

From Vibe Coding to Playable Educational Escape Rooms: Extending Room2Educ8 for AI-assisted Prototyping

Panagiotis Fotaris¹ and Theodoros Mastoras²

¹University of Brighton, UK

²University of Macedonia, Thessaloniki, Greece

P.Fotaris@brighton.ac.uk

mastoras@uom.edu.gr

Abstract: Educational escape rooms (EERs) can support active, collaborative and problem-based learning, but developing digital and hybrid versions can create technical and interaction-design challenges for educators. Generative AI has been used to support EER development through persona creation, puzzle ideation, hint generation and debriefing, yet its role in implementing established designs as playable artefacts has received less attention. This study investigates AI-assisted prototyping within Room2Educ8, a Design Thinking framework for learner-centred EER development. An exploratory qualitative multiple-case study involved 23 postgraduate User Experience Design students working in four groups. All groups designed an EER as assessed coursework and adopted vibe coding during prototyping, although its use was neither required nor demonstrated in class. They used natural-language prompts with AI-assisted coding tools to create and revise software. Data sources included assessment portfolios, playable artefacts, project demonstrations, group viva responses, retained prompts, screenshots, individual reflections and testing records. Across the cases, AI-assisted coding was introduced after target audiences, learning objectives, narratives and puzzle structures had been established. It was used to implement interfaces, answer validation, feedback, hints, timers, navigation and transitions between digital and physical tasks. Playtesting identified problems involving ambiguous instructions, alternative valid answers, feedback, hint scaffolding, progression and accessibility. Broader challenges concerned behavioural specification, control over generated outputs, pedagogical alignment, documentation and consistency across generated media. The findings show how AI-assisted coding supported the implementation and revision of interactive EERs, although development time, effort and learning outcomes were not measured. The paper proposes Room2Educ8-AIP, a preliminary extension comprising seven iterative actions within the Prototype stage: specify, generate, inspect, revise, playtest, align and document. The extension offers educators and learning designers a structured way to integrate AI-assisted coding into EER development while maintaining design control, pedagogical alignment and clear documentation of implementation decisions.

Keywords: Educational escape rooms, Game-based learning, Vibe coding, Generative AI, AI-assisted prototyping

1. Introduction

Educational escape rooms (EERs) are puzzle-based learning activities in which participants complete a mission by discovering clues, solving challenges and applying relevant knowledge or skills within a defined time frame. They adapt escape room conventions for educational purposes, with the game structure designed to support learning objectives and the needs of a target audience (Fotaris & Mastoras, 2019). Although the term “escape room” is retained, contemporary escape games often involve broader missions, such as solving a mystery or preventing a crisis (Neubig, 2024).

EERs have been used across educational contexts, with growing interest in online, chatbot-based and hybrid formats (Barra et al., 2024; Wang et al., 2025). Applications span healthcare, computing, STEM and library education, where EERs are associated with active learning, collaboration, problem-solving and engagement (Quek et al., 2024; Ugo et al., 2025). Such outcomes depend on alignment between educational purpose, player experience, puzzle structure, support and evaluation (Gordillo et al., 2025). Technology-mediated EERs add technical and interaction-design demands because puzzles, feedback and progression must function reliably across software and physical-digital components (Barra et al., 2024). This burden can limit adoption when educators must move from an idea to a playable activity without specialist design or technical support (Pornsakulpaisal et al., 2023; Veldkamp et al., 2020).

A range of EER frameworks and toolkits has been proposed to structure design work. This paper focuses on Room2Educ8, a Design Thinking framework for learner-centred EER development (Fotaris & Mastoras, 2022), and examines how AI-assisted coding tools can be integrated into its prototyping stage. Previous work has explored generative AI in Room2Educ8-based EER design, including the production of learner personas, puzzle ideas, hints and debriefing plans (Fotaris et al., 2023). However, AI was used mainly as a design assistant. Less attention has been given to the implementation stage, where a specified EER design is turned into an interactive prototype.

In this paper, generative AI refers broadly to tools that generate text, images, media or code. AI-assisted coding refers specifically to using such tools to generate, revise or debug software. Vibe coding refers more narrowly to a natural-language-led workflow in which creators describe an application's intended behaviour and interact with code-generating models to produce and modify working software (Sarkar & Drosos, 2025). AI-assisted prototyping describes the wider design activity surrounding this workflow, including specification, inspection, playtesting, alignment and documentation.

