

Playing for Quality: A Competitive Educational Game for Teaching Unit Testing in Computer Science Education

Alia El Bolock¹, Caroline Sabty², Gordon Fraser³ and Slim Abdennadher²

¹The American University in Cairo, Egypt

²The German International University, Cairo, Egypt

³Passau University, Germany

alia.elbolock@aucegypt.edu

Abstract: Software testing is a foundational competence in computer-science curricula and a notoriously disengaging one for learners; a recurring concern for the games-based learning (GBL) community is whether gamified-testing interventions actually deepen learning or merely raise activity levels. This paper presents the design and a proof-of-concept evaluation of *Test Royale*, an educational game for teaching unit testing whose mechanics are deliberately bound to learning objectives at Bloom's apply, analyse and evaluate levels: live mutation analysis (Stryker.NET) and coverage analysis (Coverlet) are surfaced as in-game feedback, and the score function is computed directly from validated quality indicators rather than from activity volume. The design is articulated through the Mechanics–Dynamics–Aesthetics (MDA) framework and grounded in the Octalysis core drives. We report a small mixed-methods pilot study with two complementary participant groups: sixteen undergraduate students in a Software Testing course (8 in a gamified condition, 8 in a non-gamified condition; Study 1) and eleven professional developers at an Egyptian fintech company (Study 2). In Study 1, the gamified condition outperformed the non-gamified condition on every measured quality indicator with large-effect-size differences (mutation score $\Delta = +8.2$, $d \approx 0.81$; branch coverage $\Delta = +10.7$, $d \approx 0.96$; line coverage $\Delta = +7.5$, $d \approx 0.84$; useful test LOC $\Delta = +12.1$, $d \approx 1.23$), with $p < .05$ for each and the pattern surviving Holm–Bonferroni correction across the five outcomes, while execution time did not differ significantly. In Study 2, the system achieved a System Usability Scale score of 80.0 (“excellent”; 95% CI [66.88, 93.12], $n = 11$), with self-reported motivation among already-motivated practitioners remaining high but not significantly increased. Given the small samples, the entanglement of game mechanics with feedback immediacy in our control condition, and the single-session design, we present these results as encouraging proof-of-concept evidence rather than confirmatory findings, and we describe a three-arm replication plan that disentangles game mechanics from feedback design.

Keywords: Games-based learning, Serious games, Unit testing, Mutation testing, Mechanics-Dynamics-Aesthetics framework, Gamification

1. Introduction

Software testing is widely recognised as a foundational competence in computer-science curricula and one of the most disengaging activities for learners and practitioners alike (Fraser, 2017; de Jesus et al., 2020). In educational settings, this manifests as shallow test suites, low coverage, and instrumental engagement with testing concepts; in industry, neglected testing has been implicated in costly software failures across safety-critical domains (Mascheroni and Irrazábal, 2018). A persistent concern in the games-based learning (GBL) community has been whether gamified-testing interventions translate higher engagement into measurable learning outcomes, or whether they merely amplify activity at the expense of quality (Straubinger et al., 2025; Fulcini et al., 2023). As Landers (2014) argues, this is the central empirical question for any gamified-learning intervention: *do the game elements influence the targeted learning behaviour, and through what mediating mechanism?*

This paper presents the design of *Test Royale*, an educational game for teaching unit testing in computer-science education, together with a proof-of-concept (PoC) mixed-methods evaluation. The design is articulated as a serious educational game (Papastergiou, 2009; Backlund and Hendrix, 2013) rather than as a developer tool with bolted-on game elements: its mechanics are derived from explicit, Bloom-tagged learning objectives (Anderson and Krathwohl, 2001) and decomposed using the Mechanics–Dynamics–Aesthetics (MDA) framework (Hunicke et al., 2004), while motivational scaffolding is informed by the Octalysis core drives (Chou, 2015) and Self-Determination Theory (Ryan and Deci, 2000).

A distinctive design choice in *Test Royale* is that the scoring function is computed directly from validated quality indicators of the test code each player writes (mutation score, branch and line coverage, useful test code, execution time). This choice is intended to align learner incentives with the quality construct that previous gamified-testing studies have struggled to influence (Straubinger et al., 2025; Blanco et al., 2023; Arnarsson and Jóhannesson, 2015). We make three contributions to the GBL community: (1) a serious-games design for unit-testing pedagogy, decomposed through MDA and mapped explicitly to Bloom-level learning objectives; (2) proof-of-concept empirical evidence from a small pilot ($n = 8$ per group in Study 1; $n = 11$ in Study 2) that quality-

bound competitive feedback can be associated with large-effect-size differences in objective test-quality indicators in an authentic in-session task, with the pattern surviving Holm–Bonferroni correction; and (3) a transparent account of the methodological challenges in evaluating gamified-testing environments, in particular the entanglement of game mechanics with feedback immediacy, and a concrete three-arm replication design intended to disentangle them.

