Revista ARACÊ
ISSN: 2358-2472

AN ANALYSIS OF THE APPLICATION OF GAMES FOR TEACHING BASIC PROGRAMMING FROM A SYSTEMATIC MAPPING OF THE LITERATURE

https://doi.org/10.56238/arev7n1-176

Submitted on: 12/22/2024 **Publication date:** 01/22/2025

Alexsandro Monteiro Nascimento Callins[1], Davi Alberto Correa do Carmo[2] and Julio Cezar Costa Furtado[3].

ABSTRACT

This article performs a systematic mapping of the use of digital games in programming education, analyzing publications from 2013 to 2024. The study seeks to identify which games are used to teach programming concepts and in which areas these games have an educational impact. Using strict inclusion and exclusion criteria, the research selected articles in databases such as ACM Digital Library and IEEE Xplore. The mapping identified 49 games aimed at teaching programming, with a predominance of genres such as indie, educational and simulation, available on platforms such as Windows, MacOS, Linux and the web. The areas of knowledge covered by the games include programming logic, data manipulation, syntax, functions, and debugging, promoting the development of these skills in a motivating environment. The results suggest that digital games are effective tools to engage students and facilitate the learning of abstract concepts in programming. Offering interactive practice and immediate feedback, these games complement traditional teaching methods, making learning more accessible and dynamic. The research concludes that the integration of games in programming education can increase students' interest and support the assimilation of complex content. In addition, it is recommended to continue studies on the impact of games on long-term skills, aiming to adapt these tools to different educational levels and teaching contexts.

Keywords: Serious Games. Programming. Education. Teaching.

[1] Graduating in Computer Science
Federal University of Amapá
Email: alexsandromnc@gmail.com
[2] Graduating in Computer Science
Federal University of Amapá
E-mail: dcarmo200010.dd@gmail.com
[3] Dr. in Computer Science
Federal University of Amapá
E-mail: furtado@unifap.br

INTRODUCTION

Digital games are a powerful tool in playful learning and have been successfully applied in several areas of knowledge (ZYDA, 2005). With the growing and dynamic technological evolution applied to the educational environment, the forms of learning have become more accessible (METTLER; PINTO, 2015), and digital games are increasingly effective and integrated into everyday life as a teaching alternative. When used as pedagogical instruments, these games contribute to the democratization of knowledge, facilitating its absorption by users (GARCÊZ; OLIVEIRA, 2022). All this contributes to the world moving in a direction where education and knowledge become less privileges of a few (ZYDA, 2005).

Technological development in the digital environment collaborates directly with education, since the novelties in this field corroborate pedagogical advances (SAMPSON et al., 2009). This collaboration makes digital games more accessible, both in terms of cost and platforms, which facilitates the learning of abstract topics such as programming (REIS; MAN; ARANHA, 2013).

Teaching programming can be challenging, given that its approaches are often far from reality (DEITEL; DEITEL, 2009). However, many means have been developed to minimize these barriers present in the teaching of programming. With the advancement in the technological and educational environment, such barriers tend to be reduced (GARCÊZ; OLIVEIRA, 2022).

In this scenario, the objective of this research is not only to discuss the use of digital games in a generic way. The focus is to perform a systematic mapping (KITCHENHAM et al., 2009) to identify which existing serious games specifically support the process of teaching and learning programming logic. These serious games, aimed at educational purposes, are developed with the intention of overcoming the traditional difficulties of teaching programming, offering a more interactive and engaging approach for students (METTLER; PINTO, 2015). From this analysis, we seek to highlight the most effective games in this context, providing a relevant collection for teachers and students who wish to use these tools in the teaching of programming logic (REIS; MAN; ARANHA, 2013).

In this context, the objective of this research is not simply to discuss the use of digital games in learning in a generic way. The main focus is to carry out a systematic mapping to identify the existing serious games that specifically support the process of teaching and learning programming logic. These serious games, aimed at educational purposes, are developed with the intention of overcoming the traditional difficulties in teaching programming, offering a more interactive and engaging approach for students. From this analysis, the most effective games in this context are highlighted, providing a relevant collection for teachers and students who wish to use these tools in the teaching of programming logic.

THEORETICAL FOUNDATION

In this section of theoretical foundation, important topics are presented for a better understanding of the work.

INTRODUCTION TO THE TEACHING OF PROGRAMMING AND EDUCATIONAL GAMES

Programming has become, over time, an increasingly requested area in the job market. According to Infojobs, in 2023 alone, a 20% growth in the search for training related to the programming area is expected. In this way, to meet all this demand, the market mobilizes through a greater supply — whether for jobs, training courses, or studies to improve teaching and learning in the area.

Although there are several ways of learning in programming, many students face difficulties in assimilating issues related to this subject since the area of programming has concepts and applications that can be considered more abstract compared to other disciplines. As an alternative to diversify and complement the learning process, educational digital games stand out. According to Victal et al. (2015), digital games are very important tools for learning, which makes them indispensable in education.

SERIOUS GAMES

A game can be defined as a recreational and stimulating activity that involves any type of competition with a set of rules, these rules being executed in a restricted environment, in which there is social involvement in a determined space and time (ZUCARELLI and COUTO, 2013), usually with a specific objective.

On the other hand, a serious game is properly a game, but designed with a specific educational purpose. It is not focused on entertainment, but on teaching certain content or training skills that can be applied in the real world. Serious games aim to integrate entertainment and learning, enabling a more serious approach to the achievement of educational goals, through the use of conflicts and challenges. (METTLER and PINTO, 2015; ZYDA, 2005).

For better understanding of games and serious games, games are created with the primary purpose of providing fun, whereas serious games aim to teach specific skills effectively. In the educational context, games have been used as tools to stimulate students and facilitate the understanding of complex concepts. Serious games, on the other hand, in the context of programming, can simulate practical coding environments, helping in the application of theoretical concepts.

