Towards Requirements for Mixed-Initiative Content Orchestration Tools for Game Development

Tyago Yuji Teoi[1], Leonardo Tórtoro Pereira[2], Claudio Fabiano Motta Toledo[1]

[1]Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (USP)
São Carlos – SP – Brasil

[2]São Paulo State University (UNESP), Institute of Geosciences and Exact Sciences
Rio Claro – SP – Brasil

yujiteoi@gmail.com, leonardo.t.pereira@unesp.br, claudio@icmc.usp.br

Abstract. Introduction*: Mixed-initiative tools enable collaboration between human designers and computational agents, while procedural content orchestration proposes to coordinate multiple facets of procedurally generated game content. Despite advances in both areas, their integration remains underexplored, particularly regarding the requirements needed to support effective human-AI collaboration.* ***Objective****: This paper identifies the functional and non-functional requirements for mixed-initiative content orchestration tools to support game development.* ***Methodology****: Requirements were derived through the triangulation of four evidence sources: interviews with game developers, a heuristic evaluation of the Overlord content orchestrator, literature analysis, and feedback from the Overlord development team.* ***Results****: We present a structured set of requirements organized into five key dimensions: user control and agency, iterative and exploratory workflows, feedback and transparency, multi-content support, and usability. These findings provide design implications for the development of future mixed-initiative tools with content orchestration.*
Keywords *Mixed-initiative interaction, Procedural Content Generation, Software Requirements, Content Orchestration, Human-AI Collaboration.*

1. Introduction

The increasing complexity of digital games has intensified the need for tools that support efficient and flexible content creation. In this context, Creativity Support Tools (CST) aim to assist designers by automating tasks and enhancing creativity [Shneiderman 2002]. Among these, Mixed-Initiative Content Creation Tools (MICCT) enable collaboration between human designers and computational agents, combining human decision-making with automated generation [Lai et al. 2022].

This work focuses specifically on tools that employ Procedural Content Generation (PCG) for game development. While other forms of content generation exist, our investigation is bounded by the technical and interactional challenges inherent to PCG and the coordination of multiple PCG generators. PCG has been widely explored to automate content creation, offering benefits such as scalability, adaptability, and reduced development cost [Shaker et al. 2016]. More recently, Content Orchestration has been proposed to coordinate multiple content generators across different creative facets, such as levels, rules, and narrative [Liapis et al. 2019]. However, most existing systems operate either as fully automatic generators or as tools focused on a single type of content.

Despite advances in both mixed-initiative systems and content orchestration, their integration remains underexplored. In particular, there is a lack of studies investigating what requirements must be satisfied to support mixed-initiative interaction in procedural content orchestration tools. Existing works tend to focus either on interaction techniques for single-content generators or on automated orchestration mechanisms, with limited attention to the needs of designers and developers when these paradigms are combined.

This work addresses this gap by focusing on the identification of functional and non-functional requirements for mixed-initiative content orchestration tools. Rather than proposing a complete system implementation, we investigate what such systems must provide to effectively support human-AI collaboration in the context of orchestrated procedural content generation. Requirements were derived through the triangulation of multiple sources, including interviews with game developers, heuristic evaluation of an existing content orchestrator, analysis of related literature, and *Overlord* authors' feedback. The following research question guided this study:

RQ What functional and non-functional requirements are necessary to support mixed-initiative interaction specifically in procedural content orchestration tools?

The remainder of this paper is organized as follows. Section 2 presents background on mixed-initiative procedural content generation, content orchestration, and related work. Section 3 describes the research methodology, including the interviews, heuristic evaluation, literature analysis, and author feedback. Section 4 presents the elicited functional and non-functional requirements and discusses their implications for the design of mixed-initiative content orchestration tools. Finally, Section 5 concludes the paper and outlines main study limitations and directions for future work.

2. Background

This section reviews studies on two fundamental concepts underlying this study: mixed-initiative tools with procedural content generation and content orchestration. It also discusses recent studies that are closely related to our work.

2.1. Mixed-Initiative Tools with PCG

Mixed-initiative tools support iterative collaboration between human designers and computational agents during content creation [Lai et al. 2022]. Early systems such as Tanagra [Smith et al. 2011] demonstrated designer-guided level generation through constraint solving, evolutionary search, and content suggestions. Later tools expanded these interactions by incorporating multiple design mechanisms, including parameter tuning, content locking, gameplay evaluation, and turn-based collaboration, as seen in the Evolutionary Dungeon Designer [Baldwin et al. 2017], Morai Maker [Guzdial et al. 2021], Baba Is Y'all [Charity et al. 2024] and SuSketch [Migkotzidis e Liapis 2022].

