pygiftgenerator: a python module designed to prepare Moodle-based quizzes

Jon Sáenz¹, Idoia G Gurtubay², Zunbeltz Izaola³ and Gabriel A López¹

¹ Department of Applied Physics II, University of the Basque Country (UPV/EHU), Sarriena Auzoa z/g, 48940-Leioa, Spain
² Department of Condensed Matter Physics, University of the Basque Country (UPV/EHU), Sarriena Auzoa z/g, 48940-Leioa, Spain
³ Escuela Universitaria de Ingeniería Dual, Instituto de Máquina Herramienta (IMH) Azkue auzoa 1, Elgoibar, Spain

E-mail: jon.saenz@ehu.eus

Received 5 May 2020, revised 10 August 2020
Accepted for publication 20 August 2020
Published 13 November 2020

Abstract
We present pygiftgenerator, a python module for systematically preparing a large number of numerical and multiple-choice questions for Moodle-based quizzes oriented to students’ formative evaluation. The use of the module is illustrated by means of examples provided with the code and drawn from different topics, such as mechanics, electromagnetism, thermodynamics and modern physics. The fact that pygiftgenerator relies on a well-established computer language, which allows functions to be combined and reused in order to solve complex problems, makes it a very robust tool. Simply by changing the input parameters, a large question bank with solutions to complex physical problems, can be generated. Thus, it is a powerful alternative to the calculated and multiple-choice questions which can be written directly in the Moodle platform. The module writes questions to be imported into Moodle and produces simple and human-readable ASCII output using the GIFT format, which enables html definitions for URLs for importing figures, or for simple text formatting (sub/superindices or Greek letters) for equations and units. This format also allows LaTeX and MathJax typing for complex equations.

Keywords: STEM, physics, active learning, Moodle, formative evaluation

(Some figures may appear in colour only in the online journal)
1. Introduction

The introduction of the European credit transfer system criteria [1], which are the standards accepted by all universities included in the European Higher Education Area guaranteeing the convergence of the different education systems in Europe, and the launch of the so-called Bologna process [2], have introduced a revolution in the teaching–learning process in European universities. The teaching–learning process has switched from teaching-centered education to learning-centered, requiring that (i) lecturers change traditional teaching methodologies to give students the leading position, and (ii) students become active learners, being more committed to their learning process, a role which enables them to arrive at the competencies and learning outcomes designed in the new curricula. To this end, all universities have adopted new technologies especially suited to offering new online teaching and learning environments to both students and lecturers. There is no doubt that this change of paradigm would not have succeeded without the introduction of all these online resources.

Reference [3] shows that online learning is optimal when it is combined with the face-to-face modality and that this effect is statistically significant. Nevertheless, lecturers should not be tempted to use online resources just as mere vehicles for delivering instruction [4, 5] and should design them to make the most of their capabilities. Doing so has demonstrated that the use of active learning techniques contributes to improving the outcomes of the learning process [6–11]. Online platforms, such as Moodle [12], allow students to tackle different activities: downloading lecture notes at home, exchanging ideas in discussion forums, uploading assignments, peer-instruction activities, chat-rooms, surveys, quizzes, plugins for access to virtual online classrooms and many others. In the last decade, there has been a significant proliferation of online courses in many universities [13, 14], although there are not so many studies devoted to massive assessment tools [15].

The use of Moodle-based quizzes for formative evaluation has been considered for years. In reference [16], the authors focused on the optimization of Moodle quizzes for online assessments. They showed that there are many important reasons to consider Moodle-based tests as an instrument during the learning process at the university level. The first reason is that they are effective learning tools from the point of view of the knowledge acquired by the students during the course. An important aspect in this regard is the automated feedback they offer, which is able to engage students better than delayed feedback. Secondly, automated grading by Moodle according to a well-constructed question bank saves significant amounts of time that instructors would otherwise invest in manually correcting exams and calculating the marks obtained by students. Automatic grading allows this scarce resource (time) to be devoted to other teaching activities, such as the preparation of experiments, classroom demonstrations, new handouts, etc. Also, the ability to randomize questions and answers inside these tests allows the production of tests which are relatively robust against plagiarism [16]. This is an important characteristic in the event that some of the quizzes are unsupervised.

