Proceedings of the Informing Science + Information Technology Education Conference

An Official Publication of the Informing Science Institute
InformingScience.org

InformingScience.org/Publications

July 20 – 28, 2025

OPEN-SOURCE GAME DEVELOPMENT: A SYSTEMATIC LITERATURE REVIEW

Howard Hall*	University of Cincinnati, Cincinnati, OH, USA	hallho@mail.uc.edu
Abigail Dogbe	University of Cincinnati, Cincinnati, OH, USA	dogbeam@mail.uc.edu
Hazem Said	University of Cincinnati, Cincinnati, OH, USA	saidhm@ucmail.uc.edu

* Corresponding author

ABSTRACT

Aim/Purpose	Open-source software has transformed how technology solutions are developed across both academia and industry, especially within the game development field.
Background	This study explores open-source tools used in creating digital games and examines the factors driving their evolution.
Methodology	A systematic literature review, following the methodology of Kitchenham and Charters, was conducted across 44 studies sourced from ACM and Scopus. A thematic analysis was then performed using a modified version of Braun and Clarke's framework.
Contribution	Themes of adapted approaches, education, development processes, cost-effectiveness, passion, academic research, and intrinsic/extrinsic reward are themes that contribute to how game development tools evolve when impacted by the application of open-source concepts.
Findings	After reviewing the seven themes, findings show that open-source game development has methods of evolution similar to other open-source projects. They are often subject to the same community formats and have similar methods of evolving tools.

Accepted by Editor Michael Jones | Received: January 31, 2025 | Revised: March 25, 2025 | Accepted: March 26, 2025.
Cite as: Hall, H., Dogbe, A., & Said, H. (2025). Open-source game development: A systematic literature review. In M. Jones (Ed.), *Proceedings of InSITE 2025: Informing Science and Information Technology Education Conference*, Article 20. Informing Science Institute. https://doi.org/10.28945/5563

Recommendations for Practitioners	Practitioners are encouraged to consider using and contributing to open-source game development tools to broaden the pool of resources for developers and researchers.
Recommendations for Researchers	Researchers are encouraged to seek out and use open-source tools for research, as game development is not a well-documented field that would benefit from more explorative studies with open-source tools.
Impact on Society	Game development is a field with opportunities for study that would benefit most from projects that allow opportunities to analyze processes and tools.
Future Research	Future work will expand to more data sources and apply a more rigorous methodology to ensure themes are accurate and understandable. Subthemes will also be discussed to describe the themes found in this paper more accurately.
Keywords	game development, game development tools, open-source software, open-source methods

INTRODUCTION AND BACKGROUND

Digital games were first conceptualized in the 1950s (Michie, 1982). Attributed to Claude Shannon, who had written a paper in 1950 on programming a digital computer to play chess (Michie, 1982), digital games would see commercialization in the 1970s following the release of the Magnavox Odyssey, the first commercial games console (Smithsonian, 2024). After the Odyssey's release, Nintendo would go on to release Donkey Kong (Hadzinsky, 2014), and Activision would become the first 3rd party development studio (Activision Blizzard, 2025). According to a retrospective piece from 2000, game development tools were often programmed for in-house use and were rarely, if at all, made available to other companies outside of directly paying for licensing (Rouse, 2000).

In the mid-1990s, educational programs for the development of games began to be proposed within academic papers (Holliday, 1995). These would have students in software engineering courses re-creating existing commercial games such as Master Mind or Minesweeper (Holliday, 1995). However, this was accomplished by programming the games while not using development tools like what would be available at a dedicated development studio.

OPEN-SOURCE SOFTWARE (OSS)

The term "open source" describes a software development model that involves contributions from developers across the globe via the Internet, with the essential requirement that the source code is accessible, as without it, the software's development or progression would be extremely difficult, if not unfeasible (Gacek & Arief, 2004). While the terms Free Open Source Software (FOSS) and Open Source Software (OSS) are closely related, they have slightly different approaches and licensing schemes for developing publicly shared software (Scacchi, 2004).

In the 1990s, Linux and Debian were released for developers specifically. Communities developed and iterated on these tools and more, such as Apache, an HTTPS server (Gonzalez-Barahona, 2021). At the end of the decade, the term "open-source software" was coined after the explosion of Netscape Communicator, which at the time of release dominated the newfound creation of web browsers alongside Internet Explorer. In the aftermath, the popularity of free and open-source software exploded (Gonzalez-Barahona, 2021).

EXISTING RESEARCH ON GAME DEVELOPMENT USING OSS TOOLS

Digital game education continued to develop into the early 2000s and beyond, with software development programs offering game development courses. Digital games during this time had also caused a resurgence in computer science and adjacent fields (Goulding, 2008). One university stated that since the addition of game development education, enrollment in their CS programs increased by 240 percent (Goulding, 2008). This created a need for the proper tools to teach students on, as early development was done purely through software development means, and developers of the time built tools in-house (Vanhatupa, 2011). Many of the engines available also could not provide all the features that would be needed to give students a thorough education on the development process. Thus, many educators and members of the academic games community began working on building their own engines, such as the CAGE engine (Vanhatupa, 2011) or the GEDI engine (Coleman et al., 2005).

In practical game development, understanding the solutions adopted by video game developers is crucial for future research aimed at improving software quality and developer productivity, which has implications for both the OSS community and potentially the broader industry (Pascarella et al., 2018). Some findings from industrial game development research such as the perception of less clear requirements among game developers, the likelihood of developed code included in future releases and difficulties in writing unit tests all suggests that OSS context of game development introduces unique aspects to game development compared to the broader industrial setting (Murphy-Hill et al., 2014).

