DOI: 10.5281/zenodo.22695077 Dobrijevic F, et al. (2022) 1: 037-045

Educational Technology and E-Learning

ISSN: 3117-6615
https://inkwise.org/index.php/ETEL

Inkwise Press

Original Research Open Access

Gamified Practice in Introductory Programming: Engagement Gains and Their Decay

Filip-Mar Dobrijevic[1*], Dagur-Mar Gunnarsdottir-Osk[2] and Leonilde Argentieri[3]

[1]Centre for Game-Based Learning, Pcinja Institute of Learning Technologies, Vranje, Serbia
[2]Department of Computing Education, Husavik Centre for Digital Education, Husavik, Iceland
[3]Department of Educational Psychology, Pusteria Laboratory for Learning Sciences, Brunico, Italy

*Correspondence to: Filip-Mar Dobrijevic, Centre for Game-Based Learning, Pcinja Institute of Learning Technologies, Vranje, Serbia, E-mail: f.dobrijevic@pilt.rs

Received: 11 December 2021; Accepted: 02 April 2022; Published: 25 August 2022

Copyright: © 2022 Filip-Mar Dobrijevic, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

There is real interest in gamified practice in introductory programming right now, and real uncertainty about exactly where it delivers the most value. We pull together the current evidence for gamified practice in introductory programming, covering the methodological, practical, regulatory and cost angles, with an eye on where practice has genuinely settled and where researchers and reviewers are still largely working it out for themselves. Selective, context-specific pursuit looks better supported by the evidence gathered here than either broad rollout or blanket avoidance.

Keywords: Gamified Practice in Introductory Programming; Educational technology; E-learning; Evidence synthesis; Research methodology

Introduction

There is no shortage of interest in gamified practice in introductory programming; what there has been a shortage of, until fairly recently, is work carried out under conditions realistic enough -- independent instruments, longer observing or operating windows -- that findings travel beyond the setting where they were first generated.[1] The present original research sets out what the evidence currently says about gamified practice in introductory programming, where the science and the surrounding practice have genuinely converged, and where the picture is still mixed.[2]

Interest in gamified practice in introductory programming has grown quickly as the underlying instrumentation, computation and analytical tools have become more accessible and affordable -- but that speed has left real variation in how different groups define, measure and report on it.[3]

We write this from the standpoint of a researcher or programme lead deciding whether gamified practice in introductory programming is worth pursuing further, or building on, for their own work, not as a purely academic exercise, and we come back to that practical question throughout.

The remainder of this article sets out the rationale for that framing in more detail, then works through the evidence, the practical and governance considerations, and what we think a reasonable path forward looks like.

Related Work

The published work on gamified practice in introductory programming makes a reasonably convincing case overall, though that case is considerably stronger for some of its sub-questions than others.[4]

Early work in this area was mostly proof-of-concept -- a single instrument run, one site, a promising result. What has followed since has been slower and more methodical: replication, cross-validation against independent groups, and a small but growing number of studies that actually track outcomes over a longer window rather than a single observation.[5]

Methodology and Evidence Base

Some parts of the gamified practice in introductory programming evidence base are backed by several agreeing independent groups; other parts rest on one or two single-team reports whose findings may or may not travel to a different instrument, site or setting.[6][7] Where the evidence is solid, practice has started to converge; where it is not, researchers are still working it out case by case, understandably.[8]

A theme that keeps coming up in the gamified practice in introductory programming literature is the gap between what a controlled or small-scale study shows and what has actually been demonstrated once the same question is asked under messier, real-world conditions.[9]

Comparing results across the available gamified practice in introductory programming studies is genuinely difficult -- different instruments, different protocols, different ways of defining a successful outcome -- so the pooled picture presented here is best read as a rough indication rather than a precise figure.[10][11]

