Enhancing the Mutation Testing Education through a Gamified Environment

Dário Darienço[1], Alessandreia Marta de Oliveira[2], Ricardo Ferreira Vilela[3], Williamson Silva[4], Maurício Massaru Arimoto[1], Pedro Henrique Valle[5]

[1]Universidade Estadual do Norte do Paraná - UENP - Bandeirantes, PR – Brasil

[2]Departamento de Ciência da Computação - UFJF - Juiz de Fora, MG – Brasil

[3]Universidade Estadual de Campinas - UNICAMP - Limeira, SP – Brasil

[4]Universidade Federal do Cariri - UFCA - Juazeiro do Norte, CE – Brasil

[5]Universidade de São Paulo - IME-USP - São Paulo, SP – Brasil

d.darienco@gmail.com, alessandreia.oliveira@ufjf.br,

rfvilela@unicamp.br, williamson.silva@ufca.edu.br,

mauricio@uenp.edu.br, pedrohenriquevalle@usp.br

Abstract. *Software testing plays a central role in quality assurance. However, challenges remain in teaching advanced concepts, such as mutation testing. The literature suggests that traditional methods do not always effectively support student engagement, motivating the adoption of active learning approaches. We present GAMutation, a gamified environment developed with the Construct 2 to support the teaching of mutation testing. To evaluate its usability, a heuristic evaluation was conducted with undergraduate students. The results indicated positive perceptions and highlighted aspects of the interface and interaction that may guide future improvements. The study reinforces the potential of gamification as a complementary strategy in software testing education.*

1. Introduction

The growing interest in high-quality software products has driven advances in Verification, Validation, and Testing activities [Fraser and Arcuri 2012, Elgrably and de Oliveira 2021, Delamaro et al. 2016, Lauvås Jr and Arcuri 2018]. Among these activities, software testing plays a central role in identifying and correcting defects, contributing to product reliability and reducing maintenance costs [Sommerville 2011, Valente 2020, de Souza et al. 2012]. To this end, testing professionals must master several techniques, including functional, structural, and defect-based testing [Delamaro et al. 2016], which reinforces the importance of qualified training [Florea and Stray 2019, Fraser and Arcuri 2012].

Despite this demand, there is evidence of a shortage of professionals in software testing [Valle et al. 2015, Rojas and Fraser 2016, Valle 2016] and of low student motivation to learn the topic [Jalil et al. 2023]. This lack of motivation is exacerbated by the perception of testing as a “destructive” activity [Fraser 2017] and by the inherent difficulty of teaching techniques that require systematic defect investigation [Martins et al. 2021].

This scenario becomes even more complex in the case of mutation testing, a criterion known for its high effectiveness but also for its conceptual and computational complexity [Delamaro et al. 2016]. From a Computing Education perspective, the challenge associated with mutation testing is not only technical but also pedagogical. It is content that requires instructors to mediate abstract concepts, multiple application steps, and defect-based reasoning processes, which makes it difficult to address through traditional lecture-based strategies. Teaching mutation testing demands pedagogical approaches that support instructors in organizing the content, gradually introducing concepts, and proposing activities that foster student engagement throughout the teaching process.

In this context, gamification strategies have proven useful in supporting mutant identification, but they do not address collaboration and motivation [Toda et al. 2019, Alzahrani and Alhalafawy 2022, Deterding et al. 2011, Nadi-Ravandi and Batooli 2022]. Several studies have applied gamification to the teaching of testing techniques [Mäntylä and Smolander 2016, Fraser 2017, de Jesus et al. 2018, Straubinger and Fraser 2022, Blanco et al. 2023], although few focus specifically on mutation testing. Among existing initiatives, Rojas et al. [Rojas and Fraser 2016] propose a gamified environment aimed at mutant identification, but without addressing mutation operators, the objectives of the criterion, or the complete process of applying the technique.

Given this gap, the objective of this study is to support the teaching of mutation testing through the development of a gamified environment called **GAMutation**. The proposal aims to make learning more motivating and to foster an understanding of the criterion, from fundamental concepts to the practical execution of the mutation process.

The remainder of this document is organized as follows: Section 2 presents related approaches. Section 1 describes the methodology adopted in this study. Section 4 details the proposed gamified environment. Section 5 presents the planning and execution of the usability evaluation. Section 6 discusses the results obtained, while Section 7 describes threats to the validity of this work. Finally, Section 8 presents the conclusions and directions for future work.