One way to meet needs for customizing engines or finding alternative development approaches would be to take advantage of the emerging open-source applications for software (Scacchi, 2004). Engines and other development tools on the market for free in the modern day often have some form of open-source contributions available. Some examples are Unity's repository with open-source features (Unity Technologies, 2024a) or the Unity Asset Store featuring free contributions from developers accompanied by source code (Unity Technologies, 2024b). Unreal Engine is source-available, meaning it provides access to the source code and allows modifications through its pricing models (Unreal Engine, 2025). On top of these two, the Godot Engine and Blender 3D modeling software are completely open source (Blender Foundation, 2025; Godot Engine, 2025). According to analytics from Steam, a digital storefront for purchasing games, of 7,392 analyzed games, 38.8% were created within Unity or Unreal Engines (Toftedahl, 2019).

However, there are few studies on open-source game development that explore the historical overlap between these fields. To document some of the challenges of developing open-source tools or what led to their evolution, this study seeks to perform a systematic literature review and thematic analysis of the history within the cross-domain of open-source and digital games development. Through this, we can uncover gaps in existing research, identify key trends in the game development domain, and analyze the impact of open-source tools in this industry. The review aims to answer the following research question: What factors led to the evolution of open-source software solutions used to develop digital games?

METHODOLOGY

This paper utilizes a systematic literature review according to the guidelines proposed by Kitchenham and Charters (2007). It includes the subsequently elaborated components: inclusion and exclusion criteria as seen in Table 1, search process, filtering and quality assessment, and the data collection process. A systematic literature review approach allows for analysis of trends in existing research that can be identified as themes in the subsequent thematic analysis process. The primary databases for this study are the ACM and Scopus Digital Libraries. ACM is widely known for finding peer-reviewed literature in the field of computing and technology (ACM Digital Library, 2025), and Scopus includes a wide selection of research articles and papers across numerous areas (Elsevier, 2025).

Table 1. Inclusion and exclusion criteria for open-source game development research

Inclusion criteria	Exclusion criteria
Articles that review the history of game development tools	Articles not related to the history of game development
Research articles and short articles	Non-research articles
Articles in English	Non-English articles
Articles related to digital games	Papers focusing on the history of domain-adjacent software
Articles discussing open-source game development tools	
Articles on the origin of open-source practices in digital game development	

Search Process

For conducting this systematic literature review, publications were retrieved from the ACM and Scopus Digital Libraries, and three different search queries were applied to both libraries.

The libraries have extensive literature that details open-source innovations in the scope of game development-specific tools and features within domain-adjacent software solutions that fall within the inclusion and exclusion criteria in Table 1, enabling a comprehensive look at technical developments within the cross-domain. Table 2 shows the metadata for these searches, which utilize the queries found in Appendix B. Queries 1 and 2 were used on the first search for literature specifically to find literature relevant to the technical aspects of game development. On a second pass, query 3 was added to the search to find more literature focused on accounts detailing the evolution of tools.

Table 2. Search metadata for open-source game development research

Database	Queries	Results
ACM	Query 1	103
Scopus	Query 1	195
ACM	Query 2	10
Scopus	Query 2	10
ACM	Query 3	0
Scopus	Query 3	25

Applying Inclusion Criteria for Quality Assessment

The application of the queries in Table 2 resulted in a total of 343 articles. The authors applied the inclusion criteria to the title and abstract of each article through a two-step process. In the first step, one author applied the inclusion criteria; in the second step, the other author reviewed the papers obtained from applying the inclusion criteria. If both authors agreed that a study met the inclusion criteria, it would be included in the quality assessment. If there was a disagreement, a discussion was held to justify whether it would be included. After this process, a total of 44 articles were selected as primary studies.

Following the guidelines established by Kitchenham et al. (2009), each paper underwent a quality assessment through a review of the full text. Each paper was evaluated by the second author to determine whether it provided sufficient information relevant to the assessment questions listed in Table 3. Whether a study ranged from "Less Focus" to "Most Focus" was determined based on the

author's observations of how much a study discussed the assessment criteria throughout the full text of the publication. Studies that did not receive a "Most Focus" in any assessment questions were excluded from the final list of primary studies. Ultimately, 28 papers met the assessment questions and were selected as primary studies. Primary study titles and designations have been listed in Appendix A.

Table 3. Quality assurance of literature on open-source game development

Assessment question	Less focus	Some focus	Most focus
To what degree does the article discuss challenges and/or developments from overcoming them?	10	12	22
To what degree does the article discuss historical accounts of open-source game development concepts?	37	6	1
To what degree does the article discuss the creation or innovation of open-source tools to facilitate game development?	19	14	11
To what degree does the article discuss methods and concepts for applying open-source to game development?	4	18	22

Thematic Analysis

Thematic analysis has been performed according to guidelines set by Braun and Clarke (2022). The analysis was also performed using MAXQDA, a tool with features to perform the process more efficiently (MAXQDA, 2025). With MAXQDA, phases 1 and 2 of Braun and Clarke's original framework could be performed simultaneously, as coding could be done during familiarization, and collation would automatically happen during that process, thanks to the smart-coding tool offered. Figure 1 shows the process we follow.

