A Data-Driven Recommender for Gamified Classroom Designs

CS 4980 Technical Report, Spring, 2025

Eric Weng
Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia, USA
qgt7zm@virginia.edu

Abstract

Gamification is a pedagogical technique that aims to remedy low motivation, engagement, and grades, which are common problems among students in the United States. Although researchers have studied gamification extensively, papers still produce inconclusive or conflicting results, with little indication of which game design elements trigger which student learning outcomes. Authors also struggle standardizing results across publications with different student demographics. More concerning is how many experiments appear to have been designed without input from either teachers or students, which leads to systems that are difficult to implement in classrooms. Thus, I developed a web application that can help teachers learn how to create gamified classrooms based on hard evidence instead of theory and speculation. My application incorporates data from past studies to generate effective recommendations for users' needs. This technical report documents how I designed my data models and recommendation algorithm and demonstrates how a typical user might use my website. I end by discussing the current limitations of my application and suggestions for improvement in the future, inviting others to take part in development.

1 Introduction

Across the United States, students are losing confidence in their education system's ability to ready them for the job market. Only 34% of undergraduate students believed they were well-prepared to seek a job, compared to the 88% of students who entered college to receive employment [7]. Modern education prioritizes good grades and memorization, which was useful in training factory laborers during the Industrial Revolution, but is no longer sufficient for today's multi-disciplinary and collaborative workplace [12]. As a result, schools suppress students' individuality and creativity, causing otherwise talented pupils to lose motivation to learn and even drop out [8].

Gamification promises to address education's deficiencies by bringing the fun and excitement of games to classrooms. By granting students more autonomy over their learning, gamification can improve student motivation and engagement, ultimately improving knowledge retention and grades [6]. Companies have been quick to capitalize on these supposed benefits, with the prominence of gamified learning apps such as Duolingo, Kahoot!, and Quizlet. Nonetheless, researchers in the field have repeatedly called for more rigorous and empirical studies [5]. Without sufficient scientific control, scholars will not be able to fully understand of each game element affects student learning, and instructors will not be well-equipped to design gamified systems.

To this end, I designed and constructed a prototype web application to serve as a teaching tool for educators. The website works by aggregating results from published and peer-reviewed studies, capturing the general trend of how different game elements affect student behavior while smoothing out outliers. Normally, users need not worry about inserting data, as the website host will curate the database. Teachers can input their classroom needs, from desired learning outcomes to student demographics, into custom scenarios and receive a recommendation of what elements to employ to best achieve those goals. Additionally, scholars could view the existing results to see what areas of research are lacking without creating scenarios. The rest of the report will explain why and how I developed my web application, demonstrate how it functions, and propose how others could extend it for future use.

2 Background and Motivation

Gamification is defined as the application of game design elements in non-game environments to promote desirable behavior, such as greater knowledge retention, class engagement, or test scores [13]. An element may range from abstract gameplay concepts, such as competition and limited resources, to tangible systems that implement those concepts, such as leaderboards and currency [3]. Gamification generally produces positive results when the elements satisfies students' primary psychological requirements for motivation: competence, autonomy, and relatedness, as outlined by self-determination theory [15].

Some challenges hindering further adoption of gamification in education are overspecialization into competition, poor experimental design, and lack of consideration for teachers' needs. Currently, much of gamification research centers on a few specific elements, with points, badges, and leaderboards (commonly abbreviated as PBL) being of specific interest [4]. The PBL elements offer plenty of external rewards through competition but risk lowering intrinsic motivation [17] and conditioning students to depend on these rewards for learning [21]. If implemented poorly, competitive features may only serve to introduce additional measures of control over student learning, rather than relaxing them [9]. Furthermore, already underperforming students may become further demotivated if they are aware of how behind they are compared to other students [2].