2. Related Work

2.1 Educational Games in Computing Education

Educational games, also called serious games or games-with-a-purpose, have been studied extensively as instructional vehicles in computing education. Papastergiou (2009) demonstrated that a digital learning game produced both higher learning gains and stronger motivation than equivalent non-gaming material in a high-school CS context. Backlund and Hendrix (2013) synthesised over fifty serious-games studies and concluded that, while the field’s evidence base is heterogeneous, well-designed educational games produce learning gains comparable to or exceeding traditional instruction in many domains. Petri and Gresse von Wangenheim (2017) systematically reviewed evaluations of computing-education games and found that most studies measure perceived learning and engagement but few report objective performance on learning artefacts, motivating the field-standard MEEGA+ instrument (Petri et al., 2018) that combines usability and player-experience subscales.

2.2 Gamification and Serious Games for Software Testing

Within software testing specifically, Fulcini et al. (2023) provide a systematic review of gamified testing tools and identify the field’s tendency to focus on white-box JUnit testing in educational settings, with measurement criteria varying widely. They explicitly call for unified, quality-oriented metrics, a call that motivates our score design. de Jesus et al. (2020) compared a traditional and a gamified software-testing course and found higher engagement in the gamified arm without a clear difference in learning outcomes. Straubinger et al. (2025) reported a recent replicability study of the IntelliGame plugin in which gamified participants wrote more tests and used debugging features more often, yet did not improve in coverage or mutation score and produced more test smells, a pattern interpreted as activity inflation rather than quality improvement. Arnarsson and Jóhannesson (2015) reported similar findings: gamified groups felt more motivated and produced more tests, while path coverage did not improve.

In contrast, some studies report quality-related gains. Costa and Oliveira (2020) described a gamified exploratory-testing framework themed around *Pirates of the Caribbean* and designed using Octalysis (Chou, 2015); learners reported high engagement and identified usability and interface defects. Blanco et al. (2023) evaluated GoRace, a gamified Olympic-style tool integrated with SQLTest, and observed higher task effectiveness and engagement in the gamified group. Across these studies a consistent pattern emerges: gamification reliably raises engagement, but its effect on the *measurable quality* of test artefacts remains contested, and several authors caution that poorly aligned reward structures can incentivise testers to optimise for visible scoring criteria rather than underlying test effectiveness (Straubinger et al., 2025; Inozemtseva and Holmes, 2014).

2.3 Frameworks for Designing and Evaluating Serious Games

Three framings recur in the GBL design literature. The Mechanics–Dynamics–Aesthetics (MDA) framework (Hunicke et al., 2004) is the most-cited formal vocabulary for serious-games design: *mechanics* are the rules and components specified by designers, *dynamics* are the run-time behaviours that emerge from player interaction, and *aesthetics* are the emotional responses targeted in the player. Self-Determination Theory (Ryan and Deci, 2000) and the related Player Experience of Need Satisfaction (PENS) instrument (Ryan et al., 2006) foreground autonomy, competence and relatedness as psychological mediators of motivation. The Octalysis framework (Chou, 2015) translates these into eight “core drives” useful for design. For evaluation, the GBL community has converged on validated player-experience instruments. The most widely used in computing-education contexts is MEEGA+ (Petri et al., 2018); GameFlow (Sweetser and Wyeth, 2005) extends Csíkszentmihályi’s flow construct to games. The present pilot used the System Usability Scale (Brooke, 1996) and items derived from the Intrinsic Motivation Inventory (Ryan, 1982) for instrument-cost reasons, with MEEGA+ and a flow measure identified as targets for the planned replication.

We position Test Royale at the intersection of these threads. Like Costa and Oliveira (2020), we adopt an Octalysis-informed competitive design. Like Straubinger et al. (2025) and Blanco et al. (2023), we measure objective code-quality indicators. Unlike most prior studies, we (i) explicitly bind the in-game score to those

quality indicators rather than to activity volume, (ii) decompose the design through the MDA framework with explicit Bloom-tagged learning objectives, and (iii) evaluate the same artefact with both undergraduate students and professional developers.

3. Research Questions, Hypotheses and Proof-of-concept Framing

3.1 Research Questions and Hypotheses

We address three research questions: **RQ1.** In an authentic in-session unit-testing task, does interaction with a quality-oriented competitive educational game result in higher objective test-quality indicators than a non-gamified workflow that uses the same underlying analysis tools? **RQ2.** When professional developers engage with the same game, how do they perceive their motivation, perceived competence and engagement? **RQ3.** Does the platform achieve usability ratings that meet or exceed the established benchmark for usable systems?

The corresponding pilot hypotheses are: **H1.** Students using *Test Royale* will produce, during an authentic 15-minute in-session task, unit-test suites of higher objective quality than students using the same underlying tools in a non-gamified workflow. **H2.** Professional developers will report *at least sustained* motivation and perceived competence after using *Test Royale*. **H3.** *Test Royale* will achieve a System Usability Scale (Brooke, 1996) score above the established benchmark of 68 (Bangor et al., 2009).

3.2 Proof-of-concept Framing

We frame the investigation explicitly as a pilot proof-of-concept study. The samples are small ($n = 8$ per arm in Study 1; $n = 11$ in Study 2), the intervention is a single 15-minute session, and the tooling is exploratory rather than fully controlled. Our claims are correspondingly hedged: where the data support an effect, we report effect sizes alongside p-values and multiple-comparison-corrected p-values (Section 7); where they do not, we say so plainly.