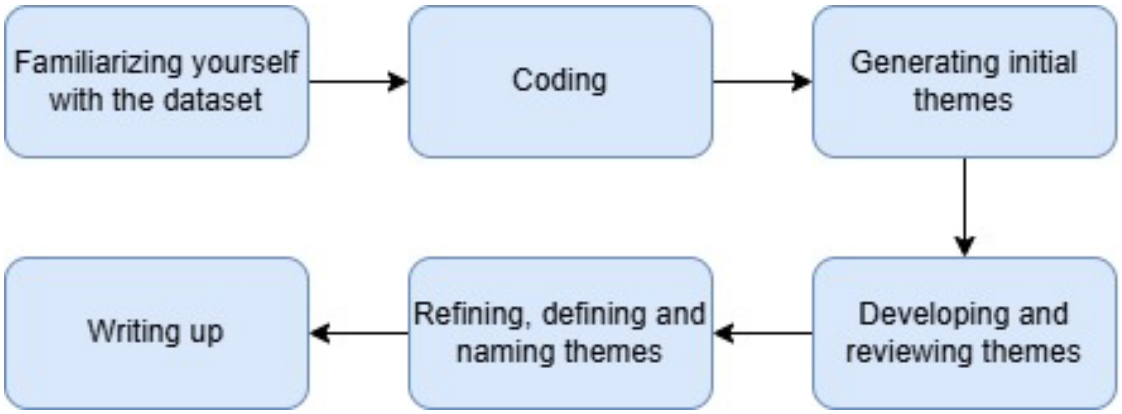

Figure 1. The thematic analysis process described by Braun and Clarke

Phase One: Familiarization and coding

Phases one and two of the process proposed by Braun and Clarke (2022) involve familiarizing oneself with the dataset and coding relevant data with concise, yet meaningful labels that can be organized. MAXQDA enabled the performance of both steps at once; thus, the structure has been modified to combine them. Through MAXQDA, we were able to highlight information within the text

and give it specific tags matching its relevance, thus reviewing relevant pieces of literature and sorting them simultaneously.

Phase Two: Generating initial themes

After coding the data set and applying automatic collation through MAXQDA, initial themes were generated based on the coded segments. This involved creating a code matrix through MAXQDA, which allowed visualization of which codes most often overlapped, as shown in Figure 2. Additionally, with the smart-coding tool available in MAXQDA, it was possible to isolate codes and draw themes from analyzing them as well.

Phase Three: Developing and reviewing themes

This phase took another pass after analyzing the data for initial themes. Here, the dataset was re-analyzed in terms of how well themes applied to coded segments, or how those themes fit in the context of the whole dataset. This determined which themes needed changes or removal if they were deemed irrelevant to the dataset.

Phase Four: Refining, defining, and naming themes

Phase four took the results of phase three and consisted of refining the themes that remained after analysis. This ensured the names given to themes were concise yet informative.

Phase Five: Writing up

This phase melds in slightly with phases two through four, as it consists of recording the thematic analysis process and writing out findings generated from it. This includes the steps of the methodology and the following results and discussion section, which explains how these findings answer the research question.

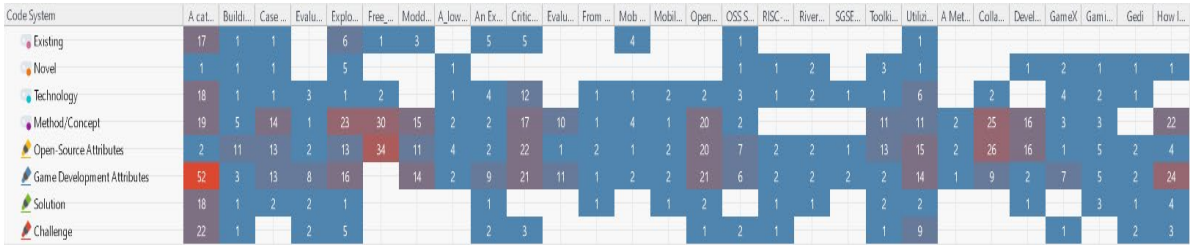

Figure 2. Code matrix visualized by MAXQDA **for the thematic analysis**

RESULTS

The results discussed in this section are on display in Table 4. Thematic overlap between each of the primary studies is shown in Table 5. For thematic coding performed in phase one, information was tagged with eight different codes to sort information based on whether it applied to open-source or game development, if it discussed a challenge or solution within those domains, whether the information pertained to a concept/method used or technologies used, and if the information existed prior to the article or is proposed as a novel contribution.

In phase two, initial themes were identified based on a review of coded segments in a grouped context. The groupings for Game Development Attributes and Open-Source Attributes were specifically reviewed as they pertained directly to information on open-source additions to game development, while other codes provided a high-level context of the premise of a given code snippet.

Table 4. Results of the thematic analysis process

Themes	Applicable studies	Sample quotes
Adapted Approaches	PS6, PS7, PS10, PS15, PS16, PS18, PS20, PS21, PS22, PS24, PS28	"Thus, game modding represents a singular software development approach positioned in the intersection of hobbyist fan-made amateur creations, free/Open-source software (FOSS) and professional game development" (PS7) "With a FOSS game engine, the developer has complete access to all of the game engine's programming source code and is free to modify the engine as needed." (PS10)
Education	PS2, PS10, PS12, PS17, PS19, PS21, PS25, PS26	"Supported by the enthusiasm driving creative coding, maker, and open hardware movements, practice-based educational formats are migrating from the extra-curricular margins toward the center of new forms of experimental learning" (PS12) "In an effort to promote open systems and to align with other courses offered in the department, we decided that the emulated processor should be RISC-V based. RISC-V was designed with research and education in mind, but is growing in industry use." (PS17)
Development process	PS3, PS6, PS7, PS9, PS10, PS13, PS14, PS15, PS16, PS17, PS18, PS19, PS20, PS21, PS26, PS27	"It is expected that game genres with high availability of components can more easily benefit from OSS reuse." (PS3) "Offering communities of users the possibility to interact and collaborate on a toolkit basis in order to co-create new products or solutions proved to be a very successful development strategy for open source software development." (PS20)
Cost-effective	PS4, PS6, PS8, PS21	"using free and open-source toolboxes, software, features, and libraries. The entire concept delivered a working application, capable of recognising hand gestures, without any specific high-end hardware requirements" (PS4) "FOSS communities develop software that's extremely valuable, generally reliable, globally distributed, made available for acquisition at little or no cost, and readily used in its associated community" (PS6)
Passion	PS5, PS6, PS15, PS16, PS20, PS21, PS23, PS24, PS28	"indie game development teams are formed upon shared aspirations and use various strategies to collaborate with friends or online strangers" (PS5) "... conclude that enjoyment-based intrinsic motivation, namely how creative a person feels when working on the project, is the strongest and most pervasive driver." (PS24)
Academic research	PS1, PS3, PS8, PS9, PS10, PS11, PS17, PS22, PS23, PS24, PS25, PS26	"GitHub hosts a significant number of open-source game repositories with approximately 188K repositories related to games. These repositories contain valuable software artifacts such as commits, issues, and pull requests, which can be leveraged to gain insights into