The authors of this contribution have been using Moodle-based multiple-choice (MC) quizzes as a formative evaluation tool for many academic years (2012–2020) with students enrolled in general physics as part of the first year STEM (science, technology, engineering and mathematics) degrees taught at the Faculty of Science and Technology at the University of the Basque Country UPV/EHU. At the end of every topic, students answer ten questions, randomly extracted for every student and with answers randomized within each question, from quite a large question bank (around 60–70 per chapter) prepared by the lecturers. The goal of these quizzes is twofold: (i) to allow the students to self-evaluate their performance during the semester; (ii) to give real-time feedback to the lecturer on the performance of each student. A
full description of this methodology can be found in [17]. A study of the results obtained with this strategy demonstrates that those students regularly taking and passing the tests increase their probability of passing the course and, on average, obtain better final marks [17]. Since 2016–2017 we have made them optional, as a training tool for honest self-evaluation. In this case, even if the students are not required to complete the quizzes, they are aware that they are for their own benefit, and the same conclusion achieved in the previous study is drawn. Those who pass the tests increase the probability of passing the subject and obtain, on average, better marks [8]. We have also used the system at the MSc level for two years.

If properly set up, the MC quizzes and numerical questions in Moodle can be a very valuable online tool for teaching and for (formative) evaluation, as already shown in our previous studies. We list some of the advantages of using them: they allow self-correction; questions and answers can be randomized; they allow immediate or delayed feedback; they can be made available only within a time-window; a countdown for solving them can be set; questions can be classified in categories and subcategories; questions can be exported from a course to be imported back to another course in many different formats; equations and images are allowed within the questions; and many more. But probably one of the most useful features of the Moodle question bank is that when it is large enough, all students are randomly assigned different questions of a similar level of complexity that evaluate the same learning outcome. This guarantees that they cannot share the answers while they fill in the quiz. More importantly, this ensures that all the students are fully evaluated in a manner respecting the principle of equal opportunities. Of course, this means that, depending on the amount of students enrolled in the course, a huge amount of questions must be generated. Thus, the workload placed upon the instructors increases dramatically.

The outbreak of the global COVID-19 pandemic in early spring of 2020 has put online teaching and learning at the focus of most educational systems worldwide. Indeed, the closure of schools and universities has literally obliged all of the teaching–learning processes to adapt to the online world unexpectedly, and in a very short time. Lecturers and students have suddenly needed to modify the way they teach and the way they learn, respectively. Moreover, due to the long period of confinement ordered by most governments, a face-to-face evaluation has not been possible at the majority of universities. Therefore, the final evaluation of the teaching–learning process has also been done using online resources, with all the drawbacks and difficulties this implies, such as students not being under surveillance to prevent them from sharing answers during the exam.

In this paper we introduce the pygiftgenerator module for python [18]. It is easy and intuitive to use and it allows instructors to prepare a large number of MC and numerical questions based on the setup of a single problem (which can also include images). Since the system works by allowing the values of the parameters involved in the problem to be changed, many different instances of the same problem can be easily produced in GIFT format. The output file (in GIFT format) holds the questions and their corresponding answers, all classified according to the Moodle categories specified by the user. They can then easily be uploaded to the question bank in the Moodle platform of any course. If the instructors carefully design the taxonomy of categories to which the questions belong, the automatic random selection of questions for every student from the same or similar families of problems is straightforward, and allows the instructor to cover the syllabus of the course.