Such work still requires human review, testing and critical oversight of generated outputs (Du et al., 2026; Lee et al., 2025; Pellas, 2026). Room2Educ8-AIP applies these practices to the specific demands of EER design by connecting generated behaviours, such as answer validation, feedback, hinting and progression, to puzzle logic, intended player reasoning and learning objectives. Without this connection, a functioning prototype may still contain weak puzzle logic, unclear feedback or poor pedagogical alignment.

The empirical setting is a postgraduate coursework task in which four student groups designed EERs using Room2Educ8. Vibe coding was not part of the brief, but all groups adopted it during prototyping. Their submissions show how a structured EER design process operated when AI-assisted coding became part of implementation. The study was guided by three research questions:

RQ1. How did students integrate vibe coding into a Room2Educ8-based EER design process?

RQ2. What opportunities and challenges emerged when vibe coding was used to prototype digital and hybrid EER artefacts?

RQ3. How can Room2Educ8 be extended to support AI-assisted EER prototyping while preserving design control and learning alignment?

The paper proposes Room2Educ8-AIP as an extension of Room2Educ8. It structures AI-assisted work in the Prototype stage around seven actions: specify, generate, inspect, revise, playtest, align and document. The proposal is informed by recurring practices and shortcomings identified across the four student submissions.

2. Background

2.1 Room2Educ8 as a Design Thinking Framework for EER Development

Room2Educ8 uses Design Thinking to organise learner-centred EER development around research, educational problem definition, scenario and puzzle design, prototyping and evaluation. It can be adapted across subjects, class sizes and physical, digital or hybrid formats, while allowing designers to revisit earlier decisions as constraints, puzzle logic and playtest findings become clearer. The framework was developed and internally evaluated with 104 postgraduate students who created 26 EERs between 2018 and 2022; the findings supported its clarity, usability and applicability across physical, digital, and hybrid formats (Fotaris & Mastoras, 2022).

Room2Educ8's Prototype and Test activities already support iterative development, but the framework predates the recent use of natural-language-led AI-assisted coding. These tools can generate interactive artefacts from natural-language instructions, but designers must still specify intended behaviour, inspect and revise the generated result, resolve errors and test the player experience. Room2Educ8 provides no detailed guidance for these implementation activities. A visually convincing prototype may nevertheless conceal incorrect answer validation, unclear feedback or misalignment with puzzle logic and learning objectives.

Figure 1 shows the original Room2Educ8 process. Prototyping and playtesting sit within a wider cycle of documentation and evaluation, so implementation is treated as part of iterative design work. Software behaviour, interface flow and feedback remain connected to learner research, learning objectives, scenario design and puzzle structure.

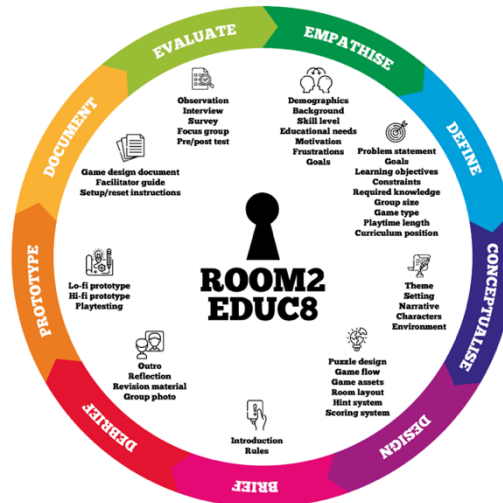


Figure 1: Original Room2Educ8 process, adapted from Fotaris and Mastoras (2022)

2.2 Generative AI in EER Design and Production

In EER research, generative AI is being used both as a design-support tool before play and as an interaction mechanism during play. In the design phase, AI can help prepare puzzles, story material, hints and feedback (Fotaris et al., 2023; Stojšić, 2024). During gameplay, it can support chatbot-mediated interaction, adaptive guidance or non-player characters (Rühl & Thurm, 2025; Wang et al., 2025). In both cases, the designer's role shifts towards selecting, revising and checking generated outputs.

