gaps and then to a dropout when it becomes too great to be filled by an approximate understanding.

Moreover, through the exercises that they return to us for correction, we observed that students tend to stay on the surface when solving problems. They also tolerate a certain approximation in the results. In programming, for example, we observe in lab class that they stop working as soon as their results seem consistent with the requirements. They do not think of borderline cases and exceptions that can fail their program. Typically, they spontaneously spend little time thinking before coding and do not visualize what they should get before running their program. They also take little time to analyze the result obtained or to think about the tests to put in place to qualify their work. Student responses to our surveys and during interviews corroborate these observations.

Besides, we were able to observe by comparing the results when the practical course followed the theoretical course and when they were distant in time, that the students have more difficulties in appropriating theoretical concepts if these are not fixed quickly by practice and put into perspective in a global strategy of learning. Student responses to our surveys show that poor management of the time elapsed between theory and practice leads them to feel that they have wasted their time in class and makes these theoretical sessions less effective pedagogically.

Finally, our students report that the final exam used to assess the acquisition of skills is often seen as a sanction. The stakes are very high (it can affect the validation of the whole semester or year) and generates a lot of stress, including for the best students. Only a part of the educational program is actually assessed considering the length of the exam and students should be at their best at that time. This leads to a feeling of mistrust in the face of the assessment.

In addition, the format of the computer assessment is, for logistical reasons, often done on paper while the learning is done on a machine. All this pushes students to defines strategies, such as cramming or rote learning, to optimize the grade obtained on the assessment to the detriment of acquiring and mastering real skills. It questions the meaning of the final exam as an objective assessment of the skills acquired.

## 2.2 Observations on the issues for teachers

It is important to involve teachers in the process of creating educational reform. The objective is to include real operational constraints in the reform so that teachers can improve the service provided to students at constant working hours.

The main problem brought up by the teachers to set up regular evaluations is a question of time cost, whether in terms of balance (time spent on evaluation versus total teaching time) or on the time spent on correction (which is directly related to the size of the group of students). In our case, the characteristic sizes vary between fifty and several hundred students.

This problem often leads to reducing the number of evaluations during the activity which mechanically increases the stakes of these few assessments. These stakes make it difficult for the teacher to write the subjects. It is indeed necessary to find a subtle balance between the exhaustiveness, the complexity, the novelty and the duration of the evaluation. Fewer evaluations also imply testing many concepts at the same time making it more difficult to appreciate sources of misunderstandings.

In addition, feedback to students takes a long time (time required for correction). This lag of feedback can be problematic to catch up students who are in learning debt. The more time passes, the worse the situation gets. It can also make the correction totally ineffective because the student is no longer in the right cognitive context when receiving the feedback.

Also because of the cost of time, it is very complicated to organize a re-assessment once a student has re-worked the concepts. Evaluation therefore loses its value as a learning tool.

The corrections of ungraded exercises in class partially solve the previous problems and are, therefore, interesting complementary tools to exams. However, as the correction is done in a synchronized way for the whole class, this tool does not adapt to the diversity of individual rhythms. Students who have already finished have to wait when they want to move on. And the others must stop to listen to the correction which deprives them of the expected thinking on the subject. Furthermore, it is impossible with these global corrections to verify that all the students have dealt with all the scenarios of the exercise.

Finally, in computer science, correcting codes manually is tedious and it is complicated to systematically check all cases or to detect plagiarism because of the volume of data to be processed.

### 2.3 Analysis of observations and solutions

In this part, we present the analysis of the previous observations to define the needs that our solution must meet.

In the data collected from our students, it appears that they have difficulty maintaining their attention on the course, tend to become demotivated and present a rather detached and passive attitude. Our main problem therefore seems to be a problem of engagement toward traditional teaching methods.

Student engagement is a long-recognized key to having a successful learning process (Astin, 1984; Hancock & Betts, 2002; McMahon & Portelli, 2004; Krause & Coates, 2008). Engagement is often described through three components: behavioral engagement, affective engagement and cognitive engagement (Fredricks et al., 2004; Appleton et al., 2008). Some authors like (Handelman et al., 2005) decline these components in a slightly different but quite similar way: interaction engagement,

emotional engagement, skills engagement and performance engagement. In a more descriptive way, (Schlechty, 1994) says that engagement means that the students are attracted to their work, that they persist in it despite difficulties, and that they take pleasure in accomplishing it.

There is three main approaches to stimulate student engagement. The first one is active learning methods (Bonwell & Eison 1991; Azzalis et al., 2009; Weltman and Whiteside, 2010). In this vast category, we find, for example, problem-based approaches that make it possible to put things into practice and to give meaning (Delisle, 1997; Lattimer & Riordan, 2011). The second one relies on web platforms to offer richer interactions. These methodes play on many characteristics to stimulate and capture attention (Lytle et al., 2006; Liu, 2007; El-Sheikh, 2009; Dixon, 2010; Mavromoustakos & Kamal, 2018). The third one uses simulation games or serious games in order to offer a more lively experience to facilitate involvement and therefore learning (Ruohomäki, 1995; Cai et al., 1997; de Freitas, 2006; Breuer & Bente, 2010; Council, 2011). Without going into educational formats in the form of games, it is possible to use game levers in learning. This is called gamification and it stands as a good tool to generate engagement, in particular by stimulating and capturing attention (Gee, 2003; Fitz-Walter et al., 2011; Lee & Hammer, 2011; Raymer, 2011).

Due to the length of our course (two semesters), problem-based or serious/simulation games approaches are not quite appropriate even if they can be used for sub-parts. We therefore turned to web-based solutions that also integrate gamification levers.

The second issue that emerges from the data collected in the previous section concerns the need for individualized learning in a large group context. Studies like Garrison & Cleveland-Innes (2005) have shown the impact of self-paced learning on the quality of learning and teaching methodologies. Our web platform must therefore be able to implement robust tools allowing a large number of students to progress at their own pace while benefiting from rapid individualized feedback on their work for a constant working time by the teacher.

These constraints on fast and individualized interactive feedback in a context of large groups resemble those found in algorithmic competitions such as ACM-ICPC (2018). These events rely on automatic correctors.

Some of these automatic correctors, like Mooshak (Leal & Silva, 2002), had already been tested in pedagogy with encouraging results (Leal & Moreira, 2000; Georgouli & Guerreiro, 2010) still valid a few years later (Fernandez Aleman, 2011; Rubio-Sanchez et al., 2012; Rodrigues et al., 2014; Rubio-Sánchez et al., 2014). They were good at processing a massive number of codes on the fly, but generally they did not make it possible to modulate the response of the corrector according to the level of the student. Nor to deal with code snippets rather than entire programs to allow tailor-made teaching on a case-by-case basis. Moreover, their human-machine interface were unsuitable for stimulation and visualization of educational progress. The reader interested in the characteristics of the correctors available at the time and created since, can refer to the studies of Ihantola et al. (2010) or Wasik et al. (2018).

In a pedagogical context where it is necessary to be able to grade the students, it is particularly important to detect the attempts of fraud to guarantee a fair treatment

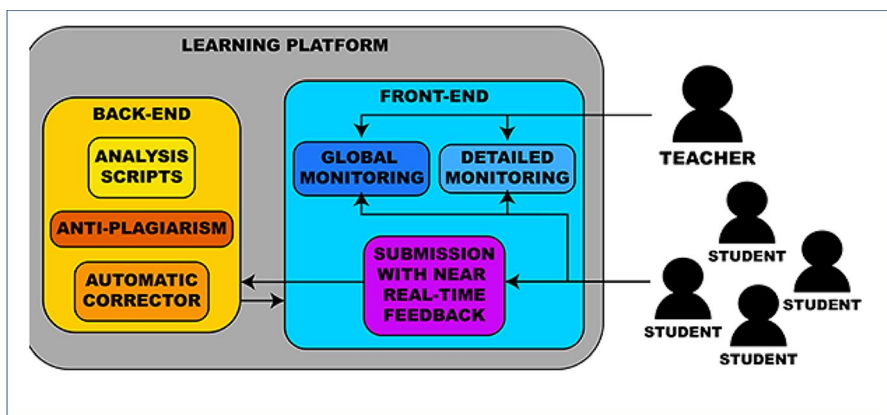
of the students. In a context of a huge number of programs submitted to an automatic corrector, this plagiarism detection must also be automated. The tools available at the time (Chowdhury & Bhattacharyya, 2016; Deokate & Hanchate, 2016) were either not suited to the comparison of source codes, or not open-source, or not popular enough to be known outside their borders.