Themes	Applicable studies	Sample quotes
		the challenges faced by users and developers, and their potential solutions." (PS1) "GamingAnywhere is publicly available online. Use of GamingAnywhere in academic research is free of charge, but researchers and developers should follow the license terms claimed in the binary and source packages." (PS26)
Intrinsic and Extrinsic reward	PS6, PS7, PS15, PS20, PS21, PS23, PS24	"... because of the nature of FOSS development, each developer or community follows their own ways, implementing features they are interested in." (PS15) "The main incentive for game modders, aside from the possibility to enhance their own game playing experience, is the ability to learn about game development and maybe use those skills later to find a job in a game studio." (PS23)

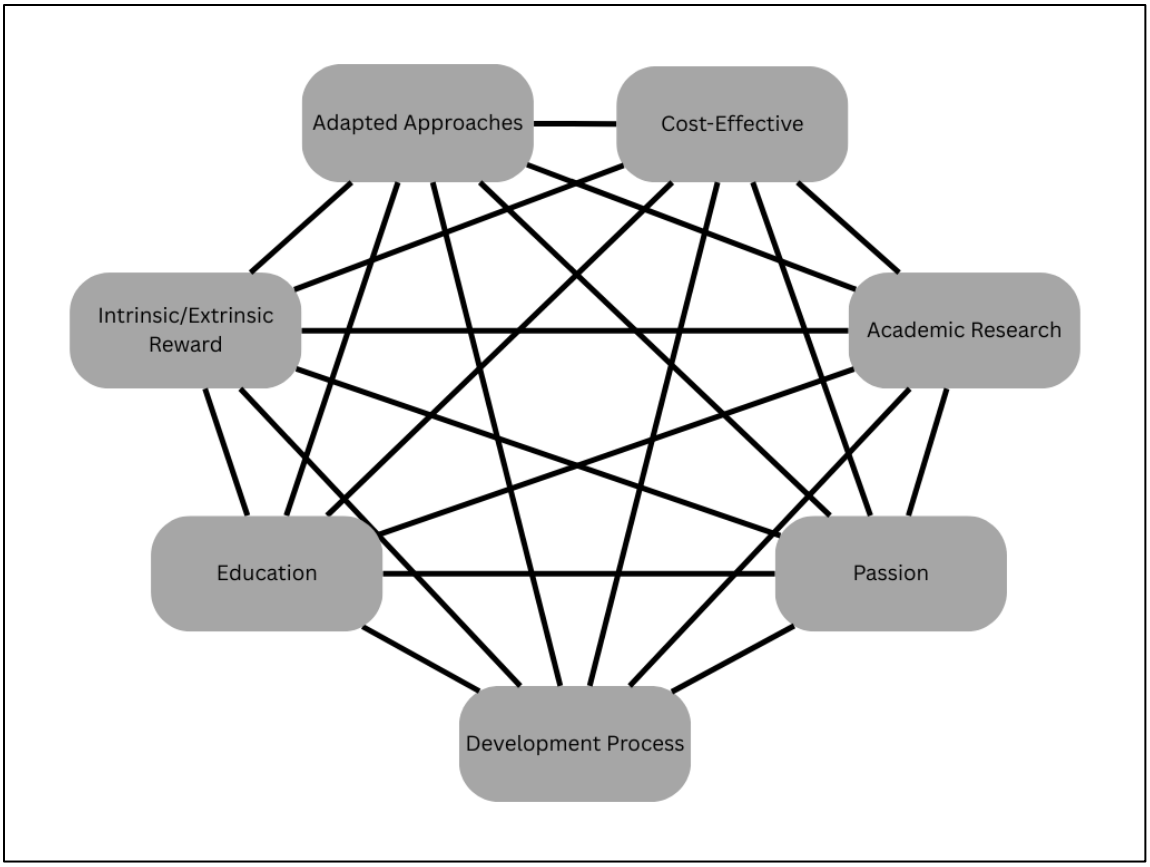

Figure 3. Visual representation of thematic overlap

Table 5. Thematic overlap in primary studies

	Adapted approaches	Education	Development process	Cost-effective	Passion	Academic research	Intrinsic/ extrinsic reward
Adapted approaches	X						
Education	PS10, PS21	X					
Development process	PS6, PS7, PS10, PS15, PS16	PS10	X				
Cost-effective	PS6, PS21	PS21	PS6, PS21	X			
Passion	PS6, PS15, PS16, PS20, PS21, PS24, PS28	PS6, PS21	PS6, PS15, PS16, PS20, PS21	PS6, PS21	X		
Academic research	PS22, PS24	PS10, PS26, PS17	PS3, PS9, PS10, PS17, PS26	PS8	PS23, PS24	X	
Intrinsic/ extrinsic reward	PS6, PS7, PS15, PS20, PS21, PS24	PS21	PS6, PS7, PS15, PS20, PS21	PS6, PS21	PS6, PS15, PS20, PS21, PS23, PS24	PS23, PS24	X

Phases three and four were performed alongside each other. Segments coded under 'Game Development Attributes' and 'Open-Source Attributes' were once again reviewed to determine which themes fit each primary study. Final themes were iterated upon during this process, as some of the segments added nuance that warranted a shift in the wording. Theme 1 was initially derived from Figure 2, as the codes with the highest number of occurrences came from 'Game Development Attributes', 'Open-Source Attributes ', and 'Methods/Concepts' codes. During phase four, the theme was altered to better suit the segments collected from the primary studies. Theme two was revised to better suit the studies after reviewing PS19, which discussed an open-source framework for developing educational games instead of developing a specific learning tool (Xu et al., 2013). Theme three was altered to better fit the breadth of technological applications gathered from segments in phase three. The rest of the themes remained unchanged from their initial versions.