THE SYSTEMATIC MAPPING OF LITERATURE

A Systematic Literature Mapping (MSL) is an in-depth investigation of a phenomenon of interest that provides specific and thorough results through the evaluation of the content and quality of the researched material (Kitchenham et al., 2007, p.), this type of mapping emphasizes the importance of a careful analysis of the literature to obtain valuable insights. In order to obtain a comprehensive view of the use of games in the teaching of programming, it was decided to carry out a mapping. The time interval of the selected studies was delimited from March 2013 to March 2024. This choice was based on the premise that a 10-year period offers an up-to-date and comprehensive perspective for the research context, in addition to covering a relevant period of related studies.

PLANNING

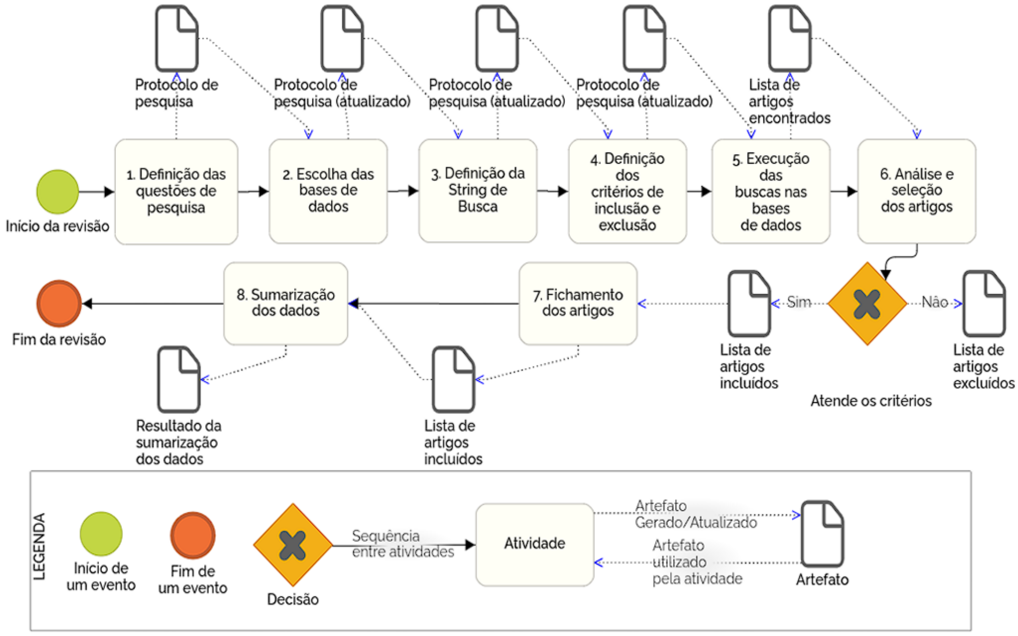
According to Grant, Mary & Booth, Andrew. (2009), factors such as the clear definition of the mapping objectives and the effective allocation of responsibilities are essential for the effectiveness of the process. The planning of the Systematic Mapping of Literature (MSL) was carried out with the purpose of identifying the fundamental stages for the effective realization of the research, analysis and summarization of the articles, as well as the elements contemplated in each stage. The planning and execution of this process aimed at the elaboration of a Protocol for Systematic Mapping of Literature (PMSL).

The PRSL procedures were defined through meetings between the researcher and the professor. The procedures can be seen in Figure 1, in their respective execution orders, as well as the relationship of the artifacts with their activities during the process.

Research questions

According to Petticrew, M., Roberts, H., research questions define what is intended to be discovered and the objectives to be achieved, being essential to ensure that the mapping is focused and relevant to the research objectives. This process is carried out through the formulation of a main question (QP) and secondary questions (QS). QP guides the research holistically, considering that the state of the art in the use of games in programming education involves several aspects. Thus, it is necessary to adopt scopes that delimit specific aspects and allow for a more thorough analysis. The QS, in turn, specify which aspects must be addressed so that the main question can be answered. The questions formulated for this study are:

Figure 1 - Stages of systematic mapping


Protocolo de pesquisa
Protocolo de pesquisa (atualizado)
Protocolo de pesquisa (atualizado)
Protocolo de pesquisa (atualizado)
Lista de artigos encontrados
Início da revisão
1. Definição das questões de pesquisa
2. Escolha das bases de dados
3. Definição da String de Busca
4. Definição dos critérios de inclusão e exclusão
5. Execução das buscas nas bases de dados
6. Análise e seleção dos artigos
Fim da revisão
8. Sumarização dos dados
7. Fichamento dos artigos
Sim
Não
Lista de artigos incluídos
Lista de artigos excluídos
Atende os critérios
Resultado da sumarização dos dados
Lista de artigos incluídos
LEGENDA
Início de um evento
Fim de um evento
Decisão
Sequência entre atividades
Atividade
Artefato Gerado/Atualizado
Artefato utilizado pela atividade
Artefato

Source: Prepared by the authors (2024)

- QP: What are the existing games that help in the teaching of the Programming discipline?
  - QP aims to identify available digital games that can contribute to the teaching of initial programming concepts, providing an overview of accessible options;
- QS1: What are the most used platforms?
  - QS1 seeks to identify the most frequently used platforms for the execution of games, in order to determine where they can be accessed;
- QS2: What are the commonly used types of games?

  - This question aims to specify which types of games were found in the analyzed works, as well as which are Open Source, which may be relevant to the accessibility and customization of educational resources;
- QS3: What is the language of the game?
  - QS3 aims to verify the languages in which the games are available, considering the issue of accessibility;
- QS4: What kind of knowledge does the game aim to train?
  - Through QS4, it seeks to understand what knowledge the games aim to develop as a primary objective, allowing the evaluation of the educational benefits offered by each game.

