2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #52223

AI-Driven Game Design for Enhancing Engagement in Calculus Learning

Ing. Guillermo Cotto, Universidad Galileo

Guillermo Cotto received the B.S. degree in Telecommunications and Telematics Networks Engineering from Universidad Galileo, Guatemala, and the M.S. degree in Information Technology, with a specialization in Information Security, from the same institution. He is currently a lecturer and course administrator at Universidad Galileo and collaborates with the Department of Applied Mathematics in the instruction of undergraduate mathematics courses for engineering students. His academic interests include engineering mathematics education, gamified learning environments, and the integration of artificial intelligence technologies to enhance student engagement and learning outcomes in STEM education.

Dr. Alberth Alvarado, Universidad Galileo

Alberth Alvarado received, with honors, the B.S. degree in Electronics and Computer Science Engineering from Universidad Francisco Marroquín, Guatemala, in 2004, and the M.S. degree in Applied Mathematics and the Ph.D. degree in Industrial Engineering from the University of Illinois at Urbana–Champaign, USA, in 2010 and 2014, respectively. He is Head of the Department of Applied Mathematics at Universidad Galileo, Guatemala, where he leads initiatives to strengthen mathematics education for engineering students. He teaches undergraduate mathematics for engineering students and graduate courses in operations research and data science. His research interests include engineering education, particularly the teaching and curriculum development of mathematics for engineers. His technical interests include game theory, convex and nonconvex optimization, and distributed optimization with applications to signal processing and communication systems.

Ing. Jose Roberto Portillo, Universidad Galileo

Roberto Portillo heads the Academic Support Department of Universidad Galileo in Guatemala and is a mathematics instructor in engineering courses. He holds a B.S. in Electronics and Computer Science (from Galileo University, Guatemala) and an M.S. in Operations Research (from Galileo University, Guatemala). He also has extensive teaching experience in engineering mathematics, which has led him to receive multiple excellence teaching awards. His research area is engineering education, where he has led numerous projects to improve students' academic performance and motivation. Additionally, he collaborates with the SENACYT (Secretaría Nacional de Ciencia y Tecnología de Guatemala) as National Contact Point (NCP) in the CELAC (Comunidad de Estados Latinoamericanos) group to promote research infrastructures collaborations between countries in Latin America and the European Union (EU).

From Worksheets to Web Games: AI-Driven Gamification for Engaging Engineering Students in Mathematics

1. Introduction

Mathematics continues to represent one of the greatest academic challenges for engineering students across Latin America [1]. For many students, mathematical concepts are perceived as abstract, overly formal, or disconnected from real experience. This perception contributes to discouragement, low motivation, and weak study habits that become visible in early engineering courses such as calculus [2]. Even more, these attitudes often emerge long before students attempt their first course; they stem from prior negative experiences and the belief that mathematics is inherently difficult or inaccessible. As a result, students begin their university studies with psychological barriers that limit confidence, curiosity, and engagement.

For engineering programs, this represents a critical problem. Mathematics constitutes the cognitive foundation for modeling, problem-solving, and analytical reasoning [3]. Students with math anxiety or avoidance do not necessarily lack competence; rather, their fears interfere with the fast pace and rigor of university-level courses, making early steps feel overwhelming. A major factor behind this disconnection is the contrast between traditional instruction—typically procedural and worksheet-centered—and the dynamic, interactive digital environments where students spend much of their time [4]. Unless deliberate efforts are made to bridge this gap, motivation declines quickly, and students struggle to develop the fluency in mathematical thinking that engineering requires.

In response to these challenges, engineering education has progressively incorporated active learning methodologies that emphasize participation, reasoning, collaboration, and conceptual exploration, see e.g., [5]. Gamification has emerged as a particularly promising strategy. By embedding elements such as challenges, narrative structures, rewards, or instant feedback into learning activities, gamification increases students' sense of autonomy, competence, and engagement [6]. For example, tools like Kahoot, Quizizz, and Jeopardy-style exercises have shown that competitive, interactive formats can reduce students' fear of failure and shift the emotional experience of learning mathematics toward enjoyment and persistence.

However, these tools share an important limitation: although they energize the classroom, they rarely maintain engagement outside of it [7]. Their design relies heavily on multiple-choice formats and quiz templates, offering limited narrative depth or authentic gameplay. Students enjoy these tools, but they do not perceive them as true games, and the motivational effect diminishes quickly. At the same time, the creation of custom educational video games has historically required programming expertise or institutional resources that are beyond the reach of most instructors [8]. As a result, educators remain constrained: they recognize the value of game-based learning but still lack the means to implement more immersive experiences that align with their course syllabus.

The workflow presented in this paper emerged from two of practical instructional constraints. First, the instructor required a digital review activity that could be created and deployed quickly, without extended programming time, specialized software, or institutional infrastructure. Second, existing classroom practices relied primarily on short quiz-based tools or group activities, which—while effective during scheduled review sessions—offered limited continuity beyond the classroom.

The instructional objective was therefore to design an individual, browser-based activity that provided immediate feedback, supported repeated practice outside class hours, and remained accessible during short study intervals in the days leading up to an assessment. These constraints motivated the use of generative artificial intelligence (AI) as a development aid. Rather than replacing instructional design decisions, AI was employed to accelerate the transformation of existing review materials into interactive formats that could be rapidly generated, tested, revised, and accessed by students both during review sessions and independently.