DISCUSSION

Factors contributing to open-source game development were categorized as themes in the results of the study. This includes adapted approaches from open-source communities, educational purposes, the development process, cost-effectiveness, passion for a project, academic research potential, and intrinsic/extrinsic rewards. Each of the themes found through applying the methodology points to factors that contribute to the evolution of open-source tools for game development. The discussion will give some context to each theme and offer a contributing factor to answering the research question.

ADAPTED APPROACHES

Open-source offers several well-documented approaches adapted to game development. This theme was identified due to a large amount of discussion around how open-source development methodologies were being applied to game development projects. A segment from PS6 directly states that "No prior model or globally accepted framework exists that defines how FOSS is developed in practice. The starting point is to investigate FOSS practices in different communities (Scacchi, 2004). PS7 discusses that there are tendencies in individual communities that are like practices found in formal

software and game development (Pereira & Silva Bernardes, 2021). One of these is discussed in detail in PS24, which explains that members of an open-source community can play several roles in the development process (Costa & Costa, 2013). This leads to a contribution-focused community that lets contributors with the largest impact have the most effect on guiding the development of features for game creation tools (Costa & Costa, 2013). PS28 discusses that open-source projects divide and conquer tasks based on personal interest, thus only features and tools that fit the goals of the community are likely to be successfully created (Pascarella et al., 2018). As seen in the studies discussed, this theme of inheriting the development processes of open-source software is an important factor in the evolution of open-source game development tools.

Education

Educational needs are met through applying open-source principles to game development. This theme addressed a common topic of segments regarding how open-source development tools were built to suit education within the space of game development. These needs are met through applying open-source principles to game development. PS2 gives a rundown of Sugar, an open-source game development tool that was built by students who then returned to educate new students on the use of the tool (Jacobs, 2010). Additionally, PS10 mentions more abstractly that this is a common way to educate students and that educators also teach using open-source libraries (Lothian & Ryoo, 2013). PS17, PS21, and PS25 present RISC-V, HEAT, and GameX, three tools created for educational game development. RISC-V is an emulation tool (Nitta et al., 2022), HEAT is a 3D game engine built inexpensively from only open-source libraries (Stowell et al., 2009), and GameX is an instructional game engine (Hoetzlein & Schwartz, 2005). PS19 discussed a serious games framework for educating players on sustainability (Xu et al., 2013). With the primary studies discussed in mind, open-source game development tools will often factor in educational uses while innovating upon tools and games or when creating new ones.

Development Process

The development life cycle and post-release of games benefit from implementing open-source technologies into their tools. The development of tools, as mentioned in PS6, generally targets the community in which they arise (Scacchi, 2004). PS7 and PS15 show a specialized example of this in game development through modding or add-ons. This is a user-driven initiative to create custom tools or auxiliary content for existing games that overlaps with open-source on a cultural level (Isitan et al., 2011; Pereira & Silva Bernardes, 2021). The remaining primary studies all discuss open source platforms or engines, all created to address the need for fulfilling purposes or generating features that commercial tools lack. Thus, another factor in evolving open-source game development tools is the need to fulfill niche purposes for a community that developers of commercial tools have not or will not implement.

Cost-Effective

Open-source game development tools provide cost-effective alternatives to commercial software licenses. Since open-source software is freely accessible, it is more cost-effective than commercial alternatives, which often offer fewer customization options and lack community support. PS4 and PS21 comment that independent games studios and educational institutions often do not have the personal capital or monetary support of a publisher for their needs. Thus, they turn to free, open-source alternatives for their development needs (Kavouras et al., 2023; Stowell et al., 2009). Linking in with themes of education and development processes, PS6 notes that, on top of being usually cheap or free, open-source solutions are a great way to acquire niche tools for game development that also come with community support (Scacchi, 2004). PS4 and PS8 both report on game development tools that were made using open-source software, an emulator, and a virtual reality development environment, where the only costs related to the hardware used in PS8 (McMahon & Schukat, 2018). Once more linking in with theme 2, PS21 mentions that the ability to use open-source solutions for educational game development would be a cost-saving measure for underfunded programs

(Stowell et al., 2009). Cost-effectiveness shows that a large factor for evolution in the open-source game development space is by making a variety of tools accessible regardless of budget.

Passion

Passion from online communities generates open-source development tools and add-ons for existing games. This theme is closely tied to the previous theme of adapted approaches and how they contribute to the evolution of tools. PS1 and PS6 assert that aspirations around a given subject greatly impact the innovations made to a tool (Agrahari et al., 2023; Scacchi, 2004). PS6 mentions how shared interests can rejuvenate a community around closed-source software if the code is redistributed and opened to the public (Scacchi, 2004). Another element of the theme of passion is the availability of community experts. In the case of PS16, the authors specifically mention participating in researching closed and open forums to gain context of the open-source game engine, Amethyst (Taibi et al., 2021). Furthermore, in PS21, the authors note the willingness of community members to assist with issues such as bugs or concerns of novice users (Ampatzoglou et al., 2012). Though it is yet to be discussed, themes of passion also tie into themes of intrinsic/extrinsic reward, as reward is another motivator for passion and investment into the development of open-source tools and communities (Piller et al., 2004). Community support for onboarding and troubleshooting makes working on tools in these open-source game communities much more accessible. Thus, the degree of passion that newcomers and open-source communities have for a project factors into its contributions to the evolution of open-source game development tools.