pygiftgenerator is thus offered as an additional option to a set of tools which help instructors to produce massive question banks for Moodle. There are some freely-available packages such as GIFTr for the R programming language, gift-wrapper for python, the
Moodle package for LaTeX, a system for converting questions in an R course using python [19] and additional tools for Excel or OpenOffice [20] which convert a list of questions already written as structured data (Excel, CSV, YAML markdown file or LaTeX, for instance) and reformat them as a set of GIFT files. These tools are not comparable to the one presented in this paper, since they simply reformat the entries in the original formatted data file. On the contrary, pygiftgenerator produces the question bank from the definition of the physical problem written as code. The system py2gift, also written for python, shares the same philosophy as our tool. Similarly to pygiftgenerator, it builds the questions directly from the python program and is not a mere reformatting tool. However, a quick look at examples of both tools shows that pygiftgenerator is easier to use because, unlike py2gift, it requires only a very basic knowledge of python. Anyone with an elementary knowledge in any programming language should be able to adapt any of the examples provided with the pygiftgenerator module to her needs without much effort. Besides, it can either use classes or built-in functions to generate a question bank. The use of functions eases the task for anyone used to programming in the computer languages extensively used by physicists, such as Fortran or C.

This paper is organized as follows: section 2 describes the methodology used by the module to generate easy-to-import GIFT files in Moodle and section 3 outlines its implementation and installation. Some applications to selected problems in different fields of physics are shown in section 4. We finish with a discussion in section 5 and conclusions in section 6.

2. Methodology

The GIFT format [21] is one of the formats used by the Moodle learning environment to allow the preparation of questions for batch import into a given course. This is relatively simple for the case of simple text questions, but the format also supports HTML tags, and therefore, basic text formatting and even remote URL-defined files (images, references) can be included in the questions. LaTeX code can also be inserted in the GIFT files, so that complex equations are also supported.

An important first step in this module is the preparation of variable input parameters for every problem. We show two different ways of treating this. In the first case, the parameter set is randomly extracted from a combination of input lists specified by the user, which allows the instructors to keep the numbers used as input parameters nice looking or within a discrete set that is physically meaningful. The way this system works by combining lists of parameters is illustrated in the file EM_DC_circuit_MC.py, distributed with the module.

A second option is to produce the input parameters by calls to random number generators from the program generating the questions. Even in this case, as shown in the example file TH_Blackbody_Planck_NQ.py, some of the input parameters may be preset through a list (representing a range of temperatures in this case), as in the previous example.

An important aspect that needs to take place before the design of the questions is the preparation of a consistent tree of categories and subcategories, which allows the instructors to perform a proper classification of topics and subtopics. This helps in the random extraction of questions by the Moodle quiz preparation system for each individual student. These categories are kept during the production of the GIFT code and are fully supported by Moodle (see figure 1). This is important when random tests for students are designed on the basis of diverse topics. It allows the program covered through the year to be properly sampled and distributes different options to every student equally.

There are many classes of questions available in Moodle, and all of them can be imported by means of the GIFT format, such as MC, true-false, short answer and numerical, to name
a few. Considering that we deal with exercises in the field of physics, we have focused on numerical and MC types only, since we found them to be the most useful for carrying out exams during the COVID-19 exam period. A similar code structure can be used for the other kinds of questions, and it might even happen that we support them in the future. However, in many cases (short answer, missing words or matching questions, for instance), the use of a module such as pygiftgenerator would not be effective at all. The reason is that the use of python as proposed in this paper, as a computer language, is perfect for questions in which numerical computation is needed, because the answer can be recalculated for different values of the parameters. This applies perfectly to numerical and MC questions. However, for short answer or matching questions, the use of a computer language does not offer a significant advantage and using it would require much more work than the effort needed to write the questions directly in GIFT format. Thus, the authors have decided to exclusively support numerical and MC questions. In the latter case, the module ensures that questions with repeated answers are not included in the output GIFT file.

The GIFT file produced by a python interpreter running a program built around the pygiftgenerator.py module can be easily imported to the question bank of a Moodle installation. A screenshot of a result of an import is shown in figure 2 and the final versions of an MC and a numerical question as displayed in the Moodle platform, in figure 3.

3. Implementation

The module pygiftgenerator can be installed by means of a standard setup.py installation script which uses the distutils tools. The code has been tested with versions 3.6, 3.7 and 3.8 of python. It is open-source distributed under the GPLv3 license and it can be obtained at the https://gitlab.com/EHU/pygiftgenerator repository.
Figure 2. Result of importing a GIFT file produced by the code in ME_binary_star_MC.py to the Moodle platform.