Generated puzzles, hints and feedback need pedagogical as well as technical review. Puzzles should draw on the intended knowledge, hints should preserve enough challenge, and feedback should help learners understand the result of their actions. Digital escape games also require checks around answer validation, progression logic and system consistency. Work on AI-generated educational escape games has stressed the need for human oversight, testing and pedagogical alignment when generative tools are used to create game content (Kara et al., 2025). More broadly, gamified mechanisms do not produce educational benefits automatically; their effects depend on pedagogical integration and context (Papadakis & Karakose, 2025).

Existing studies provide examples of AI-supported content creation and AI-mediated play, but give less attention to AI-assisted implementation: translating an EER design plan into working screens, answer handling, feedback, hints and progression logic. Pellas (2026) and Rodríguez-Rivera et al. (2025) offer the closest empirical parallels because they examine learners creating games or escape rooms with generative AI. Du et al. (2026), a preprint on vibe-coded XR prototyping, illustrates the role of prompt specification, generated code, human review and rapid testing in implementation. These studies help identify relevant production practices, while leaving open the question of whether the resulting artefacts support learning effectively. In an EER, the implemented interaction must also support the intended puzzle logic and learning experience.

2.3 Novice EER Creators and Student-generated EERs

EERs are often designed by educators for learners, but recent work also treats escape-room creation as a student design task or assessment artefact (Geissler, 2025; Rodríguez-Rivera et al., 2025; Solano & Wen, 2025). In this role, students must interpret a subject, define an audience, connect puzzles to learning goals and revise their designs when testing reveals problems. The resulting EER becomes both a learning product and a record of its development.

Student-generated EERs can support design reflection when learners work within clear constraints, test their ideas with others and document successive decisions (Rodríguez-Rivera et al., 2025). As part of this process, students may produce design documentation alongside the playable artefact, showing how learner research and educational goals were translated into puzzles, interactions and feedback.

The present study focuses on postgraduate UX students who used vibe coding to turn EER design plans into playable artefacts. Each group submitted a coursework portfolio documenting the Room2Educ8 process and the development of its EER prototype. Students indicated that they had no previous game-design or EER-design

experience. They are therefore described as novice EER creators; this description does not imply that they lacked broader technical or UX experience, which was not systematically measured.

3. Method

An exploratory qualitative multiple-case design was used, with each of the four group projects treated as a case. Twenty-three postgraduate UX students worked in groups of five or six on an assessed Design Thinking task at a British university. Groups selected one of three broad themes (security awareness, library orientation or sustainability) and chose a format appropriate to their target audience and learning goals.

EER design was introduced through a three-hour workshop combining instruction, discussion, practical activities and opportunities to play physical and digital escape games. Throughout the semester, teaching on Design Thinking was interspersed with group activities through which students applied each stage to their developing projects. Generative AI was permitted for assistive use, subject to disclosure of tools and prompts and critical checking of outputs. AI-assisted coding entered the projects through the groups' own initiative, without classroom demonstration or instruction. Students indicated that they had no previous game-design or EER-design experience and no prior familiarity with Replit, Lovable or Claude, although some had used ChatGPT for general tasks. Previous coding and front-end development experience was not systematically measured.

The primary dataset comprised all four assessment portfolios. Each contained documentation of user research, problem definition, ideation, low- and high-fidelity prototyping, user testing, iteration, accessibility and usability considerations, educational evaluation, individual reflections and generative-AI use. Each project was also examined during a one-hour demonstration of the submitted EER and a subsequent one-hour group viva. Students were asked about the AI tools used, their reasons for using them, the resulting outputs, and the advantages and limitations they encountered. The demonstrations and discussions supplemented and clarified the written evidence. All four portfolios and submitted prototypes were available for analysis, although the amount of retained prompt evidence and testing documentation varied between groups. Complete platform-exported prompt histories were not available, so claims about the development process were limited to the evidence contained in the portfolios, demonstrations and viva responses.

The group project was treated as the primary unit of analysis, with prompts, screenshots, prototype features, testing records, group accounts and individual reflections used as evidence within each case. Initial analysis was guided by the research questions, while allowing implementation practices and challenges to emerge through comparison of the four cases. Evidence was first reviewed within each project and then compared across the cases to identify shared patterns, differences and shortcomings. Direct evidence from prototypes, screenshots and retained prompts was distinguished from retrospective accounts in portfolios, reflections and viva responses. The resulting patterns were subsequently interpreted in relation to the Room2Educ8 Prototype stage and relevant prototyping literature. The seven actions were proposed where the cases revealed either a recurring implementation practice that should be made explicit or a recurring problem requiring further guidance. The first author conducted the initial analysis, while both authors discussed the interpretation of the findings and the development of the proposed Room2Educ8-AIP actions.