4. Learning Design

4.1 Learning Objectives

Test Royale targets three learning objectives at Bloom's revised cognitive levels of *apply*, *analyse* and *evaluate* (Anderson and Krathwohl, 2001). Table 1 states each objective, its Bloom level, and the in-game mechanic that operationalises it.

Table 1: Learning objectives, Bloom levels, and the operationalising game mechanics

ID	Bloom level	Learning objective	Operationalising mechanic
LO1	Apply	Apply unit-testing constructs (assertions, fixtures, parameterised tests) to a non-trivial code-under-test in C#/MSTest.	Editor and test runner; in-game starter scaffold; live test-pass/fail panel.
LO2	Analyse	Analyse mutation and coverage feedback to identify gaps in a candidate test suite.	Live mutation panel (killed/survived/timed-out/not-covered); live coverage bars.
LO3	Evaluate	Evaluate trade-offs between thoroughness, clarity and execution efficiency of test suites.	Composite scoring formula (Eq. 1) combining mutation, coverage, useful LOC and an execution-time penalty.

4.2 Mechanics, Dynamics and Aesthetics Decomposition

Following Hunnicke et al. (2004), we distinguish what the designer specifies (mechanics), what emerges at run-time (dynamics) and what the player is intended to feel (aesthetics). Table 2 maps Test Royale onto MDA with cross-references to the Octalysis core drives that motivated each element (Chou, 2015).

Table 2: MDA decomposition of Test Royale, with Octalysis core drives

Mechanic	Intended dynamic	Aesthetic / Octalysis drive
Shared room and synchronous countdown	Co-presence and shared deadline focus attention	Fellowship; Social influence
Live mutation and coverage panels	Iterative refinement of tests as players read feedback	Discovery; Empowerment of feedback

Mechanic	Intended dynamic	Aesthetic / Octalysis drive
Quality-bound score formula	Players optimise for the quality construct the educator targets	Challenge; Development and accomplishment
Tiered and threshold-based badges	Sub-goals at varying skill levels (Bronze/Silver/Gold/Platinum)	Accomplishment
Session and global leaderboard	Peer comparison and aspirational ranking	Competition; Social influence
Persistent profile and history	Self-comparison across sessions	Mastery; Ownership
Time-bounded session	Pacing pressure and disciplined termination	Submission to game state; Scarcity

4.3 Score Function as a Ranking Heuristic

The composite score (Eq. 1) is presented as a ranking heuristic rather than a measurement instrument. Its purpose is to provide a single visible quality signal during play, held constant across all gamified participants, that ranks players consistently within a session. We deliberately do *not* claim that the absolute score is a unit-bearing measurement of test-suite quality.

$$\text{Score} = 0.4 \cdot \text{MutationScore} + 0.2 \cdot \text{BranchCoverage} + 0.1 \cdot \text{LineCoverage} + 0.2 \cdot \text{UsefulLOC} - 0.1 \cdot \text{ExecutionTime} \quad (1)$$

The weights were set heuristically: (a) mutation score is the most discriminating indicator of fault-detection capability (Inozemtseva and Holmes, 2014), hence the largest single weight; (b) branch coverage captures path-traversal richness and is weighted twice as strongly as raw line coverage; (c) useful (non-trivial) test code is rewarded to discourage padding while a small execution-time penalty discourages runaway suites. Because the LOC and time terms are not commensurable in unit with the percentage terms, the formula is best understood as a designer-tuned ranking utility.

5. Test Royale: Game Design and Implementation

5.1 Game Flow

A session begins when one learner creates a room and shares its code with peers. Once a host launches the match, every player receives the same C# code-under-test together with a faulty unit-test scaffold and a countdown timer (15 minutes in the present studies). Each player writes and refines test cases individually using the MSTest framework. While playing, the system continuously runs Coverlet (for coverage) and Stryker.NET (for mutation) against each player's evolving suite and updates a personal dashboard with mutation score, branch and line coverage, useful LOC, and execution time, plus a per-mutant breakdown (killed, survived, timed-out, not-covered); see Figure 1(a). When the timer ends, the system computes the composite score (Eq. 1) and ranks players on a session leaderboard and a global leaderboard, Figure 1(b).

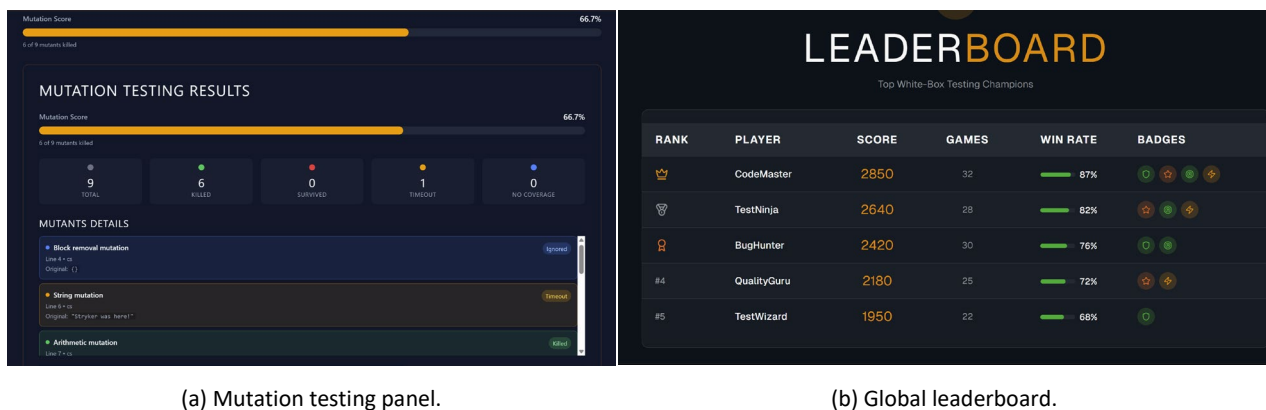


Figure 1: In-game feedback panels supporting LO2 (left): per-mutant outcomes (killed, survived, timed-out, not-covered); and exposing the social-influence and accomplishment Octalysis drives (right): global leaderboard across sessions