Another area for improvement is the experimental design of gamification studies. Researchers often test multiple elements together, making determining which element caused what response in students more difficult [5]. Even if the results are conclusive enough, authors lack methods to evaluate the long-term success of gamification, which is needed to rule out the novelty effect of the game elements [17]. Moreover, studies done on university-level CS courses are overrepresented, with 53% of all studies being done on college-age students and 19% with programming courses [23]. Understanding the effect of student demographics on learning an important research topic in the field [22], and evidence supports gender being a strong factor in how gamification affects students [18]. Students from different demographics learn differently, which means results drawn from one group of students do not necessarily generalize to all types of students.

Lastly, gamification can also be difficult for teachers to integrate into classrooms. The Technological, Pedagogical, and Content Knowledge (TPACK) educational framework states that teachers cannot improve education only by introducing new technologies or teaching practices. Instead, instructors must learn how these new skills fit into their current subject matter, existing practices, and learning technologies, as well as how these three areas affect each other [11]. The TPACK authors also advocate for teachers and students to collaborate in designing course structures. Through "learning by design", both sides can discover each other's needs and best practices [11]. In contrast, many gamification studies are conducted with little to no consideration the needs of teachers and students [16]. This lack of communication results in teachers being misinformed about gamification, leading them to often seek out techniques due to novelty rather than proven effect [10].

Despite the wealth of software claiming to boost student learning performance, accessible resources for learning how to design gamified environments remain scarce. Therefore, I saw a unique opportunity to build a tool that can help teachers determine which elements to best utilize in their classrooms. My program is not only sensitive to the subject and student age of the user's classroom, but also allows them curate a database specific to those needs.

3 Application Design

3.1 Requirements

Originally, I wanted to leverage the "learn by design" concept introduced with the TPACK framework. My initial concept envisioned users playing through gamified scenarios that required them to design classrooms to meet particular learning goals—in essence, using gamification to teach gamification. The game would rate users, after they completed each scenario, on how well they improved the desired student behaviors and provided resources to read more about each element or outcome. Next, the game would increase the difficulty of subsequent scenarios by restricting the total "cost" of the design, whether that be money, time, or student satisfaction. Users could also progress between "levels" by "unlocking" new gamification technique they could use in future scenarios or upgrading their "expertise," which would make a technique less costly in the future.

However, I could not be certain this gamified design would work better than a simple informational format, given the concerns for indecisive results in academia. Moreover, I found little literature covering the cost of implementing gamification, financial or otherwise, limiting the flexibility of the game design itself. I ended up settling on a simplified version of my earlier idea with just a scenario feature and no gamification. The user would enter their requirements into each scenario, namely desired learning outcomes and student demographics. Then, the application would draw on its library of past experimental results to suggest the elements that best fit the given purpose. Users would need a robust interface to access and manage the data, so nearly all parts of the databases exposed through views. For most models, a separate view was created that displayed a table of each record. Users could search the table by column, click on a row to access more detailed information about the record, and add new records to the table.

3.2 Database Models

The teaching tool is essentially a thin layer of views on top of a relational database, with minimal abstraction over the models. The entity-relationship (ER) diagram below (Figure 1) describes the classes/entity-sets used by the application. In my implementation, each relation has an auto-incrementing integer primary key field (ID), which is not included for brevity. The underlined attributes are assumed to be unique together (and thus a candidate key), although in reality, authors (and even titles) could conceivably share common names.