The projects were originally completed as assessed coursework. Permission to use the submissions for research and publication was obtained after marks had been confirmed, and identifying information was removed. Student-led user research and testing followed the University's low-risk ethics process for coursework projects.

4. Findings

Three patterns emerged across the four cases: vibe coding entered the projects during prototype implementation; it enabled design plans to be developed into interactive artefacts that could be tested and revised; and it introduced challenges concerning specification, design control, documentation and learning alignment. The four prototypes addressed digital footprint awareness, sustainable fashion, workplace cybersecurity and library induction. Three combined digital components with physical activities, with the cybersecurity EER operating entirely online.

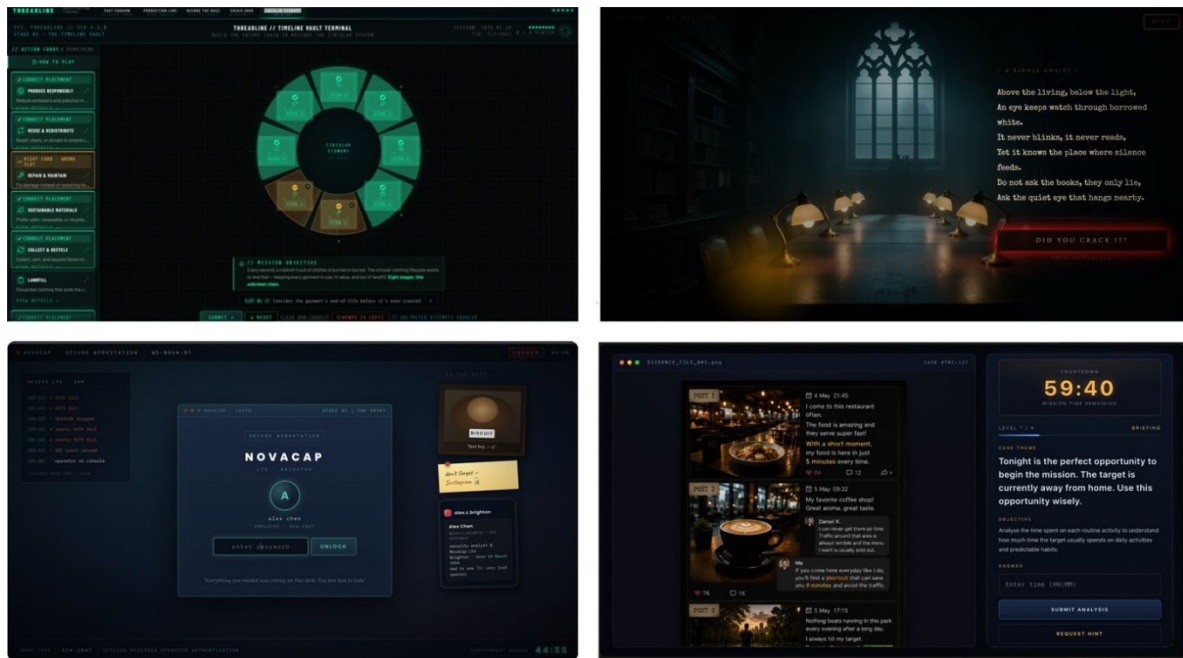


Figure 2: Examples of student-produced EER prototypes across the four topics using vibe coding

4.1 Vibe Coding in the Prototyping Process

Across the four projects, vibe coding entered the design process after groups had defined their educational context, target audience, learning goals, narrative direction and puzzle structure. It was used when these design decisions had to be developed into playable or demonstrable artefacts. Portfolio records and project demonstrations documented its use in interface construction, progression logic, answer handling, feedback, hints and transitions. The fully digital cybersecurity project depended on a browser-based interface, while the hybrid projects used digital components to coordinate or support physical activities.

Prompting became an additional design activity through which groups expressed interface content, navigation, accepted answers, feedback, hints, failure states and progression rules. Wireframes, game flows, puzzle descriptions and visual references provided implementation targets against which generated outputs could be inspected. Where descriptions of puzzle behaviour or player flow were less precise, interfaces could appear complete while still requiring correction.