5.2 Gamification Elements and Implementation

Test Royale awards tiered badges including *Mutation Slayer* (kill 80% of mutants), a tiered *Coverage Explorer* (Bronze 70%/Silver 80%/Gold 90%/Platinum 100%), and a progressive *Top Rank* series from Rookie to Legend for sustained leaderboard performance. Personal profiles record each player's recent matches, average score, win rate and earned badges.

Test Royale is implemented as a web-based multiplayer system. The backend is a Node-based service that orchestrates rooms, handles synchronous events, runs Stryker.NET and Coverlet against submitted test code in isolated containers, and persists session and profile data in MongoDB. The frontend is a responsive web client. Figure 2 shows the three-tier architecture.

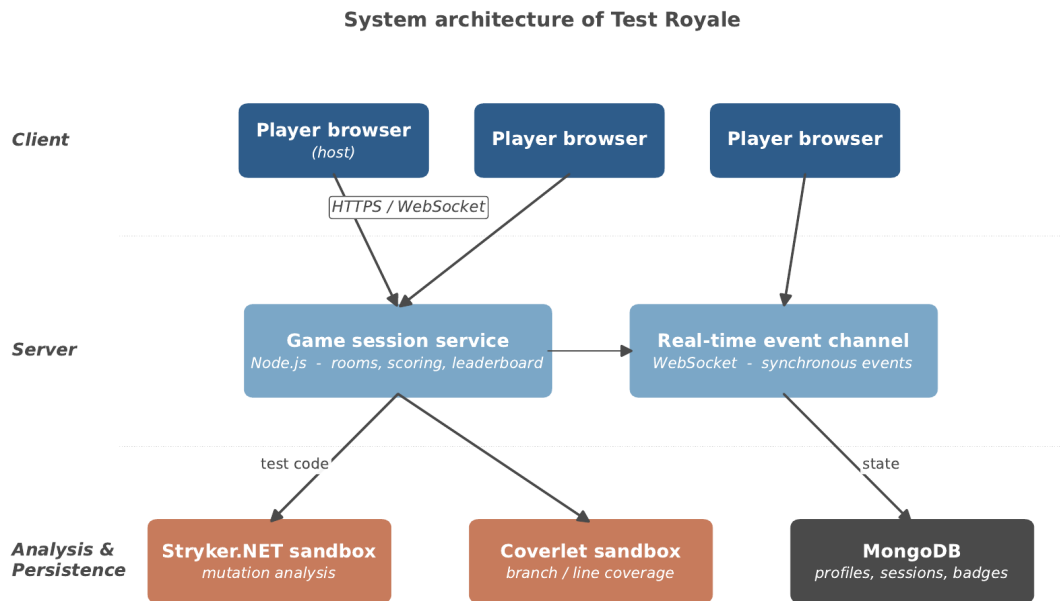


Figure 2: Three-tier system architecture of Test Royale. Player browsers communicate with a Node-based session service over HTTPS and a real-time WebSocket channel; submitted test code is executed in isolated Stryker.NET and Coverlet sandboxes; profiles, sessions and badges persist in MongoDB

6. Methodology

The evaluation followed a mixed-methods, exploratory empirical design (Kitchenham et al., 2002), with two complementary studies: a between-subjects classroom study with undergraduate students (Study 1, RQ1/H1) and an on-site industry study with professional developers (Study 2, RQ2/H2 and RQ3/H3). Both studies were small-sample pilots intended to surface effect-size estimates and design issues for a fully-powered confirmatory replication.

6.1 Study 1: Classroom Evaluation of Test Quality

Sixteen undergraduate students enrolled in a Software Testing course at the German International University (GIU) volunteered to participate. All had prior coursework in software testing but limited practical experience. Participants were randomly assigned with equal allocation to a gamified condition ($n = 8$) using *Test Royale* or a non-gamified condition ($n = 8$) using a Git-based workflow. Participation was voluntary and did not affect course grades. Both groups received the same C# code base (an *Advanced String Calculator* class with non-trivial parsing logic), the same scaffold of intentionally faulty unit tests, and the same briefing: improve the test suite to maximise its quality within a fifteen-minute window. Both groups were told the same metrics would be used to evaluate their work.