Despite these advances, most tools focus on a single content type or specific genre. Moreover, even when multiple facets are supported, these systems are typically designed as mixed-initiative from the outset, with limited discussion on how to incorporate such interaction into existing systems or generalize it across domains.

2.2. Content Orchestration

Content Orchestration refers to the coordination of procedural generators to produce coherent content across two or more distinct creative facets [Liapis et al. 2019]. Recent work has proposed orchestration frameworks, such as the surrogate-model-based approach by Karavolos et al. [Karavolos et al. 2021], that coordinate gameplay elements based on learned relationships. Some approaches also explore integration with user interaction, such as SuSketch [Migkotzidis e Liapis 2022], which combines orchestration with mixed-initiative design. Finally, Overlord is a content orchestrator that coordinates PCG across multiple domains, including levels, rules, and narrative, and adapts the content based on player profiles, covered in Section 3.1.

2.3. Related Work

Previous studies have identified design principles for mixed-initiative tools, emphasizing designer control, support for creative workflows, and compatibility with existing development practices [Lai et al. 2020]. Recent systems such as Level Building Sidekick [Aliaga et al. 2023] and Melete [Murturi et al. 2025] urther highlight modular and extensible AI-assisted design environments. Meanwhile, So et al. [So et al. 2025] investigated architectural requirements for PCG systems, and Nebeling et al. [Nebeling et al. 2025] elicited requirements for a mixed-initiative virtual cinematography tool. These works address important aspects of mixed-initiative interaction or PCG architectures but do not focus on the requirements of tools that coordinate multiple procedural generators.

However, the specific challenges of coordinating multiple procedural content facets remain insufficiently understood. This work takes an initial step toward addressing this gap by eliciting a set of functional and non-functional requirements for mixed-initiative content orchestration tools.

3. Method

This section presents the Overlord content orchestrator and the multi-method approach used to elicit requirements for mixed-initiative content orchestration. Requirements were derived by triangulating evidence from semi-structured interviews, a heuristic evaluation, literature analysis, and feedback from the Overlord system's developers. The resulting evidence was synthesized into a coherent set of requirements through a three-phase process: extraction, consolidation, and prioritization.

3.1. The Overlord Content Orchestrator

Overlord is a content orchestrator that coordinates procedural generation across multiple creative facets: Levels (e.g., dungeon layouts, item and enemy placement, and door–lock puzzles), Rules (e.g., enemy attributes, movement, and weapons), and Narrative (e.g., quests and questlines). The level generator combines multiple algorithms, including room-template-based dungeon layout generation and cellular automata for room generation. The Rules generator uses evolutionary algorithms, including MAP-Elites, to generate enemies with diverse behaviors and difficulty levels. Narrative content is generated from formal grammars that produce quests and questlines. Each generator exposes configurable parameters that allow designers to influence the characteristics of the generated content.

The orchestrator adapts the generated content to player profiles based on motivation factors [Yee e Ducheneaut 2018]. It coordinates the outputs of the individual generators to ensure cross-facet coherence by integrating the generated levels, enemies, and quests, and then performing post-processing to resolve incompatibilities and enforce design constraints. The orchestrator was originally evaluated in a top-down game [Pereira et al. 2022, Pereira 2022] and was later adapted for a 2D platformer [Teoi et al. 2024, Teoi et al. 2026]. Given its potential for broader application, we investigate the orchestrator as the foundation for a mixed-initiative tool, as described in Sections 3.3 and 3.5.

3.2. Interviews with Game Developers

First, we obtained approval for the research protocol from an institutional Research Ethics Committee (CAAE: 83177524.7.1001.5411) prior to any data collection. All procedures involving participants were conducted only after this approval. Participation was voluntary, and no financial compensation was provided. All participants were informed about the study procedures and provided informed consent prior to their participation.