To summarize all the needs expressed in this analysis section, the proposed solution must take the form of a learning platform. This platform can be web-based and uses gamification levers to stimulate student engagement. And it should include both automatic code corrector and plagiarism detector to enable fair individualized learning in the context of a large group of students.

At the same time, Khan (2013) made the same observation and started with a solution that could today be called a MOOC (Massive Online Open Course): the KhanAcademy (2008). Originally focused on mathematics and physics, the ambition of this learning platform is to be accessible to all students on the planet with internet access. This solution was closest to our needs, but it was neither free nor adapted to our face-to-face educational needs. Indeed, our needs were closer to what is now called a SPOC (Small Private Online Course) because our course was not purely online.

### 3 Description of the e-learning platform developed

We have therefore developed an internal e-learning platform (see Fig. 2). The front-end learning interface collects students codes and shows the learning progress. The back-end part is composed of an automatic corrector and an automatic plagiarism detector, both of which are fully optimized for real-time and on-the-fly source code analysis. In this experimentation, the codes submitted by the students was in C/



**Fig. 2** A learning platform must allow effective global and detailed monitoring for both the student and the teacher. Associated with real-time feedback on the student's submissions, the visualization of progress should be stimulating to encourage the student to work regularly until the acquisition of all the required skills



C++, but the platform could be adapted to learning other programming languages, as this will only impact the analysis scripts of the back-end part. This platform has enabled us to implement the two main levers of our reform: the monitoring of learning progress and the individualization of learning, both in a collective context (hundreds of students). We detail these levers in the next Sections 3.1 and 3.2 sections. As for the tools, they are detailed in Section 4.

### 3.1 Monitoring of learning progress

Self-monitoring learning progress gives students a clear view of their situation and the paths to follow to complement their skills and knowledge in the course they are taking. The global overview for teachers allows them to identify at a glance the students in difficulty, those who drop out, but also those who advance more quickly than the others. It also helps to see if the group dynamic is good and if the workload is well dimensioned. For example, a general delay in the whole group should question the teachers. In addition to the synthetic visualization proposed on the interface (see Section 4.1), we mainly worked on three axes within this context: sequencing, pace and evaluation of learning.

Regarding sequencing, we created learning sequences composed of the succession of a theoretical course (1h30), a very supervised practical session (1h30) and a longer and more autonomous session (3h). In practice, we have obtained the best results by completing one learning sequence in a single day per week. This allows students to immediately put into practice what they have seen in theory and thus more easily establish their knowledge. All teaching has been redesigned around these learning sequences and we have reworked the distribution of key concepts by favoring the rule: one learning sequence, one key concept. Thanks to the e-learning platform, students have a clear visibility over all the learning sequences that will be covered in the course.

Along with sequencing, controlling the pace of learning is also important. Thus, chapters and exercises are added as the lessons progress and not all at once at the start. This allows students to focus on the present and push them to dig deeper. In a normal situation, students aim to complete all the exercises of the learning sequence before the next one. At the end of this period, we choose to close the normal repository and open a new one to allow students to finalize their work while realizing that they are behind the schedule. This approach empowers the students and allows the teachers to detect very early cases at risk of dropping out.

The last important point for a relevant and effective monitoring concerns the evaluation. Basically, the assessment should estimate the level of skill acquisition at a given point. Thus, the really important information is whether a skill is not acquired, in the process of being acquired, acquired or mastered.

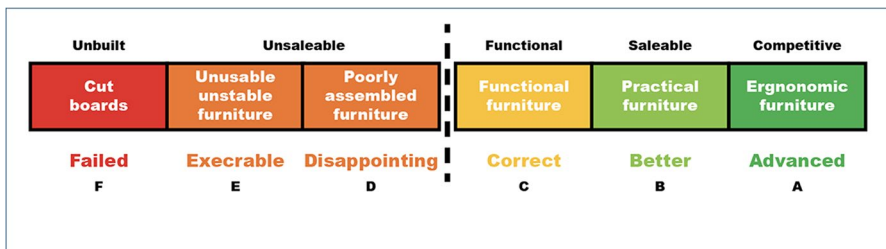
Thanks to the time freed by the tools detailed in Section 4, we have chosen to set up theoretical and practical evaluations and re-evaluations systematically for each learning sequence. This high frequency, in addition to producing regular feedback, makes it possible to de-dramatize the assessment. Besides, the

possibility of validating skills later than the normal rhythm makes it easy to manage complicated situations (accident, illness, professional work to help finance studies, etc.).

To help students exploit the feedback from these assessments and prevent “aim-for-the-average” bias, we have revised the scoring system in the form of a profile. We divide each macroscopic skill into notions belonging to three categories: fundamental, general and advanced. These categories are clearly identified in the assessments to make them transparent and useful to students. Macroscopic validation of skills is acquired when a minimum profile is reached in each category (percentages of concepts assimilated).

The low-level concepts of the fundamental category are considered essential and must be fully understood by students: failing in this category means failing the assessment. Exercises in the general category require a first level of mastery and reflection. And concepts of the advanced category require a reformulation of the knowledge.

In addition, we have abandoned the French numerical rating system used until now, which had too much granularity (0 to 20) and which masked the importance of the skills behind them. We turned to the Angloaxon alphabetic grade systems (Kumar & Sharma, 2014) which had just been introduced in France with the European Credit Transfer and accumulate System (ECTS) (Commission et al., 2009). But ECTS is based on a relative grading scale: the final numerical marks are translated into alphabetical marks by a Gaussian interpolation (A for top 10% of admitted students, etc.). However, in our learning framework, reaching the pedagogical objectives is more important than the relative rank of the student in the group. We therefore decided to use an absolute system, based on the profiles obtained in the different categories mentioned above (basic, general and advanced). We use a scale of 6 grades (A-F), but rather than letters, we have chosen to use appreciations covering the same semantic field for the teacher and the student: Advanced, Better, Correct, Disappointing, Execrable, Failed. We also provide students with a concrete analogy so that they understand the relationship between skills and grading. This original grading scale is shown in Fig. 3. The



**Fig. 3** Our new graduation scale that replaces numerical notes (0 to 20) by six appreciations that covers the same semantic field for teachers and students. A concrete analogy with the construction of furniture is given to the students to make it easier to understand why they cannot be satisfied with an “average mark”. To give an idea, being in category Correct (C) corresponds to have a grade between 60% and 80% of the maximum grade on our previous rating scale

importance of verbalizing what a rating represents rather than just a numerical rating was also highlighted by Pangaro and McGaghie (2005); Hanson et al. (2013).

### 3.2 Individualization of learning

The e-learning platform developed offers the possibility of individualized learning. That is to say adapted to the pace of the students, accompanying them at their level, and encouraging them to retrain or to test other solutions.

Indeed, dematerialization and the permanent availability of tools allow them to optimize their personal organization. They can work at the time most convenient for them or easily catch up on a delay, for example. Besides, the automatic archiving of all submissions encourages students to try other approaches or ideas without the risk of seeing their rating drop. This encourages students to rework alone the exercises worked in pairs in class. This option also allows them to retrain after a certain period of time or to retry an exam until they pass, even if the assessment is complete.

For struggling students, the automatic correction make it easy to add additional or intermediate exercises to increase training and improve comprehension. This allows them to consolidate not only their base of skills and knowledge but also to build confidence in themselves and in their ability, an essential ingredient in success without having a significant cost in time for the teacher.

In the same way, we can better manage high-performing students thanks to optional exercises called “challenges”. These exercises cover less essential parts and can be very demanding. If unsuccessful, they have no effect on the final grade. We saw in Section 2.1 that Generation Z loved challenges. The introduction of this type of exercises helped lift all students to the top.

Bringing assessment back as a teaching tool made it possible to empower students by making them responsible for their learning while facilitating a relationship of trust with the teacher. The reader interested in this last point can consult (Antibi, 2003, 2007, 2014).

Feedback on exercises is also personalized in relation to student profiles. When students are inexperienced with both programming and automatic correctors, some hints help them to better understand why they failed a given test. When they are more experienced, the feedback is more succinct. This encourages them to gradually formulate and test the right hypotheses and thus gain autonomy.

## 4 Developed tools behind the e-learning platform

We will now detail the tools as well as their main functionalities which make it possible to implement the levers seen in the previous section. We discuss the front-end learning interface in Section 4.1, the automatic code corrector in Section 4.2 and the automatic plagiarism detector in Section 4.3.

### 4.1 Front-end learning interface

In this section, we discuss the main functionalities of interest of the front-end learning interface developed for the student on the one hand Section 4.1.1 and for the teacher on the other hand Section 4.1.2.