Definition of databases

The selection of databases should be based on the scope of the mapping and the objectives of the research, considering digital libraries, repositories of scientific articles, electronic journals, among others. The databases chosen for the search were ACM Digital Library and IEEE Xplore, due to the reliability and large number of published works in the programming area, in addition to being references for the elaboration of the curriculum in computing courses. Among the criteria used to define the database, we can list:

- Availability of access through the internet;
- The availability of these databases for free consultation/download of works for UNIFAP students.
- This search performed automated searches in the ACM Digital Library and IEEE Xplore databases through the use of the search string established in the process.

Search String

In this study, the IOP method - Population (Participants), Intervention (Type of Intervention) and Outcomes (Outcomes) was used to formulate the search string, using keywords relevant to the research context. Chart 1 presents the application of this method, in line with the objectives of this study.

Table 1 - IOP Method

Population (Participants)	Computer Academics.
Intervention (Intervenção)	Use of games to support the teaching of Programming.

Outcomes	1. Areas and knowledge of programming covered; 2. Teaching methodologies; 3. Lesson plan; 4. Gaming;

Source: Prepared by the authors (2024)

Creating the Precise Search String is essential to the scope of studies pertinent to the search. Based on the main question and the secondary questions previously established, a combination of keywords was performed to determine a search string that is pertinent. For string validation, the following criteria were established: return between 140 and 305 results obtained with the databases used; And, keywords that should contain in the returned works were determined, if many different keywords were still returned, it was essential to recalibrate the search string.

The search string is the combination of terms (keywords) and logical operators "OR", "AND" and "NOT", as shown in tables 2 and 3, which are used to retrieve relevant studies in the selected databases. In the context of the search method outlined by Malcher et al. (2015), the use of the 'OR' operator is imperative to associate keywords and synonyms, while the use of the 'AND' operator is appropriate when the intention is to associate a specific set of keywords with all the elements investigated. The generated string can be seen in the table below.

Table 2 - Search string

(("Learn to Code" OR "Programming Fundamentals" OR "Programming Fundamental" OR "Programming Basic" OR "Programming Basics" OR NOT("Object Oriented")) AND (teaching OR learning OR training OR education OR tutorial) AND (Game* OR Serious Game* OR Simulator*))

Source: Prepared by the authors. (2024)

Among the researched works, publications from conferences, magazines, newspaper articles and scientific journals were included, covering the period from March 2013 to March 2024, totaling an interval of 11 years.

Inclusion and exclusion criteria

The selection of studies was guided by the inclusion and exclusion criteria, which establish clear parameters for the definition of the studies to be considered. The rigorous application of these criteria is essential to ensure relevance and consistency in the choice of studies for review (HIGGINS, J.; GREEN, S, 2008).

After the automated search in the databases, the selection phase of the works began. The screening process (pre-selection) was divided into two distinct phases. In the first phase, the titles, abstracts and keywords were analyzed. In the second phase, a detailed review of the studies was carried out to assess whether they met the criteria established for exclusion and inclusion and exclusion, as outlined in Chart 3.

Table 3 - Selection criteria for primary studies

ID	Inclusion Criteria (CI) or Exclusion Criteria (CE)
CE.1	Articles that are not freely available for consultation or download (in full version) from research sources or through manual search (for articles that are not provided in full) performed in Google (http://www.google.com.br/) and/or Google Scholar (http://scholar.google.com.br/) search tools;
CE.2	Articles that clearly do not meet the research questions.
CE.3	Repeated articles (in more than one search source) had only their first occurrence considered;
CE.4	Studies framed as summaries, keynote speeches, courses, tutorials, workshops and the like;
CE.5	Articles that do not mention the keywords of the research in the title, abstract or keywords of the article, except for works that address improvement of the software process in which the possibility of Statistical Process Control being treated throughout the work is observed;
CE.6	Exclude if the study is not inserted in the context;
CE.7	Exclude if the study is not presented in one of the accepted languages (English and Portuguese).

Source: Prepared by the authors (2024)

RESULTS OF THE LITERATURE MAPPING

In this section, the results obtained from the execution of the planning are presented, with the objective of answering the main (QP) and secondary (QS) questions previously defined.

RESULTS OF PRIMARY STUDIES

The search strings applied to the ACM Digital Library and IEEE Xplore databases resulted in a total of 755 jobs. The ACM Digital Library database returned 290 articles (38.41%), presenting the lowest number of results, while the IEEE Xplore database presented a total of 465 articles (61.58%).

Table 1 shows the total number of articles classified according to the pre-selection criteria, Exclusion Criteria (CE) and Inclusion Criteria (CI). Each row represents the total number of studies obtained from each indexer. The pre-selection phase covered the largest number of articles (561) among the eliminated researches, followed by CE.2 and CE.1, with 165 and 17 exclusions, respectively. The data in column CI.1 indicate the studies that met the established inclusion criteria.

Also in table 1, it is observed that the IEEE Xplore search engine had the highest number of articles excluded (453), followed by CE.2, which represented the main criterion for the elimination of articles with 158 exclusions. On the other hand, the ACM Digital Library excluded 285 articles during the pre-selection, with CE.2 being the criterion with the highest rate of deletion of papers, totaling 7 exclusions.