We conducted semi-structured interviews with 12 Brazilian participants. The sample included game designers, programmers, artists, musicians, and researchers with diverse levels of experience, ranging from 9 to 168 months (mean = 71.8 months, SD = 50.4 months). The sample comprised 3 participants from academia (researchers), 5 from industry (designers, developers, artists, musicians), and 4 from indie development, providing a range of perspectives on game development practices. Regarding gender, the sample included 9 participants identifying as male and 3 identifying as female. We acknowledge that the sample is not representative of the broader game development community in terms of gender diversity, and our findings should be interpreted with this limitation in mind.

The interviews focused on participants' interaction preferences and their perspectives on the use of generative and non-generative AI tools in game development. The interview script and protocol are publicly available on OSF[1]. Although we followed a structured protocol with predetermined questions, we frequently asked follow-up questions based on the interviewees' responses.

The interviews were conducted via Discord, a voice and text communication platform, and were recorded with participants' consent. Each interview was transcribed and subsequently validated by the respective participant to ensure accuracy. The qualitative analysis followed a thematic analysis approach [Braun e Clarke 2021]. The transcribed data were iteratively coded and grouped into categories (themes and subthemes). Initial codes were derived inductively from the data and subsequently refined into higher-level themes representing recurring needs and concerns. The supplementary material[1] summarizes the theme development and mapping to requirements, data saturation assessment, coding validation, and researcher bias mitigation.

A comprehensive presentation of all themes and subthemes is beyond the scope of this study. However, we present selected quotes that illustrate how key requirements emerged from the analysis:

[1] https://osf.io/9edvs

- **User Control over Generated Content:** Multiple participants emphasized that automatically generated content should support, rather than replace, the designer's work. For example, P2 stated: "[automatically generated content] can be used as an initial idea, or to help 'see alternatives'... it can assist the developer as a tool, not replace them." Similarly, P6 noted: "AI can be used in that sense... help me think of new harmonic and melodic paths... Now, making whole songs, I think that's lame." P4 added: "I receive the generated image and work on refining it." These observations motivated requirements related to manual editing, content refinement, and user control over generation outcomes.
- **Reliability and Trust in Generated Content:** Participants also highlighted concerns regarding the reliability of generated content. For instance, P5 stated: "We can't trust 100% of what AI gives us. It returned wrong information about a game we knew had that feature." P3 observed: "[CSTs] still can't maintain consistency on some basic things," while P8 remarked: "The result isn't as good as I expect... It's usually something much simpler." These concerns informed requirements related to feedback, transparency, and mechanisms to assess and validate generated content.

3.3. Heuristic Evaluation

Heuristic evaluation is a method for identifying usability problems based on established usability principles and heuristics. We conducted a heuristic evaluation of the base orchestrator (described in Section 3.1) to identify usability issues and derive requirements related to interaction and user experience. The evaluation followed the heuristics proposed by [Granollers 2018]. Participants were recruited from the researchers' network and received a brief training session on both the system and the evaluation method.

A total of three evaluators participated (two experienced industry developers and one researcher), in accordance with recommended practices [Nielsen 1994]. Each participant followed a predefined interaction flow in the Overlord system, consisting of the following steps: (1) configuring individual generators, including level, narrative, and rule generation parameters; (2) generating content based on the configured parameters; (3) inspecting the generated outputs; and (4) modifying orchestration settings to control how different content generators interact. This flow was designed to simulate a typical content creation process using the orchestrator.

Participants performed individual evaluations of the system through guided interaction, supported by interface screenshots when necessary. After the individual evaluations, participants collaboratively produced a consolidated report containing identified usability problems, their severity, and suggested improvements. These findings were used as input for requirement derivation. The main identified usability problems are summarized in Table 1.

3.4. Literature Analysis

We analyzed representative works in mixed-initiative systems and content orchestration to identify recurring interaction patterns and design practices.

This analysis focused on well-established tools such as Tanagra [Smith et al. 2011], Sentient Sketchbook [Liapis et al. 2013], Evolutionary Dungeon Designer [Baldwin et al. 2017], and SuSketch [Migkotzidis e Liapis 2022], as well as conceptual work on content orchestration [Liapis et al. 2019]. These studies were selected due to their relevance and frequent citation in the literature.

From these works, we extracted recurring design elements such as content suggestion, user control mechanisms (e.g., locking and parameter tuning), iterative

Table 1. Key Usability Problems (Severity $\geq$ 3). The consolidated document included issues classified by severity as 3 (Major), and 4 (Catastrophic). For the purpose of deriving software requirements, problems with Cosmetic and Minor severity were excluded due to their limited impact.