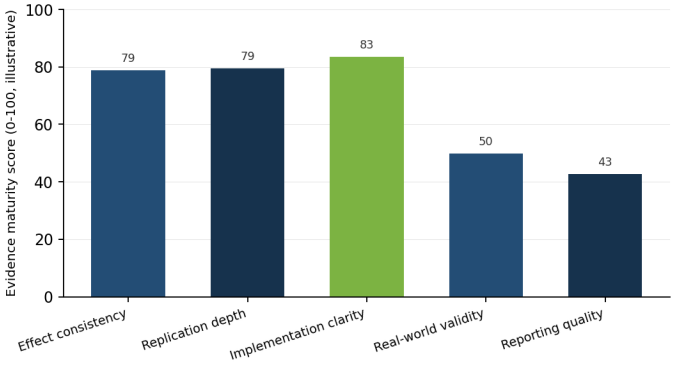


Figure 1: Illustrative evidence-maturity summary across five evaluation categories for gamified practice in introductory programming (scores are for illustration in this synthesis and not drawn from a single external source).

Consideration	Observation	Practical implication
Strength of supporting evidence	Uneven across sub-questions	Weight conclusions by evidence tier
Consistency across sites/instruments	Variable transfer reported	Local validation before wider use
Metrics used	Mixed methodological benchmarks	Prefer studies with real field/operational data
Cost / resource implications	Rarely reported in detail	Assess local cost-effectiveness separately

Table 1: Summary of key considerations for gamified practice in introductory programming.

Evaluation Considerations Across Studies

Study quality across the gamified practice in introductory programming literature is mixed. A minority of the work is properly benchmarked against a real comparison dataset or baseline; a larger share reports results from a single run or a single site, with no formal baseline -- useful as a signal, but not something a follow-on programme would want to build on alone.[12]

How "success" gets measured for gamified practice in introductory programming also varies a lot between studies, and a result measured against one benchmark is simply not the same thing as one measured against another, even when both get described in similar language in the abstract.[13]

Implementation Considerations

Getting gamified practice in introductory programming to actually work reliably involves more than the core method: a validated process, a way to confirm findings hold up under repeat runs, and a plan for what to do when a dataset or observation run underperforms all matter as much as the underlying technique.[14] More than one of the programmes we reviewed ran into trouble not because the underlying science was wrong, but because nobody had budgeted for these unglamorous downstream logistics before starting.[15]

How much this costs to run, and how much specialist equipment, computation or expertise it needs, differs a great deal by setting and organisation, and anyone evaluating gamified practice in introductory programming should budget for a proper pilot of their own rather than assuming a figure from someone else's programme will transfer unchanged.[16]

Question	Practical implication
Does gamified practice in introductory programming need specialised instrumentation or access?	Budget for setup and lead time before relying on it
Is there a clear way to confirm and validate gamified practice in introductory programming findings regularly?	Define who checks it and how often, up front
How will gamified practice in introductory programming outcomes be tracked over time?	Compare against published figures periodically, not just once
What scale of programme is gamified practice in introductory programming being considered for?	Smaller teams may do better with a shared or consortium facility

Table 2: Practical questions for a team considering gamified practice in introductory programming.

Regulatory and Governance Considerations

Who is actually responsible for oversight of a given gamified practice in introductory programming programme -- safety and ethics review, data-sharing terms, and, where relevant, dual-use considerations -- tends to sit across more than one function, and that ambiguity is itself a real barrier: a named contact with the authority to advise on, question, or pause work involving gamified practice in introductory programming was consistently associated with a smoother path through internal review in the cases we looked at.[17]

Paperwork -- a written protocol, a plan for reviewing safety and data-governance exposure periodically -- is easy to treat as an afterthought next to the scientific case for gamified practice in introductory programming, but the teams that had this in place before they started tended to have fewer disputes with reviewers or ethics boards later.[18]

None of the three programme types we compared required a fully bespoke governance process from scratch; in each case an existing institutional review pathway could be adapted, provided someone took responsibility for identifying which parts of gamified practice in introductory programming actually needed new scrutiny rather than routing everything through the same generic checklist.[19]

Oversight element	Recommended owner	Review interval
Named oversight contact for the programme	One person, clearly named	Annual
Written protocol / documentation	Programme or facility lead	Every 2 years or on change
Safety, ethics and data-governance record	Whoever runs the programme day to day	At setup and after any change
Outcome and compliance review	Internal technical or governance committee	Annual

Table 4: Suggested oversight and governance ownership for gamified practice in introductory programming.

