**Survey on Gamification Techniques in Programming Education**

Department of Computer Science and

Engineering,

Rajalakshmi Institute of Technology,

Chennai, India.

[adhihtyavarman.s.s.2021.cse@ritchennai.edu.in](mailto:adhihtyavarman.s.s.2021.cse@ritchennai.edu.in)

Department of Computer Science and

Engineering,

Rajalakshmi Institute of Technology,

Chennai, India.

[anbuchelvan.k.2021.cse@ritchennai.edu.in](mailto:anbuchelvan.k.2021.cse@ritchennai.edu.in)

Ashok M,

Center for Artificial Intelligence,

Rajalakshmi Institute of Technology,

Chennai, India.

[ashok.m@ritchennai.edu.in](mailto:ashok.m@ritchennai.edu.in)

Kumar Ramasamy,

Professor Department of Computer Science &

Engineering,

Knowledge Institute of Technology,

Salem, India.

[rkumarnkl@gmail.com](mailto:rkumarnkl@gmail.com)

***Abstract—*Gamification in education of programming increases motivation and involvement but lacks adaptable, scalable, and standardized frameworks for it. Current systems often serve specific groups, with low generalizability or scalability in contexts such as MOOCs or corporate training. The main challenges are keeping the learning rigor in harmony with motivation and aligning game elements with technical problem-solving. Most of these frameworks fail to accommodate diverse styles of learning because they are either too simple for advanced learners or too complex for beginners. The paper fills these gaps by presenting a systematic, scalable, and adaptable framework of gamification designed specifically to the needs of learners, attempting to improve effectiveness and acceptance in gamified programming education systems.**

***Keywords—Gamification, programming education, scalable frameworks, adaptable learning environments, motivation, engagement, learning rigor, game mechanics, technical problem-solving, personalized learning, massive open online courses (MOOCs), corporate training, systematic frameworks.***

### Introduction

Programming education is required to get students prepared for success in a technology-driven society. However, traditional teaching methods often fail, and some of the reasons are student disinterest, lack of accommodation for students having different levels of ability, and the inability to understand abstract concepts. These problems become more acute in large-scale education systems such as MOOCs and corporate training, as homogeneous practices cannot be followed there. Rising demand for programming skills necessitates innovative and inclusive methods of education [16].

Gamification—the application of game design elements such as points, badges, leaderboards, and challenges into non-game contexts—has plenty of promise in solving this challenge. For programming education, gamification can really enhance engagement, intrinsic motivation, and provide a completely risk-free environment for experimentation [20]. However, most of the available gamified systems have failed in scaling up and catering to the varied needs of learners [3].

This research will provide a flexible and scalable framework specifically designed for programming education in a gamification context, which aims at bringing together educational intensity and interactive components. By solving problems in personalization, scalability, and adaptability, this framework strives to achieve the full advantages of gamification and improve teaching approaches in programming [18].

### II. Methods

#### *A. Criteria for Inclusion*

This research mainly focuses on research and models that have looked into gamification within programming education. The criteria for inclusion have highlighted thematic areas of research such as scalability, personalization, and learner engagement [4]. The studies have been chosen because they are capable of handling the key educational challenges such as the enhancement of learner motivation, accommodating different levels of skill, and educational rigor.

Only entertainment-gamification-based studies with no supporting evidence linking it to programming education, technical problem-solving or educational outcome, were left out. Such an activity was done for avoiding too much of vacuousness- and failure-grounds' frameworks that poorly fit with the intended purposes of gaming. Concerns such as scalability and adaptability during research were a priority as they play an extremely important role in large-scale, diverse learning environments, like MOOCs, higher education courses, and corporate training programs. Emphasis was also placed on frameworks that supported different learner profiles from novices to experienced learners while providing excellent learning experiences.

#### *B. Information Sources*

The prominent frameworks such CodeCombat, CodinGame, and Moodle LMS can be characterized as key components of the review. Challenges in CodeCombat are crafted in story form that appeals to new learners to connect intuitively with programming; CodinGame is challenging with complexity that meets learners at multiple levels of experience.Further, published case studies, comparative analyses, and academic research referred to in the conference paper count as other sources. They all provide the basis for understanding how effective and poorly managed gamified systems work in programming education [5].