2. Related Work

Mutation testing is a structural technique that evaluates the quality of test cases by systematically introducing small changes into the code, known as mutants, which must be detected and eliminated by adequate tests [DeMillo et al. 1979, Offutt 1996, Jia and Harman 2011, Valle et al. 2017]. Because it requires analytical reasoning and a refined understanding of program behavior, its teaching often poses challenges for novice students. Although several initiatives apply gamification to software testing education, the number of studies specifically dedicated to teaching mutation testing through gamified systems remains quite limited. This reveals a relevant gap in the literature, reinforcing the need for educational solutions that make learning mutation testing more accessible and motivating.

Code Defenders [Rojas and Fraser 2017] transforms mutation testing into a competitive game between attackers, who are responsible for generating mutants, and defenders, who are tasked with producing tests capable of killing them. Its objective is to make learning more active and motivating, and studies report higher engagement and improved conceptual understanding. This is a practical approach centered on real-time interaction among players. Mutant Codebraker [Autor 2024] is, in turn, a serious game that presents challenges related to identifying, creating, and eliminating mutants. Its objective is to facilitate understanding of mutation operators and the reasoning required to kill mutants. Evaluation results indicate gains in engagement and conceptual clarity among novice students.

ProTesters [Lima et al. 2022] is a board game designed to teach the general software testing process. Although it does not address mutation testing, its objective is to support understanding of testing activities, roles, and strategies through collaborative interactions. Results indicate improvements in motivation and understanding of testing processes, demonstrating the potential of educational games in this domain. GAMUT [Farias et al. 2021], in contrast, employs narrative, progression through stages, and practical challenges to teach unit testing. The results show increased engagement and content retention. Although it does not address mutation testing, it offers relevant methodological principles for designing gamified environments that aim to teach testing techniques.

Finally, the OER for mutation testing described by Silva et al. [2023] provides structured materials to support learning the mutation criterion, including examples, exercises, and conceptual explanations. Evaluations indicate good acceptance among instructors and students, serving as theoretical support for teaching activities.

Although these initiatives demonstrate the potential of gamified approaches in software testing education, they differ from GAMutation in scope and focus. While tools such as Code Defenders emphasize competition and real-time interaction, and others focus on isolated aspects of testing, GAMutation seeks to support a structured and progressive understanding of mutation testing, covering conceptual foundations, mutation operators, mutant generation, and mutant elimination. This positioning reinforces GAMutation's contribution as an educational resource designed to guide learners through the entire mutation testing process.

3. Methodology

This study adopted an applied research design with a qualitative and exploratory approach, situated within the context of Computing Education. From this perspective, the study investigates how a gamified environment can support the teaching of mutation testing concepts, seeking to identify initial evidence of its educational potential rather than to produce statistical generalizations.

The methodological design combines activities of conception, development, and educational evaluation of a computational artifact, characterizing it as an empirical study focused on analyzing students' perceptions within an academic context. The methodology was structured into four main stages, as illustrated in Figure 1.

In the first stage, we conducted a literature review to identify approaches to software testing education, the application of gamification in educational contexts, and the teaching of mutation testing. This stage shows the pedagogical and design decisions for the environment, ensuring alignment with established practices in Computing Education.

The second stage involved the conception of the GAMutation environment, which entailed the explicit definition of educational objectives, covered content, pedagogical progression, and gamification elements employed as strategies for engagement and instructional support. In this phase, we sought to articulate technical and educational aspects so that game elements would act as mediators of the teaching process, rather than merely as aesthetic or motivational resources.

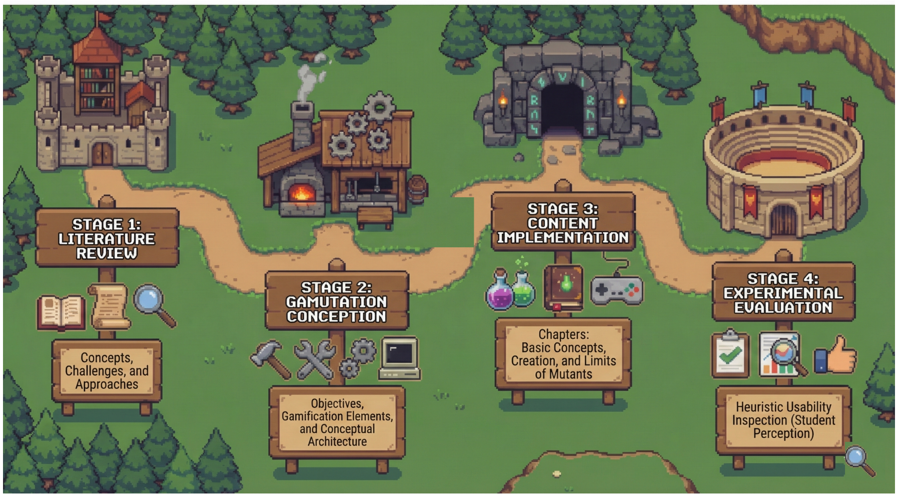


Figure 1. Methodology to conduct the study

The development of the environment was carried out using the Construct 2 game engine as the platform for implementation. This choice was motivated by its wide adoption, low entry barrier, and support for Web-based deployment, which facilitates replication, adaptation, and reuse of the environment by other educators interested in adopting or extending GAMutation in different institutional contexts. Technical decisions were guided by the need to enable the implementation of didactic content and interaction mechanics, while preserving coherence between educational objectives and the user experience.