Figure 3. (a) Multiple-choice question number 1 of figure 2 as displayed by Moodle. (b) A numerical question based on the same setup. These questions correspond to the example described in section 4.1.

4. Results: application to selected problems

We list here a selection of applications. Many more and their corresponding example files can be found in the documentation available with the module.

4.1. Mechanics

Figure 4 shows the initial position of a system of two sun-like stars of mass $m_1 = nM$ and $m_2 = M$, where $M$ is a variable parameter that we select to be close to the mass of the Sun. They are a distance $r_0$ apart and their velocities, as measured from a distant observatory (at rest), are $v_1 = v_0$ (approaching) and $v_2 = mv_0$ (receding). The value of $v_0$ is given by $v_0 = \sqrt{GM/r_0}$. The positive constants $n$ and $m$ and the different values (in astronomical units) of $r_0$ allow the instructor to produce a set of questions with different numerical values. Since both masses are of the same order of magnitude, the problem must be solved by making use of the reduced mass $\mu = nM/(n + 1)$. It can be shown that the orbit will be closed ($E < 0$) if $n > m(n + 2)/2$.
Figure 4. Initial position of the system of stars.

(this is a restriction that the input parameters must fulfill). The minimum of the effective potential energy leads to a circular orbit when \( n = m (m + 2) \).

This example easily demonstrates that, based on a single figure and by slightly changing the setup of the problem, one can prepare a large number of different questions involving varying physical quantities. For example:

(a) Given \( r_0 \) and \( m \), find the value of \( n \) for the orbit to be circular.

(b) Given a pair of \( n \) and \( m \) values, find the eccentricity of the orbit:

\[
\epsilon = \sqrt{1 + \frac{[m(m+2)−(2n+1)][m+1]}{(n+1)^2}}.
\]

(c) Given a pair of \( n \) and \( m \) values, \( M \) and \( r_0 \),

(1) Find the energy of the orbital motion: \( E = \frac{GM^2}{r_0} \left[ \frac{n}{2(n+1)}[m(m + 2) − (2n + 1)] \right] \).

(2) Find the semi-major axis of the orbit: \( a = \frac{(n+1)}{(1+2n−m(m+2))}r_0 \).

(3) Obtain the period of the orbital motion: \( T = \sqrt{\frac{4\pi^2}{GM(n+1)}}a^3 \).

(4) Find the distance \( c \) from the center of the ellipse to its foci: \( c = \epsilon a \).

(5) Obtain the distance to the apocenter: \( x_a = a(1 + \epsilon) \).

(6) Obtain the distance to the pericenter: \( x_p = a(1 − \epsilon) \).

(7) Obtain the semilatus rectum: \( p = \frac{GM(n+1)}{GM/a^2} \).

The files `ME_binary_star_NQ.py` and `ME_binary_star_MC.py`, distributed as examples by the module’s git server, show the implementation of (a), (b) and (c)(3) as numerical questions and the implementation of (c)(2) as an MC question, respectively, using simple python scripts. The same code can also be found written in a single full-featured Jupyter notebook: `ME_binary_star_all.ipynb`.

4.2. Thermodynamics

4.2.1. Equation of the state of air. Following the relevant literature [22], we can build questions related to the determination of the density of dry air \( \rho_d = \frac{P}{(R_d T)} \), with \( R_d = 287 \text{ J kg}^{-1} \text{ K}^{-1} \) the gas constant corresponding to the mixture of dry air, \( P \) the pressure in Pa and \( T \) the air temperature in K.

The density of moist air \( \rho_m = (P − e)/(R_d T) + e/(R_v T) \) is computed from the mixture of dry air and water vapour, with \( R_v = 461 \text{ J kg}^{-1} \text{ K}^{-1} \) the gas constant corresponding to water vapour and \( e \) the partial pressure corresponding to water vapour in the moist air measured in Pa.