#### *C. Search Strategy*

The search strategy designed the frameworks and studies that correspond to the inclusion criteria. Keywords such as "gamification," "programming education," "scalability," and "learner diversity" would also be used for searches conducted in academic databases. Priority was given to frameworks using adaptive mechanisms and personalization strategies [6]. Comparative studies across university courses, MOOCs, and corporate training were also considered into this search to make sure that the insights were diverse [9].

#### *D. Selection Process*

The evaluation process focused on frameworks in terms of whether they can support differentiated learning and scalability. Systems that could adapt the variability of learner abilities, including beginners to advanced users, were identified. Such frameworks include CodeCombat for beginner learners, while CodinGame increases the complexity of activities it introduces. These platforms provided adaptive learning environments that are sensitive to the varying skills and therefore ensured a personalized experience.

Other mechanisms, such as leaderboards, badges, and adaptive challenges to improve engagement, were considered. The scalability challenges also presented several gaps during the discussion; most frameworks fail to address multiple learner profiles [7]. Most research fails to provide personalized solutions that would help solve various learning needs, due either to restrictions in technological infrastructure or insufficiency in customization.

The selection process is thus supposed to screen out systems that have emphasized entertainment in the cost of scholarly intensity and, instead, focuses on those who demonstrate scalability and scholarship. In this respect, by concentrating on scalable, academically sound, and different kinds of learner profiling systems, the review selects only high-quality contributions that make scalable and personalized programming education.

