

An Open-Source Gamified Programming Learning Environment

José Carlos Paiva  

CRACS - INESC-Porto LA, Portugal

DCC - FCUP, Porto, Portugal

Ricardo Queirós   

CRACS - INESC-Porto LA, Portugal

uniMAD - ESMAD, Polytechnic of Porto, Portugal

José Paulo Leal   

CRACS - INESC-Porto LA, Portugal

DCC - FCUP, Porto, Portugal

Jakub Swacha   

University of Szczecin, Poland

Filip Miernik  

University of Szczecin, Poland

Abstract

The importance of e-learning tools facilitating the process of learning to program is growing, especially as the pandemic-caused lockdown enforced distance learning in many countries. The key success factor in this process is the provision of an instant and relevant feedback to students. In this paper, we describe a novel open-source programming learning environment featuring automatic assessment of students' solutions and customized gamification. This environment has been developed as a part of the FGPE framework.

2012 ACM Subject Classification Applied computing → Computer-managed instruction; Applied computing → Interactive learning environments; Applied computing → E-learning

Keywords and phrases learning environment, programming exercises, gamification, programming learning, automatic assessment

Digital Object Identifier 10.4230/OASICS.ICPEC.2021.5

Category Short Paper

Funding This paper is based on the work done within the Framework for Gamified Programming Education project supported by the European Union's Erasmus Plus programme (agreement no. 2018-1-PL01-KA203-050803).

1 Introduction

The role of feedback in improving knowledge and skill acquisition is considered crucial ([29] and works cited therein). In programming education, particularly, an instant feedback can be generated automatically based on the submission code, providing the student with relevant information about general concepts, task constraints, mistakes made, hints on how to proceed or even meta-cognition [16].

There is also a major concern amongst educational researchers with the disengagement of the students from the learning activities, which frequently leads to academic failure and, lastly, dropout. This issue is even more noticeable in distance learning [28], which became the default way of learning in many countries in times of the current pandemic. One way to counteract this problem is through gamification, which has been proven as a capable tool to reduce the dropout rate [10].



© José Carlos Paiva, Ricardo Queirós, José Paulo Leal, Jakub Swacha, and Filip Miernik; licensed under Creative Commons License CC-BY 4.0

Second International Computer Programming Education Conference (ICPEC 2021).

Editors: Pedro Rangel Henriques, Filipe Portela, Ricardo Queirós, and Alberto Simões; Article No. 5; pp. 5:1–5:8

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

5:2 An Open-Source Gamified Programming Learning Environment

In this paper, we describe a novel open-source programming learning environment addressing both the above-mentioned needs, featuring automatic assessment of students' solutions and customized gamification. As the environment has been developed within the Framework for Gamified Programming Education (FGPE) project [9], we called it FGPE PLE.

The rest of this paper is organized as follows. Section 2 reviews the current state of gamification in education, focusing on empirical studies describing experiments which applied game elements into a learning environment. Section 3 presents the FGPE PLE, its architecture and user interface. Finally, Section 4 summarizes the contributions of this work and indicates some future directions.

2 Related Work

Gamification of education has been a top trending in the last years [30], which can be explained by the positive boost in motivation and user experience it is accredited with. There are several case studies in the literature, particularly performed in high school and universities, covering different learning subjects, ranging from Computer Science / Information Technology (CS/IT) [34, 1, 2, 4, 11, 22, 33] to Mathematics (Maths) [8, 35, 6, 25], Foreign Languages (FL) [13, 26], Communication and Multimedia (C&M) [3, 12, 14, 15], and Medicine / Biology (M/B) [27, 20, 5]. A search for experimental studies that apply gamification methods in educational learning environments was conducted in Google Scholar and ScienceDirect databases, using combinations of the following keywords: gamification, serious games, education, and learning. From the returned results, only studies published since 2014 which have reported findings in high school and university subjects of the previously mentioned areas were considered. The result was a set of about 50 empirical research studies, of which 20 were selected according to their relevance. Table 1 presents a comparative study of the collected research papers, regarding the applied game elements.

■ **Table 1** Game elements reported in the selected research papers.