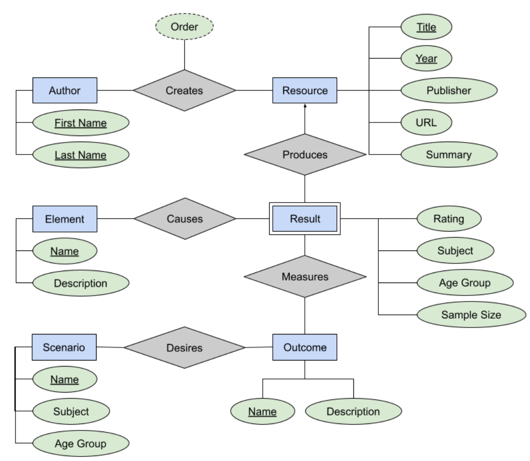

Figure 1: Application Entity-Relationship Diagram

The purpose for each database model is as follows:

(1) **Resource**: A journal article, conference paper, or book. Each **Resource** has a title, year of publication, and location of publication (journal, conference, book publisher, etc.). The resource also includes short summary of its findings, as well as a URL (such as a DOI link) so users can quickly access these resources to read more for themselves.
(2) **Author**: Someone who writes resources. The **Author** relation contains a first and last name field and participates in a many-to-many relationship with **Resource**. Keeping **Author** as a standalone entity set instead of a comma-separated text attribute inside **Resource** facilitates searching for resources published by author name.
(3) **Element**: A game design element. Each **Element** consists of a name and a text description.
(4) **Outcome**: A student behavioral change caused by elements. Each **Outcome** also has a name and a text description.

(5) **Result**: An experimental result produced by a study. Each **Result** contains a rating, subject, age group, and sample size. The rating describes how conclusive and positive the results, and is an enum with options *Very Positive*, *Somewhat Positive*, *Neutral*, *Somewhat Negative*, and *Very Negative*. The possible values for subject are *Computing*, *Engineering*, *Mathematics*, *Sciences*, *Medicine*, *Languages*, *Humanities*, and *Other* (denotes miscellaneous or unknown). The choices for age group are *Elementary (K-5)*, *Middle School (6-8)*, *High School* (9-12), *Undergraduate*, *Graduate*, and *Other*. A **Result** can be associated with at least one **Element** and at least one **Outcome**. Each **Resource** may also have multiple **Results** if one is not enough to describe the entire experiment (i.e. different outcomes had different ratings), although not all resources need have results. For example, Jane McGonigal's book *Reality is Broken* [14] would be a useful starter resource for users new to gamification, but because it is not specifically geared towards education, the database would not record any experimental results.

(6) **Scenario**: The learning needs of a classroom. All **Scenarios** may take one or more desired **Outcomes**, as well as an optional subject and age group choice like those for **Result**. Presently, the application recalculates the recommended elements every time the scenario is viewed. During development the algorithm and data set were frequently adjusted, meaning the results could rapidly become outdated with just a few changes. A future version whose algorithm is more stable could opt to cache or periodically update recommendations.

3.3 Recommender Algorithm

The recommender algorithm accepts a user's scenario and produces an ordered map of elements to their "score." The greater an element's score is, the more suitable that element is for the user's scenario. The recommender first gathers all results containing any outcomes selected in the input scenario. The resources that the results belong to are also saved for later use. Afterward, the program collects all the elements associated with those results and calculates their aggregated from several factors. Finally, the algorithm outputs the elements ranked by descending score plus the list of resources supporting the recommendations. A flowchart of the recommendation algorithm (Figure 2) is shown to the right. The exact aggregate fields and score formula are not included, as the algorithm's interface should stay the same even if the formula itself is altered.

To calculate the score of an element, the algorithm averages the following factors from all associated results:

(1) **Rating:** The most important determiner an element's suitability is how well it does in an experiment. The enum constants map linearly to integers, with *Very Negative* at -2 and *Very Positive* at +2.

(2) **Subject Similarity:** How related the experiment's subject to an instructor's subject affects how transferrable the results are. If the scenario's subject matches that of the result, then the similarity value is set to 1.1. Otherwise, the value is set to 0.9.