Academic Research

An area of academic research in digital games uses open-source games/tools to study the field by analyzing code and design patterns. This theme is derived from varied types of academic study in game development being linked to the availability of source code for games, game platforms, and tools. Traditional game development is difficult to study because it is closed-source and can't be accessed (Ullmann et al., 2023). PS1 performs a thematic analysis to find anti-patterns in open source games, which yields a list of bad development practices, challenges, and solutions (Agrahari et al., 2023). PS2 and PS22 both identify reusable software components by looking at the source code of different open-source games (Ampatzoglou et al., 2012; Jacobs, 2010). Research isn't limited to analyzing code; tools like the virtual reality development environment introduced in PS8 or the open-source community modeling approach discussed in PS24 are also academic research projects in open-source game development (Costa & Costa, 2013; McMahon & Schukat, 2018). The studies tied to this theme show different ways research can utilize open-source code and communities in order to propose evolutions and innovations of open-source game development tools.

Intrinsic and Extrinsic Reward

Open-source tool and game developers receive various intrinsic and extrinsic rewards for their work. This theme works closely with the prior theme of passion, but was distinct enough to warrant its own discussion. PS6, PS7, and PS24 explain that some open-source community members in game development spaces receive offers of employment for their work on open-source game development (Pereira & Silva Bernardes, 2021), or may even just enjoy contributing to something they are passionate for (Costa & Costa, 2013; Scacchi, 2004). Some participants in the community contribute because they want to improve the service they use (Piller et al., 2004). Conversely, in cases like PS21, a project will often fail if motivation is lost, as there is no reinforcement to keep working on a given project (Stowell et al., 2009). Intrinsic and extrinsic rewards can factor positively or negatively in the development of open-source game development tools, improving the offerings of a project or hindering its progress.

After reviewing the seven themes, we found that open-source game development evolves similarly to other open-source projects. They are often subject to the same community formats and contain similar methods of evolving their tools. These factors range from the passions and motivation of a

community to develop a solution, how the open-source methodologies apply to developing game creation tools, what needs exist in game development research and education, and how to make tools more financially accessible.

LIMITATIONS

There are several limitations to this study. One limitation was a degree of selection bias due to the thematic analysis and quality assurance performed by the first and second authors, respectively. The study also uses only the ACM and Scopus databases, where a more comprehensive literature review could be obtained with more sources of information. Additionally, discussion during the author's quality checks and the application of inclusion criteria made each independent review of the literature subject to the other's bias. There are also threats to the external validity of the study, given that themes are derived from two databases, which may be misaligned with others. The reliability of the study is outlined in regard to the search for literature, but the study lacks structure regarding the application of quality assessment and coding in phase one of thematic analysis.

CONCLUSIONS AND FUTURE WORK

Through this systematic literature review and subsequent thematic analysis, findings identified seven factors that contribute to the continued evolution of open-source game development tools: adapted approaches, education, development processes, cost-effectiveness, passion, academic research and intrinsic/extrinsic reward. Open-source game development inherits development processes from open-source software and works to support developed tools after release through modding or extending software to fit user needs. Open-source game development also factors into game development education and research with customized tools and freely available datasets for study that yield insights for game development practices. Open-source game development also provides a means for educational institutions and independent developers to access game development tools, so cost-effectiveness is an important factor in the development and availability of these tools. Lastly, the successful development of open-source game development tools is rewarding, offering developers a chance to expand their portfolio, improve a service they use, or contribute to a cause they care about.

In future work, a revised methodology will include more thorough inclusion and exclusion criteria, and options for an approach that incorporates interviews with industry professionals, seminal works, and more industry-focused literature will be explored. More rigorous application of thematic analysis tools will be used to acquire and organize supporting segments further. Data sources will be expanded beyond ACM and Scopus libraries. Subthemes will be used to elaborate further upon the themes found in this work. Furthermore, a more rigid metric for accurately performing coding and quality assessment will be provided.

REFERENCES

ACM Digital Library. (2025). https://dl.acm.org/

Activision Blizzard. (2025). *Who we are.* Activision Publishing. https://www.activisionblizzard.com/who-we-are

Agrahari, V., Shanbhag, S., Chimalakonda, S., & Rao, A. E. (2023). A catalogue of game-specific anti-patterns based on GitHub and Game Development Stack Exchange. *Journal of Systems and Software*, *204*, 111789. https://doi.org/10.1016/j.jss.2023.111789

Ampatzoglou, A., Stamelos, I., Gkortzis, A., & Deligiannis, I. (2012). A methodology on extracting reusable software candidate components from open source games. *Proceeding of the 16th International Academic MindTrek Conference* (pp. 93–100). Association for Computing Machinery. https://doi.org/10.1145/2393132.2393152

Blender Foundation. (2025). *The freedom to create.* https://www.blender.org/about/

Braun, V., & Clarke, V. (2022). *Thematic analysis: A practical guide*. Sage. https://doi.org/10.53841/bpsqmip.2022.1.33.46

Coleman, R., Roebke, S., & Grayson, L. (2005). GEDI: A game engine for teaching videogame design and programming. *Journal of Computing Sciences in Colleges*, *21*(2), 72-82

Costa, C. J., & Costa, P. (2013). Developing a game in the context of an open source community. *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 108–112). Association for Computing Machinery. https://doi.org/10.1145/2503859.2503876

Elsevier. (2025). *Scopus: A comprehensive abstract and citation database for impact makers*. https://www.elsevier.com/products/scopus

Gacek, C., & Arief, B. (2004). The many meanings of open source. *IEEE Software*, *21*(1), 34–40. https://doi.org/10.1109/ms.2004.1259206

Godot Engine. (2025). *Your free, open-source game engine*. https://godotengine.org/