4.2 Opportunities for Novice EER Production

All four groups developed interactive versions of their EER concepts. The resulting artefacts implemented features such as staged progression, immediate feedback, timers, answer validation and transitions between digital and physical activities. Their interfaces also used typography, colour, layout, transitions and narrative framing to communicate the intended atmosphere and player journey during testing and presentation.

The portfolios and demonstrations provided several examples connecting testing findings to subsequent revisions. Digital-footprint testers requested clearer and multiple hints, leading to revisions to the instructions and hint system. Sustainable-fashion testing revealed two semantically valid word orders, so the group accepted both and added further guidance. Workplace-cybersecurity testing exposed problems with capitalisation checks and the number of permitted attempts; a retained prompt requested case-insensitive validation and increased the limit from three attempts to five. Library-induction testing identified confusion around numbers, citations and feedback, leading to changes in text, layout, buttons, progression screens and system responses.

Creating interactive prototypes also required groups to make implementation decisions explicit. They had to determine what counted as a correct answer, how feedback should appear, when a stage should unlock and how players would recognise the next action. Testing the implemented behaviour made unclear rules, missing feedback and weak progression more visible. These cases demonstrate how AI-assisted coding supported the production and revision of interactive prototypes, although development time and effort were not measured.

4.3 Challenges for Design Control and Learning Alignment

Maintaining control over generated behaviour was a recurring challenge. EER interfaces must accommodate puzzle uncertainty, answer checking, feedback, hinting, progression and failure states. Prompts that concentrated on surface appearance could produce an attractive interface without adequately implementing the intended puzzle behaviour.

Several weaknesses were familiar EER-design problems whose origins could not be attributed specifically to AI-assisted coding, including ambiguous clues, uneven difficulty and insufficient scaffolding. Implementing an interaction also provided no assurance that it required the intended knowledge or skill. Groups therefore needed to check each working feature against its learning objective, puzzle rule and intended player experience.

Hint design illustrated this issue. Some prototypes initially offered limited support, while later versions introduced progressive or context-sensitive guidance. Effective implementation required attention to the timing and content of hints, the preservation of productive challenge and their connection to the reasoning or knowledge elicited by the puzzle.

Documentation varied across the four cases. Prompt records showed what students asked the tools to produce and, in some instances, how the prototypes changed. They did not always explain the reasoning behind accepting or revising an output. Documentation that connects prompts and generated outputs to intended puzzle behaviour, testing evidence and final design decisions would make this reasoning easier to inspect and reuse.

Generated media introduced a related consistency problem. Images, video and cutscenes that worked individually did not always remain coherent across the complete prototype. Character appearance, visual style or narrative transitions could vary between scenes. These cases showed the need to review generated assets within the complete player journey as well as individually.

5. Room2Educ8-AIP: An AI-assisted Prototyping Extension

Room2Educ8 provides a learner-centred process for EER development. The four cases showed that AI-assisted coding entered this process when established design decisions had to be implemented as playable interactions. This required groups to specify intended behaviour, inspect generated outputs, correct implementation problems, test the resulting interactions and check their alignment with the learning objectives. Room2Educ8-AIP incorporates these activities into the Prototype stage as a tool-agnostic iterative loop. Tool selection should also consider institutional access, cost, data protection and available technical support.

The proposed extension retains connections with the Design, Define and Document stages of Room2Educ8. Problems identified during implementation may require technical corrections or revisions to the puzzle structure, learning objective, narrative or documentation. Figure 3 presents seven actions: specify, generate, inspect, revise, playtest, align and document. These actions may be repeated whenever a designer creates or changes an element such as a digital lock, feedback screen, hint system, timer, branching path or transition between digital and physical tasks.

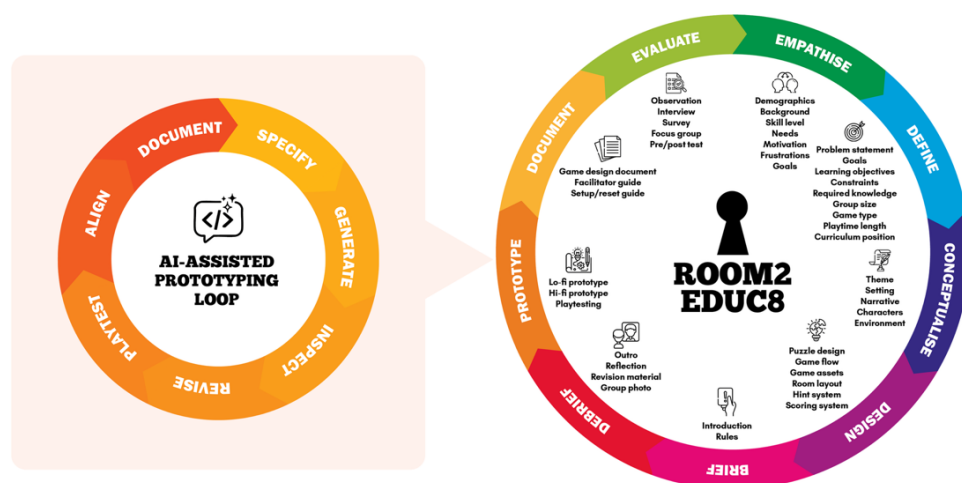


Figure 3: Room2Educ8-AIP extension

Specification establishes the intended behaviour before generation. For example, instead of asking an AI-assisted coding tool to “make a puzzle screen”, a designer could write: “Create a puzzle screen where players enter a four-digit code. Accept only 2486 as correct. If the answer is wrong, show feedback asking players to re-check the clue without revealing the answer. After three wrong attempts, display a first-level hint. When the correct code is entered, unlock the next mission screen.” This kind of prompt makes answer handling, feedback, hinting and progression explicit before generation. The generated component can then be inspected against these requirements. Mismatches can be addressed through revised prompts, simplified interactions, regenerated content or direct implementation changes.

Playtesting examines whether players understand the task, follow the intended reasoning and progress successfully. Alignment checks whether the implemented puzzle requires the intended knowledge or skill and remains consistent with the narrative. Throughout the loop, documentation records prompts, outputs, identified problems, revisions, testing findings and the reasons for key design decisions.

Room2Educ8-AIP is presented as a preliminary extension derived from the four cases and relevant prototyping literature. Future studies should examine its application across other cohorts, educational contexts and AI-assisted coding tools.

Table 1: Derivation and application of the Room2Educ8-AIP actions

Action	Finding that informed it	What the designer should do	Evidence to retain
Specify	Prompts sometimes described appearance more clearly than puzzle behaviour	Define the player action, accepted answers, feedback, hints, failure conditions and progression rule	Behaviour brief, puzzle flow and initial prompt
Generate	Groups used AI-assisted coding to convert established designs into interactive components	Use an AI-assisted coding tool to create or modify one defined prototype component	Prompt, generated output, tool used and date
Inspect	Generated interfaces could appear complete while containing behavioural errors or inconsistencies	Check puzzle rules, navigation, answer handling, feedback, accessibility and visual consistency	Inspection notes, screenshots and identified mismatches
Revise	Answer handling, hints, navigation and progression frequently required correction	Address specific problems through revised prompts, regeneration, content changes or direct editing	Revised prompt, changed output and reason for the change
Playtest	Testing revealed ambiguous instructions, valid alternative answers and progression problems	Observe whether players understand the task, follow the intended reasoning and progress successfully	Observations, user comments and resulting revisions
Align	Technically functional interactions did not automatically support the intended learning	Check that solving the puzzle requires the intended knowledge or skill and remains consistent with the narrative and player experience	Learning–puzzle mapping and alignment decisions
Document	Portfolio records did not always explain why outputs were accepted or changed	Connect prompts, outputs, revisions and testing evidence to the reasons for key design decisions	Prompt log, revision record, testing evidence and design rationale

6. Discussion and Future Work

This paper contributes to EER research by examining how established design plans are developed into playable prototypes. Previous work on Room2Educ8 has focused on learner-centred EER design, while studies of generative AI have largely considered support for activities such as persona development, puzzle ideation, hint generation and debriefing. The four cases extend this discussion to AI-assisted implementation. They show that designers must express intended interaction clearly enough for generated components to be inspected, tested and revised.

This is particularly important in EERs because small implementation decisions can affect puzzle fairness, usability and learning alignment. Answer validation, hint timing, feedback, progression rules and transitions between digital and physical activities influence how players encounter the educational content. A prototype may appear complete while still containing ambiguous rules, weak feedback or interactions that do not require the intended knowledge. Room2Educ8-AIP responds to these risks by making AI-assisted implementation an explicit and reviewable part of the Prototype stage.