Table 1 - Studies returned based on inclusion and exclusion criteria

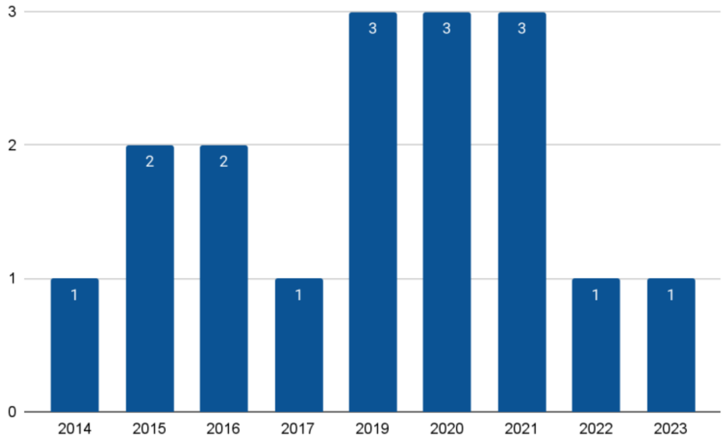
Primary Studies					
Source	Returned works	Screening	CE.2	CE.1	CI.1
IEEE Xplore	465	289	158	6	12
ACM Digital Library	290	272	7	6	5
Total	755	561	165	12	17

Source: Prepared by the authors (2024)

The result of the selection of studies resulted in 17 articles published between 2014 and 2023. Figure 2 illustrates the number of articles found per year. The years with the highest number of publications were 2019, 2020 and 2021, with 3 articles each. The years 2015 and 2016 had 2 articles each. On the other hand, the years 2014, 2017, 2022, and 2023 had the lowest rate of publications, with only 1 article in each of these years.

Regarding the distribution of the types of publications, the results show that most of the contributions were concentrated in conferences, with 13 articles (76.5%). In addition, there were publications in journals, with 2 articles (11.8%), journal articles and workshops, both with 1 article (5.9%).

Gráfica 1 - Publications per year



Source: Prepared by the authors (2024)

Table 2 classifies the participation events according to the type of publication mentioned above. Notably, in the case of conferences, the "IEEE Frontiers in Education Conference (FIE)" stood out with the largest number of papers, totaling 2 publications. This differs from the other events of other types of publications, where there was a numerical equality of 1 publication in each one. This distribution can be seen in Table 2, which presents the detailed data.

Table 2 - Places of publication.

Conference	Articles
IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)	1
IEEE Frontiers in Education Conference (FIE)	2
IEEE 20th International Conference on Advanced Learning Technologies (ICALT)	1
IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)	1
International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS)	1
ISDOC '14: Proceedings of the International Conference on Information Systems and Design of Communication	1
NordiCHI '16: Proceedings of the 9th Nordic Conference on Human-Computer Interaction	1
IDC '16: Proceedings of the The 15th International Conference on Interaction Design and Children	1
IDC '21: Interaction Design and Children	1
SoutheastCon	1
IEEE International Conference on Advanced Learning Technologies (ICALT)	1

XLIII Latin American Conference on Computing (CLEI)	1
Magazines	Articles
IEEE Access	1
IEEE Ibero-American Journal of Learning Technologies	1
Newspaper article	Articles
ACM Transactions on Computing Education	1
Workshop	Articles
IEEE Blocks and Beyond Workshop (B&B)	1

Source: Prepared by the authors (2024)

In addition, details on the universities, authors, and countries with the highest number of articles published on the subject can be found in Table 3. In the count, all the appearances of the elements were cataloged, regardless of whether they were authors or co-authors of the works. With regard to universities, all institutions of the selected works contributed equally, with each one presenting 1 publication related to the research theme. When it comes to the countries that contributed the most, the United States leads with 4 published surveys, followed by Germany, with 3 published articles. Regarding the authors who appeared the most in the selected papers, none of them was involved in more than 1 paper.

The data presented in Table 3 are valuable to identify the universities, authors and countries that stood out the most in the field addressed in this work. This information can serve as a source of guidance and collaboration for the development of related research.

Qp: What are the existing games that help in teaching the programming discipline?

This question aims to identify and catalog the available digital games that can contribute to the teaching of the initial concepts of programming, providing a comprehensive overview of the accessible options for educators and students.

The research resulted in the identification of a total of 49 games that stand out for their contributions to programming education. These games were selected based on strict criteria, which ensured the relevance and quality of the studies analyzed. The complete list of identified games, along with the areas of knowledge covered, can be consulted in Appendix A of this work. This information is valuable not only for educators looking to integrate games into their pedagogical practices, but also for researchers who want to further explore the potential of games in programming education.

Qs1: Which platforms are most used?

Taking into account the context of this research, it is essential to consider the platforms on which these games are available. Analysis of the results revealed a variety of options, with Windows being the most widely used platform, with a total of 26 (27.7%) games available. In addition, the MacOS operating system supports 20 games and SteamOS + Linux with 14 games, both aimed at desktop devices, and the web platform supporting 12 games. Figure 3 illustrates both the platforms mentioned above and others that do not support as many games as the 49 selected for this research.

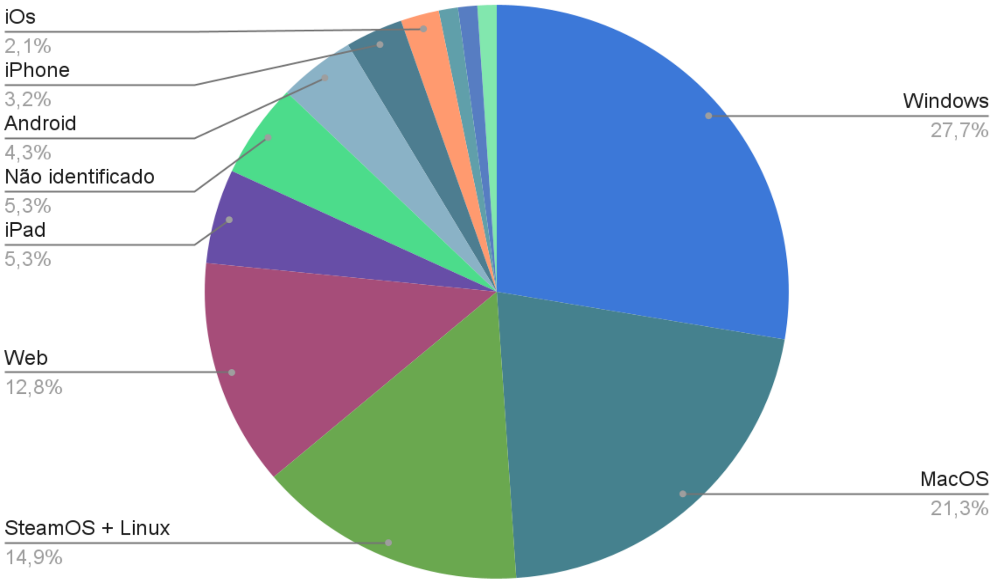
Qs2: What Are The Commonly Used Game Types?