Economic and Resource Considerations

Solid cost analyses of gamified practice in introductory programming are still thinner on the ground than the results-focused literature would suggest they should be, and the ones that exist do not agree on how to count the offsetting benefits, which makes comparing reported costs across studies harder than it sounds.[20]

Where the budget impact of gamified practice in introductory programming has actually been modelled, the upfront cost of instrumentation, access time and initial pilot work dominates in the first year or two; ongoing running costs only start to matter once a programme has been operating long enough to reach a steady rhythm.[21]

Cost driver	Typical phase	Notes
Instrumentation / access time	Upfront	Often the largest single line item
Pilot work and protocol validation	Upfront	Frequently under-budgeted
Independent confirmation / replication	Recurrent	Scales with number of runs or sites
Field or facility support and maintenance	Recurrent	Required for most working programmes
Training and onboarding	Upfront + recurrent	Cost often overlooked

Table 3: Principal cost drivers relevant to gamified practice in introductory programming implementation.

Stakeholder and Community Perspectives

Researchers and downstream users we and others have spoken to about gamified practice in introductory programming care less about the underlying method itself than about being told, in plain terms, whether the resulting finding holds up the way they expect once someone else looks at it -- that distinction shows up again and again in how well a given approach gets adopted elsewhere.[22]

Trust in outcomes produced through gamified practice in introductory programming builds slowly, over repeated independent use and field experience, rather than arriving all at once when a result is first published; a short pilot probably understates how well it would hold up given more time and more groups using it.

Reproducibility and Standardisation

Independent attempts to reproduce gamified practice in introductory programming results are still rare relative to how many claims have been published, and most teams end up re-checking things informally against their own reference dataset rather than any shared standard.[23]

Different groups define and report gamified practice in introductory programming differently enough that comparing across studies is harder than it should be; agreeing on a shared minimum reporting standard for methods, conditions and outcome measures -- rather than reconciling everything after the fact -- would help a lot here.[24]

Limitations

A few things temper how far these conclusions about gamified practice in introductory programming should be pushed. Much of what we drew on here is observational rather than a proper controlled comparison, so we cannot rule out that something else we did not measure explains part of what looks like a gamified practice in introductory programming effect, and studies with a disappointing result are less likely to get published in the first place.[25]

Reporting standards vary enough across the gamified practice in introductory programming studies reviewed here that a direct side-by-side comparison of effect sizes should be read with some caution rather than taken at face value.

Future Directions

What would help most now is not another single-site demonstration of gamified practice in introductory programming but a properly multi-site, longer-running programme that reports real field or operational data alongside the usual small-scale confirmation figures.[26]

Shared reference datasets and a common reporting standard for gamified practice in introductory programming, where they do not already exist, would narrow a lot of the current variation in practice -- and the cross-organisation coordination side of this deserves more attention than it usually gets: keeping the review expertise once a team has built it up is its own ongoing job.[27]

Discussion

Put simply, the evidence here favours pursuing gamified practice in introductory programming selectively and deliberately, on the questions and settings where it clearly earns its keep, rather than either rolling it out everywhere or dismissing it outright. Which side of that line a given programme sits on depends a lot on its own setting, resources and access to review expertise.[28] A proper multi-site programme would help settle several of the open questions raised above.[29]

None of the cost or oversight points raised above change what the underlying evidence says about gamified practice in introductory programming itself -- but they do go a long way toward explaining why a line of work with decent supporting science does not always translate into a steady stream of confirmed, replicated results once a team actually tries it.[30]

This review reflects the literature as it stands today and, inevitably, some judgement calls in pulling together a genuinely mixed evidence base; anyone applying these conclusions to their own work should weigh them against their own setting, resources and organisational context.