In a calculus course, the instructor sought a review activity that could replace traditional worksheet-based sessions without relying on web- or team-based competitions. The objective was to design an activity that students could access from their own laptops or mobile devices, while incorporating a competitive element through visible scoring and shared leaderboards.

Instead of creating new instructional content, the starting point consisted of a conventional review guide containing standard mathematical exercises. Using generative AI tools, this static document was transformed into an interactive arcade-style experience that incorporated timers, scoring systems, lives, and immediate feedback. Initial game prototypes employed simple question-and-response mechanics, which later evolved into more dynamic formats such as shooting games and platform-style challenges. Throughout all iterations, the mathematical content remained unchanged; only the mode of interaction and presentation was redesigned.

In order to address the aforementioned limitations, this paper explores a novel approach: the use of AI to transform traditional mathematics worksheets into HTML5 [9] arcade-style video games that students can play directly in their web browser or even on smartphones. Instead of manually coding games, instructors can now use generative AI to create game prototypes that embed mathematical content into well-known gameplay structures such as shooter mechanics, platform levels, or mission-based challenges. This idea gives rise to a project we call *Fun Study Guides*: interactive learning resources that preserve mathematical rigor while leveraging familiar gameplay structures to reduce math anxiety, promote self-study, and motivate students to engage actively with the course material. In essence, the general objective of this project is to enable students to learn mathematics in an environment that feels recreational, while allowing instructors to retain full control over the mathematical structure.

The central contribution of this work is not the video games themselves, but the demonstration that generative AI can serve as a practical bridge between traditional instructional materials and interactive digital learning environments. This approach enables instructors to extend active learning beyond the classroom while preserving mathematical rigor and without requiring prior programming expertise. By preserving original exercises and embedding them directly into gameplay mechanics, this AI-assisted workflow aligns with students' digital habits while significantly lowering the barriers for instructors seeking to implement digital game-based learning activities [10]. The resulting tools are accessible, low-cost, and easy to deploy, while allowing instructors to retain full control over the mathematical structure, difficulty, and learning objectives of each activity.

The objectives of this paper are threefold. First, it presents a replicable AI-assisted workflow that enables instructors to convert existing review materials into customized web-based educational games aligned with specific course content. Second, it documents four case studies corresponding to core mathematics courses in the engineering curriculum, each illustrated through a functional arcade-style game deployed on an open-access platform. Finally, it offers practical recommendations for educators interested in adopting AI-assisted game design as a scalable strategy for enhancing motivation and persistence in engineering mathematics courses.

As a clarifying note, the terms *game* and *prototype* are used interchangeably in this paper to refer to browser-based instructional artifacts generated through the proposed workflow. The term *prototype* emphasizes its exploratory and design-oriented nature, whereas *game* refers to the student-facing interactive experience.

2. AI-Assisted Development of Curriculum-Aligned Games

The development of the games was based on the instructional materials already used across the mathematics courses sequence—specifically, the *study guides* and *review worksheets* prepared for midterm and final examinations. These assignments provide students with a structured summary of the theoretical content to be assessed, including key definitions and theorems, along with a collection of exercises typically organized into three categories: *(i)* conceptual understanding, *(ii)* procedural fluency, and *(iii)* application-based (word) problems. Answer keys are also provided so that students can verify their work. Grounding the game design process on these materials ensured that the resulting prototypes remained closely aligned with course content, learning outcomes, and assessment formats.

The design of the proposed educational games followed a structured and repeatable process that preserved pedagogical integrity while enabling technical feasibility. To move from static study guides to functional interactive prototypes, this project adopted a multi-stage workflow that begins with instructor-driven content selection and then progressively incorporates AI-assisted design, coding, and refinement. Each stage serves a distinct instructional purpose. These include maintaining fidelity to course objectives, translating mathematical tasks into meaningful gameplay interactions, and iteratively improving the resulting prototypes through testing and feedback. The stages of this development pipeline are described in detail below.

Stage 1: Selection and Preparation of the Problem Bank

The design process began with the selection of problems from the course *study guides* and *review worksheets*, which are typically distributed to students in PDF files. These materials already reflect the course scope, expected problems' difficulty, and assessment style, making them a suitable starting point for the development of interactive review tools. For each game, the instructor selected a subset of problems aligned with the topics covered in the upcoming midterm or final examination. It is worth emphasizing that this selection was driven by instructional priorities rather than gameplay considerations.

The selected problems were then compiled into a concise document that articulated the conceptual focus and computational demands of each task. No rewriting, simplification, or adaptation of the mathematical content was required; instead, the AI is explicitly instructed to preserve the original problem statements and correct answers as provided (see next steps). This problem bank served as the mathematical *blueprint* guiding the development of each game and ensured that the resulting prototypes were direct extensions of existing study materials, maintaining full alignment with course expectations while modifying only the mode of interaction.

Stage 2: Specification of Game Mechanics and Instructional Constraints

After selecting the problems, the instructor outlined the high-level design specifications for each game. These specifications described the intended genre (e.g., arcade shooter, platformer, or timed survival challenge), the core interaction model, and the basic rules governing scoring, lives, penalties, and pacing. The instructor also specified how each mathematical question would be embedded within the gameplay, for example as triggered events, level transitions, checkpoints, or short problem-solving tasks activated during collisions. The prompts were implemented using ChatGPT and Gemini, accessed through its web-based interface, with the instructor iteratively refining the instructions to guide code generation and maintain alignment with instructional objectives.