Game Elements	CS/IT							Maths				FL		C&M				M/B			
	[34]	[1]	[2]	[4]	[11]	[22]	[33]	[8]	[35]	[6]	[25]	[13]	[26]	[3]	[12]	[14]	[15]	[27]	[20]	[5]	
Badges	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓				✓	
Collaboration	✓			✓			✓	✓				✓		✓	✓	✓				✓	✓
Direct Competition ¹							✓														
Content Disclosure		✓		✓		✓	✓							✓							
Exhaustible Resources								✓		✓	✓			✓	✓		✓				
Leaderboards	✓	✓		✓		✓		✓	✓	✓	✓			✓	✓	✓	✓			✓	✓
Levels							✓					✓		✓	✓	✓	✓			✓	
Points ²	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
Progress Indicator	✓	✓		✓								✓		✓		✓	✓				✓
Quests							✓					✓		✓		✓				✓	
Real-world Rewards																				✓	
Serious Games / Immersion							✓	✓	✓		✓	✓		✓	✓	✓				✓	
SLP ³			✓			✓					✓			✓	✓						
Status	✓					✓		✓												✓	

¹Only core competition, i.e., one player (or team) plays against other, if one wins, the other loses.

²Includes experience points (XP), skill points (score), influence points (rating, reputation), and automatic grades.

³Similar Learning Path (SLP) consists of using data from other students to predict and display the current progress of a student (not leaderboards).

From this study, we can observe that badges, points, and leaderboards are the most applied game elements in general. This was expected as those are the easiest elements to apply, and reuse in distinct educational scenarios. Only a few works use, for instance, quests (5), content disclosure (5), status (4), and direct competition (1) as these are harder to implement and less reusable. Surprisingly, even though there is much more work on gamification of CS/IT education compared to other areas of education, no significant difference in game elements could be found.

Regarding experimental results, the effects on students are mostly slightly positive or neutral, with a few exceptions [12, 6]. Nevertheless, in the area of CS/IT the results are overall positive, i.e., they either promote engagement, participation, dedicated time, or learning outcome.

3 Programming Learning Environment

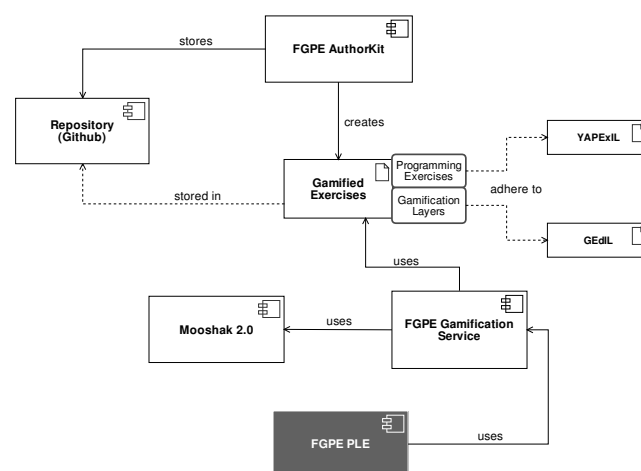
FGPE Programming Learning Environment (PLE) is an open-source progressive web application designed to engage learners while learning to program. To this end, it combines automated assessment with a meaningful composition of gamification elements, both provided by the Framework for Gamified Programming Education (FGPE) ecosystem. The PLE is effortlessly reusable across distinct courses and programming languages, being independent of the gamification layer linking the activities.

The following subsections provide an in-depth overview of the FGPE PLE. Subsection 3.1 details the architecture of the environment. Subsection 3.2 presents its user interface.

3.1 Architecture

The FGPE PLE is a web-based environment that aims to deliver a single interface to a complete ecosystem for gamified programming education. This PLE handles two types of users: students – who register and play the games (i.e., courses) in which they are enrolled, by solving the proposed challenges – and teachers – who manage the courses, enroll students, assign them into groups, and collect/analyze data about their progress. As users must also be able to access other tools of the same ecosystem, particularly authors, and use different authentication mechanisms, authentication and authorization is managed by Keycloak [17]. Keycloak is an open-source identity and access management solution providing numerous features for the applications, such as centralized user management, authentication, single sign-on and identity brokering, and user federation and social login.

Figure 1 presents the FGPE ecosystem, in which the FGPE PLE plays a key role. This ecosystem is composed of two formats, one for describing programming exercises – YAPExIL [24] – and another for gamification layers – GEdIL [31], three tools including an authoring tool, a gamification service, and an evaluation engine, and GitHub.



■ **Figure 1** Architecture of the FGPE ecosystem, highlighting the FGPE PLE.