#### 4.1.1 Features for students

Figure 4 summarizes the use cases for students. Students can log in to the learning platform either alone or with another student when they are working in pairs. Then, they can access the monitoring page which contains a visual summary of their progress in carrying out the requested exercises. Self-monitoring of acquired skills is done intuitively and instantly, in global (chapters) and in detail (exercises of a chapter) thanks to a color code which reflects the feedback from the evaluation (see Fig. 5 for the meaning of the symbols used in our monitoring).

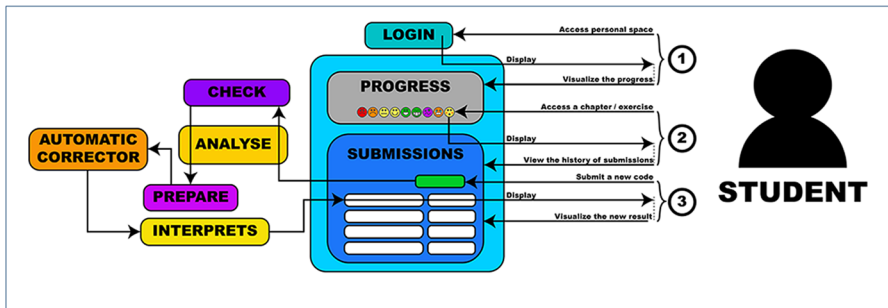


Fig. 4 Use cases of our e-learning platform for students. They must be able to log in (1), see their monitoring and statistics, access chapters and exercises to submit a code (2) and view the results obtained (3)

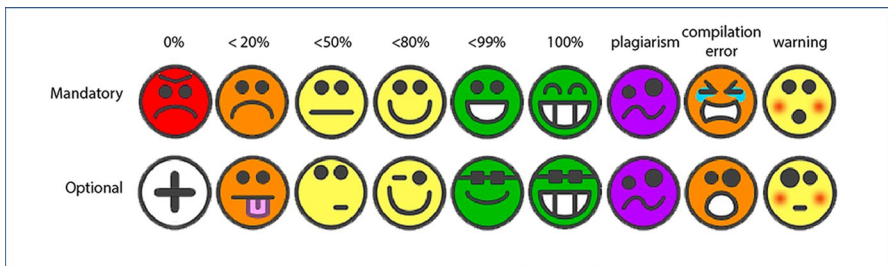


Fig. 5 We use smileys of different shapes and colors for displaying progress on our e-learning platform. Green indicates that the skills have been acquired, yellow stands for in progress, orange means a work still insufficient, red stands for no work and purple means plagiarism. A distinction is made between mandatory (“you must finish this exercise”) and optional exercises (“you may find interest in trying this exercise”) thanks to the different faces of the smileys

From the home page (see Fig. 6), students can select a chapter to access the list of different exercises. History of code submissions, associated results and feedback are accessible for a given exercise. Students can make code submissions

## Autocorrect

Hello, **student\_name** ([Logout](#))  
 In pairs ? [Identify your buddy!](#)

[student\\_name stats](#)

[student\\_name settings](#)

[P3 Loops & Branches](#)  
[P6 Iterative vs Recursive](#)

[P4 Compiler messages](#)  
[Test N°1](#)

[P5 Working with arrays](#)  
[P7 Strings](#)

### Status

**Student**

*Students can submit their programs to the corrector.*

### Groups

*You belong to the group(s): P12-12 P12-12A*

### General informations

Number of submissions:	10
Oldest submission:	10/09/2011 at 09:23:28 (Exponentiation (test))
Most recent submission:	13/11/2011 at 10:12:42 (GCD (2nd session))

### Settings

[Change my password](#)

[See the details of the submissions](#)

## P3 Loops & Branches

## P4 Compiler Messages

**Fig. 6** The home page that students see when they log into the learning platform. It displays the summary of their current evaluation (self-monitoring) as well as their statistics and it gives them access to the different chapters and exercises to submit programs

24/7, solo or in pairs, from any place (at school or elsewhere) as long as an internet connection is possible and the deposit is still authorized. Students have developed and tested their code with the tools of their choice (integrated development environments, command-line tools, etc.) before submitting a version that they think is correct.

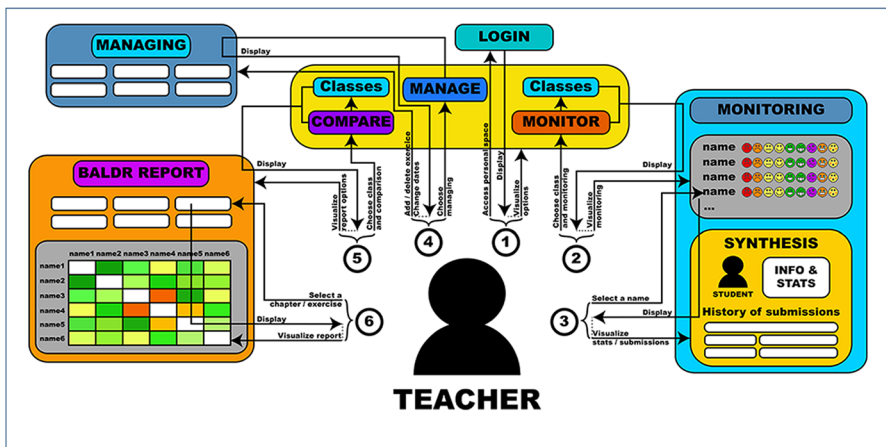
The submitted code is then analyzed by a back-end script that performs a series of actions such as checking for prohibited functions or breaking up the source code into separate functions.

After that, the submitted code enters the automatic corrector (see Section 4.2) which returns its result to the interface for formatting. Students then have feedback on the success of the exercise. The total duration, from submission to the display of the result, takes only a few seconds.

#### 4.1.2 Features for teachers

Figure 7 summarizes the use cases for teachers. When they connect to the platform, they can access the monitoring interface which contains the complete list of their students (by class) with the summary of their progress (see Fig. 8). Then, they can access the details of the submissions of each student. Finally, the results can be exported for a group, for a given exercise or for a selection of exercises in order to facilitate the creation of scoring grids.

On the exercise management interface, teachers can easily add, modify or delete chapters, exercises and test sets. A set is made up of one or more pairs of inputs and outputs. Teachers can also manage the opening and closing dates of a chapter or an exercise according to the group of students concerned, as



**Fig. 7** User cases of the platform for teachers. After logging in (1), they can choose a class to check its monitoring (2) and access the individual monitoring of a student (3). Teachers can also manage chapters and exercises (4). Finally, they can access plagiarism analyzes (5) on the submissions of one or more classes for a given exercise (6)





### 4.1.3 Summary of the features and actions implemented with regard to the educational objectives

In Section 2 we saw the issues from the perspective of both students and teachers. In Section 4.1 we saw the main functionalities offered by the developed e-learning platform. In this section, we summarize the impact of each of these features on the educational objectives to solve the issues. As seen previously in Section 3 we separate our educational objectives into three categories. The first one is focused on “sequencing” (Table 2) the second one on “pace” (Table 3) and the third one is dedicated to “evaluation” (Table 4). In these tables, we have specified who activates the features ([S] for student, [T] for teacher) and who benefits from them. Then, in the following sections, we detail the two back-end tools that support the e-learning platform.

**Table 2** Summary of the features and actions implemented with regard to the educational objectives concerning sequencing ([S] student, [T] teacher)

Sequencing	
Features / Actions implemented	Educational goals
1. Creation of learning sequences around key concepts [T]	<ul style="list-style-type: none"> <li>◦ Gives meaning [S]</li> <li>◦ Facilitates attention focus [S]</li> </ul>
2. Grouping learning sequences into macroscopic themes [T]	
3. Grouping T / visualization [T+S] of exercises created on the e-learning platform by category (theme → learning sequence)	<ul style="list-style-type: none"> <li>◦ Facilitates the identification of key points not mastered [T + S]</li> </ul>

## 4.2 Back-end automatic code corrector: Autocorrect software

The corrector is a script that is launched in its sandbox for each code submitted. It starts by compiling the code and checks that there are no error or warning. If an error occurs, the analysis is stopped. Otherwise, the analysis continues, but if there is a warning, it is signaled to encourage students to correct it even if the result is correct.

After the compilation step, the corrector runs the obtained program with the different datasets and tests associated with the corresponding exercise. Some are randomly chosen from a set of tests to prevent brute-forcing. For each test, the program output is compared with the expected output. These must be strictly identical to validate the exercise. A single failure causes the whole exercise to fail. The result is then sent to the interface to display it to students (see Fig. 9).