ID	Problem Description	Suggested Solution	Sev.
H1	No content visualization without testing	UI with content abstraction	4
H2	Changes not directly observable	UI with content abstraction	4
H3	Modifiers require technical knowledge	Unified UI for modifiers/settings	4
H4	Workflow hard to remember (fragmented)	Unified UI for modifiers/settings	3
H5	Information scattered across screens	Unified UI for modifiers/settings	3
H6	Organization not intuitive for non-techs	Unified UI for modifiers/settings	3
H7	No constraints for player parameters	Define valid input ranges	3
H8	Lack of documentation	Provide comprehensive documentation	4
H9	No help feature available	Add help linked to documentation	3
H10	Cannot save generated content	Export/reproduce content	4
H11	No preview of generated content	Implement preview rendering	4
H12	Generated content cannot be edited	Allow post-generation adjustments	4

interaction workflows, and feedback mechanisms. These elements were used as complementary input during requirement elicitation, particularly to validate and contextualize findings from interviews and heuristic evaluation.

In addition, established design principles from Creativity Support Tools [Resnick et al. 2005], mixed-initiative interaction [Horvitz 1999], and usability [Nielsen 2024] were considered to support the interpretation of findings and the formulation of requirements.

3.5. Authors' Feedback from the Overlord Development Team

We incorporated systematic feedback from the development team of the Overlord orchestrator (described in Section 3.1) as a complementary source of evidence. This feedback provided unique insights into technical feasibility, implementation constraints, and design trade-offs that emerged throughout the system's development and practical use.

The feedback was collected from six individuals who had been directly involved with the Overlord system during its development lifecycle: the two principal developers (the first and second authors of this paper), two faculty advisors (the third author and another faculty member), and two undergraduate researchers who had used the system in a related study. Further details on the feedback collection procedure and analysis are provided in the supplementary material[2].

3.6. Requirement Synthesis

To transform the evidence collected from the four sources into a coherent and defensible set of requirements, we carried out a structured synthesis process consisting of three phases: (1) extraction of candidate needs from each source, (2) consolidation into unified requirement statements, and (3) prioritisation. The synthesis was performed primarily by the first author with the other authors acting as reviewers and providing critical feedback during regular alignment meetings.

[2]https://osf.io/9edvs

The **extraction** phase was described in the aforementioned subsections, contemplating heuristic evaluation, interview with game developers, literature analysis, and authors' feedback. In the **consolidation** phase, the first author grouped the candidate statements into thematic clusters. After comparing the clusters, it was merged to address the same underlying concern, even when wording differed (e.g., "lock parts of the level" and "freeze sections against changes" were unified). Each final cluster was then expressed as a standardised requirement using the template "The tool shall [capability]" (functional) or "The tool shall [quality attribute]" (non-functional). At this stage, overlap and hierarchy among requirements were checked, and highly specific sub-requirements were either absorbed into a parent requirement or kept separate if they introduced a distinct concern.

Finally, in the **prioritisation** phase, we assigned a priority level (High, Medium, or Low) to each requirement based on two criteria: (a) the number of independent evidence sources supporting the requirement and (b) the severity of the associated usability problems identified during the heuristic evaluation. Requirements supported by at least three of the four evidence sources, or addressing a catastrophic usability issue (severity 4), were assigned High priority. Requirements supported by two sources or associated with major usability issues (severity 3) were assigned Medium priority. Requirements derived from a single source without corroborating evidence were assigned Low priority. The prioritisation procedure was applied consistently and explicitly documented to reduce arbitrariness. Further details of the quality assessment of the requirement set are provided in the supplementary material[2].

4. Results and Discussion

This section presents the identified requirements and discusses their implications, addressing the research question:

RQ: *What functional and non-functional requirements are necessary to support mixed-initiative interaction specifically in procedural content orchestration tools?*

The requirement elicitation process resulted in a set of functional and non-functional requirements, summarized in Tables 2 and 3. The requirements were grouped in the following dimensions.