Each session lasted approximately twenty minutes: a short briefing, a fifteen-minute working window, and a brief debrief. The gamified group used *Test Royale* and saw real-time feedback on the same five metrics, together with a leaderboard component within the platform. The exact granularity of social comparison during

the live session (whether each player could see other players' live scores updating, or only the post-session ranking) varied across pilot deployments and is documented imprecisely in our session logs; we therefore describe the social-comparison mechanism conservatively as post-session ranking with the possibility of live peer-score visibility, and we identify this ambiguity in Section 9. The non-gamified group cloned a Git repository containing the same code-under-test and the same underlying analysis tools (Stryker.NET, Coverlet) and obtained equivalent feedback by running the tools manually from the command line. They received no in-platform UI, no badges, no scoring formula, and no leaderboard. The same researcher supervised all sessions.

Five objective metrics were captured per participant: mutation score, branch coverage, line coverage, useful lines of test code (useful LOC), and execution time. For the gamified group, these were logged automatically; for the non-gamified group, they were extracted post-hoc by running the same tools on each participant's submitted code through a Python script. Group means were compared using independent-samples Welch *t*-tests with $\alpha = .05$. We report *t* and *p* statistics together with Cohen's *d* effect sizes (Table 4). To address the simultaneous testing of five conceptually related outcomes, we additionally report the Bonferroni-corrected threshold ($\alpha/k = 0.01$) and the Holm–Bonferroni step-down procedure (Holm, 1979), the latter being the more powerful and now widely recommended default (Aickin and Gensler, 1996).

6.2 Study 2: Industry Evaluation of Motivation and Usability

Study 2 was conducted on-site at PaySky, an Egyptian fintech company specialising in digital-payments software. Nineteen professional developers participated in the pre-test phase. Eleven completed both pre- and post-test instruments and contributed gameplay data and usability ratings; eight of the eleven contributed complete matched pre/post motivation data on the items used for the paired analysis. All participants were proficient in C# and unit testing. Each session lasted approximately thirty minutes: a senior developer's primer on testing concepts, a walkthrough of the platform, a pre-test survey, a fifteen-minute *Test Royale* session, and a post-test survey including the System Usability Scale (SUS) (Brooke, 1996) and items derived from the Intrinsic Motivation Inventory (IMI) (Ryan, 1982). All Likert items were on a five-point scale. SUS scores were computed using Brooke's standard procedure. Pre/post-test motivation scores were compared using paired-samples *t*-tests on the eight participants with complete pairs; SUS scores were compared against the benchmark of 68 (Bangor et al., 2009), with a 95% confidence interval reported alongside.

7. Results

7.1 Study 1: Test Quality

Descriptive statistics for the five quality metrics, by condition, are reported in Table 3. The gamified group outperformed the non-gamified group on every quality-oriented indicator and showed marginally faster execution times.

Table 3: Study 1: Descriptive statistics for unit-test quality metrics, by condition (M, SD)

Group	Mutation score	Branch cov. (%)	Line cov. (%)	Useful LOC	Exec. time (s)	Total score
Gamified (n = 8)	64.3 (10.5)	83.5 (12.1)	91.7 (8.3)	73.4 (9.4)	2.1 (0.7)	68.9 (8.2)
Non-gamified (n = 8)	56.1 (9.8)	72.8 (10.3)	84.2 (9.5)	61.3 (10.2)	2.6 (0.9)	59.7 (9.1)

Independent-samples Welch *t*-tests (Table 4) showed statistically significant uncorrected between-group differences in favour of the gamified group on mutation score, branch coverage, line coverage, useful LOC and total score, with large effect sizes ($d \geq 0.81$ in every case). Execution time did not differ significantly. Under the strict Bonferroni correction ($\alpha/k = 0.01$), only the comparison on *total score* ($p = 0.008$) clearly retains significance; *branch coverage* ($p = 0.010$) is at the threshold. Under the more powerful step-down Holm–Bonferroni procedure (Holm, 1979), which controls family-wise error at $\alpha = 0.05$ while retaining greater power, all five comparisons survive (Table 4). We privilege the Holm–Bonferroni result as the primary inference, while noting that the more conservative Bonferroni outcome substantially narrows the set of confidently significant comparisons.

Table 4: Study 1: Between-group comparisons (independent-samples Welch t-tests). The rightmost column reports both multiple-comparison adjustments (Bonferroni and Holm–Bonferroni) for the family of five quality-related outcomes

Metric	Δ	t	p	Cohen's d	Bonf / Holm
Mutation score	+8.2	2.36	0.024*	≈ 0.81	n.s. / sig.
Branch coverage	+10.7	2.91	0.010*	≈ 0.96	~ / sig.
Line coverage	+7.5	2.12	0.037*	≈ 0.84	n.s. / sig.
Useful LOC	+12.1	2.64	0.016*	≈ 1.23	n.s. / sig.
Execution time	-0.5	-1.45	0.162	≈ 0.62	n.s. / n.s.
Total score	+9.2	3.05	0.008**	≈ 1.07	sig. / sig.

* $p < .05$ (uncorrected); ** $p < .01$ (uncorrected). Bonf = Bonferroni threshold $\alpha/k = 0.01$; Holm = Holm–Bonferroni step-down at family-wise $\alpha = 0.05$ (Holm, 1979).

The pattern of effects therefore supports H1 in this exploratory sample under the Holm–Bonferroni-corrected analysis: students working with *Test Royale* produced unit-test suites of higher objective quality than students using the same underlying analysis tools manually, while completing the task in comparable time. Figure 3 visualises the comparison.