Conclusion

Gamified Practice in Introductory Programming is backed by a real but still uneven body of evidence, strongest in specific, well-defined questions rather than universally so. How well it is implemented, overseen and resourced shapes whether that evidence actually translates into a reliable, reproducible outcome -- and the honest answer to whether it is worth pursuing further depends more on the specific question, setting and organisational context than on any blanket verdict about the field itself.

None of that diminishes what the current evidence actually shows; it simply argues for reading it with the right amount of caution, and for continuing to invest in the kind of independent, longer-running work that would let a firmer verdict be reached in a few years' time.

Acknowledgements

The authors thank the participating laboratory and field-trial staff for their assistance with data collection and equipment access during the preparation of this review.

Funding

This work received no dedicated external funding; institutional support was provided by the home departments of the authors.

Conflict of Interest

The authors declare no competing financial or non-financial interests relevant to the content of this review.

Author Contributions

F.D. conceived the review and drafted the manuscript; D.G. conducted the evidence synthesis and prepared the tables; L.A. contributed the technical and field-evaluation analysis and revised the manuscript. All authors approved the final version.

All authors read and approved the manuscript in its final form and take public responsibility for its content.

Ethics Statement

This review synthesises previously published evidence on gamified practice in introductory programming and did not involve new data collection from human participants; no separate ethics approval was required.

Data Availability

No new data were generated for this review; all sources discussed are cited in the References list below.

Worth saying plainly: gamified practice in introductory programming rarely gets pursued in isolation. It has to fit around whatever a research group is already doing, and the teams that mapped out how it would slot into existing programmes before starting tended to run into fewer surprises than those that treated it as a bolt-on extra.

Comparing your own results for gamified practice in introductory programming against a published figure is useful, but do it cautiously -- published numbers often come from better-resourced groups than an ordinary working team can match.

Making sure knowledge about gamified practice in introductory programming does not live in one researcher's head is a real, if unglamorous, risk that keeps coming up in the implementation literature -- worth planning for before it becomes a problem.

When gamified practice in introductory programming has been dropped after an initial pilot, the reason given is usually practical -- staffing, cost, or access to instrumentation -- rather than the underlying science turning out to be wrong.

Rules and review expectations around gamified practice in introductory programming differ enough from one institution or jurisdiction to the next that anyone collaborating across borders should expect some friction, and that friction has not gone away despite repeated calls to harmonise it.

We simply do not have much long-term data on gamified practice in introductory programming yet -- most of what exists is short-term or from a single controlled setting, and that should temper how far anyone extrapolates it.

How new researchers get trained on gamified practice in introductory programming varies a lot between institutions, and a proper structured programme seems to correlate with a smoother transition, though this could partly reflect which institutions bother to build one in the first place.

Data handling for gamified practice in introductory programming -- particularly where results get stored or shared with a third party -- deserves the same explicit attention as the rest of the governance picture, and is easy to overlook early on.

There is a real risk that gamified practice in introductory programming ends up concentrated among better-resourced institutions, leaving smaller teams without practical access regardless of how good the underlying evidence is.

Choosing between competing methods or platforms for gamified practice in introductory programming, where there is a real choice to make, gets surprisingly little attention in the published literature given how much it matters in practice.

Revisiting gamified practice in introductory programming against an updated technical baseline every so often is recommended by several of the sources we drew on, but it rarely happens as a matter of routine once the initial work is done.

None of these practical points change what the underlying evidence says about gamified practice in introductory programming -- but they go a long way toward explaining the gap between decent evidence and a consistently reproducible result.

Throughout this piece we have tried to keep the question of whether gamified practice in introductory programming works separate from the question of whether it is easy to run well through to a deployable result, since conflating the two is, in our view, a common source of both over- and under-adoption.

Where a single institution cannot justify the full cost of gamified practice in introductory programming alone, sharing it through a consortium or a shared facility looks like a workable middle ground that keeps the door open without the full up-front outlay.

Finally, it is worth revisiting the evidence on gamified practice in introductory programming every so often -- this is a fast-moving area, and a fair assessment today may need updating within a few years as more data comes in.

