

***NewBot*: Jogo Educativo para o Ensino do Pensamento Computacional**

**Guilherme Silva, Júlia O. Pessoa, Ian N. M. da Silva, Aymê C. Gonçalves,
Karoline R. Garcia, André L. Brandão, Karla Vittori**

CMCC – Universidade Federal do ABC (UFABC)

Campus Santo André - Avenida dos Estados, 5001 - Bairro Santa Terezinha

Santo André - SP - Brasil - CEP: 09210-580

{guilherme.hugo, julia.pessoa, nicolas.magatti, ayme.cardoso, k.roncarat
e}@aluno.ufabc.edu.br, {karla.vittori, andre.brandao}@ufabc.edu.br

Resumo. *O pensamento computacional (PC) envolve habilidades que facilitam a resolução de um conjunto diversificado de problemas e que na era da informação e digitalização, têm se tornado fundamentais. Dentro deste contexto, este artigo apresenta um jogo educativo para o ensino de quatro habilidades do PC a crianças do ensino fundamental. Até o momento, foram implementadas as fases do jogo ligadas às habilidades de abstração, decomposição de problemas e algoritmos. As atividades futuras envolvem a implementação da fase relativa à habilidade de reconhecimento de padrões, a realização de experimentos em uma escola de ensino básico, para testar a sua eficácia educativa e a qualidade técnica, e a sua disponibilização ao público em um website dedicado.*

Abstract. *Computational thinking (CT) involves skills that facilitate the resolution of a diverse set of problems and, in the age of information and digitalization, have become fundamental. Within this context, this article presents an educational game for teaching four CT skills to elementary school children. So far, the game phases related to the skills of abstraction, problem decomposition, and algorithms have been implemented. Future activities involve implementing the phase related to the skill of pattern recognition, conducting experiments in an elementary school to test its educational effectiveness and technical quality, and making it available to the public on a dedicated website.*

1. Introdução

Na era da informação e digitalização, tem havido um aumento notável no interesse em relação ao pensamento computacional (PC) no campo da educação [Rao and Bhagat, 2024]. O Pensamento Computacional (PC) foi proposto inicialmente por Papert (1980; 1996), tendo sido posteriormente popularizado por Win (2006). Win afirmou que o PC engloba uma mistura de habilidades, abordagens, procedimentos e formas de pensar que facilitam a resolução de um conjunto diversificado de problemas, para além daqueles associados à Ciência da Computação. Além disso, ela propôs que o PC deveria ser considerado uma habilidade fundamental para todas as crianças, como ler e escrever.

Apesar de existirem vários estudos sobre o pensamento computacional [Bati 2022; García-Peñalvo et al. 2018; Giannakoulas and Xinogalos 2023; Taslibeyaz et al. 2020; Theodoropoulos and Lepouras 2020], ainda não há acordo entre os cientistas quanto à sua definição e às competências específicas que ele engloba [Gutiérrez et al. 2023]. Papert [1980] primeiramente o definiu como um conjunto de habilidades para resolver processos de forma sistemática e eficaz. Wing (2006) iria mais tarde detalhar o termo,

descrevendo-o como uma abordagem de resolução de problemas que combina a lógica com elementos da Ciência da Computação, podendo ser usada em qualquer disciplina.

Para Wing (2006), o pensamento computacional consiste em usar as habilidades de abstração e decomposição para resolver uma tarefa complexa ou projetar um sistema complexo. Wing (2011) incluiu também o pensamento algorítmico e o pensamento paralelo na definição do pensamento computacional, o que, por sua vez, envolve outros tipos de processos de pensamento, tais como o raciocínio compositivo, a correspondência de padrões, o pensamento processual e o pensamento recursivo.