The virtual temperature of air, \( T_v \), represents the temperature at which dry air has the same density as a given parcel of moist air and is computed by means of the expression:

\[
T_v = \frac{T}{1 − \frac{\rho_m}{\rho_d}}.
\]
The expressions above can be combined with equation (2) to calculate the saturation pressure of water vapour, \(e_s\) in Pa, as a function of temperature through a common approximation [23]:

\[
e_s = 611 \exp \left[ \frac{L}{R_v} \left( \frac{1}{273} - \frac{1}{T} \right) \right],
\]

where \(L = 2.5 \times 10^6\) J kg\(^{-1}\) is the latent heat of evaporation of water. This allows the computation of the relative humidity defined as the water vapour mixing ratio, \(w = \rho_v/\rho_d\), as

\[
r = \frac{100}{w_s} = 100 \frac{e}{e_s} \left( \frac{P - e_s}{P - e} \right).
\]

4.2.2. Adiabatic evolution of unsaturated air. In the last example in this section, the potential temperature

\[
\theta = T \left( \frac{P_0}{P} \right)^{R_d/c_p}
\]

is computed to determine the adiabatic evolution of air, with \(c_p\) the specific heat of air at a constant pressure. These questions must use pressures and temperatures within the given ranges of atmospheric temperature and pressure to be meaningful, as shown in example TH_adiabatic_process_NQ.py.

4.3. Modern physics

4.3.1. Relativistic collisions of particles with photons. In this example, it is considered that a photon with energy \(E\) collides against a particle at rest with a mass of \(m = E_0/c^2\) and we want to obtain the energy, \(E_m^*\), and the magnitude of the linear momentum, \(p_m^*\), of the particle as seen from the center of mass reference frame (CM-RF). The velocity of the CM-RF is given by

\[
v = \frac{Ec}{E + E_0},
\]

which yields a value of

\[
\gamma = \frac{E + E_0}{\sqrt{E_0^2 + 2EE_0}}
\]

for the CM-RF Lorentz’s factor. Thus, the particle’s energy after the collision, as seen from the CM-RF is

\[
E_m^* = (E + E_0) \sqrt{\frac{E_0}{E_0 + 2E}}
\]

and its linear momentum is

\[
p_m^* = -\frac{E}{c} \sqrt{\frac{E_0}{E_0 + 2E}}.
\]
The example file `MP_SRelativity_CMCCollision_NQ.py` contains code that produces results for collisions between photons and particles with different energies. As an improvement to previous examples, in this case, the output is written in an ASCII file by the code, and no redirection is needed. In order to get that done, the additional parameter in the file is added to the constructor of the question objects.

### 4.3.2. Photoelectric effect

Let the photoelectric work function of a metal be \( W \). If light having a wavelength of \( \lambda \) shines on the surface of the metal, photo-electrons with a kinetic energy of \( K \) will be emitted from the surface, provided that the energy of the incident beam is larger than the work function of the metal. These three magnitudes are related by

\[
\frac{hc}{\lambda} = K + W, \tag{9}
\]

where \( c \) is the speed of light and \( h \) is Planck’s constant. The photoelectric threshold wavelength for the surface is the minimum wavelength needed to eject electrons from the surface:

\[
\lambda_0 = \frac{hc}{W}. \tag{10}
\]

The maximum velocity of an ejected electron is

\[
v_{\text{max}} = \sqrt{\frac{2K_{\text{max}}}{m_e}} = \sqrt{\frac{2(hc/\lambda - W)}{m_e}}, \tag{11}
\]

where \( m_e \) is the mass of the electron and its corresponding de Broglie wavelength reads

\[
\lambda = \frac{h}{p} = \frac{h}{m_ev_{\text{max}}}. \tag{12}
\]

The example in the file `MP_photEffect_workfunc_NQ.py` allows the generation of questions for the excitation threshold wavelength, while the code provided in the file `MP_photEffect_ejectedElectrons_NQ.py` can be used to either obtain the maximum velocity of the ejected electrons or their de Broglie wavelength.