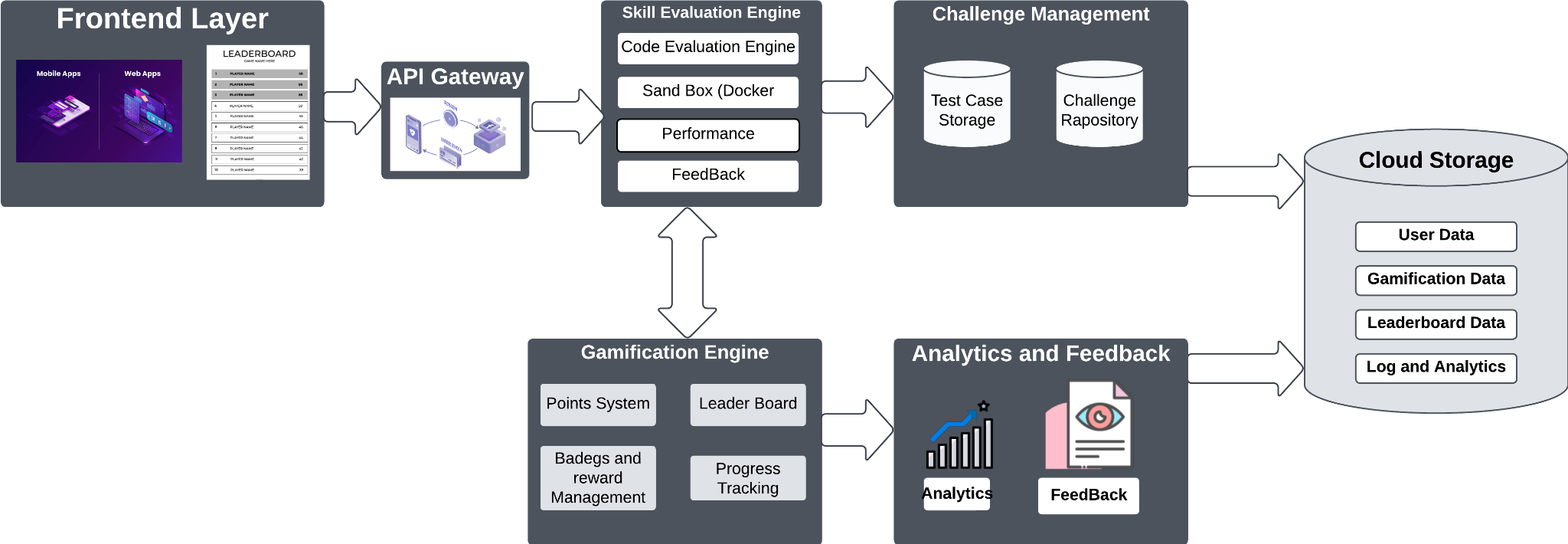
#### *E. Data Collection*

These involved critical gamification aspects such as badges, leaderboards, challenges, and coding tasks that are among the most significant features because they significantly influence the growth of learner's engagement, motivation, and adaptability in learning the programming. The research discussed how these gamification elements had impacted outcomes at the three levels of skills including novice, intermediate, and experts [11].

The purpose of collecting the data was to assess the effectiveness of gamified frameworks based on their contributions to learner motivation, the provision of a sense of achievement, and the adaptability of gamified frameworks to individual progress. Through the assessment of these influences, valuable insights regarding the role of gamification in promoting a more engaging learning environment tailored to individual learners are obtained. The collected data helped identify shortcomings in certain gamified systems that fail to scale or do not fit a variety of learner profiles. Such analysis results in a deep understanding of how gamified frameworks are effective and can be used to improve programming education in a variety of contexts.

#### *F. System Architecture*

The entire concept of this gaming system to assess the programming skills got developed around an easy-to-use front end and a reliable back end for all users. Gamifying designs include futuristic frameworks such as backend REST APIs with the microservices architecture for control of challenges and functionality to perform gaming logic and evaluation of submitted codes [13]. This design also includes a Docker-skinned sandbox for safe execution of user-submitted codes for several programming languages. Relational databases and NoSQL systems will together be used for providing organized adaptable data storage for at least analytics, user profiles, and some problems [12]. Points, badges, and leaderboards are all part of the gamified experience, with monitoring tools such as Prometheus and Grafana being relied upon to prove the integrity of the system. It's scalable for deployment over the cloud, on AWS or GCP. The System architecture is depicted in Fig. 1 for better understanding.

Fig. 1. System Architecture image.

#### *G. Risk of Bias Assessment*

The assessment of bias risk analyzed frameworks in terms of their scope for adaptation, scalability, and resultant effect upon different learner profiles. It then highlighted issues such as over-dependence on external rewards, and difficulties in scaling frameworks to much larger settings like MOOCs [15]. Thus, this ensured an equally balanced view about the advantages and disadvantages of such systems covered here.

### III. Results

#### *A. Study Selection*

The preliminary search yielded around 100 records. Out of the initial retrieved records, after screening based on the inclusion criteria, 30 studies were found applicable for review. Most of those studies excluded from this review did not address issues to do with scalability or have no capability to show academic robustness. This stringent filtering made the review include only high-quality studies that would help accomplish research objectives.

#### *B. Study Characteristics*

These included gamification elements such as leaderboards, badges, and adaptive challenges in many of those studies. The use of such gamification tactics occurred in several other learning environments, including classes delivered by a university, MOOCs, or training by a corporation [8]. After all, the findings were that gamification could also help build learners' motivation and was at least capable of allowing far more engagement for most settings [14].

The key gamification elements used in programming education studies such as leaderboards, badges, adaptive challenges, etc., are presented in Table I. The table details how each element has been used through different learning environments-MOOCs, university courses, and corporate training-along with the observed impacts on learner engagement, motivation, and personalization received through the learning process.