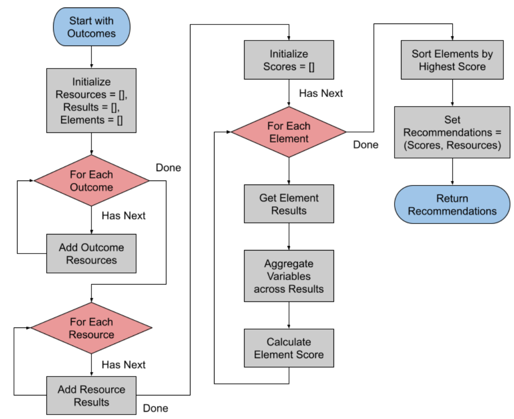

Figure 2: Application Recommender Flowchart

(3) **Age Group Similarity:** As with subject, the age group similarity equals 1.1 when the age groups match and 0.9 otherwise.

(4) **Number of Elements**: Too many elements being tested in one study "dilutes" the certainty of the result, since it becomes difficult to tell which element affects which outcome. Unlike the other factors, higher is worse, so the score is inversely proportional to the count.

(5) **Proportion of Desired Outcomes:** Likewise, too many outcomes in a result means an element could result in unwanted side effects. The proportion is defined as the number of outcomes both selected in the scenario and present in a result (desired outcomes) divided by the total number of outcomes in the same result (total outcomes).

(6) **Number of Results:** The more results supporting a particular element, the more likely it causes the desired outcome. This figure is not really an average but is aggregated from the count of results per element.

The recommender calculates an element's final score by multiplying all the variables together, although there are many other possible ways to do so. Given the limited dataset during development, there is a risk that some variables could grow quite large and drastically inflate the value of the final score. For example, simply increasing the **Number of Results** from 1 to 2 would result in a +100% increase in the score. Additionally, increasing the result count past a certain has diminishing returns (compare 50 vs 100 results in favor of an element). Therefore, the **Number of Elements** and **Number of Results** variables use the square root of the averaged value instead of the raw value to prevent overly large values from affecting the score too strongly. The formula also takes the square root of the **Proportion of Desired Outcomes**, which slightly raises values close to one (such as 4/5) while lowering those close to zero (such as 1/5). The complete formula used to compute scores is

$$\begin{aligned}\text{score}(element) = {} & \text{avg}(rating)\\ & \times \text{avg}(subject\_similarity)\\ & \times \text{avg}(age\_similarity)\\ & \times \frac{1}{\sqrt{\text{avg}(\text{count}(elements))}}\\ & \times \sqrt{\text{avg}(\text{prop}(desired\_outcomes))}\\ & \times \sqrt{\text{count}(results)}.\end{aligned}$$

To make the numerical scores more readable to the user, the scenario page displays text labels next to the scores with the following arbitrary cutoffs:

- *Excellent* for scores greater than or equal to 1.25;
- *Great*, for scores in the range $[1, 1.25)$;
- *Good*, for scores in the range $[0.75, 1)$;
- *OK*, for scores in the range $[0.5, 0.75)$;
- *Poor*, for scores in the range $[0, 0.5)$; and
- *Awful*, for scores below zero.

Earlier on, the program repeated the algorithm for each outcome individually, displaying the ranked elements under the outcome. However, this approach presented too much information to the user and forced them to sift through the recommendations to avoid elements ranked high for one outcome but low for another. Besides, a user could create a scenario with fewer elements if they wished to see recommendations separately.

3.4 Data Collection

To collect data for my application, I searched for articles on internet databases, including, but not limited to, the ACM Digital Library, IEEE Xplore, and ScienceDirect. Within ScienceDirect, I paid attention to journals relating to computing, education, or psychology, such as the aptly named *Computers & Education* and *Computers in Human Behavior*.

I prioritized literature reviews in my research since they reference numerous studies the authors have specifically curated for their analysis [1] [4] [5] [19] [20]. These articles could be found by searching for the keywords "literature," "systematic," and "review." In addition, the authors provide some commentary or description of each study that enhances my categorization of the results. More recent and specific publications could be found by searching for the keywords "gamification," "education," "student," and the name of an element or outcome. I used Zotero to manage my citations and save PDFs of each article and the LibKey Nomad browser extension to access resources freely through the UVA Library.