In the third stage, the educational content was implemented and organized into progressive chapters, covering topics ranging from introductory concepts of mutation testing to practical activities focused on generating, analyzing, and eliminating mutants. This organization aimed to foster an incremental teaching approach, allowing students to gradually advance in the complexity of the presented concepts.

Finally, the fourth stage involved planning and executing a heuristic evaluation of the environment, focusing on the analysis of usability and user experience aspects based on the perceptions of undergraduate students. Although heuristic evaluation is traditionally associated with the field of Human–Computer Interaction, in this study, it is employed as an instrument to capture educational perceptions related to content clarity, activity organization, and the potential of the environment to support the teaching and learning process.

4. GAMutation

This section presents the description of GAMutation, as well as the development of the environment and the educational content considered. It is essential to note that the target audience of GAMutation comprises undergraduate students enrolled in Computing-related programs.

4.1. Instructional Objectives

GAMutation was designed with explicit instructional objectives aimed at supporting the teaching of mutation testing in undergraduate Computing programs. The environment aims to support instructors in presenting and explaining key concepts related to mutation testing, including mutants, mutation operators, and the rationale behind the criteria. In particular, GAMutation supports the instructional process by enabling the structured presentation of mutation operators, illustrating mutant generation, and facilitating discussions on test case adequacy through interactive challenges.

These instructional objectives guided the organization of the content into progressive chapters and the selection of gamified mechanics, ensuring coherence between instructional intentions and interaction design.

4.2. Context and Problem

By observing the challenges associated with teaching mutation testing and their impact on test case design, we identified the need for solutions that effectively support instruction in this technique. From this perspective, GAMutation is a gamified environment designed to support the teaching of mutation testing. By observing the shortage of qualified professionals in the software testing domain and identifying the possible causes of this deficit, we also recognized the relevance of solutions capable of addressing or mitigating this problem. In this context, GAMutation aims to provide an educational environment that supports the teaching and learning process of mutation testing.

4.3. *Design* and Art

All resources used in the development of GAMutation were obtained through *Creative Commons* licenses that allow the free distribution of these assets. With respect to the *design* and artwork employed, the following stand out: (i) images made available on the platforms `www.itch.io` and `www.freepik.com/`; (ii) sound effects obtained from the YouTube platform under a *Creative Commons* license; and (iii) music developed by the *omitted for review*, which authorized the use of its digital music.

4.4. *Game Engine*

Construct[1] was used, an HTML5-based game development tool that enables the creation of games using a drag-and-drop approach through a visual editor and a behavior-based logic system. Construct allows both experienced programmers and non-programmers to design games that can be exported to *Web* and/or *Mobile* platforms.

4.5. Gamification Elements

The taxonomy proposed by Toda *et al.* (2019) was used to determine which gamification elements should compose the gamified environment, including levels, points, *ranking*, and time constraints. The selected elements were explored with the objective of generating a positive impact on students and supporting the teaching and learning process of mutation testing.

These gamification elements are implemented throughout the environment and across its chapters, according to the activities proposed in each stage of GAMutation. Table 1 presents the distribution of the content and the gamification elements explicitly described in each chapter.

[1] `https://construct.net/en/construct-2/`

Table 1. Content and Gamification Elements Across Chapters

Chapter	Content	Activity	Gamification Elements
1	Basic concepts (mutant definition, competent programmer hypothesis, coupling effect)	Quiz	Points for correct answers; heart loss for incorrect answers; positive/negative sound feedback; progression by correct responses.
2	Mutation testing steps	Quiz	Points for correct answers; progression by completion.
3	Mutation operators (C/Java)	Platform interaction	200-second time limit; operator collection; enemy elimination for points; points accumulation; level completion via portal.
4	Mutant generation	Platform interaction	Collection of generated mutants in scenario.
5	Mutant elimination via test cases	Test case challenge	Points gained for correct test cases; points lost for incorrect answers; progression to next mutant upon success.
All chapters	Entire environment	Applies to all chapters	Ranking system; level progression.