### 5. Discussion

Various question types such as simple calculated, calculated and calculated MC questions are supported in Moodle. Thus, it is fair to question the need of a module such as the one presented in this paper. As we see it, performing the computation in an external language such as python allows at least four objectives to be satisfied. First, the degree of complexity of the problems which can be handled can be extended beyond the ones available in the inner core of Moodle.

Second, as shown in the example in section 4.1 and in many more in the users’ manual accompanying the module, different versions of a similar problem can be prepared using a single set of parameters. With very little effort, the number of different generated problems increases, even if most of them can be solved by applying the same concepts. If the amount of time given for solving the online questionnaire is properly calibrated, the results obtained from the test should be trustworthy in terms of reflecting the students’ knowledge, even if the quiz is unsupervised.

Third, the fact that many parts of the solution are common for the same set of parameters and different questions (see, for instance, the binary star example in section 4.1) makes it easier...
to produce the set of questions by means of the solution proposed in this paper, since these variables are shared during the generation of the questions in the same piece of code, thus avoiding potential coding errors that might happen if the questions were prepared independently.

Fourth, python is a fully featured computer language with access to many external modules and packages and with the ability to build complex programs. The capability to write LaTeX code in the questions in GIFT format allows instructors to build complex equations (determinants or systems of equations, for instance). This, combined with the linear algebra routines provided by numpy, allow this system to be used to solve linear algebra problems of higher dimensions, which would be very complex to code directly in Moodle. Additionally, the use of python as the computing core allows instructors to use external auxiliary functions (see the example TH_relative_humidity_NQ.py for calculating the saturation pressure of water vapour) that are used again and again during the computation of many quantities involved in the solutions of the problems at hand. Having to rewrite them in every question would be very error-prone and time-consuming.

There are many learning environments, such as Moodle, Blackboard or Canvas, to name a few, which support the use of calculated questions. In this paper, we address the platform that our university provides to us, but after a review of the features of the calculated (formula-based) questions in these platforms, the previous statements still stand true. The solution provided here allows more complex computations with better reliability, so that the same methodology could probably be extended to formats allowed by those platforms for importing questions.

We have provided interfaces for producing numerical questions and numerical MC questions, which are the ones that we find more relevant from the point of view of teaching physics and the massive production of questions for a question bank which should allow individual and independent tests for every student. There are, of course, other options in Moodle, such as MC text questions, which allow challenging conceptual tests in the field of physics. However, from the point of view of producing those questions, our experience from the past years shows that MC conceptual text questions are better produced directly on the Moodle interface and that a python program would only complicate the steps needed to arrive at the final product (a GIFT file to be imported to the question bank). An additional advantage of using the GIFT format for importing the results into Moodle is that it is a simple ASCII-based format which can be easily read by humans before uploading the file to Moodle, so that inconsistencies can easily be detected. For this first release of the software, we have selected the GIFT format to allow easy upload of the questions to the Moodle server. Depending on the configuration of the server, this may make the inclusion of figures rather cumbersome. Images can also be encoded in the GIFT file using the base64 package, as shown in EM_circuit2_NQ.py (available from the git server). However, this leads to very large and difficult-to-read GIFT files. Most likely, future releases of pygiftgenerator will also export questions in the XML Moodle format to ease the upload of figures and other resources.

We have tested this solution by preparing several online exams for 220 first-year physics students during the COVID-19 lockdown, between June and July 2020. Since this is a full-year (2-semester) subject, we used the results of the face-to-face exams performed in the first semester for comparison. Initially, we implemented MC question-based quizzes, and later numerical question-based quizzes. In all the cases, the students were also asked to upload their hand-written exams in order to check how they had solved the exercises. From our experience, if we take into account the time invested in the preparation of the questions and the efficiency in checking the students’ performance, numerical question-based quizzes have turned out to be the best choice.
Indeed, preparing an MC question quiz requires coding three or four extra wrong answers. If we want these to include common mistakes made by the students, such as not keeping sign conventions, using wrong units, forgetting square roots or powers, etc., coding these can become an extremely difficult and time-consuming task, even more so than coding a new numerical question from scratch, for several reasons. Firstly, one has to make sure that the wrong answers are still physically meaningful, so that they cannot be easily discarded at a glance. Secondly, they have to be numerically different enough among them, so that numerical errors with the calculator cannot lead to two answers to be acceptable. We should not forget that these wrong answers are computed on a set of randomly chosen parameters, which may be suitable for the correct answer, but not for the wrong ones. Additionally, by checking the students’ handwritten answers, we realized that, even if they solved the exercise incorrectly, they could still choose the right answer in Moodle just by approximation, if they obtained a solution close enough to the correct one. As a result, in the exam where we used MC questions, we obtained fairly biased marks, compared to those in the face-to-face exams.