## 4.3 Back-end automatic plagiarism detector: Baldr software

When using the results of the automatic corrector to make a graded assessment, it is essential to be vigilant about identity theft (someone pretending to be another) or



**Table 3** Summary of the features and actions implemented with regard to the educational objectives concerning pace ([S] student, [T] teacher)

Pace	
Features / Actions implemented	Educational goals
1. Bringing the practical sessions closer to the associated theoretical sessions [T]	<ul style="list-style-type: none"> <li>◦ Facilitates attention focus [S]</li> <li>◦ Avoids the feeling of wasting time [S]</li> <li>◦ Avoids repeating the course [T]</li> </ul>
2. Gradually revealing [T] / discovering [S] the exercises on the e-learning platform according to the progress in the course	<ul style="list-style-type: none"> <li>◦ Stimulates [S]</li> <li>◦ Incites to deepen [S]</li> </ul>
3. Set a time limit for submitting exercises on the e-learning platform [T]	<ul style="list-style-type: none"> <li>◦ Stimulates [S]</li> <li>◦ Incites to finish the work [S]</li> </ul>
4. Easily schedule exercise session openings and closings in advance [T]	<ul style="list-style-type: none"> <li>◦ Saves time in organization [T]</li> </ul>
5. Reopen [T] / redo [S] exercises whose initial sessions are closed [T]	<ul style="list-style-type: none"> <li>◦ Encourages to rework [S]</li> <li>◦ Encourages to fill in the gaps [S]</li> <li>◦ Materializes underachievement [S]</li> <li>◦ Encourages to check the anchoring of skills [S]</li> </ul>
6. Add [T] / carry out [S] optional complicated exercises (challenges)	<ul style="list-style-type: none"> <li>◦ Stimulates [S]</li> <li>◦ Reinforces the level of mastery [S]</li> <li>◦ Incites to deepen [S]</li> </ul>
7. Add [T] / [S] carry out optional intermediate or complementary exercises	<ul style="list-style-type: none"> <li>◦ Adapts to the level [S]</li> <li>◦ Gives confidence [S]</li> <li>◦ Reinforces the mastery [S]</li> </ul>
8. 24/7 submission of exercises on the e-learning platform [S]	<ul style="list-style-type: none"> <li>◦ Adapts to the pace [S]</li> </ul>
9. Immediate correction with feedback	<ul style="list-style-type: none"> <li>◦ Adapts to the pace [S]</li> <li>◦ Stimulates [S]</li> <li>◦ Encourages progress [S]</li> <li>◦ Incites to finish the work [S]</li> </ul>

plagiarism (taking someone else's code and submitting it). Dematerialization facilitates this type of fraud attempt.

We can fight against identity theft by regularly organizing assessments with mandatory physical presence and then by ensuring that the results obtained are consistent with the usual results. Another positive factor in organizing such sessions is to reduce student stress by turning assessments into routine. The time cost for the teacher is minimal as it is reduced to managing only the logistics thanks to the

**Table 4** Summary of the features and actions implemented with regard to the educational objectives concerning evaluation ([S] student, [T] teacher)

Evaluation	
Features / Actions implemented	Educational goals
1. Use of a visual (colored smileys) / semantic (appreciations) rating scale [T]	<ul style="list-style-type: none"> <li>○ Gives meaning [S]</li> <li>○ Facilitates interpretation [S]</li> <li>○ Helps to better understand the points in progress and the gaps [S]</li> </ul>
2. Intuitive visualization of the individual global summary of results (theme or learning sequence) [T+S]	<ul style="list-style-type: none"> <li>○ Stimulates [S]</li> <li>○ Facilitates the identification of acquired / non-acquired [T+S]</li> <li>○ Gives a better view of the progress [T+S]</li> <li>○ Empowers [S]</li> </ul>
3. Intuitive visualization of the class global summary of results (theme or learning sequence) [T]	<ul style="list-style-type: none"> <li>○ Helps detect dropouts [T]</li> <li>○ Gives an idea of the class dynamics [T]</li> </ul>
4. Export scoring grids [T]	<ul style="list-style-type: none"> <li>○ Saves time for teaching [T]</li> </ul>
5. Automatic correction of submissions	<ul style="list-style-type: none"> <li>○ Saves time for teaching [T]</li> <li>○ Guarantees a more complete and rigorous correction [T+S]</li> <li>○ Offers greater amount of correction [S]</li> <li>○ Gives feedback adapted to the work carried out [S]</li> </ul>
6. Correct exercises in the form of multiple unit tests [T+S]	<ul style="list-style-type: none"> <li>○ Gives feedback adapted to the work carried out [S]</li> <li>○ Helps to conceptualize [S]</li> <li>○ Introduces the notions of code safety [S]</li> </ul>
7. Having the history of submissions [T+S]	<ul style="list-style-type: none"> <li>○ Helps to visualize progress or problems [T+S]</li> <li>○ Encourages testing new approaches [S]</li> <li>○ Facilitates rework (reassuring) [S]</li> </ul>
8. Accessing the correction report [T+S]	<ul style="list-style-type: none"> <li>○ Helps to understand errors [T+S]</li> </ul>
9. Adjusting the level of detail of the correction report (more or less indices) [T]	<ul style="list-style-type: none"> <li>○ Adapts to the level [S]</li> <li>○ Gives confidence [S]</li> <li>○ Empowers [S]</li> </ul>
10. Systematic automatic plagiarism test	<ul style="list-style-type: none"> <li>○ Empowers [S]</li> <li>○ Facilitates control (focus on suspicious cases) [T]</li> <li>○ Helps distinguish plagiarism vs help [S]</li> <li>○ Encourages the production of original personal work [S]</li> </ul>

The screenshot displays a feedback interface for a student's submission. On the left, a table lists exercises and their results:

Exercise	Result
Load Array	OK
Show Array	OK
Minimum	OK
Minimum (minimum at index 0)	OK
Maximum	OK
Maximum (maximum at index 0)	OK
Minimum index	OK
Minimum index (redundant minimum)	IR
Maximum index	IR
	<i>Invalid result, have you checked all cases? (see tip n°2)</i>
Mode	NF
	<i>Function not found, did you write the correct prototype? The expected prototype is: int getMode(int AArray[], int size)</i>
Mode (Mode = 20 and Mode=0)	NF
	<i>Function not found, did you write the correct prototype? The expected prototype is: int getMode(int AArray[], int size)</i>

On the right, a terminal window shows the execution of three test cases:

```

*****
Test n°0
Loading the array:
Minimum -83 has at least 2 occurrence.
Showing the array loaded :
-83 98 -41 70 47 -83 51
The first indice of the minimum (-83) of the array is: 1166454032
*****
Test n°1
Loading the array:
Minimum -89 has at least 3 occurrence.
Showing the array loaded :
92 -40 -88 -88
The first indice of the minimum (-88) of the array is: 2
*****
Test n°2
Loading the array:
Minimum -89 has at least 1 occurrence.
Showing the array loaded :
69 62 58 91 -8 -89 -46
The first indice of the minimum (-89) of the array is: 5
*****
Are you sure of these values?
If the local check does not give the same result, have you handled all cases of input parameters?

```

Fig. 9 Screenshot example of a feedback given by the corrector to a student after a submission

automatic correction. Aside from that, to fight against plagiarism, we have developed an original and specific detection tool detailed in this section.

### 4.3.1 Theoretical foundations

Plagiarism detection consists of finding common information content in different files. The whole difficulty lies in the fact that this common content is not strictly identical in its form but is in its meaning. Consequently, it is a very complex problem to quantify this notion that is in essence difficult to define or model.

In science, this notion is at the crossroads of many disciplines, such as mathematics, physics or more recently biology and computer science for example, which explains the many attempts to define it. Among them, we have retained those directly related to computer science and more specifically to algorithmic theory of information, also called Kolmogorov's theory of complexity (Li & Vitanyi, 2008).

To understand the principle, we must go back to the myth of Occam's razor (Thorburn, 1918). Originally created to manipulate theological concepts, it was later adopted by scientists around the 17th century. This principle advocates that the choice between different theories compatible with the facts must be made in favor of the simplest. The notion of information, seen through this prism, could thus be defined as the simplest theory.

But then we come up against another subjective notion: simplicity. It took the work of mathematicians like Ray Solomonoff, Gregory Chaitin, Per Martin-Löf and especially Andreï Kolmogorov and Leonid Levin (Solomonoff, 1964; Kolmogorov, 1965; Martin-Löf, 1966; Zvonkin and Levin, 1970; Levin, 1984; Chaitin, 1987, ...) around the 1970s to rigorously link the notions of information on a computer and simplicity (or more precisely complexity) and to define associated metrics (Delahaye, 2006).