Although the exact prompts used in this project varied across prototypes, they followed a consistent structure designed to maintain instructor control and pedagogical clarity. Each prompt explicitly specified: *(i)* the target mathematical topics and learning outcomes; *(ii)* the problem bank to be incorporated; *(iii)* the intended game format, mechanics and narrative framing; (iv) the rules governing scoring, timing, lives, penalties, and feedback; and *(v)* technical constraints related to browser-based execution and device compatibility. An illustrative example of such a prompt is provided in Figure 1 to demonstrate how a typical calculus review worksheet can be translated into an HTML5 game prototype. This structured prompting approach ensured that AI-generated code served instructional goals first, using gameplay mechanics to support mathematical practice rather than to drive it.

Illustrative AI Prompt Used to Generate an Initial Game Prototype
Using the attached calculus review problems, generate an HTML5 arcade-style game playable in a web browser. The game should include a visible score counter, a timer, and a limited number of lives. Each correct answer increases the score, while incorrect answers reduce one life. Do not modify the mathematical problems or their solutions. Integrate the questions into gameplay as short challenges that appear during play. The final output should be a single HTML file with embedded CSS and JavaScript.

Figure 1. Illustrative example of an AI prompt used to transform a differential calculus review worksheet into an initial HTML5 game prototype. The prompt specifies instructional constraints, gameplay mechanics, and technical requirements.

Stage 3: AI-assisted Generation of HTML5 Prototypes

Based on the instructor-defined design specifications, the AI tool generated an initial HTML5 prototype that included: *(i)* the complete HTML layout and user interface (UI) components; *(ii)* Cascading Style Sheets (CSS) [11] for visual styling and basic animations; *(iii)* functional JavaScript [12] logic implementing player movement, collision detection, scoring mechanisms, timers, and question handling; and *(iv)* minimal Portable Network Graphics (PNG) assets. Rather than serving as a finalized product, this initial version functioned as a working draft that captured the intended gameplay loop and enabled rapid testing, feedback, and iterative refinement.

Stage 4: Human Refinement and Iterative Debugging

Building on the initial AI-generated prototypes from Stage 3, the instructor systematically reviewed and refined each game through successive iterations, maintaining full instructional oversight throughout. Although generative AI accelerated initial development, human judgment remained essential for identifying and addressing common issues that emerged during classroom-oriented prototyping.

Refinement focused on adjustments that improve both technical functionality and pedagogical quality, including: *(i)* correcting issues in movement physics, timing, and collision detection; *(ii)* reorganizing functions and removing redundant code to improve maintainability; and *(iii)* tuning difficulty, pacing, scoring, and feedback frequency to ensure an appropriate level of challenge and enough time for mathematical reasoning and problem solving. Additional refinements addressed interface elements that could distract students from the underlying mathematical content, prompting revisions to visual layout and animation intensity to ensure that problem statements and feedback remained central to the gameplay experience. In practice, these adjustments were implemented through successive refinements to the AI prompts rather than through manual code edits.

Finally, the instructor verified that all mathematical expressions were rendered accurately and confirmed that every question was correctly integrated into the game interface with consistent formatting across the prototype. Through this iterative, AI-supported cycle of review and adjustment, each prototype evolved into a stable, coherent learning tool. This stage positioned AI as an implementation aid, while instructional decisions remained under the instructor's control.

Stage 5: Integration of Competitive and Motivational Features

Once the gameplay mechanics and mathematical components were stable, a simple online leaderboard was integrated using a cloud-based JavaScript Object Notation (JSON) storage service [13]. This leaderboard allowed students to compare their scores across course sections in real time, introducing a social and friendly competitive layer to the review activities. By making performance visible beyond a single gameplay session, the leaderboard provided an incentive for students to replay the games in order to improve their scores, leading to repeated exposure to the same review content.

Stage 6: Deployment and Post-release Updates

Finally, the completed game prototypes were deployed through *itch.io* [14], a widely used platform for hosting browser-based indie games, allowing students to access them on laptops, tablets, or smartphones without installing additional software. Classroom implementation provided direct student feedback, which in turn guided a set of targeted refinements, including animation adjustments, bug fixes, and difficulty tuning. These iterative updates were applied throughout the course to improve performance, enhance usability, and ensure that each game remained functional and aligned with instructional objectives.

Figure 2 summarizes the six-stage, AI-assisted workflow described in this section, highlighting the iterative role of instructor oversight throughout the development process. Taken together, the workflow described across these six stages is intentionally lightweight, instructor-driven, and transferable. It does not require specialized institutional infrastructure, Learning Management System (LMS) integration, or prior programming or game development experience. Because the process builds directly on existing instructional materials and relies on natural-language interaction with a large language model, it can be readily adapted to other mathematics courses and extended to problem-based learning contexts in other STEM disciplines. By emphasizing the transformation of familiar instructional artifacts, such as study guides and review worksheets, rather than the development of polished, commercial-style games, this framework offers a practical, replicable workflow that instructors can adapt to their own pedagogical goals and course contexts.