Gonzalez-Barahona, J. M. (2021). A brief history of free, open source software and its communities. *Computer*, *54*(2), 75–79. https://doi.org/10.1109/MC.2020.3041887

Goulding, T. (2008). Complex game development throughout the college curriculum. *ACM SIGCSE Bulletin*, *40*(4), 68–71. https://doi.org/10.1145/1473195.1473223

Hadzinsky, C. (2014). *A look into the industry of video games past, present, and yet to come* [Bachelor of Arts Thesis, Claremont McKenna College] https://scholarship.claremont.edu/cmc_theses/842/

Hoetzlein, R. C., & Schwartz, D. I. (2005). GameX: A platform for incremental instruction in computer graphics and game design. *ACM SIGGRAPH Educators Program* (p. 36). Association for Computing Machinery. https://doi.org/10.1145/1187358.1187402

Holliday, M. A. (1995). *Incremental game development in an introductory programming course* (pp. 170–175). Association for Computing Machinery. https://doi.org/10.1145/1122018.1122049

Isitan, K., Nummenmaa, T., & Berki, E. (2011). Openness as a method for game evolution. In K. Blashki (Ed.), *Proceedings of the IADIS International Conference Game and Entertainment Technologies* (pp. 100-104). Tampere University.

Jacobs, S. (2010, December). Building an education ecology on serious game design and development for the One Laptop Per Child and Sugar platforms: A service learning course builds a base for peer mentoring, Cooperative Education internships and sponsored research. *Proceedings of the 2nd International IEEE Consumer Electronics Society's Games Innovations Conference*, *Hong Kong, China*, 1–6. https://doi.org/10.1109/ICE-GIC.2010.5716882

Kavouras, I., Rallis, I., Doulamis, A., & Doulamis, N. (2023). Evaluating the feasibility of fast game development using open source tools and AI algorithms. In A. Krouska, C. Troussas, & J. Caro (Eds.), *Novel & Intelligent Digital Systems: Proceedings of the 2nd International Conference* (pp. 124–133). Springer. https://doi.org/10.1007/978-3-031-17601-2_13

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *Keele University and Durham University Joint Report*.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, *51*(1), 7–15. https://doi.org/10.1016/j.infsof.2008.09.009

Lothian, J., & Ryoo, J. (2013). Critical factors and resources in developing a Game-Based Learning (GBL) environment using Free and Open Source Software (FOSS). *International Journal of Emerging Technologies in Learning*, *8*(6), 11. https://doi.org/10.3991/ijet.v8i6.2918

MAXQDA. (2025, January 16). *All-in-one qualitative & mixed methods data analysis tool*. https://www.maxqda.com/

McMahon, M., & Schukat, M. (2018). A low-cost, open-source, BCI-VR game control development environment prototype for game based neurorehabilitation. *Proceedings of the IEEE Games, Entertainment, Media Conference*, *Galway, Ireland,* 1–9. https://doi.org/10.1109/GEM.2018.8516468

Michie, D. (1982). Game playing programs and the conceptual interface. *ACM Sigart Newsletter*, 80, 64–70.

Murphy-Hill, E., Zimmermann, T., & Nagappan, N. (2014). Cowboys, ankle sprains and keepers of quality: How is video game development different from software development? *Proceedings of the 36th International Conference on Software Engineering* (pp. 1–11). Association for Computing Machinery. https://doi.org/10.1145/2568225.2568226

Nitta, C., Kaloti, A., & Wang, S. (2022). RISC-V Console: A Containerized RISC-V Based Game Console Emulator for Education. *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1* (pp. 145–150). Association for Computing Machinery. https://doi.org/10.1145/3502718.3524791

Pascarella, L., Palomba, F., Di Penta, M., & Bacchelli, A. (2018). How is video game development different from software development in open source? *Proceedings of the 15th International Conference on Mining Software Repositories* (pp. 392–402). Association for Computing Machinery. https://doi.org/10.1145/3196398.3196418

Pereira, L. S., & Silva Bernardes, M. M. E. (2021). Modding as game development: Investigating the influences over how mods are made. *Journal of Gaming & Virtual Worlds*, *13*(2), 151–172. https://doi.org/10.1386/jgvw_00034_1

Piller, F., Ihl, C., Fuller, J., & Stotko, C. (2004, January). Toolkits for open innovation – The case of mobile phone games. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA*. https://doi.org/10.1109/HICSS.2004.1265487

Rouse, R. (2000). Gaming and graphics: Toiling with tools. *ACM SIGGRAPH Computer Graphics*, *34*(1), 5–9. https://doi.org/10.1145/563788.563790

Scacchi, W. (2004). Free and open source development practices in the game community. *IEEE Software*, *21*(1), 59–66. https://doi.org/10.1109/MS.2004.1259221

Smithsonian. (2024). *Magnavox Odyssey video game unit, 1972*. National Museum of American History. https://americanhistory.si.edu/collections/object/nmah_1302004

Stowell, T., Scoresby, J., Capell, M. R., & Shelton, B. E. (2009). Utilizing readily available and open source libraries to create a 3D game engine. *International Journal of Gaming and Computer-Mediated Simulations*, *1*(4), 20–49. https://doi.org/10.4018/jgcms.2009091502

Taibi, D., Lenarduzzi, V., Kilamo, T., & Zacchiroli, S. (Eds.). (2021). *Open source systems: Proceedings of the 17th IFIP WG 2.13 International Conference*. Springer. https://doi.org/10.1007/978-3-030-75251-4

Toftedahl, M. (2019, September 30). *Which are the most commonly used game engines?* https://www.gamedeveloper.com/production/which-are-the-most-commonly-used-game-engines-