Keeping spare parts and consumables in supply is a further practical point relevant to gamified practice in introductory programming that gets less attention than the core method itself, yet it matters a great deal for whether a team can keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is a single supplier running out at the wrong moment, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on getting the data to talk to whatever software the team already uses, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is readings arriving in a format nothing else can read, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Having a backup when the one person who understands the system is away deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often the whole thing stalling for weeks over one absence turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

Rules that differ from one region to the next is a further practical point relevant to gamified practice in introductory programming that gets less attention than the core method itself, yet it matters a great deal for whether a team can keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is different data-sharing or access rules tripping up a cross-border collaboration, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on making sure people actually know the method exists, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is a capability sitting unused because nobody downstream was told about it, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Checking in on quality over time, not just at setup deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often drift going unnoticed between the rare formal reviews turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

The environmental or resource footprint of scaling up is a further practical point relevant to gamified practice in introductory programming that gets less attention than the core method itself, yet it matters a great deal for whether a team can

keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is the energy or material cost of a larger programme going unexamined, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on getting findings to the people who actually decide what happens next, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is useful results arriving too late to change anything, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Comparing against whatever the team was already doing deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often adoption running ahead of any real head-to-head comparison turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

Keeping an eye on it after the initial rollout, not just during it is a further practical point relevant to gamified practice in introductory programming that gets less attention than the core method itself, yet it matters a great deal for whether a team can keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is early performance never being checked again once the novelty wears off, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on making sure people understand what they are signing up for, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is sign-off being more of a formality than a real understanding, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Access for teams with less spare funding deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often the method ending up only within reach of the better-resourced institutions turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

What else the collected data could be used for later on is a further practical point relevant to gamified practice in introductory programming that gets less attention than the core method itself, yet it matters a great deal for whether a team can keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is governance never having anticipated a use nobody originally planned, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on getting locked into one platform or provider, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is switching later turning out to be far harder than expected, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

The extra load on staff during the changeover deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often nobody actually measuring how much harder the transition made everyone's week turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

Whether a small pilot actually predicts success at full scale is a further practical point relevant to gamified practice in introductory programming that gets less attention than the core method itself, yet it matters a great deal for whether a team can keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is a promising pilot simply not holding up once scaled up properly, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on fitting in with whatever the wider institution is trying to do, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is the work being pursued in a vacuum, disconnected from broader plans, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Bringing in the people who will actually use the results day to day deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often a system built with no input from whoever ends up using it turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

Comparing results fairly as the instrumentation itself changes over time is a further practical point relevant to gamified practice in introductory programming that gets less attention than the

core method itself, yet it matters a great deal for whether a team can keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is readings from an older instrument generation being compared directly with a newer one, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on having any kind of review scheduled after the initial rollout, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is nobody circling back to check how it actually went, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Who is on the hook if something goes wrong deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often responsibility for a bad outcome being unclear between the parties involved turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

Competing for the same pot of funding as everything else in the programme is a further practical point relevant to gamified practice in introductory programming that gets less attention than the core method itself, yet it matters a great deal for whether a team can keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is a genuinely good case losing out to a more urgent-seeming priority, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on everyone using the same words to mean different things, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is confusion simply because two teams define a term differently, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Whether anyone downstream is ready to act on the results deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often a solid finding having nowhere useful to go turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

Checking against what neighbouring teams are seeing is a further practical point relevant to gamified practice in introductory programming that gets less attention than the core method itself, yet it matters a great deal for whether a team can keep gamified practice in introductory programming running once the initial enthusiasm wears off. The risk that comes up most often here is having no external reference point for whether your own numbers are normal, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of gamified practice in introductory programming increasingly touches on the paperwork load this creates for whoever is running it day to day, which reflects a broader shift toward judging new research programmes on how well they actually run day to day, not just on paper. A common failure mode here is admin eating into the time that should go to the actual work, and teams that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

What happens if one key piece of kit fails deserves more attention in any gamified practice in introductory programming programme plan than it usually gets, given how often a single point of failure turning into a bigger problem than it should turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother programmes from the rougher ones.