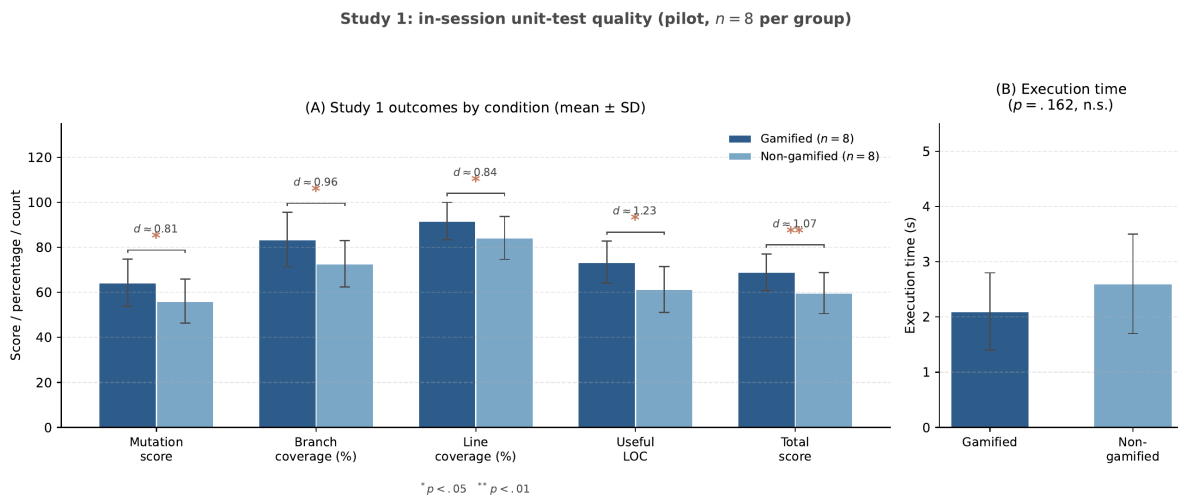


Figure 3: Study 1 outcomes by condition (mean $\pm$ SD, $n = 8$ per group). Panel (A): the gamified group showed large-effect-size advantages on every quality indicator. Panel (B): execution time did not differ significantly

7.2 Study 2: Motivation and Usability

All four motivation/perception construct means exceeded 3.9 on the five-point scale at both time points (pre and post). A paired-samples t -test on aggregated motivation scores ($n = 8$ matched pairs) indicated no statistically significant change between pre-test ($M = 4.41$, $SD = 0.42$) and post-test ($M = 4.19$, $SD = 0.70$): $t(7) = 0.91$, $p = .394$. H2 is therefore supported only in its weak form (at-least-sustained motivation): motivation did not significantly increase, but both means remained above 4.0 and engagement showed a positive shift. Three plausible non-exclusive interpretations are discussed in Section 8.

The mean SUS score across the eleven developers was 80.0 ($SD = 19.53$), within the “excellent” range (Bangor et al., 2009). Given the large standard deviation, we report a 95% confidence interval: $SE = 19.53/\sqrt{11} = 5.89$; $t(0.975, 10) = 2.228$; 95% CI = [66.88, 93.12]. The lower bound is approximately 1 point below the 68-point benchmark, meaning the difference between the sample mean and the benchmark is significant at $\alpha = 0.05$ but only narrowly so. Under a (questionable) normality assumption with $M = 80.0$ and $SD = 19.53$, an estimated 27% of the population from which this sample is drawn would score below the benchmark; in an $n = 11$ sample, this implies an expectation that approximately 3 of the 11 participants scored below the benchmark. Item-level

statistics show consistent agreement on positively worded items relating to ease of use, learnability and confidence (means 4.00–4.45), and consistent disagreement on negatively worded items relating to complexity, inconsistency and cumbersomeness (means 2.00–2.73). H3 is supported at $\alpha = 0.05$, but the narrow gap between the lower bound of the 95% CI and the benchmark, and the likely presence of sub-benchmark scorers, mean we present this support as encouraging rather than definitive.

8. Discussion

The Study 1 results suggest that pairing competitive game mechanics with quality-bound real-time feedback is associated with higher in-session test quality compared with a workflow using the same analysis tools without such feedback. The convergence of large-effect-size differences across mutation, coverage, and useful-LOC indicators, with execution time held essentially constant, is an encouraging proof-of-concept signal; the pattern survives Holm–Bonferroni adjustment, although under conservative Bonferroni only total score (and, marginally, branch coverage) retain significance. With $n = 8$ per group, a single 15-minute session, and the caveats in Section 9, the data do not support a confirmatory causal claim.

The Study 2 motivation null result admits at least three non-exclusive interpretations: a ceiling effect, since developers entered with motivation already at 4.0–4.4 on a 5-point scale (Hamari et al., 2014; Mekler et al., 2017); insufficient exposure, as a 15-minute session may be too brief to shift attitudes among experienced practitioners; and population mismatch (overjustification), in which external rewards can erode intrinsic motivation among already-motivated experts (Deci, 1971; Mekler et al., 2017). A within-subjects longitudinal design could separate these explanations.