Figure 4 reveals a varied distribution of the types of games most frequently used among the selected studies. "Indie" games lead the preference, accounting for 25 instances, possibly reflecting the growing popularity and diversity of indie developers in the gaming industry. Secondly, the educational games stand out, totaling 16 games that address this genre. In addition, there is a notable presence of simulation (10), puzzle (14) and adventure (7) games, indicating a wide variety of genres among the selected games.

The "Unidentified" categorization suggests the presence of games whose genre has not been clearly specified. This diverse landscape highlights the complexity and richness of today's gaming landscape, catering to a wide range of interests and purposes. The other types of games that did not appear very often can be seen in the figure below along with the others mentioned above.

The number of cited articles that are classified as Open Source were 4 games (Blockly Games, Pencil Code, Robot Instructus() and Stone Story RPG) that were used in the returned works, while 32 games could not be classified as Open Source.

Graph 2 - Most used platforms for games.

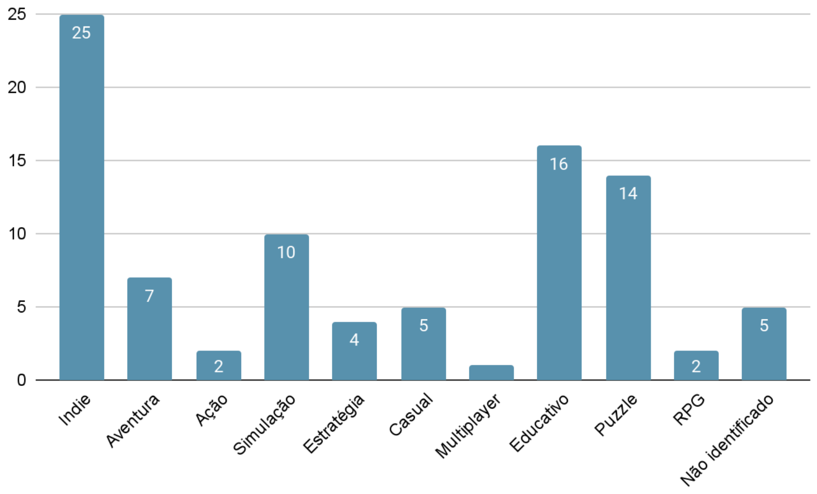


Source: Prepared by the authors (2024)

Qs3: What is the language of the game?

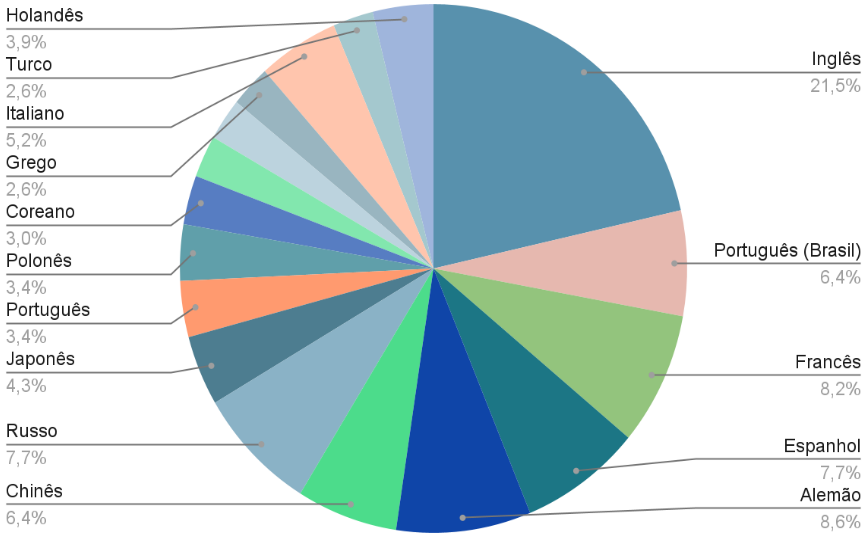
Figure 5 shows the languages available for the 49 selected educational games. It includes the 13 most frequently available languages, from those present in all games, such as English (with 49 appearances, 21.5%), to those that appear in at least 6 games. The most common languages include English, Portuguese (Brazil), French, Spanish, German, Chinese, and Russian.

Graph 3 - Types of games.



Source: Prepared by the authors (2024)

Graph 4 - Language of the games.



Source: Prepared by the authors (2024)

Qs4: What kind of knowledge does the game aim to train?

Based on the analyzed works, nine topics that are essential in learning programming were taken into account. The topics are broken down as follows: Programming logic, develops logical thinking and problem-solving skills; Syntax of programming languages, integrates syntax concepts into interactive challenges and tasks, students can learn meaningfully by applying this knowledge in a practical and realistic context; Data manipulation, players can learn to store, modify, and retrieve information, developing crucial programming skills; Functions and procedures, allows users to understand the importance of modularity and organization of code.