4.6. Content

GAMutation addresses the basic concepts of mutation testing as presented in the textbook by Delamaro, Maldonado, and Jino (2016): (i) definitions and basic concepts; (ii) steps for applying mutation testing; (iii) mutation operators; (iv) generation of mutant programs from mutation operators; and (v) elimination of mutants using test cases. Each of these concepts was transformed into chapters (Subsection 4.7) in order to better guide students during their interaction with the environment.

4.7. GAMutation Chapters

The stages of GAMutation are divided into chapters that represent the learning objects and the content to be taught about mutation testing. Currently, GAMutation consists of five chapters, organized as follows:

Chapter 1. This chapter consists of a *quiz* that addresses basic concepts of mutation testing: (i) the definition of a mutant; (ii) the competent programmer hypothesis; (iii) the coupling effect hypothesis, among other concepts. The student begins their journey with zero points; however, by selecting the correct alternatives, they earn points and consequently receive *feedback* through a positive sound effect. When incorrect alternatives are selected, hearts (a gamified element) are lost, and the student receives *feedback* through a negative sound effect. Figure 2. (a) illustrates an activity from this chapter.

Chapter 2. This chapter also consists of a *quiz*, in which the covered content includes the steps for applying the mutation testing criterion, such as: (i) the order of the steps for applying the criterion; (ii) what occurs during the execution process of the program under test; and (iii) other issues related to the application steps of this criterion. Figure 2. (b) presents an example of an activity from this chapter.

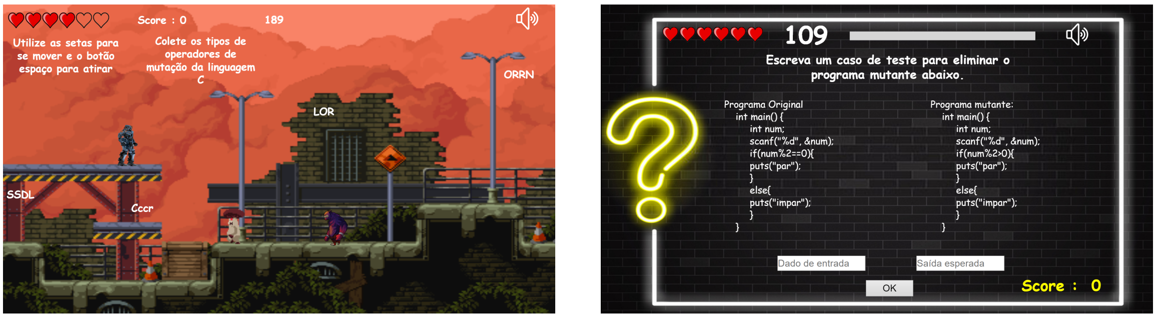
Chapter 3. This chapter addresses content related to mutation operators. In this chapter, several mutation operators are presented, and students must categorize them according to their programming language. The student controls a robot within a scenario in which mutation operators from the C and Java programming languages are scattered throughout the environment. The student must collect the C operators scattered throughout the scenario and reach the portal at the end of the level within the stipulated time limit of 200 seconds.. By eliminating enemies, the student also earns points, as shown in Figure 3(a).

(a) **(b)**

Figure 2. Examples of activities from Chapters 1 (a) and 2 (b)

Chapter 4. This chapter addresses the generation of mutants through mutation operators. It presents a platform game simulation that displays a program code and a mutation operator. In this context, the student must search for and collect the mutant programs generated within the scenario using the displayed mutation operator.

Chapter 5. This chapter addresses the elimination of mutants using test cases. It displays the code of the original program and the code of the mutant program generated using mutation operators. Thus, the student must create a test case capable of identifying the mutation. When the test case is correctly created, the student earns points and advances to the next mutant program; otherwise, points are lost, as illustrated in Figure 3.(b).



(a) **(b)**

Figure 3. Examples of activities from Chapters 3 (a) and 5 (b)

4.8. GAMutation Execution

GAMutation can be used as a supporting educational resource in courses that address Software Testing or Software Engineering, particularly during the introduction or consolidation of mutation testing concepts. The environment may be adopted as a complementary activity to traditional lectures, as a guided laboratory exercise, or as an autonomous learning resource for students to explore mutation testing concepts at their own pace.

GAMutation can be accessed through a Web browser, which facilitates its adoption in different institutional contexts without the need for local installation. It is important to note that logging in with a Google account is required to access the content of the gamified environment.

5. GAMutation Evaluation

To evaluate the educational potential of GAMutation as a supporting tool for teaching mutation testing, we conducted an inspection based on the heuristic evaluation technique [Souza and Souto 2015], following the process illustrated in Figure 4. From a Computing Education perspective, this evaluation sought to analyze not only usability aspects, but also how the interface, gamified mechanics, and content organization influence the educational experience of undergraduate students who had already completed Software Engineering courses. The evaluation was guided by the following research question: “How do students perceive GAMutation as a supporting environment for teaching mutation testing, considering aspects of usability, engagement, and content clarity?”.