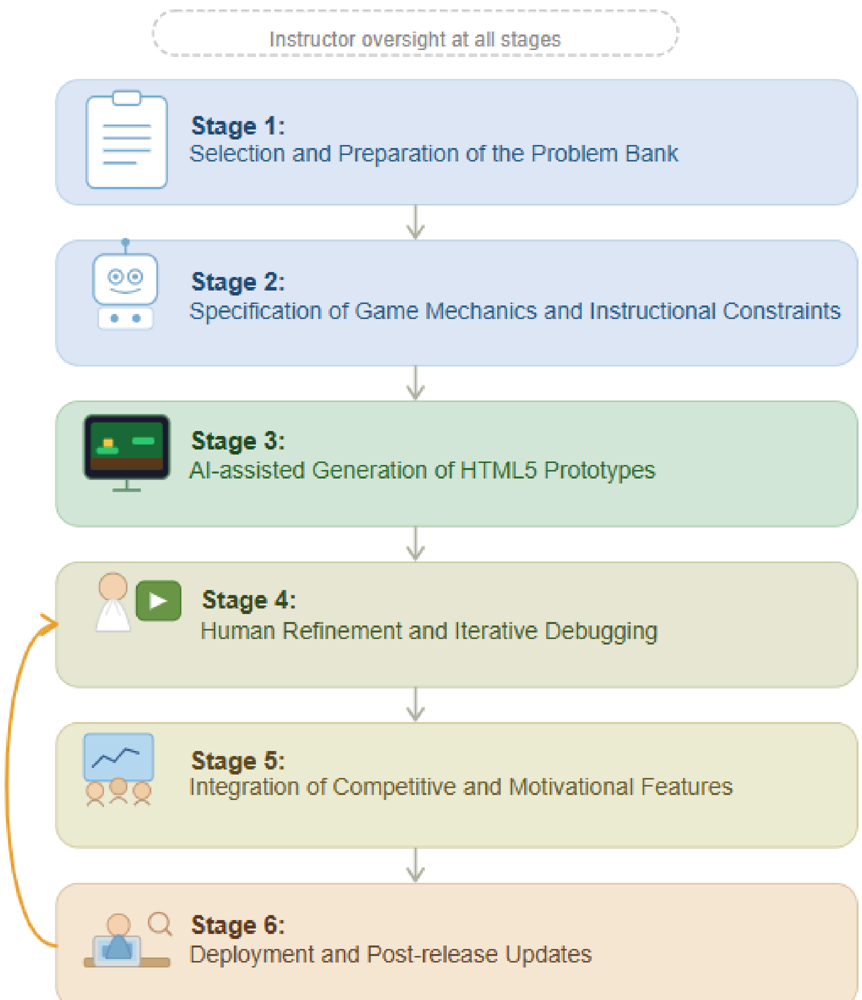

Figure 2. Overview of the AI-assisted workflow used to transform instructor-authored study guides or worksheets into interactive game-based review tools.

To complement the development workflow described above, this subsection briefly summarizes key technical decisions that supported the project's instructional goals. Rather than presenting low-level implementation details, the emphasis is on architectural and rendering choices that enabled rapid prototyping, iterative refinement, and consistent integration of mathematical tasks across multiple game genres.

Initial game prototypes were developed using a single file structure in which the HTML, CSS, and JavaScript code resided within one *index.html* file, accompanied by a small set of PNG assets. This compact architecture supported rapid experimentation and quick validation of core gameplay mechanics and the integration of mathematical content. As the number and complexity of prototypes increased, the project transitioned to a modular structure that separated styles, scripts, and assets into dedicated directories. This evolution improved maintainability, simplified debugging, and enabled the reuse of shared components, such as movement logic, user interface elements, and question banks, across multiple games.

To accommodate different types of mathematical problems and gameplay dynamics, two complementary rendering models were employed. Quiz-oriented games relied on standard HTML Document Object Model (DOM) elements (e.g., buttons, timers, and text inputs), which supported clear problem presentation and straightforward feedback. In contrast, action-oriented games, including shooters and platformers, used the HTML5 canvas element to enable continuous movement, collision detection, and smooth animation. Selecting the rendering model based on pedagogical intent ensured that mathematical content remained visible, accessible, and tightly integrated into each game’s core interaction loop.

3. Representative Game Prototypes Generated Using the AI-Assisted Workflow

To illustrate the scope of the AI-assisted prompting workflow described in Section 2, a set of representative game prototypes generated from conventional mathematics review worksheets is presented. These examples demonstrate how the same workflow can be applied across different mathematical topics, course levels, and instructional contexts by varying gameplay mechanics.

All prototypes were deployed as review tools within regular course sessions and were also made available asynchronously through the course web pages. During in-class review sessions, typically lasting approximately 50 minutes, students were encouraged to play the games repeatedly, experiment with different strategies, and attempt successive runs. Because the questions were randomly generated from predefined templates, students could replay the games without encountering identical problem sequences. Outside the classroom, the browser-based nature of the games enabled informal review at students' own pace and on their own schedule.

The prototypes were implemented across multiple courses and levels, including topics in algebra and trigonometry, differential and integral calculus of single and multivariable functions, and numerical methods. This implementation across both introductory and upper-level engineering mathematics courses highlights the workflow's adaptability to diverse curricular contexts.