By creating and calling specific roles for distinct tasks in the game, learners can gain a practical understanding of code reuse and problem-solving efficiency; Error handling and debugging, when facing errors and bugs in the context of the game, players learn to identify, diagnose and correct problems in the code, developing essential skills for solving problems in programming; Practical projects, developing practical projects within the game allows the consolidation of knowledge and construction of a portfolio of tangible experiences; Adapting to new challenges, encourages flexibility and resilience, crucial qualities for developers who frequently face new problems and technologies; Immediate feedback, players can quickly adjust their approach, better understand concepts, and reinforce learning in a practical way; Repeating and condition structure helps players develop the logic needed to create more complex and efficient programs.

Below, in table 3, you can see the number of games found related to each knowledge mentioned above.

Table 3 - Number of games found by knowledge

Number of games found per Knowledge		
Programming logic 49 matches found	Programming Syntax 23 matches found	Data Manipulation 19 matches found
Roles and Procedures 24 matches found	Error Handling and Debugging 17 games found	Practical Projects 31 Games found
Adapting to new challenges 39 Games found	Immediate Feedback 28 Games found	Repeat and condition structures 29 Games found

Source: Prepared by the authors (2024)

The complete list of identified games, along with the areas of knowledge covered, can be consulted in Appendix A of this work.

ANALYSIS ON THE MAIN GAMES

This review explored some of the top educational games and tools aimed at learning to code. These games, in addition to providing an interactive experience, also address fundamental concepts such as logic, repetition structures, functions, and error handling, helping players develop their skills in a practical and fun way.

CodinGame is a platform that offers hands-on challenges in various programming languages. According to the official documentation, "CodinGame's challenges cover a wide range of programming concepts, allowing users to practice everything from basic logic to advanced data structures" (CodinGame, 2024). Its biggest differential is immediate feedback, which allows users to correct their errors instantly and improve their skills in logic and programming syntax, as well as work with repetition structures, conditions, and data manipulation in real time. The platform also covers advanced functions and algorithms, which makes it suitable for both beginners and experienced programmers. Adapting to new challenges is made easier by a wide range of exercises ranging from easy to hard.

Developed by Microsoft Research, Code Hunt teaches programming through puzzles that require the correction of pre-written code. As officially documented, "Code Hunt offers a hands-on approach to teaching programming through interactive challenges that reinforce logic, the use of repetition structures, and error handling" (Microsoft Research, 2024). One of the game's strengths is its adaptation to new challenges and immediate feedback, allowing players to test their solutions in real-time and learn from their mistakes. The game focuses on programming logic, repetition structures, and error handling through intuitive debugging. The environment is ideal for practicing programming syntax, with a focus on loops and conditions.

Wu's Castle teaches programming logic in an adventure setting. As described in the documentation, "Wu's Castle provides an immersive environment in which programming logic and data manipulation are introduced gradually, through a fun and challenging narrative" (Wu's Castle, 2024). The game focuses on data manipulation and control flow concepts through repetition and conditional structures. Its great differential is the engaging narrative, which makes learning fun and accessible, while encouraging the use of functions and procedures to solve the challenges presented.

Code World uses code blocks to teach programming in a visual way. The documentation states that "Code World allows beginners to visualize the flow of control and understand code behavior in a clear and intuitive way" (Code World, 2024). Through logic blocks, users learn control flow, command sequence, repetition structures, and conditions. The platform is excellent for beginners, as it offers a clear and intuitive understanding of code behavior and allows them to adapt to new challenges as students progress.

In Coding Planets, players control robots on different planets, solving problems that require logic and the use of repetition structures and conditions. According to the official documentation, "Coding Planets focuses on the development of logical reasoning through the practical application of loops, conditions, and programming logic in challenging environments" (Coding Planets, 2024). The platform helps to reinforce logical reasoning and encourages the use of functions for more complex solutions. Adapting to new challenges is central to the game's mechanics, as each planet presents new problems and difficulty levels.

Focused on functions and procedures, the Instructus Robot challenges players to find creative solutions to the problems presented. According to the documentation, "Instructus Robot encourages creativity in problem-solving while challenging users to optimize their solutions for maximum efficiency" (Instructus Robot, 2024). The platform is ideal for those who want to learn code optimization and project organization efficiently. The game offers continuous feedback on proposed solutions, encouraging the process of trial and error.

In Debugger 3.16: Hack'n'Run, players act as debuggers, learning about debugging and error handling in real-time. The documentation highlights that "Debugger 3.16 puts the player in the position of a debugger, where they must identify and correct programming errors in real time, learning the value of optimization and constant testing" (Debugger 3.16, 2024). The game is designed to teach you the importance of testing, fixing, and optimizing code in a practical way. It is an excellent tool for those who want to improve their skills in finding and fixing programming errors, with a focus on logic, loops, and conditions.

Karel is a classic educational tool that introduces the fundamentals of programming, such as syntax, loops, conditionals, and error handling, in a simple and controlled way. According to the documentation, "Karel the Robot is designed to help students understand the basics of programming through performing simple and controlled tasks in a user-friendly environment" (Karel, 2024). Players program a robot to perform specific tasks, making it easier to learn logic and flow control. The simplicity of the tool is a strength for beginners who are just starting to understand how to program.

Using visual programming, Glitch Space teaches concepts of functional logic and data flow through puzzles. According to the official documentation, "Glitch Space utilizes a visual programming system to help players understand how to manipulate data and solve problems using functional logic" (Glitch Space, 2024). The game offers a tangible learning experience, with an intuitive interface that makes it easy to understand how data can be manipulated to solve problems. The focus is on programming logic and the use of code blocks to transform data efficiently.