My intent is for others to start using the website immediately, so I have included a starter dataset with all the resources and results I compiled for download alongside my code repository. To my best knowledge (and on my honor as a UVA student), all data represents my own research, and no text descriptions of resources have been plagiarized. The starter dataset does not contain any premade scenarios, aside from a few test ones for demonstrating how the recommender system functions, as users are supposed to create them on their own. Scholars might find it interesting to "recreate" past experiments with existing data and compare the generated results to the empirical ones.

4 Application Construction and Operation

4.1 Implementation

The teaching tool was implemented as website, using the Python Django framework for the backend and a local SQLite database for persistent storage. The Bootstrap library and minimal JavaScript behaviors also provided the frontend and user interface. A web application was preferred over a desktop one since the former would allow for faster sharing of information between multiple users Additionally, this web stack is relatively simple and portable, which should reduce the effort and resources needed for prospective users to host and maintain the website. The Django framework and Python language are beginner-friendly, and the few external libraries are simple to download into a virtual environment. A server backend like Django also holds several advantages over a pure HTML/CSS/JS website, including storing persistent data with a variety of DBMSes and providing convenient support for multiple user sessions and accounts later on. This software does not use any features distinctive to Python, so another programmer could reproduce the application in a different language or framework if they desired.

4.2 Views

The website is structured with the following types of views:

- **Home**, the landing page with a description of the website and gamification and links to other pages;
- **List Records**, which shows summaries of records of a given model and allows users to filter the table;
- **View Record**, which shows detailed information of a particular record;
- **Insert Record**, where users can input a new record into the database;
- **Data**, where users can upload other datasets or share their own; and
- **About**, which describes my background and why I constructed this website.

Only the **Element**, **Outcome**, **Resource**, and **Scenario** model classes have **List Records** pages. There is no separate list or input page for **Authors** and **Results** since they are viewed and inserted alongside their respective resources. Every page also shows an identical navigation bar in the header linking to all the pages and a footer showing the source code repository. A typical user workflow on the application might proceed as follows:

(1) The user uploads a dataset contained in a JSON file through the **Data** page, if they do not already have a local or remote database. The **Data** page contains safeguards to prevent users from accidentally overwriting or deleting their databases.
(2) The user searches the **List Resources** page for an author name containing "rodrigues," year equaling "2022," and summary containing "narrative" [18] (Figure 3). The search bar is case-insensitive, and only allows numbers to be inputted for resource year. There is also a "Clear Filters" button to allow the user to restart their search.

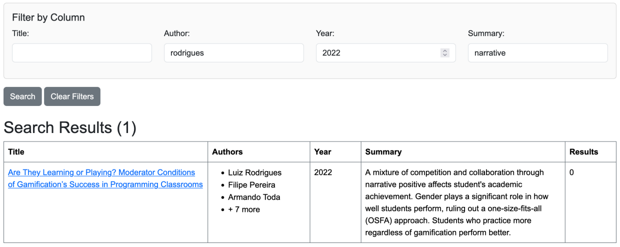

Figure 3: Application List Resources Page

(3) The user adds a result for the resulting resource with elements "Badges," "Narrative," and "Teamwork," outcome "Grades," rating *Very Positive*, subject *Computing*, age group *Undergraduate*, and sample size 399 (Figure 4).

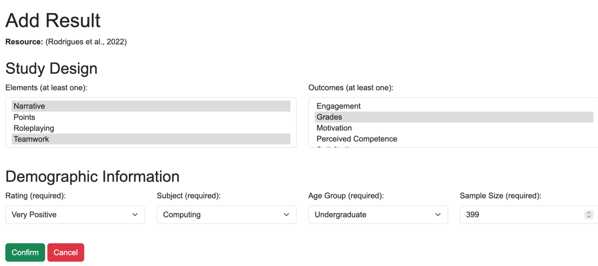

Figure 4: Application Add Resource Result Page

(4) The user creates a **Scenario** with elements "Engagement" and "Grades", subject *Computing*, and Age Group *Undergraduate* (Figure 5).