Rather than treating each prototype as an isolated artifact, the games were organized into four gameplay families, each aligned with the cognitive demands of the mathematical content in the respective target course. This organization allowed the mechanics, pacing, and interaction model to be tailored to specific instructional goals while also facilitating iteration, code component reuse, and alignment with course syllabi. Each gameplay family is described in detail below.

- *Arcade-quiz Prototypes:* For the first prototype, we prioritized immediacy over complexity. This first family includes games that combine classic arcade aesthetics with rapid question–answer cycles to emphasize procedural fluency through repetition under time constraints. Short problems are presented sequentially, with performance reinforced through scoring systems, lives, timers, and immediate visual feedback. This format is particularly well-suited for topics such as limits, differentiation rules, integration techniques, or basic algebraic manipulation, where speed and operational proficiency play an important role in mastery. Immediate feedback after each response allows students to self-correct during gameplay and refine strategies across multiple attempts, as illustrated by the representative differential and integral calculus prototype shown in Figure 3.

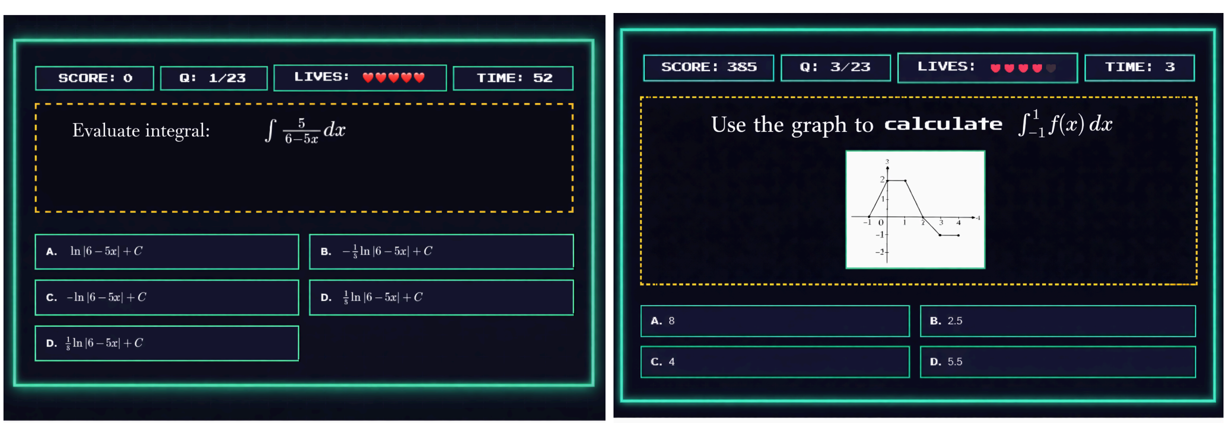

Figure 3. Arcade–quiz prototype generated from a calculus review worksheet, showing embedded questions, score tracking, and immediate feedback during gameplay.

- *Navigation-based Arcade Prototypes:* This category includes games in which students control movement through an obstacle-rich environment while solving embedded mathematical challenges. In these games, progression is directly linked to correct problem solving, i.e., successful responses unlock abilities, restore resources, or enable further advancement, while incorrect answers slow progress without abruptly ending the game. This structure requires students to alternate between spatial navigation and mathematical reasoning, supporting sustained attention and multi-step engagement, as incorrect responses result in delayed progress rather than simple point loss. A typical example of a game within this category, featuring player navigation combined with vector functions problems, is presented in Figure 4.

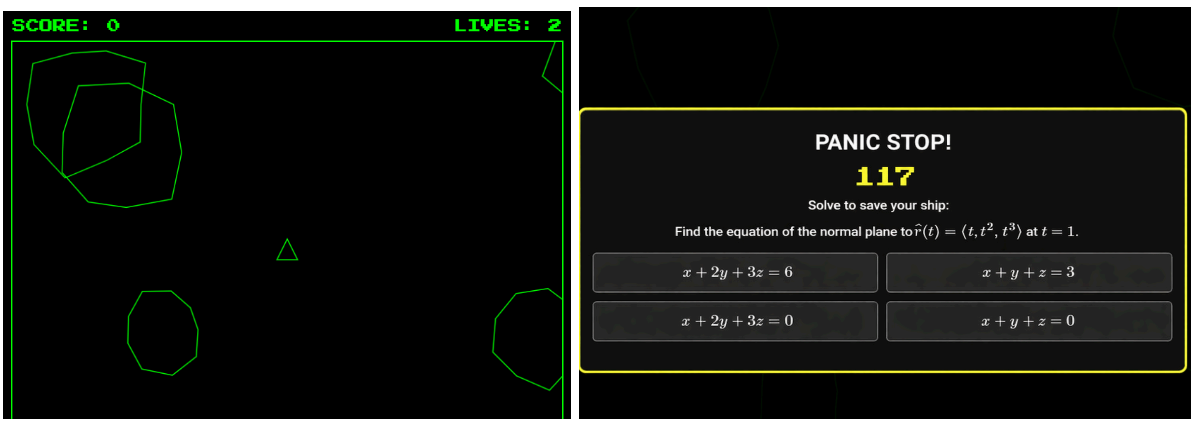

Figure 4. Navigation-based arcade prototype where progression depends on correctly solving embedded mathematical challenges.