Screeps: World is a multiplayer platform that allows players to program colonies in JavaScript, which requires constant adaptation to a dynamic and competitive environment. According to the official documentation, "Screeps is a real-time programming game, where players control colonies entirely through scripts written in JavaScript, promoting complex problem-solving in a multiplayer environment" (Screeps, 2024). The game encourages continuous code optimization and the use of functions and procedures for efficient resource control. The game's complexity challenges players to solve problems creatively and tackle hands-on projects in a real-time environment.

Although the games analyzed above have their strengths highlighted, it is worth remembering that they are not limited to this specific knowledge. These are treated as main because they simultaneously cover the nine indicators selected after a systematic mapping of the literature, which contribute significantly to the teaching of basic programming.

CONCLUSIONS

This study highlighted the effectiveness and potential of digital games as specific pedagogical tools for teaching programming. From a systematic review of the literature, a set of 49 distinct games was identified, categorized by genre, platform and type of knowledge addressed, revealing themselves to be widely adapted for the teaching of key concepts in programming, such as logic, data manipulation, syntax, functions and error handling. The diversity of these games, combined with their interactive and practical characteristics, allows for an instructional approach that not only complements traditional methods, but also provides a dynamic and highly engaging learning environment.

Despite the positive results, the present study has some limitations that should be considered when interpreting its results. First, the systematic review was carried out exclusively in the ACM Digital Library and IEEE Xplorer databases, chosen for their relevance in the area of computing. However, this choice restricts the scope of the studies analyzed, excluding works that could be found in other academic databases. In addition, the focus of this review was directed to the use of games for the teaching of basic programming concepts, such as logic, data manipulation and syntax, limiting the applicability of the results in intermediate or advanced levels of programming. More complex concepts, such as object-oriented programming, data structures, and advanced algorithms, require specific approaches and, possibly, differentiated games. Therefore, future studies could explore games that address these contents to assess their effectiveness in teaching more sophisticated skills in programming. Finally, there is the issue of temporal influence. The period covered by the review, while comprehensive, may reflect a recent or one-off trend in educational game development. Thus, it is possible that rapid changes in the game market and pedagogical practices affect the relevance of the findings over time, requiring periodic reviews to keep the study up to date with emerging technological and educational innovations.

As for the impact of the work, from an academic point of view, this study contributes substantially to the field of Computer Education, offering a careful and comprehensive analysis of the use of games in the teaching of programming. By mapping and systematizing the available games, this research provides a solid basis for educators and institutions to incorporate, with a theoretical and practical basis, game resources in their curricula. This detailed survey allows future researchers and teachers to adopt a more informed view of which games are most effective in each area of programming, thus optimizing teaching practices and enhancing the development of critical skills.

In addition, the work contributes to the expansion of research frontiers in the area of serious games, reinforcing the need for deeper investigations into the effectiveness of digital games and their applicability in the academic context. The results of this systematic review also indicate that the use of games in teaching can be an effective strategy to increase student retention and performance in subjects with a high level of abstraction, such as programming.

Socially, the use of games in the teaching of programming represents a democratizing approach, by facilitating access to technical knowledge in an inclusive and attractive way. Digital games, due to their accessibility on widely used platforms (such as Windows, MacOS and Linux) and their versatility of application, have the potential to reduce barriers to entry for learning in technology, particularly in communities with limited access to formal educational resources. By making programming teaching more interactive and motivating, games contribute to more effective digital inclusion, enabling individuals from different backgrounds and socioeconomic backgrounds to develop valuable skills for the job market. Thus, the implementation of educational games in programming can be seen as a tool for social transformation, with the potential to positively impact the training of trained professionals and reduce inequalities in access to technological education.

This study paves the way for multiple strands of future research. It is recommended that further research explore the effectiveness of digital games for the development of programming skills in the long term, with a focus on longitudinal studies that allow the evaluation of students' progress and the retention of knowledge acquired through these games.

In addition, it would be relevant to investigate the potential of personalization of games for different learning styles and student profiles, in order to optimize the adaptation of programming teaching to a heterogeneous audience. Another promising axis for future research involves the analysis of the psychological and motivational impacts of the use of games in teaching, understanding how these resources can influence students' self-esteem, continued interest and perception of effectiveness in the area of computing.

These suggested directions aim to strengthen the understanding of the role of digital games as facilitators in learning to program and consolidate innovative pedagogical practices that are adaptable and efficient for the various educational contexts. Continued research in this field is essential to ensure that new generations have access to technology learning that is both technically robust and socially inclusive

REFERENCES

1. Code World. (2024). Code World: Visual programming environment. https://www.codeworld.com/

2. CodinGame. (2024). CodinGame: Programming challenges. https://www.codingame.com/

3. Coding Planets. (2024). Coding Planets: Logic-based problem-solving. https://www.codingplanets.com/

4. Debugger 3.16. (2024). Debugger 3.16: Real-time debugging game. https://www.debugger316.com/

5. Deitel, P., & Deitel, H. (2009). Como programar em C (3rd ed.). Pearson Prentice Hall.

6. Garcêz, A. V., & Oliveira, J. M. L. (2022). The benefits of using digital games to teach programming to children. Research, Society and Development, 11(17), Article e123111735149.

7. Glitch Space. (2024). Glitch Space: Visual programming puzzles. https://www.glitchspace.com/

8. Grant, M., & Booth, A. (2009). A typology of reviews: An analysis of 14 review types and associated methodologies. Health Information and Libraries Journal, 26(2), 91–108. https://doi.org/10.1111/j.1471-1842.2009.00848.x

9. Higgins, J., & Green, S. (2008). Cochrane handbook for systematic reviews of interventions. Wiley-Blackwell.

10. Karel. (2024). Karel the Robot: Educational programming tool. https://www.karelrobot.com/

11. Kitchenham, B., Brereton, P., 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

12. Malcher, P., Ferreira, D., Oliveira, S., & Vasconcelos, A. (2015). Um mapeamento sistemático sobre abordagens de apoio à rastreabilidade de requisitos no contexto de projetos de software. Revista de Sistemas de Informação da FSMA, 16, 3–15.