User Control and Agency emerge as a central dimension. Interview participants emphasized the need to maintain autonomy over the generation process, which is reflected in requirements such as generation triggering (FR-01), manual editing (FR-02), content locking (FR-11), and configuration of procedural generation parameters (FR-05). In an orchestrated setting, user control extends beyond individual artifacts: a designer's action on one content type can impose constraints on others. The system must therefore provide mechanisms to coordinate locking across facets and to inspect how orchestration decisions shape generated content, ensuring that computational generation supports, rather than replaces, the designer's role [Horvitz 1999, Lai et al. 2022].

The need for **Iterative and Exploratory Workflows** was also consistently observed. Requirements such as content suggestions (FR-12) and support for iterative interaction cycles (FR-07) indicate that the system must enable continuous refinement. The provision of visual representation of content and its variations (FR-06) and the

Table 2. Functional requirements for a mixed-initiative content orchestration tool. HE(S) = Heuristic Evaluation(Severity), IN = Interviews, LA = Literature Analysis, and AF = Authors' Feedback

No	Requirement	Description	Source	Priority
FR-01	Trigger PCG anytime	Allow users to trigger procedural content generation at any stage of the design process	IN; LA; AF	High
FR-02	Manually edit generated content	Allow manual editing of generated content	IN; HE; LA; AF	High
FR-03	Feedback and metrics	Provide metrics and feedback about generated content	HE; LA; AF	High
FR-04	Multi-content orchestration	Support orchestration of multiple content types (e.g., levels, rules, narrative)	HE; LA; AF	High
FR-05	Configure PCG parameters	Allow configuration of procedural generation parameters	IN; LA; AF	High
FR-06	Visualize content variations	Provide visual representation of content and its variations	IN; HE; LA	High
FR-07	Iterative interaction cycles support	Support iterative interaction cycles between user and system	LA; AF	Medium
FR-08	Save/load sessions	Allow saving and loading of content creation sessions	LA; AF	Medium
FR-09	Cross-facet dependency visualization	Allow users to visualize cross-facet dependencies and conflicts	HE(4); AF	High
FR-10	Inspect orchestration decisions	Allow users to inspect how orchestration decisions shape generated content	HE(3)	Medium
FR-11	Lock content parts	Allow users to lock parts of the content to prevent modification by the system	IN, LA	Medium
FR-12	Suggest alternative content	Provide alternative content suggestions based on the current design state	LA	Low
FR-13	Coordinated locking	Support coordinated locking across content facets	LA	Low

ability to revisit previous decisions (FR-08) further support this exploratory behavior. Importantly, in a multi-content pipeline, exploration must reveal how local refinements ripple across the whole—a need absent in single-generator tools. This aligns with principles from Creativity Support Tools [Shneiderman 2002, Resnick et al. 2005], where exploration and iteration are fundamental.

Feedback and Transparency are critical to enable effective human-AI collaboration. Requirements related to content metrics and feedback (FR-03), timely system responses (NFR-02), and transparent feedback about generation processes (NFR-03) reflect the need to make generation processes understandable. In the orchestration context, transparency takes on added complexity: the system must reveal not only how individual generators behave, but also how their outputs interact. Reliability concerns raised during interviews further reinforce the importance of mechanisms that allow users to assess and validate generated content, particularly when multiple interdependent generators are involved.

Usability is fundamental to the effectiveness of mixed-initiative systems. Requirements such as learnability (NFR-01), responsiveness (NFR-02), and reliability (NFR-04) directly impact the fluidity of interaction. Findings from the heuristic evaluation indicate that poor feedback, unclear system behavior, or delayed responses can disrupt the user experience—effects that are magnified when multiple generators must be orchestrated. Orchestration-specific usability breakdowns observed in the evaluation

Table 3. Non-functional requirements for a MI content orchestration

No	Requirement	Description	Source	Priority
NFR-01	Learnability	Provide intuitive and easy-to-learn interaction mechanisms	HE; LA; AF	High
NFR-02	Responsiveness	Provide timely feedback during user interaction	HE(4); LA	High
NFR-03	Transparent generation process	Provide transparent feedback about content generation processes	HE(4); LA	High
NFR-04	Reliability	Secure stable and predictable behavior during generation and editing	HE; IN	Medium
NFR-05	Cross-facet Consistency	Maintain consistency across content facets after user modifications	HE; IN	Medium
NFR-06	Extensibility	Support integration of new procedural generators or algorithms	AF	Low
NFR-07	Performance	Maintain performance as content complexity increases	LA	Low
NFR-08	Modularity	Follow a modular architecture separating interaction, orchestration, and generation components	AF	Low
NFR-09	Documentation support	Provide links to a comprehensive documentation	HE(4)	High

are addressed by dedicated requirements for cross-facet consistency and dependency visualization (see the next dimension).