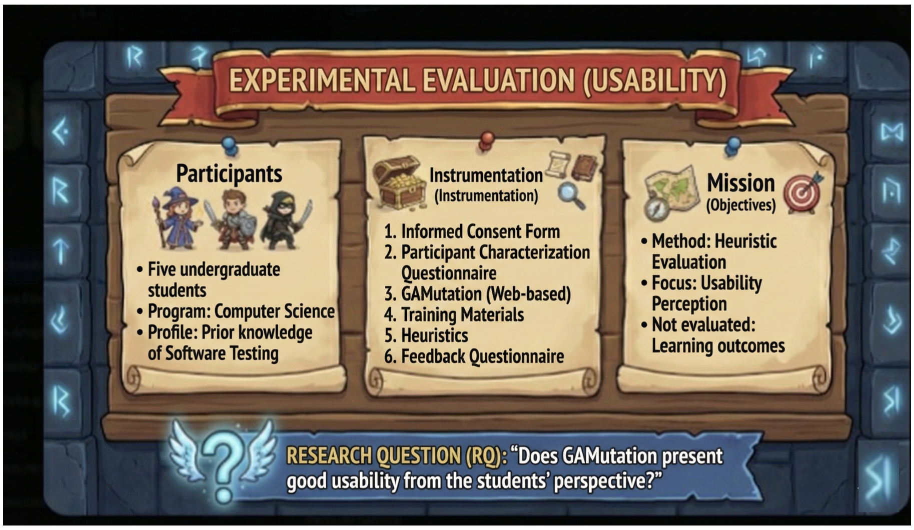


Figure 4. Proposed Evaluation

Five undergraduate students participated in the evaluation, which is the recommended number for heuristic inspections. All participants volunteered and signed an informed consent form that presented the educational nature of the study and the confidentiality conditions. The characterization questionnaire enabled the identification of participants' educational profiles, including their prior knowledge levels about mutation testing and their familiarity with gamified environments, aspects relevant to interpreting their perceptions of GAMutation as a teaching and support resource.

The results presented in Table 2 indicate that 80% of the participants had basic knowledge of mutation testing, while 20% reported intermediate knowledge, with no participants at an advanced level. This profile reinforces the suitability of GAMutation as a tool to support introductory teaching of the topic. Regarding gamification, a higher level of familiarity was observed: 20% had a basic level, 40% an intermediate level, and 40% an advanced level. This characteristic suggests that participants were able to critically evaluate how gamified elements influenced their motivation, engagement, and understanding of the educational content.

The study employed the following instrumentation: (i) the informed consent form; (ii) the characterization questionnaire; (iii) the GAMutation gamified environment, made available on a Web platform; (iv) training material on mutation testing, gamification, and heuristic evaluation; (v) the set of 28 heuristics proposed by Souza and Souto

[Souza and Souto 2015], evaluated on a severity scale ranging from 0 (no problem) to 4 (catastrophic problem); and (vi) an open-ended feedback form for collecting qualitative perceptions. After the training, participants used GAMutation to apply the heuristics and identify usability problems. At the end of the activity, they completed the feedback form, recording their overall perceptions of the environment, its functionalities, and the clarity of the proposed interactions.

Table 2. Profile of the participants in the heuristic evaluation

Knowledge Level	Mutation (%)	Gamification (%)
Basic	80%	20%
Intermediate	20%	40%
Advanced	0%	40%

6. Results Analysis and Discussion

According to Souza and Souto (2015), evaluation heuristics can be employed to support the design of educational gamified environments, providing input not only for improving the interface, but also for reflecting on how design choices may influence users' educational experience. Through the heuristic evaluation, we identified 12 usability problems, as presented in Table 6, with severity levels 3 (Severe) and 4 (Catastrophic).

Table 3. Heuristic Evaluation Results

ID	Problem Description	Sev.
HA02	Difficulty in identifying all available functionalities.	3
HA06	Requires improvement.	3
HA12	The imposed difficulty and time constraints may negatively affect content assimilation.	3
HA15	Insufficient balance of difficulty levels.	3
HA15	Only one difficulty level available.	4
HA16	Advancement to the next chapter is only possible after completing the current one.	3
HA16	Winning is only possible by answering correctly.	4
HA16	Difficulty in finding alternative ways to win.	3
HA18	Excessive information displayed on the screen.	3
HA21	Lack of communication among users.	4
HA23	Lack of support for user groups or communities.	4
HA24	Users cannot meet each other within the environment.	4
HA28	High difficulty level.	3
HA28	Potential frustration for users unfamiliar with gamified educational environments.	3

Regarding HA02 ("*The user should be able to easily turn the gamified environment on or off, view options, obtain help, save, and pause at different stages*"), participants reported difficulties in identifying all the functionalities available in GAMutation. From an educational perspective, this limitation may compromise student autonomy during the learning process, as navigation barriers hinder access to resources necessary for understanding mutation testing concepts. Although interface elements were positioned with the intention of facilitating use, participants' reports indicate the need for design adjustments to better support the educational experience.