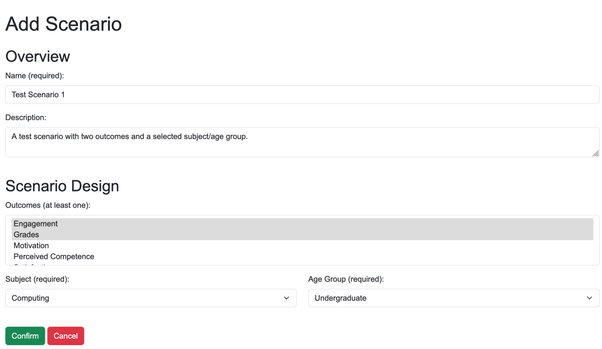

Figure 5: Application Create Scenario Page

(5) The user views the recommendations for the newly-created scenario (Figure 6).

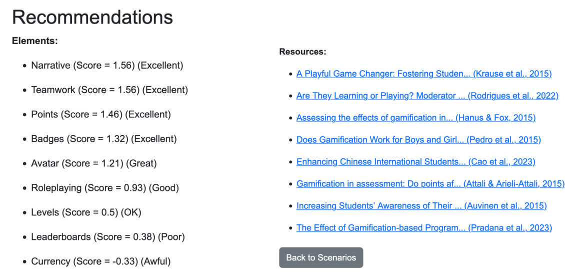

Figure 6: Application View Scenario Recommendations Page

5 Discussion

5.1 Use Cases

For gamification to achieve greater success in education, teachers (and even students) should be more actively involved in its research, down to the very design of studies. But before this can happen, teachers need to be well-informed on how gamification affects students in their particular subject area and student demographic, as well as what it can and cannot do. I hope that my website can serve as an accessible starting point for educators interested in gamification, showing them the basics and allowing them the basics without forcing them to dive into endless academic papers. My program, if well-tuned, can train educators to approach gamification more objectively and analytically, thus avoiding the pitfalls of hype. Furthermore, through the data management feature, educators and scholars alike can collectively build one large dataset that can be hosted on one installation.

5.2 Future Improvements

While the recommender shows promise, the quality of its algorithm is still limited by the quality and breadth of the data. With more data gathered over time, the recommender would satisfy the central limit theorem and become more representative of the larger population of studies. One notable weakness of the current algorithm is that all aggregate variables are measured at different scales. Without sufficient scaling, a scenario could easily generate recommendations with most scores exceeding 10, rendering the predefined cutoff of 1.25 utterly meaningless. For this reason, result sample size (where larger sample size means less variance and thus more certainty) could not be modeled in the algorithm, as the resulting scores became too large. A more ideal approach would be to transform all the variables to a standard normal (Z) distribution. Then, the application would simply compute a weighted average instead of taking the product of all variables. The raw scores would similarly be standardized to a bell curve, which would help restrict the possible range of scores. Moreover, the score labels could be assigned to certain percentiles instead of to subjective cutoffs. For instance, *Excellent* could correspond to the 90th percentile, *Great* to the 80th percentile, and so on.

Additionally, the algorithm considers connected subjects such as *Computing* and *Engineering* to be as distinct as disconnected subjects such as *Computing* and *Humanities*. A possible solution would be to establish a graph/matrix with distances between subjects representing their relatedness. Future models may also consider other demographic factors—gender, ethnicity, technological literacy, learning disorders, and funding—and geographic factors—country and language of instruction if enough data can be found. All demographic fields plus subject can easily be implemented as separate relations if users want greater and finer customization over the available options.