As mentioned above, preparing numerical questions is much simpler and with the same or much less time investment we can prepare a larger amount of varied questions for each category evaluating the same concepts. Moreover, students cannot guess the correct answer because they have to fill in an empty box, as shown in figure 3(b), as opposed to figure 3(a). The results obtained by this type of exam correlated very well with the previous results of the face-to-face exams. Although these qualitative conclusions have been drawn from a limited experience on a small sample, we believe they point in the right direction. A more accurate quantitative evaluation on a larger sample will be the goal of a future study.

6. Conclusions

In this contribution, we presented a python-based flexible module for producing a large number of numerical and MC numerical questions oriented to evaluating students online. This has been very relevant during the lockdown due to the COVID-19 pandemic, which has enforced the closure of most schools and universities. It may also be of special relevance if a new lockdown is needed due to a potential new outbreak. In particular, new online tools need to be designed to evaluate students’ learning processes. If these tools are meant to substitute for on-site supervised written exams, they require a minimum guarantee that students will not be able to share answers while solving the exam because of the fact that no one is supervising them in a room. This python module allows the production of a large question bank. All the questions produced are different, but can be classified into categories and types. This way, the probability of students getting quizzes with exactly the same questions is very low, and can be made negligible just by increasing the amount of items prepared. The module is quite flexible, and it can be used to cover different branches of physics, such as mechanics, thermodynamics, special relativity and quantum mechanics, as shown here. We have selected python because it is a simple programming language. It is very user-friendly for people who are not programming experts. Its learning curve is not too steep. Besides that, the module presented here is quite easy to use, which makes us think that it can be used by instructors worldwide without a significant investment of time in learning its working. However, the system is powerful; the examples provided in the documentation of the module show that it can produce many different questions from the same set of parameters with a unique identification and organized in categories. Equations written in LaTeX and MathJax, or even figures, can be imported by Moodle from the GIFT files generated.
Acknowledgments

JS acknowledges support by the Spanish Government’s MINECO grant and ERDF (Grant No. CGL2016-76561-R) and the UPV/EHU (Grant No. GIU17/02). IG acknowledges the University of the Basque Country UPV/EHU (Grant No. GIU18/138) for financial support.

ORCID iDs

Jon Sáenz https://orcid.org/0000-0002-5920-7570
Idoia G Gurtubay https://orcid.org/0000-0002-7060-5174
Zunbeltz Izaola https://orcid.org/0000-0001-6203-5418
Gabriel A López https://orcid.org/0000-0002-5429-7490