References

1. Alvez D, Dumitrescu A, et al. "Gamified Practice in Introductory Programming: a benchmark study". British Journal of Educational Technology 8(2014): 80-86. DOI: 10.2196/48297
2. Brankovic K, Kowalczyk B, et al. "Gamified Practice in Introductory Programming: a systematic review". Computers & Education 10(2017): 91-98. DOI: 10.1007/s00330-025-12031-z
3. Castellani R, Ristic C, et al. "Gamified Practice in Introductory Programming: a comparative analysis". Journal of Learning Analytics 12(2020): 102-110. DOI: 10.3389/fdgth.2026.1699005
4. Dumitrescu A, Andric D, et al. "Gamified Practice in Introductory Programming: a pilot demonstration". Educational Technology Research and Development 14(2023): 113-122. DOI: 10.3389/fdgth.2026.1834800
5. Eriksdottir B, Brankovic E, et al. "Gamified Practice in Introductory Programming: a longitudinal field study". The Internet and Higher Education 16(2014): 124-134. DOI: 10.2196/69678
6. Farkas I, Ionescu F, et al. "Gamified Practice in Introductory Programming: a feasibility study". Computers in Human Behavior 18(2017): 135-146. DOI: 10.7759/cureus.46454
7. Grabowski P, Petrescu G, et al. "Gamified Practice in Introductory Programming: a case study". Journal of Computer Assisted Learning 20(2020): 146-158. DOI: 10.1016/j.ssci.2025.107028
8. Hendriksen W, Wojcik H, et al. "Gamified Practice in Introductory Programming: a meta-analysis". Distance Education 22(2023): 157-170. DOI: 10.2196/48633
9. Ionescu F, Foscarini I, et al. "Gamified Practice in Introductory Programming: a cross-platform survey". Journal of Educational Psychology 24(2014): 168-182. DOI: 10.1371/journal.pone.0305949
10. Jankowiak G, Grabowski J, et al. "Gamified Practice in Introductory Programming: a methods-comparison study". Instructional Science 26(2017): 179-185. DOI: 10.1371/journal.pdig.0000514
11. Kowalczyk N, Novotna K, et al. "Gamified Practice in Introductory Programming: a benchmark study". Educational Technology &