A distinguishing aspect of this work is the requirement for **Multi-Content Orchestration Support**. The coordination of multiple content types (FR-04) introduces additional complexity compared to traditional mixed-initiative tools. This includes managing dependencies between generators and maintaining consistency across content facets—challenges directly addressed by cross-facet consistency (NFR-05), dependency visualization (FR-09), inspection of orchestration decisions (FR-10), and coordinated locking (FR-13). These operational demands are also reflected in non-functional requirements such as modularity (NFR-08) and extensibility (NFR-06), which are necessary to support system evolution and integration of new generators. Thus, orchestration is not merely the sum of independent generators, but a coordination layer with its own interaction demands. In Table 4, a comparison is presented between traditional mixed-initiative PCG tools and proposed content orchestration tools along the requirements of this dimension.

Table 4. Traditional MI Tools vs. Content Orchestration Tools

Requirement	Traditional MI Tools	Content Orch. Tools
Multi-content coordination (FR-04)	Rare	Must Have
Cross-facet consistency (NFR-05)	Rare/Absent	Must Have
Cross-facet dependency visualization (FR-09)	Rare/Absent	Must Have
Inspect orchestration decisions (FR-10)	Absent	Should Have
Coordinated locking (FR-13)	Absent	Should Have
Modularity (NFR-08)	Rare/Absent	Could Have
Extensibility (NFR-06)	Rare	Could Have

Overall, these findings suggest that the design of mixed-initiative content orchestration tools should not be approached as a simple extension of existing systems, but rather as a complex interaction problem involving multiple interdependent requirements. The identified dimensions define a design space that can guide future research and development in this area.

5. Conclusion

This paper investigated the requirements for mixed-initiative procedural content orchestration tools in the context of game development. By focusing on requirements elicitation, this work provides a structured understanding of the capabilities such systems should support to enable effective human–AI collaboration.

The results comprise a set of functional and non-functional requirements derived through the triangulation of interviews, heuristic evaluation, literature analysis, and feedback from the authors of the Overlord content orchestrator. These requirements highlight key aspects such as *user control and agency*, *iterative and exploratory workflows*, *feedback and transparency*, *usability*, and *multi-content coordination support*.

The findings indicate that supporting mixed-initiative content orchestration requires more than well-designed interfaces to facilitate collaboration between human designers and computational agents. Content orchestration introduces an additional layer of complexity by requiring the coordination of multiple procedural generators while enabling designers to control and understand the dependencies among them. Consequently, cross-facet consistency and dependency visualization, locking control over coordinated content, orchestration inspection support, modularity, extensibility, and coordination across multiple creative facets emerge as central design concerns.

This study contributes an empirically grounded set of requirements that can serve as a foundation for the design and evaluation of future mixed-initiative content orchestration tools. As an initial step toward this goal, the findings establish a baseline for understanding the challenges of combining mixed-initiative interaction with multi-faceted procedural content generation.

Nevertheless, this work has limitations. We cannot claim that the elicited requirements apply equally across all PCG paradigms, including search-based, machine learning-based, constructive, rule-based, and LLM-based approaches. Moreover, the requirements have not been validated by external practitioners, limiting claims of completeness and industry-wide representativeness. However, triangulating four independent evidence sources strengthens their reliability. Future work will validate the requirements through expert surveys and industrial case studies and assess their applicability across PCG paradigms and domains.

Declaration of Generative AI and AI-assisted technologies

We used LLMs such as *Grammarly* and DeepSeek to improve the article's overall quality. We take full responsibility for all content produced with AI assistance.

Acknowledgments

This research was funded by the National Council for Scientific and Technological Development (CNPq). This study was also financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil* (CAPES) – Finance Code 001.

References

Aliaga, C., Vidal, C., Sepulveda, G. K., Romero, N., González, F., e Barriga, N. A. (2023). Level building sidekick: An ai-assisted level editor package for unity. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, volume 19, pages 392–399. AAAI Press.

Baldwin, A., Dahlskog, S., Font, J. M., e Holmberg, J. (2017). Mixed-initiative procedural generation of dungeons using game design patterns. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 25–32.