Three implications follow for GBL research and practice. *Bind the score to the construct*: if a gamified-learning intervention rewards activity volume, it tends to produce more activity; if it rewards quality, it may produce more quality, consistent with Landers (2014)’s theory of gamified learning. *Distinguish engagement from quality reporting*: the GBL community would benefit from routinely reporting both. *Match mechanic to learner*: the contrast between students (who improved on quality indicators) and developers (whose motivation did not significantly increase) suggests that the same gamified design may not serve both populations equally. Finally, quality-bound competitive scoring raises a concern that the score becomes a target for gaming: empty assertions raise coverage but not mutation score (the 0.4 weight on mutation makes this non-dominant); hard-coded outputs tend to fail surviving mutants; suite padding is partially discouraged by the useful-LOC term and execution-time penalty. A fuller anti-cheating analysis is part of the planned extension.

9. Threats to Validity

We organise threats following Wohlin et al. (2012). *Internal validity*: the most serious threat in Study 1 is a confound between game mechanics and feedback design. The gamified and non-gamified arms differed simultaneously on feedback timing, feedback visualisation, and game elements; the results tell us the integrated gamified environment produced higher quality than the integrated non-gamified workflow but do not isolate the contribution of gamification per se. A three-arm replication (Section 10) is proposed to address this. The exact granularity of social comparison was also documented imprecisely in our pilot logs.

External validity: both studies are pilots with small samples ($n = 8$ per arm in Study 1; $n = 11$ in Study 2; $n = 8$ matched pairs for H2), a single code-under-test, a single language (C#), a single tool stack, and a single fifteen-minute session. We make no claim of generalisation to other populations, codebases or durations.

Construct validity: the composite score (Eq. 1) is presented as a designer-tuned ranking heuristic rather than a unit-bearing measurement. The motivation instrument was abbreviated from the IMI; MEEGA+ (Petri et al., 2018) and a flow construct (Sweetser and Wyeth, 2005) were not measured.

Conclusion validity: t -tests with $n = 8$ have limited power and produce effect-size estimates sensitive to outliers (Gelman and Carlin, 2014); Cohen’s d values should be read as wide-confidence-interval estimates. Two further issues: (i) under strict Bonferroni only total score (and, marginally, branch coverage) survive, while under Holm–Bonferroni all five survive at family-wise $\alpha = 0.05$ (Aickin and Gensler, 1996); (ii) the SUS 95% CI lower bound is approximately 1 point below the benchmark, and under normality approximately 3 of the 11 participants are expected to have scored sub-benchmark.

10. Conclusion and Future Work

This paper presented *Test Royale*, an educational game for teaching unit testing whose mechanics are tied directly to validated quality indicators of the test artefacts learners produce. We articulated the design through

the MDA framework (Hunicke et al., 2004) and explicit Bloom-tagged learning objectives (Anderson and Krathwohl, 2001), and reported a small mixed-methods pilot with sixteen undergraduate students and eleven professional developers. The classroom pilot produced large-effect-size differences in objective in-session quality indicators compared with a non-gamified control, with the pattern surviving Holm–Bonferroni correction across the five outcomes; the industry pilot achieved excellent usability (with a wide 95% confidence interval whose lower bound is at the benchmark) and sustained-but-not-significantly-increased motivation. These are exploratory proof-of-concept findings, and a fully-powered replication is the next step.

The planned extension comprises a three-arm between-subjects replication separating (A) the full gamified system, (B) a non-gamified control with equivalent real-time feedback, and (C) a non-gamified control with manual CLI feedback, to isolate the contribution of game mechanics from that of feedback immediacy; a longitudinal multi-session design with delayed post-tests on novel code-under-tests to assess retention and transfer; a larger and more diverse sample across multiple institutions and industry contexts; adoption of validated GBL evaluation instruments (MEEGA+ and a flow measure) alongside the SUS and IMI items; a controlled study of cooperative versus competitive variants (Sanchez and Mandran, 2017; Hanus and Fox, 2015); and a pre-registered analysis plan including both parametric (Welch’s *t*) and non-parametric (Mann–Whitney *U*) tests with pre-specified multiple-comparison correction. We will release the platform as open-source software to support replication.

Acknowledgements

We thank the students of the Software Testing course at the German International University and the engineers of PaySky for their voluntary participation, and we are grateful to colleagues and reviewers who provided feedback on earlier versions of this work.

Ethics Declaration: Participation in both studies was voluntary and based on informed consent. Participants were informed of the study’s purpose, duration, the data to be collected, and their right to withdraw at any stage without penalty. No personally identifying information beyond names linking pre- and post-test responses was collected, and gameplay logs were stored securely for analysis only. No financial incentive was offered, and students were assured that participation would not affect course grades. The studies followed standard ethical guidelines for exploratory empirical research in computing education. A formal institutional review board (IRB) clearance was not obtained for these pilot evaluations, given their low-risk, classroom-and-industry character; we acknowledge that a confirmatory replication would warrant a formal IRB review at the host institutions.

AI Declaration: The authors used an AI assistant (Anthropic’s Claude) during the preparation of this paper for the following purposes: drafting, editing and proofreading of prose; reformatting and copy-editing the manuscript across template versions; checking statistical computations (e.g., recomputation of Bonferroni and Holm–Bonferroni-adjusted thresholds, derivation of the SUS 95% confidence interval) against the descriptive statistics reported in the paper; and assistance with figure layout. All study design, data collection, primary analysis, interpretation of findings, and final editorial decisions were made by the authors, who take full responsibility for the content of the paper. No AI tool was used to generate or fabricate any data, results, or citations.