Ullmann, G. C., Guéhéneuc, Y.-G., Petrillo, F., Anquetil, N., & Politowski, C. (2023, May). An exploratory approach for game engine architecture recovery. *Proceedings of the IEEE/ACM 7th International Workshop on Games and Software Engineering, Melbourne, Australia*, 8–15. https://doi.org/10.1109/GAS59301.2023.00009

Unity Technologies. (2024a). *Unity asset store*. https://assetstore.unity.com/

Unity Technologies. (2024b). *Developer relations*. https://github.com/orgs/UnityTechnologies/repositories

Unreal Engine. (2025). *Accessing Unreal Engine source code on GitHub*. https://www.unrealengine.com/en-US/ue-on-github

Vanhatupa, J.-M. (2011). Game engines in game programming education: Experiences from use of the CAGE game engine. *Proceedings of the 11th Koli Calling International Conference on Computing Education Research* (pp. 118–119). Association for Computing Machinery. https://doi.org/10.1145/2094131.2094156

Xu, Y., Johnson, P. M., Moore, C. A., Brewer, R. S., & Takayama, J. (2013). SGSEAM: Assessing serious game frameworks from a stakeholder experience perspective. *Proceedings of the First International Conference on Gameful Design, Research, and Applications* (pp. 75–78). https://doi.org/10.1145/2583008.2583018

APPENDICES

APPENDIX A: FULL LIST OF PRIMARY STUDIES

ID	Primary study title
PS1	A catalogue of game-specific anti-patterns based on GitHub and Game Development Stack Exchange
PS2	Building an Education Ecology on Serious Game Design and Development for the One Laptop Per Child and Sugar Platforms
PS3	A Case Study on the Availability of Open-Source Components for Game Development
PS4	Evaluating the Feasibility of Fast Game Development Using Open Source Tools and AI Algorithms
PS5	Exploring Indie Game Development: Team Practices and Social Experiences in A Creativity-Centric Technology Community
PS6	Free and Open Source Development Practices in the Game Community
PS7	Modding as game development: Investigating the influences over how mods are made
PS8	A low-Cost, Open-Source, BCI-VR Game Control Development Environment Prototype for Game based Neurorehabilitation
PS9	An Exploratory Approach for Game Engine Architecture Recovery
PS10	Critical Factors and Resources in Developing a Game-Based Learning (GBL) Environment Using Free and Open Source Software (FOSS)
PS11	Evaluation of object-oriented design patterns in game development
PS12	From Technomania to the School of Things: Taking Control of Your Own Game
PS13	Mob Programming and Simultaneous Style Pair Programming in the Development of a Battle Royale Game: An Action Research
PS14	Mobile Bluetooth-Based Game Development using Arduino on Android Platform
PS15	Openness as a Method for Game Evolution
PS16	OSS Scripting System for Game Development in Rust
PS17	RISC-V Console: A Containerized RISC-V Based Game Console Emulator for Education
PS18	RiverGame - a game testing tool using artificial intelligence
PS19	SGSEAM: Assessing Serious Game Frameworks from a Stakeholder Experience Perspective
PS20	Toolkits for Open Innovation - The Case of Mobile Phone Games
PS21	Utilizing Readily Available and Open Source Libraries to Create a 3D Game Engine
PS22	A Methodology on Extracting Reusable Software Candidate Components from Open Source Games
PS23	Collaborative Framework for Browser Games Development
PS24	Developing a Game in the Context of an Open Source Community
PS25	GameX: A Platform for Incremental Instruction in Computer Graphics and Game Design
PS26	GamingAnywhere: The First Open Source Cloud Gaming System
PS27	Gedi: A game engine for teaching videogame design and programming
PS28	How Is Video Game Development Different from Software Development in Open Source?

APPENDIX B: SEARCH QUERIES

Query	ACM format	Scopus format
Query 1	Title: (games) AND Abstract: ("open-source" OR "open source") AND Fulltext: (programming)	TITLE(games) AND ABS("open-source" OR "open source") AND ALL(programming)
Query 2	Title: (game development) AND Abstract: ("open-source" OR "open source") AND Fulltext: (programming)	TITLE(game development) AND ABS("open-source" OR "open source") AND ALL(programming)
Query 3	Title:("game" AND "development") AND-Title:("open source" OR "open-source") AND Abstract:("open source" OR "open-source") AND Keyword:("open source" OR "open-source") AND AllField:("evolution" OR "history" OR "review")	TITLE(game AND development) AND \newline TITLE-ABS-KEY("open source" OR "open-source") AND ALL("evolution" OR "history" OR "review")

AUTHORS

Howard Hall is a first-year Information Technology PhD student at the University of Cincinnati. He graduated from his IT undergraduate program in the Spring of 2024 and now works as a Graduate Teaching Assistant in the University of Cincinnati School of Information Technology. Howard has experience with projects centered around game development and design. Howard has collaborated on research projects in the civic technology space, such as common operational picture tools for crisis response and habitat preservation and monitoring tools for the Ohio Department of Transportation.

Abigail Dogbe is a first-year Information Technology PhD student at the University of Cincinnati. She successfully defended her thesis and graduated from her IT master's program in the Summer of 2024. She is now a graduate research assistant at the University of Cincinnati School of Information Technology. She is devoted to open-source and her community, highlighting open-source projects and communities through her "Everything Open Source" initiative and empowering women through her work with PyLadies Ghana.

Dr Hazem Said is a Professor of Information Technology and the director of the School of Information Technology at the University of Cincinnati. Dr Said founded the UC Information Technology Solutions Center in 2012, consulting with government, public, and private organizations, and leads teams to investigate, develop, and support a variety of information technology solutions. Dr Said is a co-founder and co-director of the Ohio Cyber Range Institute and the Justice, Law, and Information Technology Institute. Dr Said is the recipient of over 200 grants and contracts totaling over $30 million and has authored over 27 articles on topics related to information technology education.