Papert (1980) introduziu o PC sob o conceito de construcionismo, o qual enfatiza a importância dos alunos como participantes ativos em sua aprendizagem [Gutiérrez et al. 2023]. Desde então, diferentes estudos têm sido produzidos buscando formas de desenvolver o pensamento computacional em crianças [Gutiérrez et al. 2023]. Uma grande parte deles envolve o desenvolvimento de jogos sérios para este fim [Bati 2022; Giannakoulas and Xinogalos 2023; Papadakis 2021; Silva et al. 2019; Sun et al. 2021]. O uso de jogos na educação tem potencial para aumentar a motivação e o envolvimento dos alunos no processo de aprendizagem, permitindo a aquisição de um conhecimento prático duradouro [Videnovik et al. 2023].

Dentro deste contexto, esta pesquisa propõe o desenvolvimento de um jogo educativo para ensinar o pensamento computacional a crianças do ensino fundamental. Jogos educativos constituem uma categoria de jogos sérios [Susi et al. 2007] que priorizam o ensino sobre o entretenimento e favorecem a abordagem educacional do "aprender fazendo". Incorporar jogos educativos no processo de ensino traz uma série de vantagens [Giannakoulas and Xinogalos 2023]. Estes jogos oferecem aos alunos uma experiência de aprendizagem única, onde eles podem se envolver de forma ativa em um ambiente interativo que desperta a sua motivação. Ao mesmo tempo, eles recebem feedback relevante, enfrentam desafios e recebem o apoio adequado para ajudar no seu processo de aprendizagem.

No que diz respeito aos alunos do ensino fundamental, o efeito do uso de jogos educativos para introduzir conceitos de programação e promover o pensamento computacional é amplificado [Giannakoulas and Xinogalos 2023]. De acordo com Zaharija e colaboradores (2013), os jogos podem assumir um papel crucial no ensino de conceitos de programação de computadores a jovens estudantes, uma vez que os métodos tradicionais de ensino podem se mostrar inadequados para a sua faixa etária, e não ser capazes de lidar de modo eficaz com desafios como a sua capacidade de atenção limitada.

No jogo proposto nesta pesquisa, chamado *NewBot*, os jogadores são desafiados a explorar um mundo aberto, resolvendo quebra-cabeças em quatro fases principais, cada uma focada em um pilar do pensamento computacional: i) abstração; ii) algoritmos; iii) decomposição de problemas e iv) reconhecimento de padrões. O principal objetivo do *NewBot* é ensinar o PC de uma maneira divertida e envolvente. Ao completar os desafios e explorar o mundo, os jogadores desenvolvem habilidades essenciais que são aplicáveis tanto na programação quanto na resolução de problemas em geral.

Este se caracteriza como um trabalho em andamento, e possui como resultados iniciais o desenvolvimento das fases do jogo *NewBot* relacionadas à três habilidades do PC: i) abstração; ii) algoritmos e iii) decomposição de problemas.

Este documento é organizado como segue. Na Seção 2, são mencionados os trabalhos relacionados com o tema desta pesquisa, seguidos na Seção 3 pela descrição do jogo educativo desenvolvido e as fases implementadas até o momento. Na Seção 4, é realizada uma análise do jogo desenvolvido, seguida finalmente pela Seção 5, que apresenta os próximos passos desta pesquisa.

2. Trabalhos Relacionados

2.1. Definições do Pensamento Computacional

O pensamento computacional aborda diferentes habilidades lógicas de resolução de problemas [Gutiérrez et al. 2023]. Embora não exista uma definição padrão das habilidades ligadas ao PC, vários autores concordam sobre alguns dos seus elementos comuns. Esta pesquisa se baseou nos 7 autores [Barr and Stephenson, 2011; Bocconi et al. 2016; Castro et al., 2021; Ho et al. 2021; Hoyles and Noss 2015; Turchi et al., 2019; Weintrop et al. 2016] considerados por Gutiérrez e colaboradores (2023) para definir as principais habilidades que o pensamento computacional aborda, quais sejam: i) gestão de dados; ii) decomposição de problemas; iii) abstração; iv) algoritmos (pensamento algorítmico); v) automação; vi) paralelização; vii) solução de problemas; viii) pensamento recursivo; ix) uso de heurísticas; x) depuração de erros (*debugging*); xi) modelagem e simulação; xii) pensamento sistemático e xiii) reconhecimento de padrões.

As habilidades do PC mais comumente mencionadas na literatura [Gutiérrez et al., 2023; Rao and Bhagat, 2024] e consideradas nesta pesquisa são: i) algoritmos; ii) decomposição de problemas; iii) abstração e iv) reconhecimento de padrões. Portanto, de acordo com Gutiérrez e colaboradores (2023), essas habilidades específicas devem ser priorizadas para a elaboração de um jogo sério que busca desenvolver o pensamento computacional.

2.2. Objetivos de Aprendizado Relacionados com o Pensamento Computacional

A literatura apresenta diversas abordagens para o ensino das habilidades ligadas ao pensamento computacional. Para alunos com idade inferior a 18 anos, os objetivos de aprendizado ligados ao ensino do PC utilizando jogos, de acordo com o estudo realizado por Gutiérrez e colaboradores (2023), são os seguintes: i) estruturas sequenciais, condicionais e de repetição [Ayman et al. 2018; Schez-Sobrinho et al. 2020; Elshahawy et al. 2020]; ii) comunicação, solução de problemas, planejamento, raciocínio e socialização [Cano et al. 2021]; iii) solução de problemas, algoritmos, abstração, decomposição de problemas, pensamento recursivo, uso de heurísticas, representação de dados e avaliação de soluções [Turchi et al. 2019]; iv) atitudes ligadas a cursos de ciência, tecnologia, engenharia e matemática (STEM em inglês) [Guenaga et al., 20] e

v) algoritmos, modularidade, estruturas de controle, representação, design de processos e depuração de erros (*debugging*) [Utesch et al., 2020]. De acordo com o estudo realizado por Gutiérrez e colaboradores (2023), a maior parte dos autores analisados se concentra no desenvolvimento de jogos sérios que ensinam conceitos básicos de programação e a resolução de problemas.

2.3. Uso de Jogos para Desenvolver o Pensamento Computacional

Wassila e Tahar (2012) classificaram os jogos digitais de acordo com as habilidades do pensamento computacional que eles visam desenvolver nos alunos. Os autores sugerem que os professores podem escolher o jogo que se encaixa nas necessidades e objetivos das aulas. Eles selecionaram três tipos principais de jogos para esse fim: i) ação (por ex. labirinto, combate e plataforma); ii) estratégia (ex. aventura) e jogos híbridos (ex. aventura em tempo real e simulações). Os tipos de jogos que se encaixam na maioria dos casos são quebra-cabeças, simulações e jogos de estratégia.

Theodoropoulos e Lepouras (2020) estudaram o uso de jogos educacionais para ensinar conceitos de programação de computadores e fomentar habilidades ligadas ao PC em estudantes com idade entre 3 e 18 anos. Após analisar 44 artigos publicados sobre este tema entre 2009 e 2019, eles verificaram que a principal vantagem do uso de jogos é a motivação e o engajamento que este ambiente oferece aos alunos. Através dos jogos, o ensino e aprendizagem das habilidades ligadas ao PC incorporam vários elementos para motivar os alunos, como o cenário interativo, a narrativa e o controle dos jogadores. Um outro elemento positivo verificado em vários estudos (29 dos 44 analisados) é o sistema de feedback na forma de ranking com indicadores de processo (pontos, níveis etc.). Este feedback imediato demonstra ter um impacto considerável na motivação dos alunos e no sucesso do seu aprendizado. Além disso, os jogos digitais podem ser usados para ensinar e aprender PC sem a necessidade de conhecimento prévio sobre programação.

Theodoropoulos e Lepouras (2020) também ressaltaram a importância da avaliação do aprendizado adquirido através dos jogos. Muitos estudos por eles analisados realizaram esta avaliação através de questionários e estes apresentaram algumas limitações no sistema de pontuação, pois vários aspectos do jogo, como avaliação e controle do jogo de acordo com as necessidades, foram ignorados.

Além disso, os autores consideram que a avaliação do desempenho dos jogadores deve incluir vários fatores, como o tipo de jogo utilizado, as habilidades do PC ensinadas, os conceitos de programação envolvidos e o contexto do jogo. Dentro deste contexto, os autores sugerem que os jogos poderiam incorporar mecânicas para pontuação de desempenho que incluam categorias como objetivo, implementação, integração e tipo de avaliação principal [Shute and Ke 2012].

Desta forma, o estudo realizado por Theodoropoulos e Lepouras (2020) demonstrou que utilizar jogos digitais para o ensino do PC apresenta resultados favoráveis, mas que estes, porém, são altamente dependentes tanto do ambiente de aprendizagem quanto dos indivíduos envolvidos.

Giannakoulas and Xinogalos (2023) realizaram um estudo para identificar as pesquisas empíricas mais recentes sobre a aplicação de jogos educativos para ensinar habilidades do PC e conceitos básicos de programação a jovens estudantes, bem como explorar os efeitos desses jogos no desenvolvimento das habilidades mencionadas. Para isso, eles examinaram 61 estudos empíricos focados na faixa etária do ensino fundamental, no período de 2010 até setembro de 2022.

Em relação às habilidades de PC que os jogos analisados buscam cultivar nos alunos, a maioria deles visa desenvolver o pensamento algorítmico, a decomposição, a abstração, bem como o reconhecimento de padrões e a modularidade, que são habilidades reconhecidas por vários pesquisadores [Atmazidou and Demetriadis, 2016; Brennan & Resnick, 2012; Grover and Pea, 2018; Shute et al., 2017] como elementos fundamentais do PC. Além disso, em termos de depuração de erros (*debugging*), foi observado que há uma escassez de estudos empíricos focando especificamente neste aspecto do PC.

A maioria dos estudos que investigaram a eficiência ou o impacto dos jogos na aquisição das habilidades do PC (39 de 47) relatou resultados positivos, demonstrando que jogos educativos podem ajudar alunos do ensino fundamental a desenvolver habilidades do PC ou a entender conceitos básicos de programação. Os resultados desta pesquisa demonstraram um forte impacto positivo do uso de um jogo educativo como ferramenta de ensino para habilidades do PC e de conceitos básicos de programação para crianças do ensino fundamental. Os alunos estabeleceram uma atitude positiva em relação à programação através de atividades baseadas em jogos, as quais demonstraram ser um poderoso aspecto motivador para as crianças, engajando-as em atividades ligadas ao pensamento computacional.

3. Jogo *NewBot*

3.1 Visão Geral

O *NewBot* é um jogo educativo projetado para crianças do ensino fundamental, com o objetivo de introduzir e aprofundar os conceitos de pensamento computacional. Inspirado em um jogo educacional desenvolvido anteriormente [Higuchi et al, 2021], o *NewBot* adota uma estética de puzzle. O jogo é constituído por uma série de desafios, com o intuito de desenvolver as habilidades do pensamento computacional relacionadas com: i) algoritmo; ii) decomposição de problemas; iii) abstração e iv) reconhecimento de padrões.

A narrativa do jogo propõe uma progressão de desafios, que são divididos em quatro fases. Estas são distribuídas em um cenário de mundo aberto, o que permite que o jogador crie seu próprio percurso e encare os desafios de cada fase na ordem que preferir, promovendo assim o aprendizado por meio da exploração e da experimentação. Cada fase apresenta um novo conjunto de puzzles relacionado a uma das habilidades do PC consideradas.