References

- Aickin, M. and Gensler, H. (1996). Adjusting for multiple testing when reporting research results: The Bonferroni vs Holm methods. *American Journal of Public Health*, 86(5):726–728.
- Anderson, L. W. and Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives*. Longman, New York.
- Arnarsson, D. and Jóhannesson, Í. H. (2015). Improving unit testing practices with the use of gamification. Master’s thesis, Chalmers University of Technology.
- Backlund, P. and Hendrix, M. (2013). Educational games – are they worth the effort? a literature survey of the effectiveness of serious games. *Proceedings of the 5th International Conference on Games and Virtual Worlds for Serious Applications*.
- Bangor, A., Kortum, P., and Miller, J. (2009). Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3):114–123.
- 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.
- Brooke, J. (1996). Sus: A “quick and dirty” usability scale. In Jordan, P. W., Thomas, B., Weerdmeester, B. A., and McClelland, A. L., editors, *Usability Evaluation in Industry*, pages 189–194. Taylor and Francis, London.
- Chou, Y.-k. (2015). *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. Octalysis Media.

- Costa, I. E. F. and Oliveira, S. R. B. (2020). The use of gamification to support the teaching-learning of software exploratory testing: An experience report based on the application of a framework. In 2020 IEEE Frontiers in Education Conference (FIE), pages 1–9. IEEE.
- de Jesus, G. M., Ferrari, F. C., Paschoal, L. N., de Souza, S. d. R. S., de Paula Porto, D., and Durelli, V. H. S. (2020). Is it worth using gamification on software testing education? an extended experience report in the context of undergraduate students. *Journal of Software Engineering Research and Development*, 8:6:1–6:19.
- Deci, E. L. (1971). Effects of externally mediated rewards on intrinsic motivation. *Journal of Personality and Social Psychology*, 18(1):105–115.
- Fraser, G. (2017). Gamification of software testing. In 2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST), pages 2–7. IEEE.
- Fulcini, T., Coppola, R., Ardito, L., and Torchiano, M. (2023). A review on tools, mechanics, benefits, and challenges of gamified software testing. *ACM Computing Surveys*, 55(14s):1–37.
- Gelman, A. and Carlin, J. (2014). Beyond power calculations: Assessing type s (sign) and type m (magnitude) errors. *Perspectives on Psychological Science*, 9(6):641–651.
- Hamari, J., Koivisto, J., and Sarsa, H. (2014). Does gamification work? – a literature review of empirical studies on gamification. In 47th Hawaii International Conference on System Sciences, pages 3025–3034. IEEE.
- Hanus, M. D. and Fox, J. (2015). 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.
- Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.
- Hunicke, R., LeBlanc, M., and Zubek, R. (2004). Mda: A formal approach to game design and game research. In *Proceedings of the AAAI-04 Workshop on Challenges in Game AI*. AAAI Press.
- Inozemtseva, L. and Holmes, R. (2014). Coverage is not strongly correlated with test suite effectiveness. In *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*, pages 435–445. ACM.
- Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., and Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28(8):721–734.
- Landers, R. N. (2014). Developing a theory of gamified learning: Linking serious games and gamification of learning. *Simulation & Gaming*, 45(6):752–768.
- Mascheroni, M. A. and Irrazábal, E. (2018). Continuous testing and solutions for testing problems in continuous delivery: A systematic literature review. *Computación y Sistemas*, 22(3):1009–1038.
- Mekler, E. D., Brühlmann, F., Tuch, A. N., and Opwis, K. (2017). Towards understanding the effects of individual gamification elements on intrinsic motivation and performance. *Computers in Human Behavior*, 71:525–534.
- Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education*, 52(1):1–12.
- Petri, G. and Gresse von Wangenheim, C. (2017). How games for computing education are evaluated? a systematic literature review. *Computers & Education*, 107:68–90.
- Petri, G., Gresse von Wangenheim, C., and Borgatto, A. F. (2018). Meega+, systematic model to evaluate educational games. In Lee, N., editor, *Encyclopedia of Computer Graphics and Games*. Springer.
- Ryan, R. M. (1982). Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory. *Journal of Personality and Social Psychology*, 43(3):450–461.
- Ryan, R. M. and Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1):68–78.
- Ryan, R. M., Rigby, C. S., and Przybylski, A. (2006). The motivational pull of video games: A self-determination theory approach. *Motivation and Emotion*, 30(4):344–360.
- Sanchez, E. and Mandran, N. (2017). Exploring competition and collaboration behaviors in game-based learning with playing analytics. In *Lecture Notes in Computer Science*, volume 10474, pages 467–472. Springer.
- Straubinger, P., Fulcini, T., Garaccione, G., Ardito, L., and Fraser, G. (2025). Gamifying testing in intellij: A replicability study. *Proceedings of the ACM on Software Engineering*, 2(ISSTA):2407–2429.
- Sweetser, P. and Wyeth, P. (2005). Gameflow: A model for evaluating player enjoyment in games. *Computers in Entertainment*, 3(3):1–24.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer.