

Evolving a Digital Game for Programming Education of Adolescent Refugees in São Paulo

Lorenzo Fortes Santos Bernardo¹, Lucas Franco Ramos¹, Alessandreia Marta de Oliveira²,
Pedro Henrique Valle¹

¹Universidade de São Paulo - USP – São Paulo – SP – Brasil

²Departamento de Ciência da Computação - UFJF – Juiz de Fora – MG – Brasil

alessandreia.oliveira@ufjf.br, {lorenzofortes, pedrohenriquevalle, lucas.f}@usp.br

Abstract. Introduction: *The increasing number of adolescent refugees in Brazil highlights the need for inclusive Computing Education strategies that address linguistic, cultural, and motivational barriers while fostering social integration and digital skills. Objective:* This work presents *CodeJumper*, a digital educational game designed to support introductory programming learning through a contextualized narrative and engaging gameplay, tailored to adolescent refugees. **Methodology:** We adopted a multi-phase approach including target audience analysis, culturally grounded game design aligning learning objectives and mechanics, iterative implementation using GameMaker, and within-subject evaluation using SUS and LORI instruments. **Results:** Results indicate high usability (SUS = 82.5) and good perceived educational quality (LORI = 4.12), with improvements needed in accessibility and feedback. The findings highlight a strong alignment between learning objectives and game mechanics, while also identifying opportunities for improvement, particularly regarding accessibility features and feedback design..

Keywords Digital Educational Games, Computational Thinking, Computing Education, Programming Education.

1. Introduction

The phenomenon of forced displacement has escalated considerably in recent years, reaching critical levels in 2023, with more than 117 million individuals forcibly displaced worldwide, according to the United Nations High Commissioner for Refugees (UNHCR). In this context, Brazil has played a significant role as a host country, with an estimated population of refugees exceeding 710,000. In 2024 alone, 68,159 asylum applications were registered, of which 39.8% were submitted by Venezuelan nationals¹.

At the national level, the state of São Paulo accounted for the highest number of immigrant registrations, surpassing 45,000 in 2024, with approximately 30,000 concentrated in the capital. Among these, around 17,000 correspond to asylum requests. The demographic composition of this population is particularly noteworthy: children and adolescents under 18 years of age represent 41.8% of recognized refugees in Brazil, highlighting the urgency of developing public policies specifically targeted to this age group [Cavalcanti et al. 2025]. Furthermore, georeferenced analyses of refugees assisted by Caritas Archdiocesan of São Paulo² indicate that, although services and support

¹<https://www.acnur.org/br/dados-refugiados-no-brasil-e-no-mundo>

²<https://caritassp.org.br/>

structures are concentrated in central areas, the Venezuelan population is predominantly distributed across the southern and eastern regions of the city [ACNUR 2020].

Concomitantly, the Brazilian technology sector has experienced significant expansion, accompanied by an estimated shortage of approximately 500,000 professionals [TI INSIDE 2025]. In this scenario, programming education emerges as a promising strategy to foster socioeconomic inclusion among socially vulnerable populations. In particular, Digital Educational Games (DEGs) have been identified as effective tools for teaching programming logic in an engaging and interactive manner [Gundersen e Lampropoulos 2025, Genesio et al. 2024], including applications in contexts involving refugee populations [Sirin et al. 2018a, Ghailani et al. 2025].

In light of this context, this work presents CodeJumper, a Digital Educational Game designed to support the teaching of programming concepts, such as logic, variables, conditional structures, and loops, to refugee high school adolescents, using Python as the reference language. The proposed approach seeks to deliver such content in an accessible and engaging manner, grounded in a narrative that reflects everyday experiences. The game is being developed using the GameMaker platform³ and adopts a platform-based structure, incorporating puzzles at the end of each level to reinforce learning outcomes.

This work contributes to Computing Education by providing design principles and empirical evidence on educational games for vulnerable contexts. Our contributions are threefold: (i) a DEG featuring a narrative aligned with refugee adolescents' lived experiences; (ii) design guidelines that progressively articulate narrative, mechanics, and programming content; and (iii) the identification of challenges and opportunities for this audience based on an expert evaluation using the LORI and SUS instruments to assess usability, pedagogical effectiveness, and user support.

The remainder of this work is organized as follows. Section 2 discusses related work on Computing Education initiatives targeting refugee populations. Section 3 presents the research methodology adopted in this study. Section 4 describes the design and development of CodeJumper, including its narrative, mechanics, and learning objectives. Section 5 details the evaluation process and discusses the obtained results. Section 6 addresses threats to validity. Finally, Section 7 presents the conclusions and directions for future work.

2. Related Work

The scientific literature presents several initiatives investigating the use of DEGs to support the teaching of algorithms and programming logic. Existing work can be broadly grouped into three main approaches: (i) structured puzzle-based environments, (ii) code manipulation and abstraction-focused tools, and (iii) narrative-driven immersive games. In general, these approaches adopt distinct instructional and interaction strategies, combining game mechanics, narrative elements, and feedback systems to foster the development of Computational Thinking. In this context, games such as *Maze Code* [Neto et al. 2021] and *Project Éden* [Oliveira et al. 2021] structure the learning process through progressively challenging tasks within controlled environments, where learners construct algorithmic solutions using commands and control structures, supported by

³<https://gamemaker.io/pt-BR>

immediate feedback. Similarly, approaches such as *ParsonFlow* [Santos et al. 2025] and *Hello Food* [Macena et al. 2023] emphasize the organization and manipulation of code fragments or logical sequences, reducing syntactic complexity and prioritizing structured reasoning.

Other initiatives, such as *Codeland* and *CODE_DUNGEON* [Oliveira e Boff 2023], incorporate narrative-driven environments and mechanics inspired by role-playing games (RPGs), in which user progression is intrinsically linked to the resolution of programming challenges, thereby enhancing engagement through immersive contexts. Additionally, initiatives like *Mermãs Digitais* [Reis et al. 2025] aim to adapt content complexity and interface design to the characteristics of this audience, employing visual abstractions, puzzles, and automated feedback mechanisms. Collectively, these studies demonstrate the effectiveness of DEGs for programming learning, but they predominantly focus on controlled environments or simplified interactions, often limiting contextualization and real-world relevance.

Prior work on Computing Education for refugees highlights the use of digital environments to promote learning and social inclusion, although the pedagogical approaches adopted vary considerably [Yamashita et al. 2024a, Yamashita et al. 2024b]. Representative initiatives include Project Hope [Sirin et al. 2018b], which employs digital platforms to support programming education and cognitive development among Syrian children, and the Nairobi Play Project [Arawjo et al. 2019], which promotes intercultural exchange through the collaborative design of Scratch games. Similarly, projects involving Tibetan youth demonstrate that culturally grounded programming activities can reinforce cultural identity while increasing learner engagement [Paljor et al. 2025]. Despite these contributions, most existing initiatives emphasize digital literacy or game development activities rather than using Digital Educational Games (DEGs) as the primary instructional medium for structured computing education.

Most existing studies have been conducted in extracurricular or international contexts, leaving formal educational settings, particularly in countries such as Brazil, largely unexplored. Furthermore, current approaches rarely integrate three key dimensions in a single educational intervention: (i) a narrative grounded in refugees' lived experiences, (ii) Digital Educational Games (DEGs) as the primary instructional medium, and (iii) a systematic progression of programming concepts. To address these gaps, this paper proposes *CodeJumper*, a culturally contextualized DEG that combines a structured programming curriculum with gameplay-centered learning to support programming education for adolescent refugees in Brazil.

3. Research Method

To achieve the objectives of this study, we adopted a multi-phase research methodology that integrates context analysis, game design, development, and evaluation. This approach was defined to guide the conception and assessment of *CodeJumper*, aimed at supporting programming education for refugee adolescents, as illustrated in Figure 1.

3.1. Phase 1: Context Analysis and Student Profile

Initially, we conducted a context analysis to understand the sociocultural characteristics of the target audience. This phase involved a documentary analysis to identify aspects

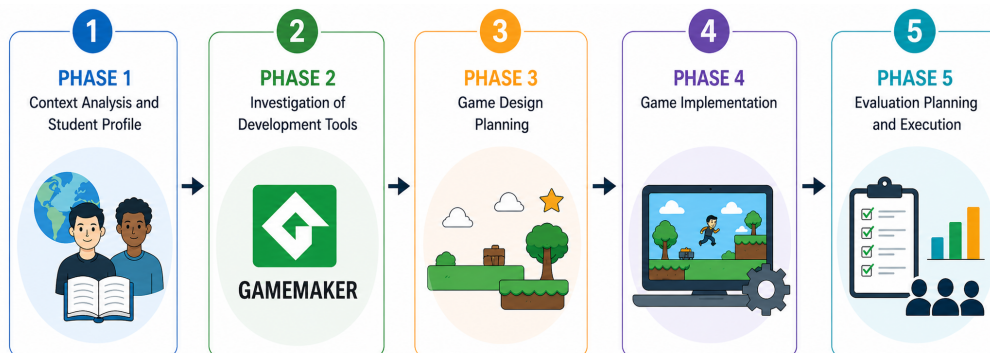


Figura 1. Methodology for developing the CodeJumper

such as nationality, language, adaptation challenges, and educational difficulties faced by refugee students. The collected information was systematized to support pedagogical and technical decisions related to the design of CodeJumper.

3.2. Phase 2: Investigation of Development Tools

After analyzing the target audience, we investigated tools for developing the digital educational game. This phase included a comparative analysis of different engines and languages, including Unity, Godot, and Construct. We considered criteria such as learning curve, educational support, multiplatform export, accessibility, and suitability for game development. Based on this analysis, we selected the GameMaker engine for its support for 2D platform game development and its flexibility for integrating educational mechanics.

3.3. Phase 3: Game Design Planning

Based on the information obtained in the previous phases, we planned CodeJumper's design. In this phase, we defined a narrative aligned with the cultural context of refugee students by creating representative characters and scenarios that closely reflect the reality of the target audience. We also defined the educational content to be addressed in the game, focusing on introductory programming concepts such as variables, conditional structures, and logical reasoning. These contents were integrated into the game mechanics through progressive challenges and educational puzzles, ensuring alignment between pedagogical objectives and gameplay.

3.4. Phase 4: Game Implementation

We implemented CodeJumper iteratively, allowing continuous refinement of the mechanics, interface, and educational content. We organized development into incremental cycles, during which we created graphical, audio, and interaction elements aligned with the defined pedagogical objectives. During this phase, we conducted continuous internal testing to validate the game logic, resolve inconsistencies, and ensure coherence between the gameplay experience and learning objectives.

3.5. Phase 5: Evaluation Planning and Execution

To evaluate CodeJumper, we conducted an empirical study focusing on usability and educational quality using the System Usability Scale (SUS) and the Learning Object Review Instrument (LORI). We invited five participants to take part in the evaluation.

3.6. Phase 6: Data Collection and Analysis

We collected data using structured questionnaires based on the SUS and LORI instruments, along with open-ended questions for qualitative feedback. The quantitative data were analyzed using descriptive statistics, including mean and standard deviation for SUS and average scores for each LORI dimension. Additionally, we analyzed participants' qualitative comments to identify recurring themes related to usability, pedagogical aspects, and user support. The combined quantitative and qualitative analysis enabled us to identify strengths, limitations, and opportunities for improving CodeJumper.

4. CodeJumper Game

This work proposes the digital educational game (DEG) CodeJumper⁴, designed to teach introductory programming concepts to refugee high school adolescents. Through a playful, contextually grounded approach, the game integrates narrative, progressive challenges, and logic-based *puzzles* to provide an accessible, engaging learning experience. In this way, it aims to contribute to digital inclusion and to expand the educational and professional opportunities of this audience.

CodeJumper is a platform game in which each level presents a *puzzle* related to programming concepts, whose resolution enables progression in the narrative. The challenges gradually increase in difficulty as the player's learning curve progresses. The game is being developed using the GameMaker engine, which supports the creation of interactive environments and the integration of educational mechanics focused on teaching programming logic.

4.1. Characterization and Narrative

We conceived CodeJumper as a DEG that integrates action, exploration, and *puzzle*-solving to support the teaching of introductory programming concepts to refugee high school adolescents. The activities are organized progressively, such that each level addresses a subset of programming logic concepts, enabling incremental learning throughout the narrative.

The story follows Miguel, a 15-year-old Venezuelan refugee who lives with his mother in Brazil. In his daily routine, which includes leaving home, going to school, and performing everyday tasks such as going to the market, the character faces challenges that are solved through *puzzles* based on programming concepts.

This structure aims to contextualize the content by bringing it closer to the target audience's reality, fostering identification with the character, and promoting learning in a playful, progressive manner.

4.2. Game Mechanics and Learning Progression

The mechanics of CodeJumper integrate platform game elements with educational programming challenges. The player explores scenarios, overcomes obstacles, and collects coins while solving logic-based *puzzles*, which reinforce the addressed concepts.

⁴The game is available at: <https://loreko.itch.io/codejumper>

The game is structured into sequential levels that follow Miguel's daily routine, with a gradual increase in complexity that includes variables, operators, conditionals, and loops. The narrative begins with Miguel going to school and continues until his return home, integrating challenges into real-life situations. Before each level, a brief explanation of the content is provided.

In **Level 1**, Miguel moves toward the bus stop, overcoming obstacles and collecting coins (Figure 2a). Upon arrival, the driver requests his assistance in removing several boxes blocking the bus, initiating a puzzle in which the boxes represent variables that must be organized according to their respective types (Figure 2b). After completing this task, Miguel disembarks from the bus (**Level 2**) and proceeds to the school. Upon arrival, he finds the gate closed, and as a condition for entry, he must bypass the system by solving a *puzzle* involving logical, assignment, and comparison operators. Upon successfully completing this puzzle, Miguel gains access to the school (Figure 2c).

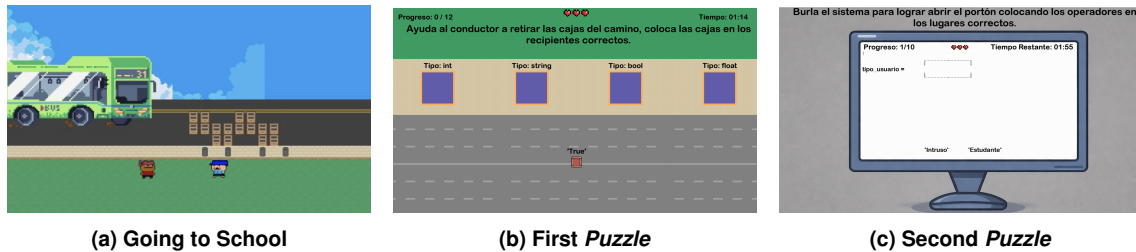


Figura 2. Examples of in-game interactions

After a long day of classes, Miguel leaves and heads toward the market (**Level 3**), passing in front of the São Paulo Museum of Art - MASP (Figure 3a). Before reaching the market, a guard stops him and asks for help in managing traffic on the avenue. To accomplish this, Miguel must solve a *puzzle* involving the use of conditional structures to organize the flow of cars (Figure 3b). Continuing toward the market (**Level 4**), Miguel overcomes additional challenges and obstacles along the way. Upon arrival, in order to access the required items, he must solve a puzzle that addresses repetition structures, including nested loops (Figure 3c).

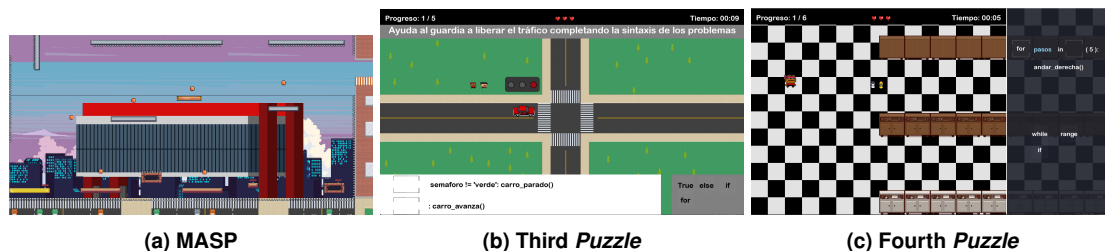


Figura 3. Additional examples of puzzles in the game

Finally, after leaving the market, Miguel heads toward his home. Upon arriving (**Level 5**, under development), he interacts with his mother, who asks questions about his day (*puzzle 5*) with the objective of reinforcing the concepts learned throughout the experience. The narrative concludes with the preparation of a traditional Venezuelan dish, strengthening cultural ties and promoting a sense of identity and belonging.

4.3. Learning Objectives

We designed CodeJumper to facilitate the introduction of fundamental programming concepts and Computational Thinking for refugee high school students, aiming to make this initial contact more accessible and engaging through progressively structured challenges embedded in a contextualized narrative.

Our approach emphasizes the development of key Computational Thinking skills, such as problem decomposition, pattern recognition, and logical reasoning, encouraged through the resolution of puzzles distributed across the game levels. In parallel, we progressively introduce core programming concepts, including variables, conditional statements, and loops, all situated within meaningful narrative-driven contexts. Additionally, we aim to stimulate students' interest in Computing, particularly among those with little or no prior exposure. By doing so, we aim to enhance engagement and expand their educational and professional opportunities.

5. Evaluation of CodeJumper

5.1. Planning and Execution

To evaluate the effectiveness of the CodeJumper game, we designed an evaluation focusing on two complementary perspectives: (i) usability and (ii) educational quality. For this, we adopted two instruments: the System Usability Scale (SUS) [Brooke et al. 1996] and the Learning Object Review Instrument (LORI) [Nesbit e Li 2004]. The System Usability Scale (SUS) is a well-established questionnaire used to assess the perceived usability of products, systems, and interfaces. It consists of 10 items rated on a 5-point Likert scale, ranging from “Strongly Disagree” to “Strongly Agree”. The Learning Object Review Instrument (LORI) is a structured framework developed by Nesbit et al. (2002) designed to evaluate the quality of multimedia learning objects and educational software. It serves as a tool for peer review, instructional design, and user evaluation by assessing educational resources across nine distinct dimensions. We selected the SUS to measure participants' perceived usability of the CodeJumper, providing a global usability score, while we used LORI to evaluate the pedagogical quality of the game across multiple dimensions, such as content quality, motivation, feedback, usability, and reusability.

We initially informed participants of the study's objectives and obtained their consent before proceeding. The participants were computer science students and graduates aged 18 to 25 who had experience with educational games. The execution consisted of three main steps: (i) participants completed a profile questionnaire to collect demographic data and information about their prior experience with programming and digital games; (ii) participants interacted with CodeJumper for approximately 20 to 40 minutes to explore the game and complete the proposed challenges; and (iii) participants completed the SUS questionnaire (10 items, using 5-point Likert scale) and the LORI questionnaire (8 dimensions, using 5-point Likert scale), and optional open-ended questions to collect qualitative feedback.

5.2. Analysis of Results

After applying these two instruments, we analyzed the SUS scores following the standard procedure. The results indicate a **mean SUS score of 82.5** (SD = 10.31), which can

be interpreted as *high usability* according to established metrics. This score places CodeJumper in the “*Good*” to “*Excellent*” range of usability, suggesting that participants found the game easy to use, well-structured, and accessible. Considering the sample size (n = 5), the results still indicate a consistent perception of usability among participants.

These findings suggest that the design decisions adopted in CodeJumper , particularly those related to interface organization and interaction flow, were effective in supporting user interaction. The high usability score is especially relevant in an educational context, as usability issues can negatively impact learning outcomes. Therefore, the SUS results indicate that the CodeJumper does not establish significant usability barriers that would hinder the learning experience. In short, Table 1a summarizes the overall SUS results.

Tabela 1. SUS and LORI Results

(a) SUS Results		(b) LORI Results by Dimension	
Metric	Value	Dimension	Mean
Mean SUS Score	82.5	Content Quality	4.50
Standard Deviation	10.31	Learning Goal Alignment	4.75
Usability Interpretation	Good–Excellent	Feedback and Adaptation	3.75
Sample Size	5 participants	Motivation	4.00
		Presentation Design	4.25
		Usability	4.50
		Accessibility	3.25
		Reusability	4.00

Regarding the LORI instrument, we analyzed the results using descriptive statistics. The overall mean score across all dimensions was **4.12** (on a 1–5 scale), indicating a *high perceived educational quality*. This result suggests that, beyond usability, CodeJumper is also perceived as an effective educational resource. Table 1b summarizes the average scores for each dimension.

A deeper analysis of the LORI dimensions reveals important insights. The highest-rated dimension, **Learning Goal Alignment (4.75)**, indicates that participants clearly perceived the connection between gameplay and the intended learning objectives. **Content Quality (4.50)** and **Usability (4.50)** reinforce the robustness of the game as both a learning resource and an interactive system. These findings are consistent with the SUS results, indicating coherence between usability perception and educational evaluation. The positive score for **Presentation Design (4.25)** also suggests that visual and structural aspects of the game contribute to user engagement and comprehension.

On the other hand, lower scores in **Accessibility (3.25)** and **Feedback and Adaptation (3.75)** highlight important areas for improvement. The accessibility results suggest that the game may not yet fully support a diverse range of users, particularly those with different needs or skill levels. Similarly, the feedback and adaptation dimension indicates that the game could provide more effective guidance, hints, or adaptive responses during gameplay. These aspects are particularly relevant in educational contexts, where timely and meaningful feedback is essential for learning.

We also analyze qualitative responses to complement these findings, providing

deeper insights into the user experience. Participants consistently emphasized the game's engaging nature, particularly highlighting how its platform-style mechanics enhance immersion. For instance, P1 reported that "I found the integration between the content and gameplay interesting", suggesting that the gameplay effectively connects interaction with learning elements. Similarly, P2 highlighted "the platform-style mechanics and the loop level", reinforcing the idea that the game structure and progression mechanics are engaging and meaningful. In the same direction, P3 noted that "the platform aspect is quite interesting", while P4 emphasized that "the game has smooth movements and is very fluid", indicating that both gameplay design and animation quality positively contribute to the overall user experience.

Regarding the educational aspects, participants also perceived a meaningful connection between the game and programming concepts. Despite the positive feedback, participants also identified several limitations. A recurring issue concerns instructions and onboarding. P4 directly pointed out "very basic instructions", while P1 suggested that "the tutorial presentation could be improved", indicating that the current tutorial and guidance mechanisms may be insufficient for supporting users, particularly in the initial stages of interaction. These findings suggest that, although the gameplay itself is intuitive, the onboarding process could be significantly improved.

Feedback mechanisms were also identified as an area for improvement. P3 suggested that "a message could be added when finishing the game", while P1 recommended "improving the signaling of what needs to be done", indicating that the game could provide clearer and more immediate feedback to guide user actions. These findings align with the lower score observed in the Feedback and Adaptation dimension.

Participants also proposed concrete improvements to enhance the overall experience. These include improving tutorial clarity and instructional design (P1, P4), refining gameplay mechanics such as jumping (P2, P5), enhancing feedback and progression indicators (P1, P3), and improving visual consistency (P5). These suggestions are particularly valuable as they directly point to actionable design improvements. Overall, the qualitative findings reinforce the quantitative results obtained through SUS and LORI.

6. Threats to Validity

During the planning and execution of this work, several measures were adopted to mitigate potential threats to validity, following established guidelines in empirical software engineering research [Wohlin et al. 2012]. Nevertheless, such threats cannot be fully eliminated and should be carefully considered when interpreting the results. Regarding **construct validity**, a potential concern is whether the selected instruments adequately capture the intended constructs. To mitigate this issue, we adopted well-established, validated instruments: the SUS for usability and the LORI for educational quality. However, both instruments rely on self-reported perceptions, which may introduce subjectivity and response bias. In addition, although the LORI dimensions cover multiple aspects of educational quality, they do not directly assess learning outcomes such as knowledge acquisition, retention, or transfer over time.

Concerning **internal validity**, potential threats include learning effects, participant fatigue, and variability in prior experience. As the evaluation was conducted

in a single session, participants' responses may reflect initial impressions rather than sustained interaction with the system. Moreover, differences in participants' prior knowledge of programming or familiarity with digital games may have affected their interaction and evaluation. To mitigate these effects, we adopted a standardized evaluation protocol and ensured that all participants followed the same procedure.

With respect to **external validity**, the main limitation concerns the generalizability of the findings. The work involved a small sample of five participants, all of whom were specialists rather than members of the target audience (i.e., refugee adolescents). As a result, the findings may not fully represent how the intended users would perceive or interact with the game. Additionally, the evaluation was conducted in a controlled environment, which may differ from real-world educational settings, particularly those involving socially vulnerable populations.

Finally, regarding **conclusion validity**, the small sample size limits statistical power and the robustness of the conclusions. The use of descriptive statistics without inferential analysis limits the ability to make stronger claims about the game's effectiveness. Despite these limitations, the study provides initial evidence and actionable insights to guide future improvements and evaluations involving the target audience.

7. Conclusion and Future Work

This work presented CodeJumper, a DEG designed to support the teaching of introductory programming concepts to refugee adolescents through a narrative-driven approach and progressive challenges. By integrating platform mechanics with educational puzzles, the game aims to provide an accessible and engaging learning experience tailored to students in vulnerable contexts.

The evaluation results provide initial evidence of the game's potential. High usability scores and positive perceptions of educational quality indicate effective alignment between gameplay and learning objectives. Overall, these findings suggest that CodeJumper represents a promising approach for programming education in socially vulnerable contexts. Beyond the game itself, this work contributes by articulating design strategies that integrate narrative, gameplay, and programming learning while addressing cultural and educational challenges faced by refugee populations.

As future work, we plan to incorporate the identified improvements into the game and conduct empirical studies with the target audience. These studies will evaluate the impact of CodeJumper on learning outcomes, engagement, and inclusion, as well as its effectiveness in real-world educational settings.

Acknowledgments

The authors acknowledge financial support from the University of São Paulo through the Programa Unificado de Bolsas (PUB-USP).

Use of Artificial Intelligence

Generative Artificial Intelligence tools were used only for language review and minor editorial support. Gemini was used for text refinement, and Grammarly for minor proofreading. All methodological decisions, analyses, and conclusions remain the sole responsibility of the authors.