Braun, V. e Clarke, V. (2021). *Thematic Analysis: A Practical Guide*. SAGE Publications.

Charity, M., Dave, I., Khalifa, A., e Togelius, J. (2024). Baba is y'all 2.0: Design and investigation of a collaborative mixed-initiative system. *IEEE Transactions on Games*, 16(1):75–89.

Granollers, T. (2018). Usability evaluation with heuristics, beyond nielsen's list. In *ACHI 2018, The Eleventh International Conference on Advances in Computer-Human Interactions*, pages 60–65.

Guzdial, M., Chen, J., Chen, S.-Y., e Riedl, M. (2021). A general level design editor for co-creative level design. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 13(2):81–83.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In *International Conference on Human Factors in Computing Systems*.

Karavolos, D., Liapis, A., e Yannakakis, G. (2021). A multifaceted surrogate model for search-based procedural content generation. *IEEE Transactions on Games*, 13(1):11–22.

Lai, G., Latham, W., e Leymarie, F. F. (2020). Towards friendly mixed initiative procedural content generation: Three pillars of industry. In *Proceedings of the 2020 IEEE Conference on Games (CoG)*, pages 1–8. IEEE.

Lai, G., Leymarie, F. F., e Latham, W. (2022). On mixed-initiative content creation for video games. *IEEE Transactions on Games*, 14(4):543–556.

Liapis, A., Yannakakis, G. N., Nelson, M. J., Preuss, M., e Bidarra, R. (2019). Orchestrating game generation. *IEEE Transactions on Games*, 11(1):48–68.

Liapis, A., Yannakakis, G. N., e Togelius, J. (2013). Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of the 8th Conference on the Foundations of Digital Games*, pages 213–220.

Migkotzidis, P. e Liapis, A. (2022). Susketch: Surrogate models of gameplay as a design assistant. *IEEE Transactions on Games*, 14(2):273–283.

Murturi, S., Pellicone, T., Yee-King, M., e Gillies, M. (2025). Melete: Exploring the components of mixed-initiative artificial intelligence pipelines for level design. In *Proceedings of the Fourth International Conference on Hybrid Human-Artificial Intelligence*, volume 4074, pages 427–447. CEUR-WS.

Nebeling, M., Wu, L., e Maddali, H. T. (2025). Xcam: Mixed-initiative virtual cinematography for live production of virtual reality experiences. In *Proceedings of* *the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.

Nielsen, J. (1994). The theory behind heuristic evaluations.

Nielsen, J. (2024). 10 usability heuristics for user interface design.

Pereira, L. T. (2022). *Arquitetura e desenvolvimento de um sistema de geração procedural de múltiplos conteúdos para jogos eletrônicos em tempo real*. PhD thesis, Universidade de São Paulo.

Pereira, L. T., Viana, B. M. F., e Toledo, C. F. M. (2022). A system for orchestrating multiple procedurally generated content for different player profiles. *IEEE Transactions on Games*, pages 1–11.

Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., e Eisenberg, M. (2005). Design principles for tools to support creative thinking. *Report of Workshop on Creativity Support Tools*, 20.

Shaker, N., Togelius, J., e Nelson, M. J. (2016). *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer.

Shneiderman, B. (2002). Creativity support tools. *Commun. ACM*, 45(10):116–120.

Smith, G., Whitehead, J., e Mateas, M. (2011). Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):201–215.

So, A., Souza, F., Souza, A., e Fabri, J. (2025). Software architecture requirements for procedural content generation in digital games: A systematic mapping. In *Anais do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 668–679, Porto Alegre, RS, Brasil. SBC.

Teoi, T., Pereira, L., e Toledo, C. (2024). Towards adapting a content orchestrator to a different gamegenre: Generating levels, rules, and narrative for diverse player profiles from a top-down adventure to a 2d platformer. In *Anais do XXIII Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 396–415, Porto Alegre, RS, Brasil. SBC.

Teoi, T. Y., Pereira, L. T., e Toledo, C. F. M. (2026). Adapting a content orchestrator across game genres: From top-down adventure to 2d platformer. *Entertainment Computing*, 58:101148.

Yee, N. e Ducheneaut, N. (2018). 485Gamer motivation profiling: uses and applications. In *Games User Research*. Oxford University Press.