5:4 An Open-Source Gamified Programming Learning Environment

The FGPE ecosystem supports seven distinct types of programming exercises, in particular: `BLANK_SHEET`, which asks the students to develop their solution from scratch; `EXTENSION`, which presents a partial solution for the student to complete; `IMPROVEMENT`, which provides a correct initial source code that needs some optimization; `BUG_FIX`, that gives a buggy solution for the student to find the right code; `FILL_IN_GAPS`, which provides code with missing parts; `SPOT_BUG`, that asks students to indicate the location of the bugs in a buggy solution; and `SORT_BLOCKS`, which asks students to sort the several blocks of code of a solution. To this end, programming exercises are defined in YAPExIL [24], a novel JSON format introduced to address the lack of support for non-traditional programming exercises in existing formats. Furthermore, YAPExIL does not target a specific evaluation engine, heading all efforts to maximize expressiveness and facilitate the conversion from/to other programming exercise formats.

Gamification layers are specified separately from programming exercises using GEdIL [31], an open format for the specification of gamification layers for educational contents. Even though it does not target a specific course, GEdIL was initially designed to cover particular requirements of gamification for programming courses [32], including a vast collection of rewarding mechanisms, such as points, badges, virtual items, and leaderboards to provide extrinsic motivation, as well as unlockable and secret content, different activity modes (e.g., speedup and duels), among others. The game dynamics and mechanics are, then, parsed and applied in the referenced collection of activities by the gamification service.

FGPE Gamification Service [21] is a GraphQL service designed to fulfil this goal, consuming a GEdIL layer and transforming it into a game, played by enrolled students. Hence, the service has complete support for various rewarding mechanisms such as points, badges, coupons, virtual items, leaderboards, locked and secret content, and different activity modes (e.g., time-bomb and speedup). Furthermore, the FGPE Gamification Service is the single point of access of the FGPE PLE to this ecosystem. Consequently, it should not only apply gamification rules, but also deliver the challenge and activity statements as well as handle the automated assessment of activities. To this end, the service uses plugins, i.e., consumers of the evaluation engine APIs, that leverage on the evaluation engine connected that is adequate to assess that type of activity.

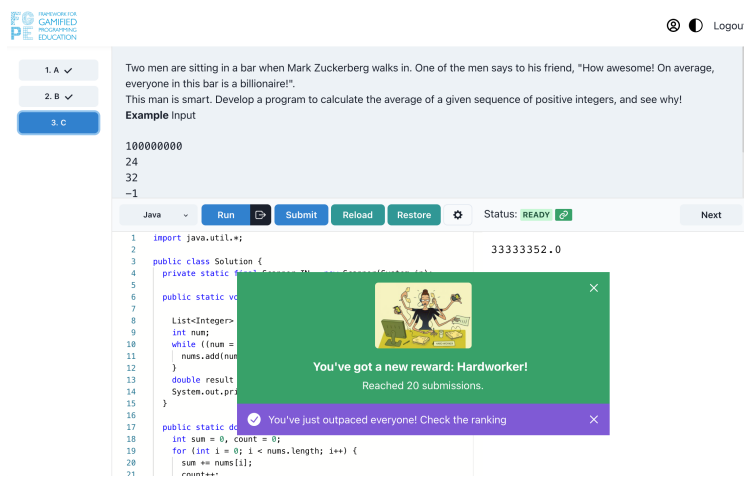
An automated evaluation engine guarantees accurate and timely feedback to students. Such an engine is responsible for marking and grading exercises. In this ecosystem, an evaluation engine will: (1) receive the identification of the student, a reference to the exercise, and the student's attempt to solve it; (2) load the exercise by reference from the repository (if necessary); (3) compile the solution and run the tests against the student's program, comparing the obtained output to the expected; and (4) build report of the evaluation with passed/failed test cases, the grade, and, possibly, some feedback. Therefore, Mooshak 2 has been selected. Mooshak [18] is an open-source web-based system for managing programming contests, which includes automatic judging of submitted programs. Version 2 inherits all features from its original version, improving feedback and attaching support for different types of exercises. Moreover, Mooshak supports custom static and dynamic analysis scripts which enables assessment for the non-traditional programming exercises previously described.

Finally, having defined two new formats, this ecosystem required a tool to facilitate the authoring of new programming exercises adhering to and importing exercises in existing formats into such new programming exercise format as well as their connection with gamification rules. FGPE AuthorKit [23] is a web application designed to support the authors through the entire process of preparing gamified programming exercises, including (1) creation of exercises with their associated metadata, (2) design of the gamification techniques for a

specific exercise or their collection, (3) definition of the content structure and sequencing rules, and (4) importing and exporting the content in several popular programming exercise formats, using YAPEXIL and GEdIL as the base. The user can share content with other peers, internally or via a GitHub repository where all exercise data is synchronized.