With so many variables involved, a linear regression model or some other form of machine learning could more accurately and efficiently determine the weights. Properly labeling the target score values in the training data will take significant investment. The enum constants for the **Result** ratings are admittedly subjective, and the p-value or some statistic may be a more objective and less vague measure. To rapidly gather such a large amount of data, researchers could also utilize web scraping APIs or LLM text summarizers. However, they must take care not to access only authorized resources and to not plagiarize descriptions. As the algorithm grows more complex, a later version could save the recommendations to the database to avoid recomputing them for each page load. The developers would need to determine what happens to the recommendations after enough new data is added or the model is significantly updated. Either the application could automatically refresh the recommendations, or users would be prompted to manually recalculate them or create a copy of the scenario.

Lastly, to facilitate open and unrestricted improvements to my application, I have published the full source code and dataset publicly at https://github.com/qgt7zm/capstone-project/. Both the code and dataset are released under the permissive MIT license, which allows users to run, study, modify, and redistribute freely, as long as they include the license notice.

Acknowledgments

I would like to thank Professor Mark Sherriff of the UVA CS department, for serving my capstone project advisor. I would also like to thank my fourth-year STS professors, Rider Foley and MC Forelle, for guiding me through the other portions of my senior thesis. I did not receive any funding for this project.

References

[1] Ar Anil Yasin and Asad Abbas. 2021. Role of gamification in Engineering Education: A systematic literature review. In *2021 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, Vienna, Austria, 210–213. doi:10.1109/EDUCON46332.2021.9454038

[2] Tapio Auvinen, Lasse Hakulinen, and Lauri Malmi. 2015. Increasing Students' Awareness of Their Behavior in Online Learning Environments with Visualizations and Achievement Badges. *IEEE Transactions on Learning Technologies* 8, 3 (July 2015), 261–273. doi:10.1109/TLT.2015.2441718 Conference Name: IEEE Transactions on Learning Technologies.

[3] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining Gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, Vol. 11. Association for Computing Machinery, New York, NY, USA, 9–15. doi:10.1145/2181037.2181040

[4] Christo Dichev and Darina Dicheva. 2017. Gamifying education: what is known, what is believed and what remains uncertain: a critical review. *International Journal of Educational Technology in Higher Education* 14, 1 (Dec. 2017), 9. doi:10.1186/s41239-017-0042-5

[5] Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. 2015. Gamification in Education: A Systematic Mapping Study. *Educational Technology & Society* 18 (July 2015), 75–88. https://www.researchgate.net/publication/270273830_Gamification_in_Education_A_Systematic_Mapping_Study

[6] Berkeley Center for Teaching & Learning. 2015. Implementing Gamification. Retrieved March 28, 2025 from https://teaching.berkeley.edu/teaching-guides/running-your-course/implementing-gamification

[7] Strada Education Foundation. 2018. New Survey Reveals Crisis of Confidence in Workforce Readiness Among College Students. Retrieved March 28, 2025 from https://web.archive.org/web/20240526222148/https://stradaeducation.org/press-release/new-survey-reveals-crisis-of-confidence-in-workforce-readiness-among-college-students/

[8] Anthony D. Fredericks. 2021. How Education Quashed Your Creativity. Retrieved March 28, 2025 from https://www.psychologytoday.com/us/blog/creative-insights/202108/how-education-quashed-your-creativity-0

[9] Michael D. Hanus and Jesse Fox. 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 (Jan. 2015), 152–161. doi:10.1016/j.compedu.2014.08.019

[10] Muhammad Kamarul Kabilan, Nagaletchimee Annamalai, and Kee-Man Chuah. 2023. Practices, purposes and challenges in integrating gamification using technology: A mixed-methods study on university academics. *Education and Information Technologies* 28, 11 (April 2023), 1–33. doi:10.1007/s10639-023-11723-7

[11] Matthew J. Koehler and Punya Mishra. 2005. What Happens When Teachers Design Educational Technology? The Development of Technological Pedagogical Content Knowledge. *Journal of Educational Computing Research* 32, 2 (March 2005), 131–152. doi:10.2190/0EW7-01WB-BKHL-QDYV Publisher: SAGE Publications Inc.