- *Mission-based Prototypes:* The games in this family are designed to support higher-order learning objectives by prioritizing decision making and method selection over response speed. Progression is driven by the completion of structured objectives that require students to analyze the mathematical structure of each task and choose appropriate solution strategies. This design aligns naturally with the goals of upper-level mathematics courses, where understanding when and why particular methods apply is as important as performing the associated computations. An example of this category is shown in Figure 5, which presents a mission-based game developed for a numerical methods course. In this prototype, students are positioned as *agents* who advance through a sequence of *exploratory missions* by selecting appropriate numerical techniques, such as root finding, interpolation, or numerical integration. By integrating narrative and problem-based elements, the game emphasizes strategic reasoning and method selection, reinforcing conceptual understanding rather than time-pressured computation.

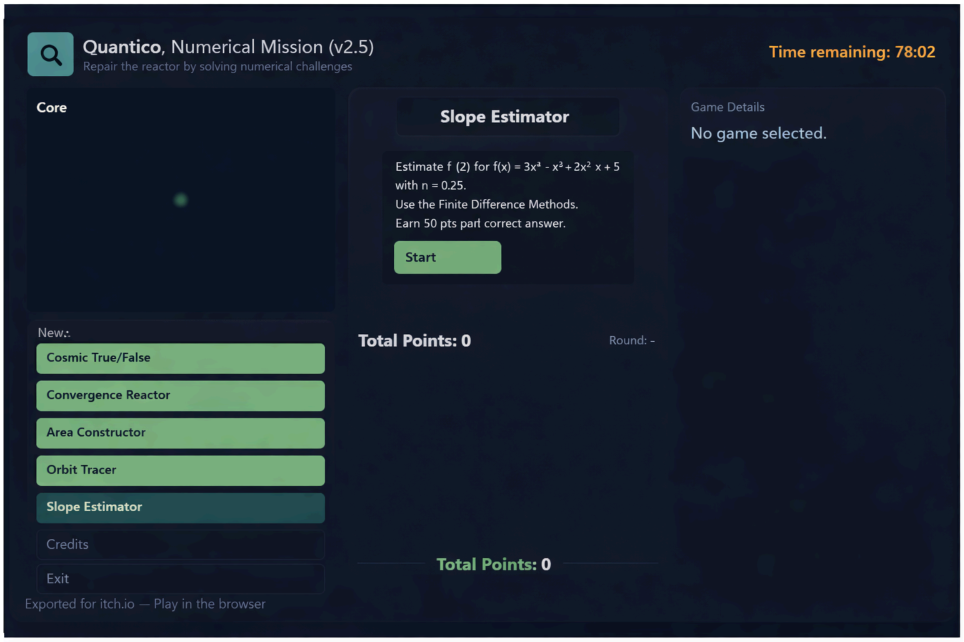

Figure 5. Mission-based game prototype illustrating narrative-driven progression in a numerical methods context.

- **Platform-style prototypes:** This family of games emphasizes progressive problem solving through exploration, in which students unlock new sections or overcome obstacles by correctly completing short, sequential tasks. Although individual challenges are intentionally concise, progression through the environment requires students to complete chains of *related problems*, supporting *multi-step reasoning* across successive game segments rather than isolated responses. The format is particularly effective for reinforcing foundational skills, as it combines frequent low-stakes practice with visible progress to promote persistence and reduce anxiety associated with incorrect responses. By allowing students to recover quickly from errors and continue advancing, the design encourages repeated engagement while gradually building confidence and procedural coherence in early mathematics courses. An example of this design is shown in Figure 6, which presents a platform-style prototype developed for an introductory mathematics course. In this implementation, the game targets analytical geometry through a sequence of interconnected tasks centered on parabolas. Students may first be asked to identify key elements such as the vertex, followed by subsequent prompts addressing the focus, directrix, or other defining features, with each correct response enabling further progression. In this way, the platform structure supports stepwise problem solving, where individual questions contribute to the construction and understanding of a complete geometric representation rather than functioning as isolated exercises.

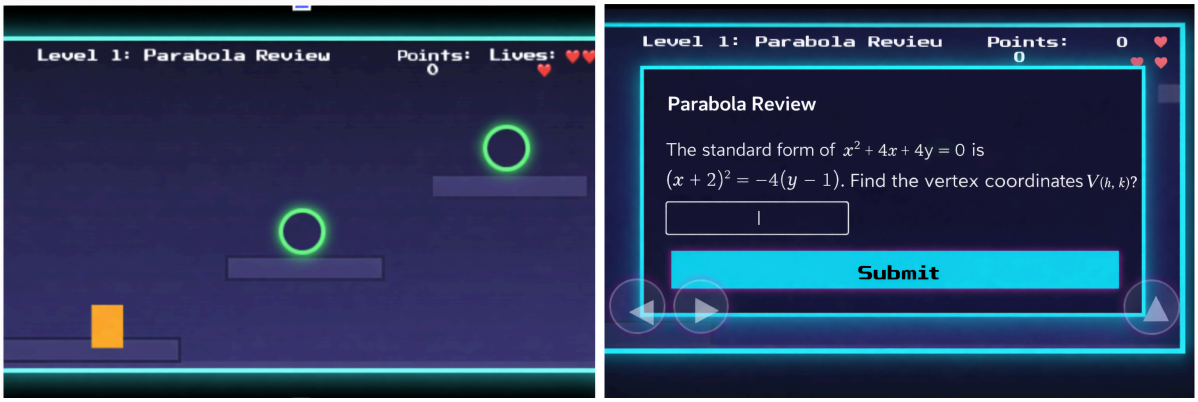

Figure 6. Platform-style prototype for introductory mathematics courses, where progression is tied to successful problem solving.