3.2 User Interface

The FGPE PLE is a progressive web application developed in ReactJS [7] – “a JavaScript library for building user interfaces” –, meaning that it is an application software delivered through the web intended to work on any platform that utilizes a standards-compliant browser, including both desktop and mobile devices. It uses the Apollo Client, a community-driven effort to build an easy-to-understand, flexible, and powerful GraphQL client, to manage and consume remote data from the FGPE Gamification Service without worrying about infrastructure code for networking and caching.



■ **Figure 2** Exemplary screenshot of the FGPE PLE User interface.

The User Interface (UI), whose exemplary view is presented in Figure 2, follows a component-driven development approach, which consists in dissecting and splitting the interface into small repeatable patterns, develop these patterns, and join them together to form larger components and pages. The main components of the PLE are the following.

Code Editor is based on Monaco Editor [19], the editor that powers Visual Studio Code.

This editor allows students to code in almost any programming language, starting from a skeleton provided by the exercise author, taking advantage of a vast set of features such as syntax highlighting, parameter hints, smart code navigation, and code completion. For challenges that can be solved in more than one programming language, a language switch is attached at the top of the editor allowing the student to choose the programming language of the editor.

Console is where the results and feedback from executions and submissions is presented. The execution consists in taking the inputs provided by the student through a popup and running the program against these inputs. Submissions follow a similar process but run against the complete set of test cases provided by the exercise author. Unlike an execution, a submission is considered for statistics and can unlock new resources of the course.

Statement Viewer is where the activity statement is displayed. This component can display either Markdown, HTML, or raw text files.

Leaderboard is the component responsible for displaying the usernames and scores, sorted according to certain metrics. Leaderboards can be challenge-scoped or course-scoped, use any of the available metrics to sort users by, and optionally have group visibility.

Push Notifications are small rectangle boxes displayed based on events received from registered GraphQL subscriptions. For instance, received rewards, results of both processed submissions and validations, and other updates.

Profile is the student's space to show off her achievements, including badges, virtual items, experience points, leaderboards' top positions, and course progress.

4 Conclusion

Although gamification has its effectiveness demonstrated in engaging students with learning activities, to the best of the authors' knowledge, until now there has been no programming learning environment that would be, at the same time, open-source, reusable across distinct courses on various programming languages, and enabling instructors to compose their own gamification layer providing a good selection of components to choose from.

This paper introduces a novel programming learning environment having all the traits mentioned above. It is implemented as a progressive web application that constitutes a single interface to a complete ecosystem for gamified programming education developed within the Framework for Gamified Programming Education project [9]; particularly, it renders exercises in the YAPEXIL format [24] and connects with the FGPE GS [21], which processes students' submissions and gamification rules defined in the GEdIL [31] format, generating a relevant instant feedback to the students. Both the programming exercises and gamification layers can be created with the companion FGPE AuthorKit tool [23].

The work described here constitutes the final artifact to complete the ecosystem proposed by the Framework for Gamified Programming Education project [9]. A complete validation of the work developed in this project with a large group of students is the immediate next step in our agenda. Furthermore, the FGPE project will have its continuation in the FGPE Plus project, which aims to make the PLE embeddable in any LTI-compliant Learning Management System as well as enhance the PLE's user experience on mobile devices.

References

- 1 Paul E. Anderson, Thomas Nash, and Renée McCauley. Facilitating Programming Success in Data Science Courses Through Gamified Scaffolding and Learn2Mine. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '15*, pages 99–104, New York, NY, USA, 2015. ACM. doi:10.1145/2729094.2742597.
- 2 Tapio Auvinen, Lasse Hakulinen, and Lauri Malmi. Increasing students' awareness of their behavior in online learning environments with visualizations and achievement badges. *IEEE Transactions on Learning Technologies*, 8(3):261–273, 2015. doi:10.1109/tlt.2015.2441718.
- 3 Gabriel Barata, Sandra Gama, Joaquim Jorge, and Daniel Gonçalves. Studying student differentiation in gamified education: A long-term study. *Computers in Human Behavior*, 71(Supplement C):550–585, 2017. doi:10.1016/j.chb.2016.08.049.
- 4 Andrija Bernik, Goran Bubas, and Danijel Radosevic. A Pilot Study of the Influence of Gamification on the Effectiveness of an e-Learning Course. In *Central European Conference on Information and Intelligent Systems*, pages 73–79, Varaždin, Croatia, 2015. University of Zagreb, Faculty of Organization and Informatics.