3.2 Estrutura do Jogo

O jogo é estruturado em torno de quatro fases, quais sejam: i) Algoritmo; ii) Decomposição de problemas; iii) Abstração e iv) Reconhecimento de padrões.

Na primeira fase, relacionada à habilidade de algoritmo do PC, a movimentação do personagem depende da criação de um algoritmo a partir da combinação de 3 tipos de setas, que representam os movimentos: para frente, para a esquerda e para a direita, como pode ser visto na Figura 1. Para completar a fase, o jogador terá que analisar o mapa do local em que o personagem se encontra e criar a sequência correta de comandos de movimentação (setas), para que o personagem chegue ao local desejado. Esta fase visa trabalhar a habilidade de criar um algoritmo para resolver um problema.

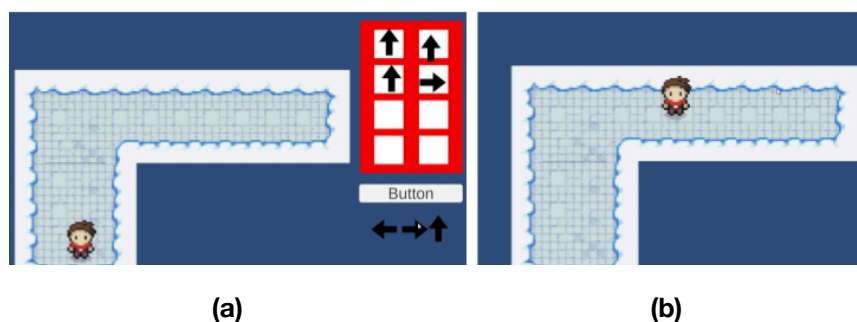


Figura 1 - (a) Fase Algoritmo. (b) Apresentando o resultado do algoritmo criado.

Na Fase Decomposição de problemas, o jogador se depara com um labirinto, com obstáculos que impedem a passagem do seu personagem, como pode ser visto na Figura 2. Este labirinto é dividido em 3 fileiras, cada uma com uma combinação de obstáculos, que pode ser alterada ao clicar nas setas localizadas nos cantos do cenário. Para chegar até o destino, o jogador precisa modificar o labirinto, clicando nas setas, até montar um caminho sem obstáculos, que o personagem possa percorrer.



Figura 2 - (a) Labirinto da decomposição; (b) Resolvendo o labirinto.

Na Fase Abstração, o personagem poderá se mover pelo cenário de três maneiras distintas: i) na direção diagonal; ii) nas direções horizontal e vertical e iii) em L. Para escolher a maneira como irá se movimentar, é necessário clicar na sua peça de xadrez correspondente, sendo: i) bispo para movimentos na diagonal; ii) torre para movimentos horizontais e verticais e iii) cavalo para movimentos em L, como em um jogo de xadrez.

Ao selecionar a peça desejada, cujas opções aparecem abaixo do personagem na Figura 3, surgem quadrados coloridos ao redor do personagem indicando os locais que ele poderá alcançar com aquele movimento. O jogador, então, deverá clicar no local

desejado. A principal ideia desta mecânica é que o jogador consiga abstrair os diferentes padrões de movimentação do personagem e utilizá-los para alcançar o objetivo.

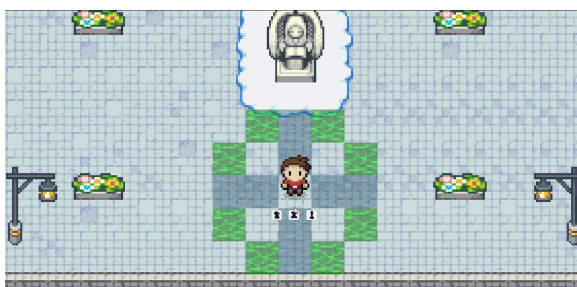


Figura 3 - Abstração, com os quadrados indicando movimentações possíveis.