Table I. Gamification Elements and Impact-Programming Education Studies

|  |  |  |  |
| --- | --- | --- | --- |
| **Gamification Element** | **Usage in Study** | **Learning Environment** | **Outcome** |
| Leaderboards | Used to track learner progress and rank students | University courses, MOOCs | Increased competition, engagement, and motivation among learners |
| Badges | Awarded for completing tasks and achieving milestones | Corporate training, MOOCs | Recognition of achievements, enhanced learner satisfaction |
| Adaptive Challenges | Tailored to the learner’s skill level, increasing difficulty over time | University courses, online platforms | Improved learner retention and engagement, personalized learning experience |
| Coding Challenges | Practical tasks requiring learners to write code based on given problems | University courses, coding bootcamps | Enhanced problem-solving skills, active participation in learning |
| Points | Earned through completing tasks, quizzes, and challenges | MOOCs, online programming platforms | Encouraged continued engagement, reinforced intrinsic motivation |

#### *C. Synthesis Methods*

The selected studies helped perform a narrative synthesis aiding the identification of dominant trends and themes. The qualitative synthesis enhanced understanding of how gamification impacts engagement and behavior among learners [19]. Collectively, these methods offer a holistic look into the effects of gamification in programming education [17].

### IV. Discussion

#### *A. Potential Benefits*

Using wide incentives, the gamified framework motivates the learners heavily. Through interesting and engaging activities, it reinvigorates them and encourages an open cooperative educational environment. This kind of method provides a learner-centered experience wherein learning occurs according to the pace set by the individual learner himself as accurate instantaneous feedback is provided [2]. As learners are well equipped with the creation of a scalable and universally accessible system, learners are well prepared to face practical programming challenges irrespective of the geographical location or the background they hail from [1]. Apart from this, the peer feedback and collaborative ventures would enhance teamwork, communication skills, and create a sense of community among participants [10]. All these could drastically alter the way students learn and get in touch with the concepts of programming.

#### *B. Addressing Criticisms and Potential Challenges*

The gamified system has to balance extrinsic rewards with intrinsic motivation such that mastery of the subject matter always remains at the focus of the learner. Challenges have to be so crafted that they are not frustrating to learners, particularly new learners, and increase gradually. The design has also to be scalable without suffering a loss in terms of performance or user experience. This will ensure that the game-like activities significantly improve the technical competencies and are not distractions [4]. Additionally, the framework has to adapt to a significant variety of cultural and education settings in order to get it deployed effectively and pertinently within different contexts [14].

#### *C. Future Research Directions*

The key future research lines in gamification of programming education include: Longitudinal studies will determine if gamification leads to a long-lasting retention of programming skills. In addition, it is critical to test the effectiveness of adaptive learning algorithms for the real-time adaptation of a system by learners based on their different progression speeds [6]. Comparing between educational institutions- such as corporate or academic setups-would help compare and validate the outcomes of this framework across different entities. Such research is warranted in terms of possibilities inbuilt into gamification for collaborative programming and teamwork as well as the ability to adapt to different learning preferences and international contexts [18].

### V. Conclusion

This research looks into how gamification could influence the enhancement of programming education. The inclusion of game design would significantly increase motivation, engagement, and overall experience for learners. Currently, however, the gamified systems have weaknesses regarding scalability, adaptability, and conformity to rigorous technical instruction. To overcome such weaknesses, this paper presents a systematic, scalable, and adaptable framework designed exclusively for programming education. This framework should aim at finding a balance between educational completeness and interesting aspects, making learners motivated and challenged. It is through efficient management of aspects of personalization, scalability, and adaptability that the proposed framework has the potential to bring about significant improvements in the effectiveness and responsiveness of gamified programming education systems. Further research is required to confirm the influence of the framework on long-term educational outcomes, balance the best possible extrinsic and intrinsic motivation, and assess the adaptive learning algorithms' effectiveness in real-world scenarios.