- 5 Mads T. Bonde, Guido Makransky, Jakob Wandall, Mette V. Larsen, Mikkel Morsing, Hanne Jarmer, and Morten O. A. Sommer. Improving biotech education through gamified laboratory simulations. *Nature biotechnology*, 32(7):694–697, 2014. doi:10.1038/nbt.2955.
- 6 Katheryn R. Christy and Jesse Fox. Leaderboards in a virtual classroom: A test of stereotype threat and social comparison explanations for women’s math performance. *Computers & Education*, 78(Supplement C):66–77, 2014. doi:10.1016/j.compedu.2014.05.005.
- 7 Facebook. React: A JavaScript library for building user interfaces. <https://reactjs.org>, 2021. accessed on 16 Jan 2021.
- 8 Usef Faghihi, Albert Brautigam, Kris Jorgenson, David Martin, Angela Brown, Elizabeth Measures, and Sioui Maldonado-Bouchard. How gamification applies for educational purpose specially with college algebra. *Procedia Computer Science*, 41(Supplement C):182–187, 2014. 5th Annual International Conference on Biologically Inspired Cognitive Architectures, 2014 BICA. doi:10.1016/j.procs.2014.11.102.
- 9 FGPE Consortium. Framework for Gamified Programming Education. <https://fgpe.usz.edu.pl>, 2018. accessed on 19 Mar 2021.
- 10 Wad Ghaban and Robert Hendley. How Different Personalities Benefit From Gamification. *Interacting with Computers*, 31(2):138–153, March 2019. doi:10.1093/iwc/iwz009.
- 11 Lasse Hakulinen, Tapio Auvinen, and Ari Korhonen. The Effect of Achievement Badges on Students’ Behavior: An Empirical Study in a University-Level Computer Science Course. *International Journal of Emerging Technologies in Learning*, 10(1):18–29, 2015. doi:10.3991/ijet.v10i1.4221.
- 12 Michael D. Hanus and Jesse Fox. Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education*, 80:152–161, 2015. doi:10.1016/j.compedu.2014.08.019.
- 13 Tatsuhito Hasegawa, Makoto Koshino, and Hiromi Ban. An English vocabulary learning support system for the learner’s sustainable motivation. *SpringerPlus*, 4(1):99, 2015. doi:10.1186/s40064-015-0792-2.
- 14 Caitlin Holman, Stephen J. Aguilar, Adam Levick, Jeff Stern, Benjamin Plummer, and Barry Fishman. Planning for Success: How Students Use a Grade Prediction Tool to Win Their Classes. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 260–264, New York, NY, USA, 2015. ACM. doi:10.1145/2723576.2723632.
- 15 Jincheul Jang, Jason J. Y. Park, and Mun Y. Yi. Gamification of online learning. In Cristina Conati, Neil Heffernan, Antonija Mitrovic, and M. Felisa Verdejo, editors, *Artificial Intelligence in Education*, pages 646–649, Cham, 2015. Springer International Publishing.
- 16 Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. Towards a Systematic Review of Automated Feedback Generation for Programming Exercises. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 41–46, Arequipa Peru, 2016. ACM. doi:10.1145/2899415.2899422.
- 17 Keycloak. Keycloak: open source identity and access management solution. <https://www.keycloak.org>, 2014. accessed on 9 Jan 2021.
- 18 José Paulo Leal and Fernando Silva. Mooshak: A Web-based multi-site programming contest system. *Software: Practice and Experience*, 33(6):567–581, 2003. doi:10.1002/spe.522.
- 19 Microsoft. Monaco Editor. <https://microsoft.github.io/monaco-editor/>, 2021. accessed on 16 Jan 2021.
- 20 Christa R Nevin, Andrew O Westfall, J Martin Rodriguez, Donald M Dempsey, Andrea Cherrington, Brita Roy, Mukesh Patel, and James H Willig. Gamification as a tool for enhancing graduate medical education. *Postgraduate Medical Journal*, 90(1070):685–693, 2014. doi:10.1136/postgradmedj-2013-132486.
- 21 José Carlos Paiva, Alicja Haraszczuk, Ricardo Queirós, José Paulo Leal, Jakub Swacha, and Sokol Kosta. FGPE Gamification Service: A GraphQL Service to Gamify Online Education.