Concerning HA06 ("*The interface of the gamified environment should be consistent in terms of navigation, design, and dialogues*"), participants pointed out the need for interface improvements. From an educational standpoint, inconsistencies in navigation and visual elements may increase students' cognitive load, diverting attention away from mutation testing concepts and toward understanding how the environment operates. Although buttons and interface elements were implemented to simplify interaction, further refinements are required to better support the learning process.

With respect to HA12 ("*The user should obtain fair results*"), participants reported that the imposed time limits and difficulty levels could hinder the assimilation of the proposed educational content, particularly for novice learners. From an educational perspective, excessive time constraints may negatively affect the learning process, particularly for conceptually complex topics such as mutation testing.

Regarding HA15 ("*The gamified environment should offer different difficulty levels*"), participants reported a lack of balance among the proposed difficulty levels. Educationally, insufficient progression may compromise student motivation and persistence, as tasks that are either too simple or overly complex tend to reduce engagement. In GAMutation, chapters were designed with simple to moderate difficulty to encourage continuity; however, the feedback indicates that more gradual progression is needed.

Regarding HA16 ("*The gamified environment should provide multiple ways to win*"), participants indicated that GAMutation offers limited strategies to achieve the proposed objectives. From an educational standpoint, restricted paths may limit active exploration and experimentation, which are fundamental for learning complex concepts such as mutation testing. Although some game mechanics provide bonus rewards, participants suggested that diversifying strategies could better accommodate different learning styles.

Based on HA18 ("*Gamified environments involving stories and characters should arouse interest in what they represent*"), participants reported difficulties in maintaining attention due to the amount of information displayed, particularly from Chapter 4 onward. From an educational perspective, excessive visual information may negatively affect attention and conceptual understanding. Nevertheless, the spatial organization of code elements was designed to minimize confusion, reflecting an effort to strike a balance between conceptual fidelity and pedagogical clarity.

Regarding HA21 ("*The gamified environment should support communication*"), participants reported that communication mechanisms among GAMutation users were absent. In Computing Education, communication among students may play a significant role in supporting learning by fostering interpretation exchange, clarifying doubts, and constructing shared meaning around complex concepts such as mutation testing. Although synchronous or asynchronous communication was not considered in the initial design, its absence limits opportunities for pedagogical mediation and discussion.

With respect to HA23 ("*The gamified environment should support groups or communities*"), participants indicated that GAMutation lacked support for group or community interaction. From an educational standpoint, the absence of such structures may restrict collaborative learning practices, in which students discuss strategies, share experiences, and collectively construct knowledge. Although the environment was designed for individual use, participants' feedback suggests that incorporating group features could enhance the system's pedagogical potential.

Regarding HA24 ("*The gamified environment should include support for participants to find each other*"), participants reported the absence of spaces in GAMutation that allow users to encounter one another. Educationally, environments that make the presence of other learners visible can contribute to a sense of belonging and increased engagement, even when direct interaction is not the primary focus.

Finally, for HA28 ("*The pace of the gamified environment should challenge participants without frustrating them*"), participants reported that difficulty levels and time limits may generate frustration, particularly among students unfamiliar with gamified environments. From a pedagogical perspective, striking a balance between challenge and frustration is crucial to sustaining engagement. Although GAMutation was designed to be challenging, participants' feedback indicates a need for adjustments that are more sensitive to learners' profiles.

Despite the aforementioned issues, positive aspects of using GAMutation for teaching mutation testing were also identified. Participants reported feeling challenged while interacting with the environment, striving to achieve its objectives without losing focus on learning. They highlighted that distributing content across different gamified mechanics increased attention, as new topics were introduced through varied interaction styles, further enhancing the challenge.

Participants also noted that real-time *feedback* elicited positive emotions when answering correctly and negative emotions when failing within the allotted time. Additionally, competitiveness fostered by the *ranking* system encouraged engagement throughout the learning process. Overall, the results indicate that GAMutation's main benefits are associated with increased motivation, engagement, and attention during mutation testing instruction. Although learning gains were not measured, the observed evidence suggests that the environment provides relevant pedagogical support, reinforcing its potential as a complementary teaching tool.