References

[1] 2015–2017 Directorate-general for education youth, sport, and culture (European Commission) ECTS User’s Guide (Accessed: 8 April 2020)
[2] European Commission/EACEA/Eurydice 2015 The European Higher Education Area in: Bologna Process Implementation Report (Luxembourg: Publications Office of the European Union)
[3] Means B, Toyama Y, Murphy R, Bakia M and Jones K 2010 Evaluation of evidence-based practices in online learning. A meta-analysis and review of online learning studies Technical Report US Department of Education
[4] Kozma R B 1994 Will media influence learning: reframing the debate Educ. Technol. Res. Dev. 42 7–19
[5] Clark R E 1994 Media will never influence learning Educ. Technol. Res. Dev. 42 21–9
[6] Borondo I, Benito R M and Losada I C 2014 Adapting physics courses in an engineering school to the B-learning philosophy Eur. J. Eng. Educ. 39 496–506
[7] Freeman S, Eddy S L, McDonough M, Okoroafor N, Smith M K, Jordt H and Wenderotha M P 2014 Active learning increases student performance in science, engineering, and mathematics Proc. Natl Acad. Sci. USA 111 8410–5
[8] Sáenz J, López G A, Martinez-Perdiguero J, Alonso I, Leonardo A and Gurtubay I G The use of Moodle quizzes by students of first year physics in the university: mandatory versus optional quizzes ICERI2017 Proc. 10th Int. Conf. of Education, Research and Innovation (16–18 November 2017) (IATED) pp 5469–72
[9] Gurtubay I G, Sáenz J, López G A, González-Rojo S J, Unzueta I, García-Goiricelaya P, Alonso I, Leonardo A and Martinez-Perdiguero J Learning physics from wrong preconceptions through daily-life related experiments INTEDE2018 Proc. 12th Int. Technology, Education and Development Conf. (5–7 March 2018) (IATED) pp 5703–11
[10] Gurtubay I G, Sáenz J, Alonso I, Lafuente-Bartolome J, López G A, Martinez-Perdiguero J and Leonardo A Training physics student’s intuition through daily-life related experiments EDUCLEARN18 Proc. 10th Int. Conf. on Education and New Learning Technologies (2–4 July 2018) (IATED) pp 5742–50
[11] Gurtubay I G, Sáenz J and López G A 2019 Laurel vs Hardy: a PBL activity for motivating first year university students in physics 3rd EuroSoTL Conf.: Exploring new fields through the scholarship of teaching and learning (Servicio Editorial de la Universidad del País Vasco) pp 206–13
[12] 2020 Moodle Project (Accessed: 7 April 2020)
[13] Broadbent J and Fuller-Tyszkiewicz M 2018 Profiles in self-regulated learning and their correlates for online and blended learning students Educ. Tech Res. Dev. 66 1435–55
[14] Liagkou V and Stylios C 2018 A trustworthy and privacy preserving model for online competence evaluation system Contemporary Complex Systems and Their Dependability Proc. 13th Int. Conf. on Dependability and Complex Systems DepCoS-RELCOMEX (2–6 July 2018) (Branów, Poland) (Springer) pp 338–47
[15] Corrigan-Gibbs H, Gupta N, Northcutt C, Cuttrell E and Thies W 2015 Deterring cheating in online environments ACM Trans. Comput. Hum. Interact. 22 1–23
[16] Gamage S H P W, Ayres J R, Behrend M B and Smith E J 2019 Optimising Moodle quizzes for online assessments *Int. J. Stem Educ.* **6** 27

[17] López G A, Sáenz J, Leonardo A and Gurtubay I G 2016 Use of the Moodle platform to promote an ongoing learning when lecturing general physics in the physics, mathematics and electronic engineering programmes at the University of the Basque Country (UPV/EHU) *J. Sci. Educ. Technol.* **25** 575–89

[18] van Rossum G 1995 Python tutorial *Technical Report* CS-R9526 (Amsterdam: Centrum Voor Wiskunde en Informatica (CWI))

[19] Mir-Torres A, Roselló-Llompart F A, Miró-Julià M, Capó-Capó T, Alberich Martí R and Julià J M 2009 Learning R in an autonomous manner with R-QUEST *IADIS Int. Conf. (WWW/Internet)* 47–51

[20] 2020 Gift format - Moodledocs [https://docs.moodle.org/39/en/GIFT_format](https://docs.moodle.org/39/en/GIFT_format) (Accessed: 2 August 2020)

[21] 2018 *Moodle Project Moodle Docs 3.8. GIFT Format* (Accessed: 7 April 2020)

[22] Petty G W 2008 *A First Course in Atmospheric Thermodynamics* (Madison, WI: Sundog Publishing)

[23] Wallace J M and Hobbs P V 2006 *Atmospheric Science: An Introductory Survey* 2nd edn (Amsterdam: Elsevier)