Any computer object (text file, source code, multimedia, binary, etc.) is ultimately a series of bits with a value of 0 or 1. In this universe, any program capable of generating

exactly this series of bits becomes an exact variant of the computer object and the information it contains. This remains true regardless of the original nature of the file, which is intuitively compatible with the universality of information.

If we assume that each program is a theory and that the simplest is the smallest program (in number of bits), we can now rigorously calculate the notion of information (Kirchherr et al., 1997) even if its content is not definable, as it is the case with the notion of entropy in signal processing. This is called the Kolmogorov complexity (Li & Bitanyi, 1997).

From an algorithmic point of view, we can exploit any regularity in an object (i.e. in the sequence of bits) to shorten the programs that describe it. For example, if an object is symmetrical, it suffices to describe the pattern and the symmetry to apply. However, the exploitation of patterns to reduce data is exactly the goal of compression algorithms. Among these, lossless compression algorithms make excellent estimators of Kolmogorov complexity (Cilibrasi & Vitanyi, 2005).

Plagiarism detection can be included in a more general problem, which is automatic classification. In the computer universe, Kolmogorov complexity arises as a natural criterion of classification. The first works on this subject focused on the classification of genetic sequences. It defines the notion of informational distance between sequences of characters as being the size of the shortest program allowing to transform one string into another (Bennett et al., 1998; Varré et al., 1999). The idea was then perfected and simplified to be now known and used under the name of similarity distance (Li et al., 2004). The first uses of this distance focused on the classification by compression of pieces of music (Cilibrasi et al., 2004).

### 4.3.2 In practice: application to source codes

Let us illustrate the principle with an example. Let  $A$  and  $B$  be two files and  $K(A)$  and  $K(B)$  their compressed version by a lossless algorithm  $K$  (see Eq. 1). Each file is composed of a common part  $C_{AB}$  and a distinct part  $D$ .

$$\begin{cases} A &= C_{AB} + D_A \\ K(A) &= K(C_{AB}) + K(D_A) \end{cases} \quad \begin{cases} B &= C_{AB} + D_B \\ K(B) &= K(C_{AB}) + K(D_B) \end{cases} \quad (1)$$

The result of the concatenation of the two files ( $A + B$ ) and its compressed version ( $K(A + B)$ ) are presented in Eq. 2: the compressor  $K$  eliminates all redundancies.

$$\begin{cases} A + B &= 2 \times C_{AB} + D_A + D_B \\ K(A + B) &= K(2 \times C_{AB}) + K(D_A) + K(D_B) = K(C_{AB}) + K(D_A) + K(D_B) \end{cases} \quad (2)$$

In plagiarism detection, what interests us is obviously the common part  $C_{AB}$  and the weight it represents with respect to the totality of the files  $A$  and  $B$ . Equation 3 shows us how to get  $K(C_{AB})$  with the elements we can calculate.

$$\begin{aligned} K(C_{AB}) &= K(A + B) - K(D_A) - K(D_B) \\ \Rightarrow K(C_{AB}) &= K(A + B) - K(A) + K(C_{AB}) - K(B) + K(C_{AB}) \\ \Rightarrow K(C_{AB}) &= K(A) + K(B) - K(A + B) \end{aligned} \quad (3)$$

So we just need to compress each file  $A$  and  $B$  as well as their concatenation ( $A + B$ ). The next step is to calculate a distance between the informational contents of files  $A$  and  $B$  from  $K(C_{AB})$ . The similarity distance proposed by Cilibrasi et al. (2004) can be developed to obtain a generalized form (see Eq. 4). We use this last formula on the compressed codes compiled for our plagiarism detection algorithm. The smaller the similarity distance, the more similar the files are and therefore the greater the likelihood of plagiarism.

$$d(A, B) = \frac{K(C_{AB}) - \min(K(A), K(B))}{\max(K(A), K(B))} \quad (4)$$

In our case, we are trying to detect plagiarism in a relatively small set (scale of few tens to hundred individuals). So we can afford a systematic calculation of all distances between all the files two by two. This gives us a table of distances. The following section details how to use it.

### 4.3.3 Using Baldr

The implementation in Java of the previous theoretical work was carried out by our colleague Hubert Wassner. This cross-platform open source software is called Baldr, in reference to the Scandinavian god of light and truth. Its technical operating principle is explained by Wassner (2014). In this section, we mainly focus on the educational benefits of this software.

Analyzing a rendering of several tens or hundreds of files is quite simple: specifying the directory containing them for automatic processing is the only requirement. The analysis is very fast (on the order of a second). The first result displayed is a histogram of the inter-file similarity distances (see Fig. 10, top).

This graph allows us to quickly get an overall idea of the risk of fraud and its extent. Indeed, as the most suspicious cases correspond to the smallest distances, we can focus on the leftmost columns of the histogram. The area of this class indicates the number of files concerned. So ideally, if no peak stands out on the left, one can be reasonably confident that there is no fraud. Otherwise, the attention of the teachers is focused on these contentious renderings. They have the option at this stage to adjust the width of the histogram classes to avoid detection bias due to poor sampling.

The next step of the analysis displays the table of similarity distances (see Fig. 10, bottom). The distances appear in a color gradient from red for the smallest to green for the largest. It is therefore very easy to identify suspected cases. And finally, teachers can automatically launch a code comparator on these few cases so they can validate whether or not it is indeed a fraud attempt (see Fig. 11). By default, the software used by Baldr is Kompare (2016).



Fig. 10 Screenshots of two graphic results rendered by Baldr. On the top, the distribution of the distances measured between the codes. On the bottom, extract from the 2D visualization of the distances between each program pair (symmetrical table). The color graduation makes it possible to focus easily on the most suspicious codes

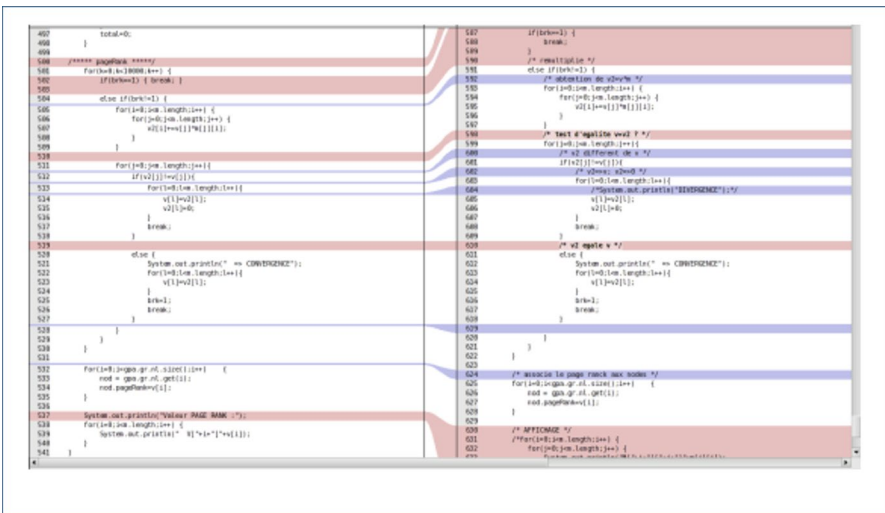


Fig. 11 Screenshot of the visual comparison of two suspicious codes so a human can confirm if it is plagiarism

## 5 Experience feedback and results

As we have seen in the previous sections, the main objective of the presented reform is to adapt to Generations YZ the teaching practices created for Generations XY. To do this, this reform uses two main levers:

1. first, individualize learning in a large group context to ensure that the students reach the expected technical level,
2. second, increase student engagement with a more empowering and stimulating pedagogy.

We have assessed the effectiveness of the actions implemented thanks to the quantitative and qualitative indicators.

As a reminder, the population studied corresponds to 100 to 200 students per year, who are around 18 years old and who are doing their first year in higher education in computer science. The quantitative indicators were calculated annually on 100% of students and the qualitative indicators come from annual systematic surveys to which more than 80% of students responded. The study covers a total period of 15 years divided into a period of 7 years before the reform and another of 7 years after. The students of the first period serve as a control group to evaluate the progress of the students of the second period.

During this period, the extrinsic factors did not change before and after the reform: same administrative functioning of the school, same team of teachers, same sequencing in relation to other subjects, same number of hours of lessons, etc. Furthermore, the starting level (a priori knowledge) of the students was not different over the years concerned by the study. Thus, we can reasonably conclude that the results observed are mainly due to the reform.

### 5.1 Technical level & Individualization

The evolution of the average and the median of the student grades and the percentage of students reaching the educational objectives are presented for the whole period of the study in Fig. 12. In a first global analysis, all the indicators came back positive and even improved compared to the initial situation.