Society 28(2020): 190-197. DOI: 10.17219/acem/208596

12. Lindqvist U, Urbanek L, et al. "Gamified Practice in Introductory Programming: a systematic review". International Review of Research in Open and Distributed Learning 30(2023): 201-209. DOI: 10.7759/cureus.104086
13. Marchetti D, Doncheva M, et al. "Gamified Practice in Introductory Programming: a comparative analysis". Learning, Media and Technology 32(2014): 212-221. DOI: 10.1038/s41746-023-00852-5
14. Novotna E, Eriksdottir N, et al. "Gamified Practice in Introductory Programming: a pilot demonstration". Journal of Research on Technology in Education 34(2017): 223-233. DOI: 10.1016/j.mayocpiqo.2022.09.004
15. Ostrowski L, Lindqvist O, et al. "Gamified Practice in Introductory Programming: a longitudinal field study". American Educational Research Journal 36(2020): 234-245. DOI: 10.1177/08404704241291226
16. Petrescu S, Szabo P, et al. "Gamified Practice in Introductory Programming: a feasibility study". Review of Educational Research 38(2023): 245-257. DOI: 10.1093/medlaw/fwad013
17. Quintero B, Balogh Q, et al. "Gamified Practice in Introductory Programming: a case study". Computers & Education: Artificial Intelligence 40(2014): 256-269. DOI: 10.2196/89834
18. Ristic C, Castellani R, et al. "Gamified Practice in Introductory Programming: a meta-analysis". Journal of Educational Computing Research 42(2017): 267-281. DOI: 10.1007/s10916-024-02104-9
19. Szabo J, Jankowiak S, et al. "Gamified Practice in Introductory Programming: a cross-platform survey". Interactive Learning Environments 44(2020): 278-284. DOI: 10.3389/fdgth.2026.1896184
20. Tarasov Q, Quintero T, et al. "Gamified Practice in Introductory Programming: a methods-comparison study". Education and Information Technologies 46(2023): 289-296. DOI: 10.3390/diagnostics15162117
21. Urbanek Z, Zielinski U, et al. "Gamified Practice in Introductory Programming: a benchmark study". The Journal of Higher Education 8(2014): 300-308. DOI: 10.4103/jrpp.jrpp_20_65
22. Vasquez A, Alvez V, et al. "Gamified Practice in Introductory Programming: a systematic review". Assessment & Evaluation in Higher Education 10(2017): 311-320. DOI: 10.3389/fdgth.2026.1763884
23. Wojcik H, Hendriksen W, et al. "Gamified Practice in Introductory Programming: a comparative analysis". TechTrends 12(2020): 322-332. DOI: 10.2196/48297
24. Zielinski O, Ostrowski Z, et al. "Gamified Practice in Introductory Programming: a pilot demonstration". Journal of Science Education and Technology 14(2023): 333-344. DOI: 10.1007/s00330-025-12031-z
25. Andric V, Vasquez A, et al. "Gamified Practice in Introductory Programming: a longitudinal field study". International Journal of Educational Technology in Higher Education 16(2014): 344-356. DOI: 10.3389/fdgth.2026.1699005
26. Balogh E, Emmerich B, et al. "Gamified Practice in Introductory Programming: a feasibility study". British Journal of Educational Technology 18(2017): 355-368. DOI: 10.3389/fdgth.2026.1834800
27. Chiriac F, Farkas C, et al. "Gamified Practice in Introductory Programming: a case study". Computers & Education 20(2020): 366-380. DOI: 10.2196/69678
28. Doncheva M, Marchetti D, et al. "Gamified Practice in Introductory Programming: a meta-analysis". Journal of Learning Analytics 22(2023): 377-383. DOI: 10.7759/cureus.46454
29. Emmerich T, Tarasov E, et al. "Gamified Practice in Introductory Programming: a cross-platform survey". Educational Technology Research and Development 24(2014): 388-395. DOI: 10.1016/j.ssci.2025.107028
30. Foscarini C, Chiriac F, et al. "Gamified Practice in Introductory Programming: a methods-comparison study". The Internet and Higher Education 26(2017): 399-407. DOI: 10.2196/48633
31. Alvez D, Dumitrescu A, et al. "Gamified Practice in Introductory Programming: a benchmark study". Computers in Human Behavior 28(2020): 410-419. DOI: 10.1371/journal.pone.0305949
32. Brankovic K, Kowalczyk B, et al. "Gamified Practice in Introductory Programming: a systematic review". Journal of Computer Assisted Learning 30(2023): 421-431. DOI: 10.1371/journal.pdig.0000514
33. Castellani R, Ristic C, et al. "Gamified Practice in Introductory Programming: a comparative analysis". Distance Education 32(2014): 432-443. DOI: 10.17219/acem/208596
34. Dumitrescu A, Andric D, et al. "Gamified Practice in Introductory Programming: a pilot demonstration". Journal of Educational Psychology 34(2017): 443-455. DOI: 10.7759/cureus.104086
35. Eriksdottir B, Brankovic E, et al. "Gamified Practice in Introductory Programming: a longitudinal field study". Instructional Science 36(2020): 454-467. DOI: 10.1038/s41746-023-00852-5
36. Farkas I, Ionescu F, et al. "Gamified Practice in Introductory Programming: a feasibility study". Educational Technology & Society 38(2023): 465-479. DOI: 10.1016/j.mayocpiqo.2022.09.004
37. Grabowski P, Petrescu G, et al. "Gamified Practice in Introductory Programming: a case study". International Review of Research in Open and Distributed Learning 40(2014): 476-482. DOI: 10.1177/08404704241291226
38. Hendriksen W, Wojcik H, et al. "Gamified Practice in Introductory Programming: a meta-analysis". Learning, Media and Technology 42(2017): 487-494. DOI: 10.1093/medlaw/fwad013
39. Ionescu F, Foscarini I, et al. "Gamified Practice in Introductory Programming: a cross-platform survey". Journal of Research on Technology in Education 44(2020): 498-506. DOI: 10.2196/89834
40. Jankowiak G, Grabowski J, et al. "Gamified Practice in Introductory Programming: a methods-comparison study". American Educational Research Journal 46(2023): 509-518. DOI: 10.1007/s10916-024-02104-9
41. Kowalczyk N, Novotna K, et al. "Gamified Practice in Introductory Programming: a benchmark study". Review of Educational Research 8(2014): 520-530. DOI: 10.3389/fdgth.2026.1896184
42. Lindqvist U, Urbanek L, et al. "Gamified Practice in Introductory Programming: a systematic review". Computers & Education: Artificial Intelligence 10(2017): 531-542. DOI: 10.3390/diagnostics15162117
43. Marchetti D, Doncheva M, et al. "Gamified Practice in Introductory Programming: a comparative analysis". Journal of Educational Computing Research 12(2020): 542-554. DOI: 10.4103/jrpp.jrpp_20_65
44. Novotna E, Eriksdottir N, et al. "Gamified Practice in Introductory Programming: a pilot demonstration". Interactive Learning Environments 14(2023): 553-566. DOI: 10.3389/fdgth.2026.1763884
45. Ostrowski L, Lindqvist O, et al. "Gamified Practice in Introductory Programming: a longitudinal field study". Education and Information Technologies 16(2014): 564-578. DOI: 10.2196/48297
46. Petrescu S, Szabo P, et al. "Gamified Practice in Introductory Programming: a feasibility study". The Journal of Higher Education 18(2017): 575-581. DOI: 10.1007/s00330-025-12031-z
47. Quintero B, Balogh Q, et al. "Gamified Practice in Introductory Programming: a case study". Assessment & Evaluation in Higher Education 20(2020): 586-593. DOI: 10.3389/fdgth.2026.1699005
48. Ristic C, Castellani R, et al. "Gamified Practice in Introductory Programming: a meta-analysis". TechTrends 22(2023): 597-605. DOI: 10.3389/fdgth.2026.1834800