A última fase, de reconhecimento de padrões, se encontra na etapa de planejamento.

3.3 Ficha Técnica

O desenvolvimento do jogo foi realizado utilizando o motor gráfico *Unity Engine*, versão 2022.3.21, com a linguagem de programação *C#*. A escolha da *Unity* foi baseada em sua extensa gama de ferramentas, que oferece suporte abrangente para as diversas fases de desenvolvimento de jogos, incluindo prototipagem, design e implementação final. Além disso, a popularidade da *Unity* na indústria de jogos facilita o acesso a uma ampla comunidade de desenvolvedores e recursos de suporte técnico, o que contribui significativamente para a resolução de problemas e o aprimoramento contínuo do projeto.

Outro fator determinante na escolha da *Unity* foi sua capacidade de exportar o jogo para múltiplas plataformas de maneira eficiente. A *Unity* permite a geração de executáveis compatíveis com diversos sistemas operacionais, como *Windows* e *macOS*, além de suportar dispositivos móveis, incluindo aqueles baseados em *Android* e *iOS*. Essa flexibilidade de distribuição é crucial para alcançar um público-alvo mais amplo e garantir que o jogo seja acessível em diferentes dispositivos, proporcionando uma experiência consistente e de alta qualidade independentemente da plataforma utilizada.

4. Análise

O *NewBot* foi inspirado em um jogo desenvolvido anteriormente [Higuchi et al. 2021], compartilhando o objetivo de ensinar habilidades do PC por meio de uma experiência de jogo educativo. No entanto, enquanto o jogo inicial se concentra predominantemente na habilidade de algoritmos, com todas as fases seguindo uma mecânica similar, o *NewBot* busca explorar uma gama mais ampla de habilidades relacionadas ao PC. Cada fase do *NewBot* é projetada para desenvolver uma habilidade específica, como decomposição de problemas, abstração, algoritmos, e reconhecimento de padrões, proporcionando uma abordagem mais diversificada e abrangente.

Originalmente, considerou-se a possibilidade de continuar o desenvolvimento do jogo inicial. No entanto, devido a diferenças nas versões de software, questões de

compatibilidade e a oportunidade de criar uma narrativa linear única para o novo jogo, optou-se por desenvolver o *NewBot* como um projeto independente. Essa decisão permitiu uma maior liberdade criativa e técnica, resultando em um jogo que não apenas preserva o valor educativo do original, mas também expande suas possibilidades pedagógicas e de entretenimento.

5. Próximos Passos

Os próximos passos envolvem três atividades: i) desenvolvimento de novos desafios em cada habilidade do PC já implementadas; ii) desenvolvimento da fase relativa à habilidade de reconhecimento de padrões; iii) disponibilização do jogo e iv) testes.

A estrutura do jogo desenvolvido permite que os desafios aumentem gradualmente em complexidade, oferecendo ao jogador uma curva de aprendizado suave. Essa abordagem garante que, ao enfrentar desafios cada vez mais difíceis, o jogador possa aplicar os conhecimentos adquiridos nas fases anteriores, aprimorando suas habilidades e desenvolvendo uma compreensão mais profunda sobre o PC. Por se tratar de um trabalho em andamento, o método a ser utilizado na escolha das quatro fases, bem como a abordagem e todos os desafios propostos dentro de cada fase, ainda se encontram em discussão. Após a conclusão do desenvolvimento, o jogo será disponibilizado para o público por meio de um website dedicado e haverá a possibilidade de download do jogo através das principais lojas de aplicativos para cada sistema operacional.

Além da distribuição do jogo, será elaborado um manual contendo as soluções das fases do jogo, o qual servirá como apoio para educadores, auxiliando-os a integrar o jogo no processo de aprendizagem dos alunos.