In more detail analysis, students from the reform period better master technical key skills. For example, 25% more students feel comfortable using Linux after the reform, 21% more students feel comfortable using pointers, 17% more students say they know how to write programs using the greedy paradigm and 14% more students say they know what Bachmann-Landau notation corresponds to. As the content of the courses has not changed significantly, this significant difference can be attributed to the reform.

According to the teaching team, the major changes concerning the evaluation of student work have been particularly decisive in this evolution of operational technical mastery. As seen in previous sections, these changes relate to the frequency of assessments, their systematic nature and their individualization.

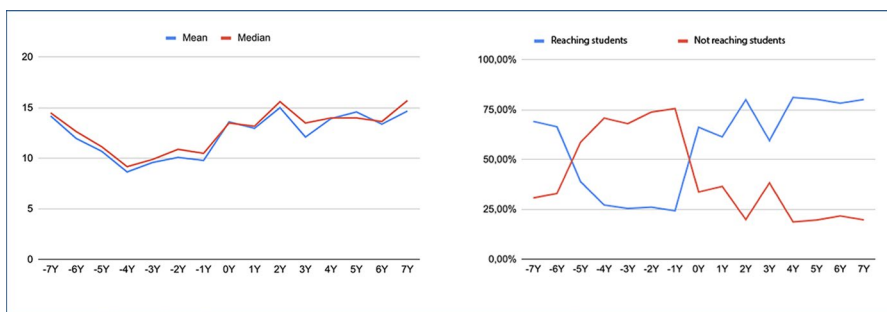
Regarding the frequency of evaluation, we have concretely gone from a monthly evaluation at best to a continuous evaluation. A side effect of this change of pace is a profound change in the way students experience evaluation. If we combine the parts of the questionnaire concerning the stress linked to individual evaluations, 67% of students on average declare that they are not very stressed by them (contrary to 24% on average before the reform). Moreover, the answers of the students also show that most of them (72% against 46% before the reform) see a real correlation between their personal work and their success in the assessments.

Regarding the systematic nature of the assessment, this approach prepares students for what is practiced in industry with systematic unit testing and gives them good programming safety habits. For example, an average of 79% of codes submitted mid-year fail the “divide by 0” test on exercises that include a divide by a variable. This percentage drops to 18% at the end of the year. Students also correct this type of error much faster.

Regarding the individualization of the assessment, the introduction of intermediate exercises makes it possible to catch up with students who would have dropped out normally. 18% of students carry out these intermediate optional exercises. Besides, the challenging optional exercises stimulate good students who would normally be bored. More than 30% of students tackle these challenges and 7% achieve them in full. From a personal satisfaction standpoint, students report that they are proud to have completed these challenging exercises (one testimonial even speaks of the pleasure comparable to beating the end-of-level boss in a video game).

Baldr is also very effective at detecting the most original solutions. If these contributions regularly come from the same students, then Baldr also makes it easy and quick to identify the most promising students.

Besides, the number of students having multiplied by 1.5 since the start of the reform with an average increase of +15% per year, we can also conclude that the solutions put in place are robust to the significant increase in workload.



**Fig. 12** Analysis of the evolution of indicators over a period of 15 years. The reforms and the software suite were implemented in year 0. On the left, the average and median of student grades. On the right, the evolution of the percentage of students reaching the educational objectives



## 5.2 Empowerment & Stimulation

As part of our reform, we have based the empowerment and stimulation of students on two main levers: helping them to understand the meaning of what they are doing on the one hand and developing their autonomy on the other.

Concerning the meaning of their teaching, the students better perceive the overall plan and the logical links between the lessons (62% instead of 48%). They are also sensitive and grateful that we take the time to explain to them the meaning of what they do (84% say it is important to them).

Likewise, the transition from numerical grades to appreciations was effective in raising the objectives that the students set for themselves to validate an assessment. We observed an immediate and positive psychological effect to this clarification. Indeed, about 40% of students were previously satisfied with a 9/20 because they thought they were close to validation. With the corresponding grade “Disappointing” about 80% of students in this category react to have a better grade.

Concerning the autonomy, it goes through self-assessment and personal involvement. About self-assessment, 94% of students find that colored smileys are very practical and effective in use. Some students (12%) even asked for more evaluations to have a finer control of their progress. 71% of them find that the automatic corrector gives a reliable estimate of their technical level. We expected a higher percentage, but since a significant part of the students (31%) find that the rigidity of the corrector on the format of the results is oversized (an extra space in the response is sufficient to decline a submission), it is likely that this has weighed on this aspect.

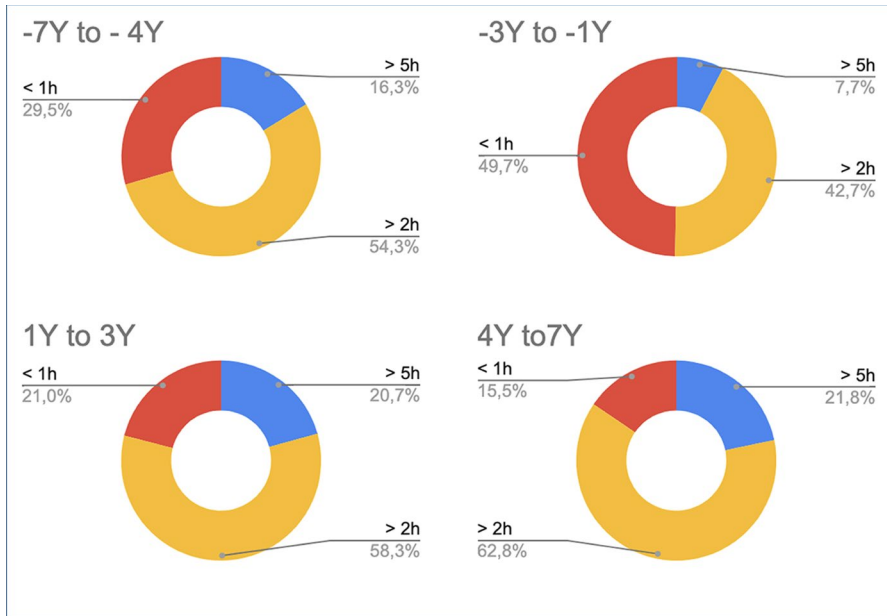
In order for the self-assessment to be reliable and trustworthy, it is important to effectively combat code submission fraud. For this, an early live demonstration of the effectiveness of the Baldr tool in front of all the students is done. 93% of students find that this demonstration convinced them that it is easier to work without cheating than to cheat without being detected. Thanks to this effort of prevention, transparency and systematic use of the tool, attempts at fraud are almost non-existent.

So that the time spent working is faced in pleasant conditions for the students, we have taken particular care to make the educational platform fun and addictive. 67% of the students appreciate this point. It should explain why the platform was adopted very quickly and rather perceived as a serious game. For 83% of students the discovery of new visuals as they progressed had a dynamic effect.

74% of students say the short delay between code submission and interface feedback helps them maintain their attention. This prompts the students to immediately rework their code to obtain a full validation (gaming effect), as the following testimony shows: “I felt like I was being challenged and I had a week to do it. I really liked it.”

The empowerment of students and their stimulation generated by the educational platform have resulted in an increase in the work provided by the students. Indeed, more than 20% of students declare a higher personal working time (of which around 10% is much higher) as can be seen at Fig. 13.

This personal working time is partly used by 92% of the students to finish the exercises within the given time (1 week), by 16% of the students to rework and resubmit old exercises in order to ensure that the skills are well mastered, and by 9%



**Fig. 13** Evolution of the average amount of weekly personal work declared by students over the four quarters of the total study period

of the students to catch up with a dropout (prolonged absence, period of demotivation, personal problems, etc.).

## 6 Conclusion & Discussion

A few years ago, we observed a significant drop in the success of students taking engineering courses in computer science. The analysis of the problems encountered led us to an educational reform. In this article, we focused on two main levers to better adapt to the particularities of digital natives students: the development of a more active, stimulating and empowering learning and the individualization of learning in a collective context. We presented how we implemented these levers through a learning platform. The front-end learning interface includes real-time visualization of progress and use gaming levers. The back-end part consists of an automatic corrector and an original automatic plagiarism detector based on similarity distance of computer sciences information.

15 years of systematic surveys allowing the extraction of quantitative and qualitative indicators have shown two main results.

First, there is a general rise in the technical level of the students of the reform period with a constant working time for the teacher in a large group context. The analysis shows that this is mainly thanks to more frequent, systematic and complete evaluations that include more understandable assessments and exercises

fitted to individuals. Among other statistics, there is a 30% increase in the average and median of student grades in just a few years after the reform.