7. Threats to Validity

When conducting an empirical study in Computing Education, it is essential to assess the validity of the results using the dimensions proposed by Wohlin et al. (2012). This study has an exploratory nature and aims to investigate initial evidence regarding the use of a gamified environment as a strategy to support mutation testing education, rather than producing statistically generalizable findings.

Regarding **conclusion validity**, the main threat concerns the small sample size, a recurring limitation in empirical studies in Informatics in Education and Computing Education [Silva et al. 2020, Silva et al. 2017]. This characteristic restricts the robustness of quantitative inferences; therefore, results should be interpreted as preliminary evidence. We intend to mitigate this threat by replicating the study with a larger number of participants and across different educational contexts.

Construct validity may be affected by the choice of evaluation methods and instruments. Although there is a risk that such instruments may favor GAMutation, this threat is mitigated by the use of well-established and widely validated techniques in the literature, particularly heuristic evaluation applied to educational gamified environments.

With respect to **internal validity**, the use of undergraduate students as participants may have influenced the results due to heterogeneous prior experiences with software testing, gamification, and interface evaluation. Nevertheless, participant anonymity and the absence of external incentives contributed to reducing potential behavioral biases.

Finally, **external validity** is limited by the academic context in which the study was conducted and by participants' profiles, which may restrict generalization to other scenarios, such as industrial settings or courses with different curricular characteristics. To mitigate this threat, prior training was provided; however, further studies in diverse contexts are necessary to strengthen external validity.

8. Final Considerations and Future Work

This study proposed the development and evaluation of a gamified environment, GAMutation, to support the teaching of mutation testing. The environment was developed using Construct 2 and made available online. To assess its usability, a heuristic evaluation was conducted with five undergraduate students who had experience with heuristic evaluation and mutation testing topics.

Regarding the proposed research question ("How do students perceive GAMutation as a supporting environment for teaching mutation testing, considering usability, engagement, and content clarity?") the results indicate that the environment is perceived positively as an educational resource. Participants highlighted GAMutation's potential to foster engagement, sustain attention, and support conceptual understanding through diverse gamified mechanics and real-time feedback. Nevertheless, limitations in usability, progression, and communication and social interaction mechanisms were identified, revealing concrete opportunities for pedagogical improvement.

Among the main contributions of GAMutation, we highlight: (i) the provision of an educational resource specifically designed to support the teaching of mutation testing within Computing Education; (ii) the promotion of student motivation and engagement when dealing with a conceptually complex testing criterion; (iii) pedagogical insights for instructors regarding students' interaction patterns, difficulties, and progress when learning mutation testing; and (iv) the encouragement of student reflection on the relevance of mutation testing and its effectiveness in revealing subtle software defects.

As future work, we intend to implement a new version of GAMutation that incorporates the feedback received during the heuristic evaluation. We also plan to conduct a case study with undergraduate students during mutation testing instruction to gather feedback from both students and instructors regarding the environment's quality. Additionally, new functionalities are planned, including awarding trophies to high-performing students and integrating a chat feature to enhance user interaction. Finally, we intend to make the environment available on the mobile platform.

AI Use Disclosure

AI tools were used solely for editorial support, including language refinement, minor proofreading, and the generation of illustrative figures. All scientific decisions, analyses, and conclusions are the sole responsibility of the authors.

References

Alzahrani, F. K. J. and Alhalafawy, W. S. (2022). Benefits and challenges of using gamification across distance learning platforms at higher education: a systematic review of research studies published during the covid-19 pandemic. *Journal of Positive School Psychology*, 6(10):1948–1977.

Autor, N. (2024). Mutant codebraker: Um jogo para apoiar o ensino de teste de mutação. Trabalho de Conclusão de Curso. Disponível em: repositório institucional.

Blanco, R., Trinidad, M., Suárez-Cabal, M. J., Calderón, A., Ruiz, M., and Tuya, J. (2023). Can gamification help in software testing education? findings from an empirical study. *Journal of Systems and Software*, 200:111647.

de Jesus, G. M., Ferrari, F. C., de Paula Porto, D., and Fabbri, S. C. P. F. (2018). Gamification in software testing: A characterization study. In *Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing*, pages 39–48.

de Souza, D. M., Maldonado, J. C., and Barbosa, E. F. (2012). Aspectos de desenvolvimento e evolução de um ambiente de apoio ao ensino de programação e teste de software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 23.

Delamaro, M., Maldonado, J., and Jino, M. (2016). *Introdução ao teste de software*. Elsevier, Rio de Janeiro, 2 edição edition.