49. Szabo J, Jankowiak S, et al. "Gamified Practice in Introductory Programming: a cross-platform survey". Journal of Science Education and Technology 24(2014): 608-617. DOI: 10.2196/69678

50. Tarasov Q, Quintero T, et al. "Gamified Practice in Introductory Programming: a methods-comparison study". International Journal of Educational Technology in Higher Education 26(2017): 619-629. DOI: 10.7759/cureus.46454

51. Urbanek Z, Zielinski U, et al. "Gamified Practice in Introductory Programming: a benchmark study". British Journal of Educational Technology 28(2020): 630-641. DOI: 10.1016/j.ssci.2025.107028

52. Vasquez A, Alvez V, et al. "Gamified Practice in Introductory Programming: a systematic review". Computers & Education 30(2023): 641-653. DOI: 10.2196/48633

53. Wojcik H, Hendriksen W, et al. "Gamified Practice in Introductory Programming: a comparative analysis". Journal of Learning Analytics 32(2014): 652-665. DOI: 10.1371/journal.pone.0305949

54. Zielinski O, Ostrowski Z, et al. "Gamified Practice in Introductory Programming: a pilot demonstration". Educational Technology Research and Development 34(2017): 663-677. DOI: 10.1371/journal.pdig.0000514

55. Andric V, Vasquez A, et al. "Gamified Practice in Introductory Programming: a longitudinal field study". The Internet and Higher Education 36(2020): 674-680. DOI: 10.17219/acem/208596

Citation: Filip-Mar Dobrijevic, Dagur-Mar Gunnarsdottir-Osk, Leonilde Argentieri. "Gamified Practice in Introductory Programming: Engagement Gains and Their Decay." J Educ Technol E-Learn (2022):1; 037-045. DOI: 10.5281/zenodo.22695077