Second, students are more involved and have significantly increased their personal working time. The key reasons highlighted by the analysis are a better understanding of the global schema and of their individual positioning towards the expectations, as well as individualized and stimulating feedbacks. Indeed, we observe a 20% increase in personal weekly working time declared by students after the reform.

There are, however, several limitations to this study. Firstly, there is an essential part of face-to-face interaction in the system put in place. There is therefore no guarantee that this experiment would have given good results with a completely online approach. Secondly, there is a risk that students get used to and get tired of the platform over time, which lowers their interest and therefore their engagement (this concerns 4% of the students in our study for example). We must therefore be vigilant so that the front end often evolves and remains attractive to successive generations, because their interests change very quickly. Finally, a last limitation is to be vigilant about the amount of content accessible to students. Indeed, if we propose too much content, we could obtain the opposite effect of what is sought by demotivating some student profiles.

In terms of perspective, we can discuss the tools available to create this kind of learning platform nowadays. At the time, we had developed everything from scratch. Today, there are sufficiently modular and adaptable automatic correctors available, such as DOMjudge (2010), which can be used directly in the back-end Wasik et al. (2018); Pham and Nguyen (2019). The same goes for plagiarism detectors: many softwares have been developed in this direction (Gomes & Matos, 2020; Iffath et al., 2021) and can also be easily integrated into the back-end. However, what is still lacking today are off-the-shelf tools that combine these features as well as feedback and progress management through a configurable interface. The learning management platform (Moodle, 2022), for example, is practical for organizing sequencing and quiz, but less to display a self-monitoring feedback, or to automatically correct codes. Nevertheless, the development tools available today make it possible to quickly obtain dynamic and attractive interfaces. This is a transition that we are currently making.

**Acknowledgements** We would like to thank our two colleagues at the time, Hubert Wassner who developed the automatic plagiarism detector and Antoine Gademer who led this reform with us and participated in the development of the e-learning platform and the automatic corrector.

**Funding** This research received no external funding.

**Data availability** Open-source anti-plagiarism software developed, Baldr, accessible at: <https://sourceforge.net/projects/baldr/>. Language: Java. Operating system: platform independent. Licence: GPLv2.

## Declarations

**Conflicts of interest** The authors declare that they have no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as

you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- ACM-ICPC. (2018). International Collegiate Programming Contest: an algorithmic programming contest for college students. <https://icpc.global/>. Accessed 2022.
- Antibi, A. (2003). *La Constante Macabre Ou Comment A-t-on Découragé des Générations D'élèves ?* France: Math'Adore.
- Antibi, A. (2007). *Pour en Finir Avec la Constante Macabre Ou Les Notes : La Fin du Cauchemar*. France: Math'Adore.
- Antibi, A. (2014). *Pour des élèves Heureux en Travaillant Ou Les Bienfaits de L'évaluation Par Contrat de Confiance*. France: Math'Adore.
- Appleton, J. J., Christenson, S. L., & Furlong, M. J. (2008). Student engagement with school: Critical conceptual and methodological issues of the construct. *Psychology in the Schools*, 45(5), 369–386. <https://doi.org/10.1002/pits.20303>
- Astin, A. W. (1984). Student involvement: A developmental theory for higher education. *Journal of College Student Personnel*, 40, 518–529.
- Azzalin, L., Sato, S., de Mattos, M., Fonseca, F., & Giavarotti, L. (2009). Active learning versus traditional teaching. *Journal of Biochemistry Education*, 7(2), 415–436.
- Bennett, C., Gacs, P., Li, M., Vitanyi, P., & Zurek, W. (1998). Information distance. *IEEE Transactions on Information Theory*, 44(4), 1407–1423.
- Bonwell, C. C. & Eison, J. A. (1991). Active learning: Creating excitement in the classroom. In: ERIC Digest Higher Education Reports
- Bradbury, N. A. (2016). Attention span during lectures: 8 seconds, 10 minutes, or more? *Advances in Physiology Education*, 40(4), 509–513.
- Breuer, J., & Bente, G. (2010). Why so serious? On the relation of serious games and learning. *Journal for Computer Game Culture*, 4(1), 7–24.
- Cai, Y., Goei, S. L. & Trooster, W. (1997). *Simulation and Serious Games for Education*. Association for Supervision & Curriculum Development, Singapore.
- Chaitin, G. (1987). *Algorithmic Information Theory*. England: Cambridge University Press.
- Chowdhury, H. A. & Bhattacharyya, D. K. (2016). Plagiarism: Taxonomy, tools and detection techniques. In: 19th National Convention on Knowledge, Library and Information Networking (NACLIN)
- Cilibrasi, R., & Vitanyi, P. (2005). Clustering by compression. *IEEE Transactions on Information Theory*, 51(4), 1523–154.
- Cilibrasi, R., Vitanyi, P., & de Wolf, R. (2004). Algorithmic clustering of music based on string compression. *Computer Music Journal*, 28(4), 49–67.
- Commission, E., Directorate-General for Education, S. Youth (2009). *Culture: ECTS Users' Guide*. Publications Office, ??? <https://doi.org/10.2766/88064>
- Council, N. R. (2011). *Learning Science Through Computer Games and Simulations*. The National Academies Press, ??? <https://doi.org/10.17226/13078>
- de Freitas, S. (2006). Learning in immersive worlds: a review of game-based learning. Technical report
- Dede, C. (2005). Planning for neomillennial learning styles. *Educause Quarterly*, 28(1), 7–12.
- Delahaye, J.-P. (2006). *Complexité: aux Limites des Mathématiques et de L'informatique*. France: Belin / Pour la science.
- Delisle, R. (1997). *How to Use Problem-based Learning in the Classroom*. Association for Supervision & Curriculum Development, United States Delisle.
- Deokate, B., & Hanchate, D. B. (2016). Software source code plagiarism detection: A survey. *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, 3(1), 3747–3750.

- Dixon, M. D. (2010). Creating effective student engagement in online courses: What do students find engaging? *The Journal of Scholarship of Teaching and Learning*, 10(2), 1–13.
- DOMjudge (2010) An automated system to run programming contests. <https://www.domjudge.org/>. Accessed 2022.
- Duşe, C.-S. & Duşe, D.-M. (2016). The teacher of the generation z. In: 7th International Conference Edu World
- Eastwood, J. D., Frischen, A., Fenske, M. J., & Smilek, D. (2012). The unengaged mind: defining boredom in terms of attention. *Perspectives on Psychological Science*, 7(5), 482–495.
- El-Sheikh, E. (2009). Techniques for engaging students in an online computer programming course. *Journal of Systemics, Cybernetics and Informatics* 7
- Fernandez Aleman, J. L. (2011). Automated assessment in a programming tools course. *IEEE Transactions on Education*, 54(4), 576–581. <https://doi.org/10.1109/TE.2010.2098442>
- Fitz-Walter, Z., Tjondronegoro, D. & Wyeth, P. (2011). Orientation passport: Using gamification to engage university students. In: Proceedings of the 23rd Australian Computer-Human Interaction Conference. OzCHI '11, pp. 122–125. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2071536.2071554>
- Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Educational Research*, 74(1), 59–109. <https://doi.org/10.3102/00346543074001059>
- Garrison, D. R., & Cleveland-Innes, M. (2005). Facilitating cognitive presence in online learning: Interaction is not enough. *American Journal of Distance Education*, 19(3), 133–148. [https://doi.org/10.1207/s15389286ajde1903\\_2](https://doi.org/10.1207/s15389286ajde1903_2)
- Gee, J. (2003). *What Video Games Have to Teach Us About Learning and Literacy*. England: Palgrave MacMillan.
- Georgouli, K., & Guerreiro, P. (2010). Incorporating an automatic judge into blended learning programming activities. In X. Luo, M. Spaniol, L. Wang, Q. Li, W. Nejdil, & W. Zhang (Eds.), *Advances in Web-Based Learning - ICWL 2010* (pp. 81–90). Berlin, Heidelberg: Springer.
- Gibson, J. W., Greenwood, R. A., & Murphy, J. E. F. (2009). Generational differences in the workplace: personal values, behaviors, and popular beliefs. *Journal of Diversity Management*, 4(3), 1–9.
- Gomes, K. & Matos, S. (2020). Detection of programming plagiarism in computing education: A systematic mapping study. In: XXXI Simpósio Brasileiro de Informática na Educação, pp. 1633–1642
- Gu, X., Zhu, Y., & Guo, X. (2013). Meeting the “digital natives”: Understanding the acceptance of technology in classrooms. *Educational Technology & Society*, 16(1), 392–402.
- Hancock, V., & Betts, F. (2002). Back to the future: Preparing learners for academia success in 2004. *Learning and Leading with Technology*, 29, 1–14.
- Handelsman, M. M., Briggs, W. L., Sullivan, N., & Towler, A. (2005). A measure of college student course engagement. *The Journal of Educational Research*, 98(3), 184–191.
- Hanson, J., Rosenberg, A. & Lane, J.L. (2013). Narrative descriptions should replace grades and numerical ratings for clinical performance in medical education in the united states. *Frontiers in Psychology* 4. <https://doi.org/10.3389/fpsyg.2013.00668>
- Howe, N., & Strauss, W. (2000). *Millennials Rising: The Next Great Generation*. USA: Vintage.
- Iffath, F., Kayes, A.S.M., Rahman, M.T., Ferdows, J., Arefin, M.S. & Hossain, M.S. (2021). Online judging platform utilizing dynamic plagiarism detection facilities. *Computers* 10(4)
- Ihantola, P., Ahoniemi, T., Karavirta, V. & Seppälä, O. (2010). Review of recent systems for automatic assessment of programming assignments. In: Proceedings of the 10th Koli Calling International Conference on Computing Education Research. Koli Calling '10, pp. 86–93. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/1930464.1930480>
- Jones, C. & Shao, B. (2011). The net generation and digital natives: implications for higher education. Higher Education Academy
- Khan, S. (2013). *The One World Schoolhouse: Education Reimagined*. USA, Twelve.
- KhanAcademy (2008). The KhanAcademy e-learning platform. <https://fr.khanacademy.org/>. Accessed 2022.
- Kirchherr, W., Li, M., & Vitányi, P. (1997). The miraculous universal distribution. *The Mathematical Intelligencer*, 19(4), 7–15.
- Kolmogorov, A. N. (1965). Three approaches to the definition of the concept quantity of information. *International Journal of Computer Mathematics*, 2(1), 157–168.
- Kompare (2016). A graphical interface that displays the differences between two source files. <https://apps.kde.org/fr/kompare>. Accessed 2022.