DeMillo, R. A., Lipton, R. J., and Perlis, A. J. (1979). Social processes and proofs of theorems and programs. *Communications of the ACM*, 22(5):271–280.

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15.

Elgrably, I. S. and de Oliveira, S. R. B. (2021). A diagnosis on software testing education in the brazilian universities. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE.

Farias, L., Santos, R., and Barbosa, D. (2021). Gamut: A game-based learning approach for teaching unit testing. In *Proceedings of the Brazilian Symposium on Software Quality (SBQS)*, pages 120–131.

Florea, R. and Stray, V. (2019). The skills that employers look for in software testers. *Software Quality Journal*, 27(4):1449–1479.

Fraser, G. (2017). Gamification of software testing. In *2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)*, pages 2–7, Buenos Aires, Argentina. IEEE.

Fraser, G. and Arcuri, A. (2012). Whole test suite generation. *IEEE Transactions on Software Engineering*, 99.

Jalil, S., Rafi, S., LaToza, T. D., Moran, K., and Lam, W. (2023). Chatgpt and software testing education: Promises & perils. In *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 4130–4137. IEEE.

Jia, Y. and Harman, M. (2011). An analysis and survey of the development of mutation testing. *IEEE Transactions on Software Engineering*, 37(5):649–678.

Lauvås Jr, P. and Arcuri, A. (2018). Recent trends in software testing education: A systematic literature review. In *Norsk IKT-konferanse for forskning og utdanning*.

Lima, J., Silva, M., and Oliveira, P. (2022). Protesters: A board game for teaching the software testing process. In *Anais do Simpósio Brasileiro de Qualidade de Software (SBQS)*, pages 1–12.

Mäntylä, M. V. and Smolander, K. (2016). Gamification of software testing-an mlr. In *Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings 17*, pages 611–614. Springer.

Martins, L., Brito, V., Feitosa, D., Rocha, L., Costa, H., and Machado, I. (2021). From blackboard to the office: A look into how practitioners perceive software testing education. In *Evaluation and Assessment in Software Engineering*, pages 211–220.

Nadi-Ravandi, S. and Batooli, Z. (2022). Gamification in education: A scientometric, content and co-occurrence analysis of systematic review and meta-analysis articles. *Education and Information Technologies*, 27(7):10207–10238.

Offutt, A. J. (1996). A practical system for mutation testing. *Journal of Software Testing, Verification and Reliability*, 6(1):3–23.

Rojas, J. M. and Fraser, G. (2016). Code defenders: A mutation testing game. In *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 1–6, Chicago. IEEE.

Rojas, J. P. and Fraser, G. (2017). Code defenders: A mutation testing game. In *Proceedings of the IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 404–409.

Silva, A., Pereira, C., and Souza, B. (2023). An open educational resource supporting mutation testing teaching. In *Anais do Simpósio Brasileiro de Qualidade de Software (SBQS)*, pages 1–10.

Silva, W., Gadelha, B., Steinmacher, I., and Conte, T. (2020). Towards an open repository for teaching software modeling applying active learning strategies. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pages 162–172. IEEE.

Silva, W. A. F., Steinmacher, I. F., and Conte, T. U. (2017). Is it better to learn from problems or erroneous examples? In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*, pages 222–231. IEEE.

Sommerville, I. (2011). Engenharia de software, 9a. *São Palo, SP, Brasil*, page 63.

Souza, É. R. and Souto, E. (2015). Utilizacao de heuristicas de jogos para avaliacao de um aplicativo gamificado. *Proceedings of SBGames*, pages 666–673.

Straubinger, P. and Fraser, G. (2022). Gamekins: Gamifying software testing in jenkins. In *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*, pages 85–89.

Toda, A. M., Klock, A. C., Oliveira, W., Palomino, P. T., Rodrigues, L., Shi, L., Bittencourt, I., Gasparini, I., Isotani, S., and Cristea, A. I. (2019). Analysing gamification elements in educational environments using an existing gamification taxonomy. *Smart Learning Environments*, 6(1):1–14.

Valente, M. T. (2020). Engenharia de software moderna. *Princípios e Práticas para Desenvolvimento de Software com Produtividade*, 1.

Valle, P., Barbosa, E. F., and Maldonado, J. (2015). Um mapeamento sistemático sobre ensino de teste de software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 71.

Valle, P. H. D. (2016). *Jogos educacionais: uma contribuição para o ensino de teste de software*. PhD thesis, Universidade de São Paulo.

Valle, P. H. D., Toda, A. M., Barbosa, E. F., and Maldonado, J. C. (2017). Educational games: A contribution to software testing education. In *2017 IEEE Frontiers in education Conference (FIE)*, pages 1–8. IEEE.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.