13. Mettler, T., & Pinto, R. (2015). Serious games as a means for scientific knowledge transfer: A case from engineering management education. IEEE Transactions on Engineering Management, 62(2), 256–265. https://doi.org/10.1109/TEM.2015.2413592

14. Microsoft Research. (2024). Code Hunt: Interactive coding game. https://www.microsoft.com/en-us/research/project/code-hunt/

15. Petticrew, M., & Roberts, H. (2006). Systematic reviews in the social sciences: A practical guide. Blackwell Publishing.

16. Reis, T., Medeiros, J., & Aranha, E. (2013). Ensino de programação utilizando jogos digitais: Uma revisão sistemática da literatura. Revista Novas Tecnologias na Educação, 11(1).

17. Robô Instructus. (2024). Robô Instructus: Creative coding challenges. https://www.roboinstructus.com/

18. Sampson, M., McGowan, J., Cogo, E., Grimshaw, J., Moher, D., & Lefebvre, C. (2009). An evidence-based practice guideline for the peer review of electronic search strategies. Journal of Clinical Epidemiology, 62(9), 944–952. https://doi.org/10.1016/j.jclinepi.2008.10.012

19. Screeps. (2024). Screeps: World multiplayer programming game. https://www.screeps.com/

20. Wu’s Castle. (2024). Wu's Castle: Programming logic game. https://www.wuscastle.com/

21. Zyda, M. (2005). From visual simulation to virtual reality to games. Computer, 38(9), 25–32. https://doi.org/10.1109/MC.2005.297

22. Zucarelli, I., & Couto, L. (2013). Jogo de tabuleiro em incentivo à alimentação infantil. Universidade São Judas Tadeu.

APPENDIX A

Table 1 - Identified Games

Gaming	Knowledge								
	LP	SP	MD	FP	TED	PP	AND	FI	ERC
CodinGame	S	S	S	S	S	S	S	S	S
Wu's Castle	S	S	S	S	S	S	S	S	S
Algorithms: Explained and Anim	S	N	N	N	N	N	N	N	S
Coding Planets	S	S	S	S	S	S	S	S	S
Human Resource Machine	S	N	N	S	S	S	S	S	S
7 Billion Humans	S	N	N	PARK	PARK	S	S	S	S
MIT App Inventor	S	S	S	S	N	S	S	S	N
Blockly Games	S	S	S	N	N	S	S	PARK	S
Code.org	S	S	N	N	N	S	S	N	S
Gameblox	S	S	S	S	S	S	PARK	N	S
Pencil Code	S	S	S	S	N	S	S	N	S
Microsoft Make Code	S	S	PARK	N	N	S	N	PARK	S
Code Spell	S	S	S	S	PARK	S	PARK	S	N
Baba is You	S	N	N	N	N	N	N	N	N
SpaceChem	S	N	N	N	N	N	S	S	N
Cyber Sentinel	S	N	N	N	N	N	S	S	N
Neon Noodles	S	N	N	N	N	N	S	S	N
Autonauts	S	N	N	N	N	PARK	S	S	S
Robô Instructus()	S	S	S	S	S	S	S	S	S
Marco Run	S	N	N	N	N	N	S	S	S
Karel (IDE)	S	S	S	S	S	S	S	S	S
Daisy the Dinosaur	S	N	N	N	PARK	S	S	S	S
Cargo-Bot	S	N	N	PARK	N	PARK	S	S	S
Code Hunt	S	S	S	S	S	S	S	S	S
Code Combat	S	S	N	S	S	S	S	S	N
Move The Turtle	S	N	N	N	N	S	S	S	S
Robozzle	S	N	N	N	N	PARK	S	N	N
Goose - Elephant with me	S	N	N	N	N	N	N	N	N
The Odyssey of Phoenix	S	S	N	S	N	N	S	N	N
LightBot	S	N	N	N	N	PARK	S	S	S
Minecraft Education Edition (M:EE)	S	N	N	S	N	S	S	S	S
Code Craft	S	N	S	S	N	S	PARK	N	S
I hate this game	S	N	N	N	N	N	N	N	N
Markov Alg.	S	N	N	N	N	N	N	N	N
The Endless Mission	S	PARK	S	S	N	S	PARK	N	PARK
The Magic Circle	S	N	N	N	N	N	N	N	N

Code World	S	S	S	S	S	S	S	S	S
Debugger 3.16: Hack'n'Run	S	S	S	S	S	S	S	S	S
Glitch space	S	S	S	S	S	S	S	S	S
Something bot	S	N	N	PARK	N	PARK	S	N	N
Algotica Translations	S	S	N	N	N	N	S	N	N
Jump Step Step	S	N	N	N	N	N	S	N	N
Bot Land (F2P)	S	S	N	N	S	S	S	S	S
Cool Headed	S	N	N	N	N	N	N	N	N
Glyphs Apprentice	S	N	N	N	N	N	S	N	N
Screeps: World	S	S	S	S	S	S	S	S	S
TIS-100	S	S	S	S	N	N	S	N	S
Robot Training	Not Found								
MagiPlay	S	N	N	N	N	N	N	N	N

Source: Prepared by the authors (2024)

Table 2 - Legend of the Abbreviations of Table 4 of the Subjects in the Games

Knowledge		
LP - Programming Logic	SP - Programming Syntax	MD - Data Manipulation
FP - Roles and Procedures	TED - Error Handling and Debugging	PP - Practical Projects
AND - Adapting to New Challenges	FI - Immediate Feedback	ERC - Repeat and Condition Structures

Source: Prepared by the authors (2024)