Referências

- ACNUR (2020). Mapa de georreferenciamento de pessoas em situação de refúgio atendidas pela caritas arquidiocesana de são paulo – 2020.
- Arawjo, I., Mogos, A., Jackson, S. J., Parikh, T., e Toyama, K. (2019). Computing education for intercultural learning: Lessons from the nairobi play project. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24.
- Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- Cavalcanti, L., Oliveira, T., e Macedo, M. (2025). Dados consolidados da imigração no brasil 2024. Série migrações, Observatório das Migrações Internacionais (OBMigra); Ministério da Justiça e Segurança Pública, Brasília, DF. Disponível em: https://portaldeimigracao.mj.gov.br/images/DADOS_CONSOLIDADOS_2024_25.pdf. Acesso em: 25 jan. 2026.
- Genesio, N. O. S., de Oliveira, A. M., Oliveira, E. W., e Valle, P. H. D. (2024). Panorama de estudos sobre jogos educacionais digitais em educação em computação. In *Workshop sobre Educação em Computação (WEI)*, pages 737–749. SBC.
- Ghailani, I. G., Malaise, Y., e Signer, B. (2025). Jsstories: Improving social inclusion in computer science education through interactive stories. *CoRR*, arXiv:2504.04006.
- Gundersen, S. W. e Lampropoulos, G. (2025). Using serious games and digital games to improve students' computational thinking and programming skills in k-12 education: A systematic literature review. *Technologies*, 13(3):113.
- Macena, J., Pires, F., e Pessoa, M. (2023). Hello food: um jogo para exercitar conceitos de programação e pensamento computacional. In *Anais Estendidos do XII Congresso Brasileiro de Informática na Educação*, pages 173–176, Porto Alegre, RS, Brasil. SBC.
- Nesbit, J. C. e Li, J. (2004). Web-based tools for learning object evaluation. In *Proceedings of the International Conference on Education and Information Systems: Technologies and Applications*, volume 2, pages 334–339. Orlando, Florida.
- Neto, J. G., Marcolino, P. P., Ferreira, P., e Barbosa, D. (2021). Maze code: Retórica procedural aplicada ao ensino de lógica de programação. In *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 519–528, Porto Alegre, RS, Brasil. SBC.
- Oliveira, G. e Boff, E. (2023). Code_dungeon: um serious game para auxiliar no aprendizado de programação. In *Anais Estendidos do XXII Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 581–590, Porto Alegre, RS, Brasil. SBC.
- Oliveira, Y., Toledo, C., e Pereira, L. (2021). Project Éden: platform for introductory programming concepts. In *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 566–573, Porto Alegre, RS, Brasil. SBC.
- Paljor, Y., Proctor, C., Honwad, S., Decker, A., e Hoadley, C. (2025). Culturally sustaining computing education for tibetan diaspora youth: Engaging heritage identity through computing. In *Proceedings of the 19th International Conference of the Learning Sciences-ICLS 2025*, pp. 880-888. International Society of the Learning Sciences.

- Reis, K., Ângelo Feitosa, Junior, J. S., Araújo, M., Aquino, S., e Sa, V. (2025). Teste de usabilidade: Perspectivas futuras da experiência de meninas em um jogo de lógica de programação. In *Anais Estendidos do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 146–152, Porto Alegre, RS, Brasil. SBC.
- Santos, A., Souza, C., e Araujo, L. (2025). Parsonflow: Um jogo baseado em problemas de parson para auxiliar o processo de ensino-aprendizagem em programação. In *Anais Estendidos do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 196–201, Porto Alegre, RS, Brasil. SBC.
- Sirin, S., Plass, J. L., Homer, B. D., Vatanartiran, S., e Tsai, T. (2018a). Digital game-based education for syrian refugee children: Project hope. *Vulnerable Children and Youth Studies*, 13(1):7–18.
- Sirin, S., Plass, J. L., Homer, B. D., Vatanartiran, S., e Tsai, T. (2018b). Digital game-based education for syrian refugee children: Project hope. *Vulnerable Children and Youth Studies*, 13(1):7–18.
- TI INSIDE (2025). Brasil chega a 2025 com déficit de meio milhão de profissionais de TI e busca soluções. Acesso em: 28 jan. 2026.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., e Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.
- Yamashita, V. T., Vilarinho, L., Silva, A., Feital, M., Valle, P. H. D., de Melo Quintela, B., e de Oliveira, A. (2024a). Jogos educacionais digitais e a inclusão das mulheres na programação: Um mapeamento sistemático. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1028–1042. SBC.
- Yamashita, V. T., Vilarinho, L. L., Gonçalves, L. B., de Melo Quintela, B., Renhe, M. C., Chaves, L. J., Valle, P. H. D., e de Oliveira, A. M. (2024b). Desenvolvimento de um jogo digital para apoiar o ensino-aprendizagem de algoritmos: Estratégias para engajar mulheres no ensino superior em computação. In *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 1281–1292. SBC.