Testing across the four cases identified design problems and informed changes to instructions, hints, answer handling, interface elements, feedback and progression. However, testing procedures, participant numbers and instruments differed across the projects. The findings therefore support claims about the design issues revealed through playtesting, but do not provide comparable evidence of learning effectiveness or a common estimate of learning gain.

When AI-assisted prototyping is introduced into coursework or professional development, it should be planned from the outset. Expectations should address institutional access, cost, data protection, copyright, accessibility, technical support and documentation. The voluntary adoption of paid or token-based tools in some projects shows that access conditions should not be left entirely to individual students. Educators should also consider how implementation work is distributed within groups and how evidence of prompting, inspection and revision will be recorded and assessed.

The study examined four assessed group projects from one postgraduate UX cohort. Participants were novice EER creators rather than educators designing for their own teaching contexts, and their prior coding experience was not systematically measured. Portfolio detail varied, and implementation was often concentrated among a small number of group members, so the findings concern group outputs rather than uniform individual skill development. The study also lacked a non-AI comparison, a standardised measure of development time and a common instrument for evaluating learning outcomes.

The authors' involvement in developing Room2Educ8, and the first author's role in leading the module, provided detailed knowledge of the framework and assignment context but may also have influenced the interpretation of evidence supporting an extension.

Future research should introduce Room2Educ8-AIP before development begins and examine how educators, pre-service teachers and learning designers use each action across different subjects and settings. Studies could investigate whether the extension improves behavioural specification, revision traceability, learning alignment and reuse. Comparisons with non-AI workflows could examine differences in development time, effort, accessibility and skill acquisition. Further work should evaluate the resulting EERs with common measures of usability, player experience and learning.

Overall, the cases show that AI-assisted implementation involves more than generating a functioning interface. Designers must retain control over puzzle behaviour, player guidance, accessibility, documentation and educational purpose. Room2Educ8-AIP makes these responsibilities visible within Room2Educ8 and offers designers a structured process for developing, reviewing and evaluating AI-assisted EER prototypes.

Acknowledgement

For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

Ethics Declaration: Student-led user research and testing followed the University's established process for low-risk coursework. Permission for the research use of submitted work was sought only after assessment and marking had been completed. Students were informed that participation was voluntary and that refusal would not affect their marks or academic standing. Identifying information was removed from the materials used in the study.

References

- Barra, E., López-Pernas, S., Gordillo, A., Pozo, A., Muñoz-Arcenales, A., & Conde, J. (2024). Empowering Database Learning Through Remote Educational Escape Rooms. *IEEE Internet Computing*, 28(1), 18-25. <https://doi.org/10.1109/MIC.2023.3333199>
- Du, R., Hersh, B., Li, D., Numan, N., Qian, X., Chen, Y., Zhou, Z., Chen, X., Ren, J., Bettridge, R. T., Chen, X. A., Faruqi, F., Toh, S., & Kim, D. (2026). Vibe Coding XR: Accelerating AI + XR Prototyping with XR Blocks and Gemini. *arXiv*, Article arXiv:2603.24591. <https://arxiv.org/abs/2603.24591>
- Fotaris, P., & Mastoras, T. (2019). Escape rooms for learning: A systematic review. Proceedings of the 13th International Conference on Game Based Learning, ECGBL 2019, Odense, Denmark.
- Fotaris, P., & Mastoras, T. (2022). Room2Educ8: A framework for creating educational escape rooms based on design thinking principles. *Education Sciences*, 12(11), Article 768. <https://doi.org/10.3390/educsci12110768>
- Fotaris, P., Mastoras, T., & Lameris, P. (2023). Designing Educational Escape Rooms With Generative AI: A Framework and ChatGPT Prompt Engineering Guide. Proceedings of the 17th European Conference on Games Based Learning, ECGBL 2023, Lisbon, Portugal.