### VI. References

1. Rabiyya Huseynzada; Elviz Ismailov. (2024). Game-based learning: a revolution in the teaching of software development. Proceedings of Azerbaijan High Technical Educational Institution, doi: 10.36962/pahtei38032024-190
2. Tri Wahyuningsih; Eko Sediyono; Kristoko Dwi Hartomo; Irwan Sembiring. (2024). The role of gamification implementation in improving quality and intention in software engineering learning. doi: 10.11591/edulearn.v18i1.20823
3. Mahfudzah Othman; Aznoora Osman; Siti Zulaiha Ahmad; Natrah Abdullah. (2024). Integrating segmenting and gamification principles in the design of interactive gamified programming assessments for low achievers. The Turkish Online Journal of Distance Education, doi: 10.17718/tojde.1315427
4. Nicoleta-Gabriela Barcan; Adrian Alexandrescu; Tatiana Turcanu. (2024). Gamification of the Learning Process for Acquiring Logical Thinking in Programming. doi: 10.1109/roedunet64292.2024.10722314
5. Emilio Ormeño; Sergio F Ochoa; Cláudio Alessio; Ferrarini Cintia (2024). Damo's Quest: A Gamified Environment for Learning Programming. doi: 10.1109/clei64178.2024.10700402
6. Joana Martinho Costa. (2023). Using game concepts to improve programming learning: A multi‐level meta‐analysis. Computer Applications in Engineering Education, doi: 10.1002/cae.22630
7. Huilin Cao; Linchun Tan. (2023). Student education assessment model based on programming cognitive diagnosis. doi: 10.1145/3661638.3661694
8. Christian Garcia Villegas; Nilson Lemos Ag*uero* (2023). The Gamification of E-learning Environments for Learning Programming. JOIV : International Journal on Informatics Visualization, doi: 10.30630/joiv.7.2.1602
9. ZhiWei Qu; JiaMing Liu; Lu Che; YanXiu Su; Wei Zhang. (2023). Research on the Application of Gamification Programming Teaching for High School Students’ Computational Thinking Development. doi: 10.1109/ICEIT57125.2023.10107843
10. Fajar Pradana; Punaji Setyosari; Saida Ulfa; Tsukasa Hirashima; Faizatul, Amalia. (2023). The Effect of Gamification-Based Programming E-Learning Platform on Students' Achievement. doi: 10.1145/3626641.3626675
11. Rawan M. Abu-Hammad; Thair Hamtini. (2023). Gamification Approach for Making Online Education as Effective as In-Person Education in Learning Programming Concepts. International Journal of Emerging Technologies in Learning (ijet), doi: 10.3991/ijet.v18i07.37175
12. Bilge Say; Haluk Altunel; Mehmet Kosa; Müjde Koca-Atabey. (2023). Evaluation of an industrial case of gamification in software quality improvement. International Journal of Serious Games, doi: 10.17083/ijsg.v10i3.594
13. Sandro Speth; Leon Hofmeister; Steffen Becker; Uwe Breitenbücher. (2023). Gamify-IT – A Web-Based Gaming Platform for Software Engineering Education. doi: 10.1109/seeng59157.2023.00009
14. Vanitha Thurairasu. (2022). Gamification-Based Learning as The Future of Language Learning: An Overview. European Journal of Humanities and Social Sciences, doi: 10.24018/ejsocial.2022.2.6.353
15. Zehui Zhan; Luyao He; Yao Tong; Xinya Liang; Shihao Guo; Xixin Lan. (2022). The effectiveness of gamification in programming education: Evidence from a meta-analysis. Computers & Education: Artificial Intelligence, doi: 10.1016/j.caeai.2022.100096
16. Danijel Kučak; Martina Kučak. (2022). Gamification in Computer Programming Education – Systematic Literature Review. doi: 10.23919/mipro55190.2022.9803457
17. José Carlos Paiva; José Paulo Leal; Ricardo Queirós. (2020). Fostering Programming Practice through Games. Information-an International Interdisciplinary Journal, doi: 10.3390/INFO11110498
18. Luis Norberto Zapata Puerta; Maria Clara Gomez-Alvarez. (2020). Gamified teaching model to promote learning of algorithms and programming. doi: 10.23919/CISTI49556.2020.9141027
19. Stamatios Papadakis. (2020). Evaluating a game-development approach to teach introductory programming concepts in secondary education. International Journal of Technology Enhanced Learning, doi: 10.1504/IJTEL.2020.10024287
20. Jakub Swacha; Ricardo Queirós; José Carlos Paiva. (2019). Towards a Framework for Gamified Programming Education. doi: 10.1109/ISET.2019.00038