Notice that, across all gameplay families, the mechanics were intentionally kept simple to support quick, repeatable practice rather than extended gameplay experiences. These representative prototypes demonstrate that the proposed AI-assisted workflow is not limited to a specific game genre or course level; instead, it equips instructors with a flexible and replicable method for transforming existing instructional materials (not restricted to study guides or review worksheets) into interactive, browser-based study tools. By maintaining alignment with mathematical content and instructional objectives, the workflow supports repeated practice, immediate feedback, and student engagement. Despite differences in visual design and gameplay mechanics, all prototypes share common design principles, including student accessibility, opportunities for repeated practice, and sustained instructor control, which were consistently applied across courses and implementation contexts.

The following section discusses the broader instructional implications of this approach, including observed benefits, practical limitations, and considerations for integrating AI-assisted game design within engineering mathematics curricula.

4. Discussion

This paper positions generative AI as a practical design aid for instructors seeking to extend active learning beyond the classroom. The contribution of this work is intentionally designed and methodology-focused. Rather than reporting learning gains, engagement metrics, or comparative outcomes, the paper documents a replicable workflow (see Section 2) that enables the rapid transformation of traditional instructional materials (e.g., review worksheets) into gamified study tools. The representative prototypes presented in Section 3 illustrate how this workflow can be applied across mathematics courses at different levels and instructional contexts.

The proposed approach addresses well-known constraints in engineering education: limited instructor time and restricted access to specialized technical resources. By using the 'Prompt Problem' approach, a new paradigm in which solutions are built by formulating natural-language prompts for code-generating Large Language Model (LLMs) [15], instructors can retain full control over mathematical content and instructional goals while substantially reducing development effort. It is important to highlight that within this workflow, generative AI acts not as a pedagogical decision-maker but as an *implementation assistant* that accelerates prototyping and revision. This distinction is critical, as it preserves pedagogical responsibility with the instructor while using AI for efficiency.

Traditional review activities in mathematics courses often take one of two forms: in-class problem-solving sessions constrained by scheduled time, or take-home review worksheets assigned for independent practice. In the latter case, students frequently receive little or no immediate feedback, or they must seek guidance independently through external resources, including general-purpose AI tools, over which instructors have limited control. As a result, the feedback students receive can vary widely in quality and alignment with course expectations. In contrast, the video-game prototypes introduced in this paper provide continuous practice in which feedback is predefined by the instructor and delivered immediately within the activity itself. Moreover, because the games remain available online, students can revisit key concepts, solve practice problems, and self-assess their performance from their phones, tablets or laptops outside class hours, during short study intervals, commutes, or even informal review moments before assessments. From a design perspective, this shift extends review activities beyond fixed classroom sessions and transforms them into flexible, on-demand study tools that align more closely with students' everyday digital habits [16], while preserving instructor supervision.

An additional implication of this work is that effective game-based study tools do not require complex mechanics or extended development cycles. Across all gameplay families presented, the designs intentionally emphasize simplicity, immediate feedback, and close alignment with course content. The examples suggest that modest mechanics, when carefully aligned with instructional goals, are sufficient to create approachable review experiences. This observation aligns with prior research in educational technology, indicating that students are more likely to adopt and continue using tools that are easy to access and designed around clearly defined learning tasks rather than complex interfaces or mechanics [4].

Furthermore, our design aligns with established research validating simple mechanics for skill acquisition. Just as linear board games [17] and fast-paced classics like *Tetris* [18] successfully reinforce cognitive skills, our prototype leverages immediate arcade feedback to build procedural fluency. This confirms that this kind of model remains a powerful instructional tool [19], bridging the gap between rote repetition and the conceptual engagement seen in educational titles like Wuzzit Trouble [20].

From an instructional perspective, the proposed workflow supports experimentation and incremental adoption. Instructors can iterate quickly, refine difficulty or pacing through prompt adjustments, and deploy updated versions without disrupting existing course structures or assessment practices. This flexibility allows game-based elements to be explored for their pedagogical value without committing to extensive course redesign.

Finally, by building all games directly from instructor-authored materials, the proposed approach preserves alignment with course content and assessments. Problem selection and structure, mathematical notation, and solution accuracy remain instructor-defined, ensuring that the games function as extensions of established instructional practice instead of parallel or disconnected activities. In this sense, the workflow introduced in this work demonstrates how generative AI can be integrated into mathematics instruction in a controlled and purposeful manner, supporting the course syllabus without altering its underlying structure.

5. Conclusion and Future Work

This paper introduced an AI-assisted design workflow that enables instructors to convert classical mathematics assignments into browser-based games with minimal technical overhead. The contribution of this work is the articulation of a practical and replicable process through which generative AI supports the implementation of interactive learning tools that complement established instructional practices and materials.