Para garantir a eficácia educativa e a qualidade técnica do jogo, serão realizados testes abrangentes, começando com a submissão do projeto ao Comitê de Ética. Após a aprovação, serão conduzidos experimentos em uma escola de ensino básico. Este processo de testes é uma etapa fundamental para assegurar que o jogo cumpra seu objetivo principal de ensinar as quatro habilidades do pensamento computacional consideradas e utilizará .

Os testes serão compostos de avaliações qualitativas e quantitativas, focando na jogabilidade, experiência do usuário e eficácia pedagógica em que as possibilidades consideradas poderão ser de Costello e Edmonds [2007] e de Oliveira et al. [2021]. Os dados coletados serão usados para permitir ajustes precisos e garantir que o jogo ofereça uma experiência de aprendizagem envolvente e eficaz. Além disso, serão realizados testes técnicos rigorosos para identificar e corrigir possíveis bugs, otimizar o desempenho e refinar a usabilidade, garantindo uma experiência de jogo suave e sem falhas. Esses testes técnicos incluirão avaliações de compatibilidade em várias plataformas e dispositivos, assegurando que o jogo funcione de maneira consistente em todas as configurações suportadas.

Referências

- Atmatzidou, S. and Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75B, 661–670. <https://doi.org/10.1016/j.robot.2015.10.008>
- Ayman, R., Sharaf, N., Ahmed, G. and Abdennadher, S. (2018). MiniColon; Teaching kids computational thinking using an interactive serious game. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 79–90. Springer Verlag.
- Barr, V. and Stephenson, C. (2011) Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *ACM Inroads* 2(1), 48–54. <https://doi.org/10.1145/1929887.1929905>
- Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-021-10700-2>
- Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Paper presented at the annual American educational research association meeting, Vancouver, BC, Canada (p. 25).
- Bocconi, S., Chiocciariello, A., Dettori, G., et al. (2016). Developing computational thinking in compulsory education. JRC Science for Policy Report 68, European Commission.
- Cano, S., Naranjo, J.S., Henao, C., et al. (2021). Serious game as support for the development of computational thinking for children with hearing impairment. *Appl. Sci. (Switzerland)* 11, 1–19. <https://doi.org/10.3390/app11010115>
- Castro, L.M.C., Shoib, H., Douglas, K.A., et al. (2021). Computational thinking frameworks used in computational thinking assessment in higher education. A systematized literature review. In: *ASEE Annual Conference and Exposition, Conference Proceedings*.
- Costello, B. and Edmonds, E. (2007). A study in play, pleasure and interaction design. *Proceedings of the 2007 Conference on Designing Pleasurable Products and Interfaces*. 76-91.
- Elshahawy, M., Bakhaty, M. and Sharaf, N. (2020). developing computational thinking for children with autism using a serious game. In: *Proceedings of the International Conference on Information Visualisation*. Institute of Electrical and Electronics Engineers Inc., pp 761–766.
- García-Peñalvo, F. J., and Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. In *Computers in Human Behavior (Vol. 80, pp. 407–411)*. Elsevier Ltd. <https://doi.org/10.1016/j.chb.2017.12.005>
- Giannakoulas, A. and Xinogalos, S. (2023). Studying the effects of educational games on cultivating computational thinking skills to primary school students: a systematic literature review. *J. Comput. Educ.* <https://doi.org/10.1007/s40692-023-00300-z>

Grover, S. and Pea, R. (2018). Computational Thinking: A Competency Whose Time Has Come. In S. Sentance, E. Barendsen & C. Schulte (Eds.). *Computer Science Education: Perspectives on Teaching and Learning in School* (pp. 19–38). London: Bloomsbury Academic. <https://doi.org/10.5040/9781350057142.ch-003>

Guenaga, M., Mentxaka, I., Garaizar, P., et al. (2017). Make world, a collaborative platform to develop computational thinking and STEAM. In: Zaphiris, P., Ioannou, A. (eds.) *Learning and collaboration technologies*, pp. 50–59. Springer International Publishing, Cham, Technology in Education.