- Geissler, S. (2025). AI-proof Active Assessment and Future Skills Training with NoCode-Edu Escape Rooms. Proceedings of the 11th International Conference on Higher Education Advances (HEAD'25), Valencia, Spain.
- Gordillo, A., Santamaría Urbieto, A., López Pernas, S., Gutiérrez Gómez-Calcerrada, S., Milošević, M., Ćosić, M., Barra Arias, E., & Elmoazen, R. (2025). *Best Practices for Creating and Conducting Educational Escape Rooms: A Comprehensive Guide for Teachers* [Manual]. Universidad Politécnica de Madrid. <https://doi.org/10.20868/UPM.book.88943>
- Kara, M., Cömert, Z., & Samur, Y. (2025). Designing AI-Enhanced Educational Digital Escape Room Games (EDERGs): A Framework for Teacher Implementation. Proceedings of the 19th European Conference on Games Based Learning, ECGBL 2025, Levanger, Norway.
- Lee, H.-P. H., Sarkar, A., Tankelevitch, L., Drosos, I., Rintel, S., Banks, R., & Wilson, N. (2025). *The impact of generative AI on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers*. Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems, <https://doi.org/10.1145/3706598.3713778>
- Neubig, C. (2024). What Educational Escape Rooms can Learn from the World's Best Rooms. Proceedings of the 18th European Conference on Games Based Learning, Aarhus, Denmark.
- Papadakis, S., & Karakose, T. (2025). Gamification and Student Achievement: Potential Benefits, Limitations, and Effective Use in Educational Environments. *Educational Process: International Journal*, 19, e2025529. <https://doi.org/10.22521/edupij.2025.19.529>
- Pellas, N. (2026). From Prompt to Play: Examining Computational Thinking Through Vibe Coding in Game Making for Pre-Service Teacher Education. *Multimodal Technologies and Interaction*, 10(5), 57. <https://doi.org/10.3390/mti10050057>
- Pornsakulpaisal, R., Ahmed, Z., Bok, H., de Carvalho Filho, M. A., Goka, S., Li, L., Patki, A., Salari, S., Sooknarine, V., Yap, S. W., & Moffett, J. (2023). Building digital escape rooms for learning: From theory to practice. *The Clinical Teacher*, 20(2), e13559. <https://doi.org/10.1111/tct.13559>
- Quek, L. H., Tan, A. J. Q., Sim, M. J. J., Ignacio, J., Harder, N., Lamb, A., Chua, W. L., Lau, S. T., & Liaw, S. Y. (2024). Educational escape rooms for healthcare students: A systematic review. *Nurse Education Today*, 132, 106004. <https://doi.org/10.1016/j.nedt.2023.106004>
- Rodríguez-Rivera, P., Rodríguez-Ferrer, J. M., & Manzano-León, A. (2025). Designing digital escape rooms with generative AI in university contexts: A qualitative study. *Multimodal Technologies and Interaction*, 9(3), 20. <https://doi.org/10.3390/mti9030020>
- Rühl, L., & Thurm, D. (2025). Can you help us? Task-specific LLM-based support in a digital mathematical escape game. Proceedings of the 14th Congress of the European Society for Research in Mathematics Education (CERME14), Bozen-Bolzano, Italy.
- Sarkar, A., & Drosos, I. (2025). *Vibe coding: Programming through conversation with artificial intelligence*. Proceedings of the 36th Annual Conference of the Psychology of Programming Interest Group (PPIG 2025). <https://arxiv.org/abs/2506.23253>
- Solano, G. L., & Wen, W. (2025). Integrating Digital Escape Room Design into Teacher Education: Cultivating Instructional Creativity in Preservice Teachers. *Excelsior: Leadership in Teaching and Learning*, 17(2), 73-96. <https://doi.org/10.14305/jn.19440413.2024.17.2.04>
- Stojšić, I. (2024). Possible utilization of generative artificial intelligence tools for creating digital educational escape rooms. *SCIENCE International Journal*, 3(1), 43-47. <https://doi.org/10.35120/sciencej0301043s>
- Ugo, C. C., López-Pernas, S., Gordillo, A., & Tukiainen, M. (2025). Escape rooms for learning programming: A systematic literature review. *Review of Education*, 13(3), e70123. <https://doi.org/10.1002/rev3.70123>
- Veldkamp, A., van de Grint, L., Knippels, M.-C. P. J., & van Joolingen, W. R. (2020). Escape education: A systematic review on escape rooms in education. *Educational Research Review*, 31, 100364. <https://doi.org/10.1016/j.edurev.2020.100364>
- Wang, Y.-F., Hsu, M.-H., & Yue-Feng Wang, M. (2025). Gamified mobile learning: An escape room chatbot to enhance medical terminology acquisition. *Health Education Journal*, 84(2), 174-188. <https://doi.org/10.1177/00178969241313442>