- In *Proceedings of the 9th World Conference on Information Systems and Technologies*, Terceira Island, Azores, Portugal, 2021. Springer. (to appear).
- 22 José Carlos Paiva, José Paulo Leal, and Ricardo Alexandre Queirós. *Enki: A Pedagogical Services Aggregator for Learning Programming Languages*, page 332–337. ACM, New York, NY, USA, 2016.
 - 23 José Carlos Paiva, Ricardo Queirós, José Paulo Leal, and Jakub Swacha. FGPE AuthorKit – A Tool for Authoring Gamified Programming Educational Content. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '20, page 564, New York, NY, USA, 2020. ACM. doi:10.1145/3341525.3393978.
 - 24 José Carlos Paiva, Ricardo Queirós, José Paulo Leal, and Jakub Swacha. Yet Another Programming Exercises Interoperability Language (Short Paper). In Alberto Simões, Pedro Rangel Henriques, and Ricardo Queirós, editors, *9th Symposium on Languages, Applications and Technologies (SLATE 2020)*, volume 83 of *OpenAccess Series in Informatics (OASISs)*, pages 14:1–14:8, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/OASISs.SLATE.2020.14.
 - 25 Mads Kock Pedersen, Anette Svenningsen, Niels Bonderup Dohn, Andreas Lieberoth, and Jacob Sherson. DiffGame: Game-based Mathematics Learning for Physics. *Procedia - Social and Behavioral Sciences*, 228(Supplement C):316–322, 2016. 2nd International Conference on Higher Education Advances, HEAd'16, 21-23 June 2016, València, Spain. doi:10.1016/j.sbspro.2016.07.047.
 - 26 Bernadette Perry. Gamifying French Language Learning: A Case Study Examining a Quest-based, Augmented Reality Mobile Learning-tool. *Procedia - Social and Behavioral Sciences*, 174(Supplement C):2308–2315, 2015. International Conference on New Horizons in Education, INTE 2014, 25-27 June 2014, Paris, France. doi:10.1016/j.sbspro.2015.01.892.
 - 27 Robin K. Pettit, Lise McCoy, Marjorie Kinney, and Frederic N. Schwartz. Student perceptions of gamified audience response system interactions in large group lectures and via lecture capture technology. *BMC medical education*, 15(1):92, 2015. doi:10.1186/s12909-015-0373-7.
 - 28 Bardh Prenkaj, Giovanni Stilo, and Lorenzo Madeddu. Challenges and Solutions to the Student Dropout Prediction Problem in Online Courses. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3513–3514, New York, NY, USA, 2020. ACM. doi:10.1145/3340531.3412172.
 - 29 Valerie J. Shute. Focus on Formative Feedback. *Review of Educational Research*, 78(1):153–189, 2008. doi:10.3102/0034654307313795.
 - 30 Jakub Swacha. State of Research on Gamification in Education: A Bibliometric Survey. *Education Sciences*, 11(2):69, 2021. doi:10.3390/educsci11020069.
 - 31 Jakub Swacha, José Carlos Paiva, José Paulo Leal, Ricardo Queirós, Raffaele Montella, and Sokol Kosta. GEdIL–Gamified Education Interoperability Language. *Information*, 11(6):287, May 2020. doi:10.3390/info11060287.
 - 32 Jakub Swacha, Ricardo Queirós, José Carlos Paiva, and José Paulo Leal. Defining requirements for a gamified programming exercises format. *Procedia Computer Science*, 159:2502–2511, 2019. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019. doi:10.1016/j.procs.2019.09.425.
 - 33 Alexandru Topîrceanu. Gamified learning: A role-playing approach to increase student in-class motivation. *Procedia Computer Science*, 112(Supplement C):41–50, 2017. Proceedings of KES 2017, Marseille, France. doi:10.1016/j.procs.2017.08.017.
 - 34 Andika Y. Utomo, Afifa Amriani, Alham F. Aji, Fatin R. N. Wahidah, and Kasiyah M. Junus. Gamified E-learning model based on community of inquiry. In *Advanced Computer Science and Information Systems (ICACSIS), 2014 International Conference on*, pages 474–480, Jakarta, Indonesia, 2014. IEEE, IEEE. doi:10.1109/icacsis.2014.7065830.
 - 35 Ibrahim Yildirim. The effects of gamification-based teaching practices on student achievement and students' attitudes toward lessons. *The Internet and Higher Education*, 33(Supplement C):86–92, 2017. doi:10.1016/j.iheduc.2017.02.002.