- Krause, K.-L., & Coates, H. (2008). Students' engagement in first-year university. *Assessment and Evaluation in Higher Education*, 33(5), 493–505. <https://doi.org/10.1080/02602930701698892>
- Kumar, P., & Sharma, R. (2014). Grading system in education: An overview. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2438412>
- Lattimer, H., & Riordan, R. (2011). Project-based learning engages students in meaningful work. *Middle School Journal*, 43(2), 18–23. <https://doi.org/10.1080/00940771.2011.11461797>
- Leal, J. P. & Moreira, N. (2000). Using Matching for Automatic Assessment in Computer Science Learning Environments, Web-based Learning Environments.
- Leal, J. P. & Silva, F. M. A. (2002). Managing programming contests with mooshak. In The 8th International Conference of European University Information Systems (Eunis).
- Lee, J., & Hammer, J. (2011). Gamification in education: What, how, why bother? *Academic Exchange Quarterly*, 15, 1–5.
- Levin, L. A. (1984). Average case complete problems. *SIAM Journal on Computing* 15
- Li, M., & Bitanyi, P. (1997). *An Introduction to Kolmogorov Complexity and Its Applications*. USA: Springer.
- Li, M., & Vitanyi, P. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*. USA: Springer.
- Li, M., Chen, X., Li, X., Ma, B., & Vitanyi, P. (2004). The similarity metrics. *IEEE Transactions on Information Theory*, 50(12), 3250–3264.
- Liu, F. (2007). Personalized learning using adapted content modality design for science students. In: Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore! ECCE '07, pp. 293–296. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/1362550.1362612>
- Lytle, S., Lytle, B. & Brophy, J. (2006). Student engagement in large undergraduate online courses. In: Proceedings of World Conference on ELearning in Corporate, Government, Healthcare, and Higher Education, pp. 1330–1333
- Martin-Löf, P. (1966). The definition of random sequences. *Information and Control*, 9(6), 602–619.
- Mavromoustakos, S. & Kamal, A. (2018). Student engagement practices for computer science students in online learning environments. In: Kumar, A.V.S. (ed.) *Optimizing Student Engagement in Online Learning Environments*, pp. 83–99. IGI Global, ????. <https://doi.org/10.4018/978-1-5225-3634-5.ch004>
- McCrimble, M. (2009). *The ABC of XYZ: Understanding the Global Generations*. Australia: McCrimble.
- McMahon, B., & Portelli, J. P. (2004). Engagement for what? beyond popular discourses of student engagement. *Leadership and Policy in Schools*, 3(1), 59–76. <https://doi.org/10.1076/lpos.3.1.59.27841>
- Moodle (2022) A learning management system. <https://moodle.org/>. Accessed 2022.
- Nicholas, J. A. (2020). Preferred learning methods of generation z. Northeast Business and Economics Association 46th Annual Conference.
- Oblinger, D. (2003). Understanding the new students: Boomers, genxers, millennials. *EDUCAUSE Review*, 38(4), 37–47.
- Pangaro, L. N. & McGaghie, W. C. (2005). Evaluation and Grading of Students. In: Fincher, R., Cox, S., DaRosa, D.A., Lynn, D.J., Morgenstern, B.Z., Pangaro, L.N., Sierles, F.A. (eds.) *Evaluation and Grading of Students*, pp. 133–250. Alliance for Clinical Education, ???
- Pham, M. T. & Nguyen, T. B. (2019). The DOMJudge Based Online Judge System with Plagiarism Detection. In: IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), pp. 1–6
- Raymer, R. (2011). Gamification: Using game mechanics to enhance elearning. *ELearn* 2011(9). <https://doi.org/10.1145/2025356.2031772>
- Rodrigues, M., Marques, F. & Martins, C. (2014). Teaching advanced computer programming topics using an automatic validation tool: Challenges and solutions. In: EDULEARN14 Proceedings. 6th International Conference on Education and New Learning Technologies, pp. 1541–1547. IATED, ???
- Rosegard, E., & Wilson, J. (2013). Capturing students' attention: An empirical study. *Journal of the Scholarship of Teaching and Learning*, 13(5), 1–20.
- Rubio-Sanchez, M., Kinnunen, P., Pareja-Flores, C. & Velazquez-Iturbide, J. A. (2012). Lessons learned from using the automated assessment tool "mooshak". In: International Symposium on Computers in Education (SIIE)

- Rubio-Sánchez, M., Kinnunen, P., Pareja-Flores, C., & Velá-Iturbide, A. (2014). Student perception and usage of an automated programming assessment tool. *Computers in Human Behavior*, *31*, 453–460. <https://doi.org/10.1016/j.chb.2013.04.001>
- Ruohomäki, V. (1995). Viewpoints on learning and education with simulation games, pp. 13–25. Springer. [https://doi.org/10.1007/978-1-5041-2870-4\\_2](https://doi.org/10.1007/978-1-5041-2870-4_2)
- Sadiku, M. N. O., Shadare, A. E. & Musa, S. M. (2017). Digital natives. *International Journal of Advanced Research in Computer Science and Software Engineering* 7(7)
- Schlechty, P. (1994). *Increasing Student Engagement*. Missouri Leadership Academy, United States.
- Smith, E. E., Kahlke, R. & Judd, T. (2020). Not just digital natives: Integrating technologies in professional education contexts. *Australasian Journal of Educational Technology* 36(3)
- Solomonoff, R. J. (1964). A formal theory of inductive inference. *Information and Control*, *7*(1), 1–22.
- Tapscott, D. (1997). *Growing up Digital: The Rise of the Net Generation*. USA: McGraw-Hill.
- Thorburn, W. M. (1918). The myth of occam’s razor. *Mind* XXVI, *1*(3), 345–353.
- Varré, J.-S., Delahaye, J.-P., & Rivals, E. (1999). The transformation distance : a dissimilarity measure based on movements of segments. *Bioinformatics*, *15*(3), 194–202.
- Wasik, S., Maciej, A., Jan, B., Artur, L., & Sternal, T. (2018). A survey on online judge systems and their applications. *ACM Computing Surveys*, *51*(1), 1–34.
- Wassner, H. (2014). Baldr l’outil anti-fraude anti-plagiat. <http://wassner.blogspot.com/2014/05/baldr-lou-til-anti-fraude-antiplagiat.html>
- Weltman, D. & Whiteside, M. (2010). Comparing the effectiveness of traditional and active learning methods in business statistics: Convergence to the mean. *Journal of Statistics Education* 18(1). <https://doi.org/10.1080/10691898.2010.11889480>
- Zvonkin, A., & Levin, L. (1970). The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, *25*(6), 83–124.

**Publisher’s note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.