Our future work follows two complementary directions. The first focuses on extending the design capabilities of the workflow itself. Planned efforts include: *(i)* refining prompt strategies to support adaptive difficulty and personalized feedback, *(ii)* exploring collaborative and multi-user gameplay formats, and *(iii)* expanding the range of gameplay structures that can be generated. These extensions aim to broaden the instructional design space available to instructors without increasing technical complexity. On the other hand, the second direction concerns a systematic evaluation of instructional impact, which was beyond the scope of this study. More precisely, our future research will involve developing frameworks to examine how different game designs, interaction patterns, and usage contexts relate to student engagement, motivation, and learning outcomes. Such analysis is necessary to move beyond design feasibility and toward an evidence-based understanding of whether, when, and how AI-assisted game-based tools are effective in mathematics education.

In summary, this work positions generative AI not as a substitute for pedagogical expertise but as a practical implementation partner that allows instructors to rapidly prototype, iterate, and deploy interactive learning tools based on their own course materials. As a result, the workflow introduced in this paper expands the instructional design space for engineering mathematics education while remaining aligned with course syllabi and responsive to contemporary student learning habits shaped by short, interactive, and on-demand digital engagement.

References

[1] B. V. Dorneles, "Mathematical learning and its difficulties in Latin-American countries," in International Handbook of Mathematical Learning Difficulties: From the Laboratory to the Classroom, Cham: Springer, 2019, pp. 201–212.

[2] N. Akhter and N. Akhter, "Learning in mathematics: Difficulties and perceptions of students," J. Educ. Res., vol. 21, no. 1, 2018.

[3] Y. Abd Algani, "Solving mathematics anxiety, lack of confidence and negative attitude with artificial intelligence models: Insights from stakeholders," *J. Math. Educ. Teach. Pract.*, vol. 5, no. 2, pp. 89–100, 2024.

[4] A. Körei, S. Szilágyi, and Z. Török, "Integrating didactic games in higher education: Benefits and challenges," *Teach. Math. Comput. Sci.*, vol. 19, no. 1, pp. 1–15, 2021, doi: 10.5485/TMCS.2021.0517.

[5] P. Doolittle, K. Wojdak, and A. Walters, "Defining active learning: A restricted systematic review," *Teach. Learn. Inq.*, vol. 11, 2023.

[6] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining 'gamification'," in *Proc. 15th Int. Acad. MindTrek Conf.*, Tampere, Finland, 2011, pp. 9–15.

[7] Y. Abd Algani, "Enhancing mathematics education with ChatGPT-4: Personalized problem-solving and consistent learning," *J. Math. Educ. Teach. Pract.*, vol. 5, no. 1, 2024.

[8] A. De Gloria and R. Veltkamp, Eds., *Games and Learning Alliance: 4th International Conference, GALA 2015, Revised Selected Papers*. Cham: Springer International Publishing, 2016, doi: 10.1007/978-3-319-40216-1.

[9] B. Lawson and R. Sharp, *Introducing HTML5*. Berkeley, CA: New Riders, 2011.

[10] M. Prensky, "Digital game-based learning," *Comput. Entertain.*, vol. 1, no. 1, p. 21, 2003.

[11] H. W. Lie and B. Bos, *Cascading Style Sheets: Designing for the Web*. Boston, MA: Addison-Wesley Professional, 2005.

[12] D. Flanagan and P. Matilainen, *JavaScript*. Madrid: Anaya Multimedia, 2007.

[13] L. Bassett, *Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON*. Sebastopol, CA: O'Reilly Media, 2015.

[14] Itch Corp., "itch.io." [Online]. Available: https://itch.io. [Accessed: Jan. 20, 2026].

[15] P. Denny, J. Leinonen, J. Prather, A. Luxton-Reilly, T. Amarouche, B. A. Becker, and B. N. Reeves, "Prompt problems: A new programming exercise for the generative AI era," in *Proc. 55th ACM Tech. Symp. Comput. Sci. Educ.*, vol. 1, Portland, OR, USA, 2024, pp. 296–302.

[16] A. Johri, H. J. Teo, J. Lo, M. Dufour, and A. Schram, "Millennial engineers: Digital media and information ecology of engineering students," *Comput. Hum. Behav.*, vol. 33, pp. 286–301, 2014.

[17] G. B. Ramani and R. S. Siegler, "Promoting broad and stable improvements in low-income children's numerical knowledge through playing number board games," *Child Dev.*, vol. 79, no. 2, pp. 375–394, 2008.

[18] M. S. Terlecki, N. S. Newcombe, and M. Little, "Durable and generalized effects of spatial experience on mental rotation: Gender differences in growth patterns," *Appl. Cogn. Psychol.*, vol. 22, no. 8, pp. 996–1013, 2008.

[19] F. Ke, "A qualitative meta-analysis of computer games as learning tools," in *Handbook of Research on Effective Electronic Gaming in Education*, R. E. Ferdig, Ed. Hershey, PA: IGI Global, 2009, pp. 1–32.

[20] H. Pope and C. Mangram, "Wuzzit Trouble: The influence of a digital math game on student number sense," *Int. J. Serious Games*, vol. 2, no. 4, pp. 5–22, 2015.