[12] Karthik Krishnan. 2020. Our education system is losing relevance. Here's how to update it. Retrieved March 28, 2025 from https://www.weforum.org/agenda/2020/04/our-education-system-is-losing-relevance-heres-how-to-update-it/

[13] Joey J. Lee and Jessica Hammer. 2011. Gamification in Education: What, How, Why Bother? *Academic Exchange Quarterly* 15 (2011), 146. https://api.semanticscholar.org/CorpusID:220866115

[14] Jane McGonigal. 2011. *Reality is Broken: Why Games Make Us Better and How They Can Change the World*. Penguin Books.

[15] Elisa D. Mekler, Florian Brühlmann, Alexandre N. Tuch, and Klaus Opwis. 2017. Towards understanding the effects of individual gamification elements on intrinsic motivation and performance. *Computers in Human Behavior* 71 (June 2017), 525–534. doi:10.1016/j.chb.2015.08.048

[16] Alberto Mora, Daniel Riera, Carina González, and Joan Arnedo-Moreno. 2017. Gamification: a systematic review of design frameworks. *Journal of Computing in Higher Education* 29, 3 (Dec. 2017), 516–548. doi:10.1007/s12528-017-9150-4

[17] Elias Ratinho and Cátia Martins. 2023. The role of gamified learning strategies in student's motivation in high school and higher education: A systematic review. *Heliyon* 9, 8 (Aug. 2023), e19033. doi:10.1016/j.heliyon.2023.e19033

[18] Luiz Rodrigues, Filipe Pereira, Armando Toda, Paula Palomino, Wilk Oliveira, Marcela Pessoa, Leandro Carvalho, David Oliveira, Elaine Oliveira, Alexandra Cristea, and Seiji Isotani. 2022. Are They Learning or Playing? Moderator Conditions of Gamification's Success in Programming Classrooms. *ACM Trans. Comput. Educ.* 22, 3, Article 30 (June 2022), 27 pages. doi:10.1145/3485732

[19] Judy Julieth Ramírez Ruiz, Ana Dolores Vargas Sanchez, and Oscar Rafael Boude Figueredo. 2024. Impact of gamification on school engagement: a systematic review. *Frontiers in Education* 9 (Dec. 2024), 10 pages. doi:10.3389/feduc.2024.1466926 Publisher: Frontiers.

[20] Sujit Subhash and Elizabeth A. Cudney. 2018. Gamified learning in higher education: A systematic review of the literature. *Computers in Human Behavior* 87 (Oct. 2018), 192–206. doi:10.1016/j.chb.2018.05.028

[21] Armando M. Toda, Pedro H. D. Valle, and Seiji Isotani. 2018. The Dark Side of Gamification: An Overview of Negative Effects of Gamification in Education. In *Higher Education for All. From Challenges to Novel Technology-Enhanced Solutions*, Alexandra Ioana Cristea, Ig Ibert Bittencourt, and Fernanda Lima (Eds.). Springer International Publishing, Cham, 143–156. doi:10.1007/978-3-319-97934-2_9

[22] Kasper Welbers, Elly A Konijn, Christian Burgers, Anna Bij de Vaate, Allison Eden, and Britta C Brugman. 2019. Gamification as a tool for engaging student learning: A field experiment with a gamified app. *E-Learning and Digital Media* 16, 2 (March 2019), 92–109. doi:10.1177/2042753018818342 Publisher: SAGE Publications.

[23] Nilüfer Zeybek and Elif Saygı. 2024. Gamification in Education: Why, Where, When, and How?—A Systematic Review. *Games and Culture* 19, 2 (2024), 237–264. doi:10.1177/15554120231158625