Gutiérrez, S.A.C., Dávila, G.A. and Quintana, H. (2023). Implementation of a Serious Game to Develop Computational Thinking Skills. In: Pereira, R., Bianchi, I., Rocha, Á. (eds) *Digital Technologies and Transformation in Business, Industry and Organizations. Studies in Systems, Decision and Control*, vol 497. Springer, Cham. https://doi.org/10.1007/978-3-031-40710-9_9

Higuchi, V., Rocha, R. V., Bezerra, D. S., and Goya, D. H. AlgoBot: jogo sério para o desenvolvimento do pensamento computacional. <https://www.sbgames.org/proceedings2021/WorkshopG2/219258.pdf>

Ho, W.K., Looi, C.K., Huang, W., et al. (2021). Computational thinking in mathematics: to be or not to be, that is the question. In: *Mathematics—Connection and beyond: Yearbook 2020 association of mathematics educators*, pp 205–234. World Scientific.

Hoyles, C. and Noss, R. (2015). Revisiting programming to enhance mathematics learning. In: *Math+ Coding Symposium*. Western University.

Oliveira, R. N. R., and Rocha, R. V. (2021). AvaliaJS: Planejamento da Avaliação do Desempenho de Alunos em Jogos Sérios. X Congresso Brasileiro de Informática na Educação (CBIE 2021). <https://www.academia.edu/download/90778157/18031.pdf>

Papadakis, S. (2021). The Impact of Coding Apps to Support Young Children in Computational Thinking and Computational Fluency. A Literature Review. In *Frontiers in Education* (Vol. 6). <https://doi.org/10.3389/educ.2021.657895>

Papert, S. (1980). Computers for children. In *Mindstorms: Children, computers and powerful ideas*.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*. <https://doi.org/10.1007/BF00191473>

Rao, T.S.S. and Bhagat, K.K. (2024). Computational thinking for the digital age: a systematic review of tools, pedagogical strategies, and assessment practices. *Education Tech Research Dev*. <https://doi.org/10.1007/s11423-024-10364-y>

Schez-Sobrino, S., Vallejo, D., Glez-Morcillo, C., et al. (2020). RoboTIC: a serious game based on augmented reality for learning programming. *Multimed Tools Appl* **79**, 34079–34099. <https://doi.org/10.1007/s11042-020-09202-z>

Shute, V. J., and Ke, F. (2012). Games, learning, and assessment. In *Assessment in game-based learning* (pp. 43–58). Springer. doi:10.1007/978-1-4614-3546-4_4

Shute, V. J., Sun, C. and Asbell-Clarke, J. (2017). Demystifying computational thinking. In *Educational Research Review* (Vol. 22, pp. 142–158). Elsevier Ltd. <https://doi.org/10.1016/j.edurev.2017.09.003>

Silva, V., Diniz, J., and França, S. (2019). Jogos Digitais como Estratégia para Desenvolver o Pensamento Computacional nos Anos Finais do Ensino Fundamental. *Anais do Congresso sobre Tecnologias na Educação*.

Sun, L., Guo, Z., and Hu, L. (2021). Educational games promote the development of students' computational thinking: a meta-analytic review. <https://doi.org/10.1080/10494820.2021.1931891>

Susi, T., Johannesson, M. and Backlund, P. (2007). *Serious Games : An Overview, Institutionen för kommunikation och information, Skövde*.

Taslibeyaz, E., Kursun, E., and Karaman, S. (2020). How to develop computational thinking: A systematic review of empirical studies. *Informatics in Education*. <https://doi.org/10.15388/INFEDU.2020.30>

Theodoropoulos, A., and Lepouras, G. (2020). Digital Game-Based Learning and Computational Thinking in P-12 Education: A Systematic Literature Review on Playing Games for Learning Programming (pp. 159–183). <https://doi.org/10.4018/978-1-7998-4576-8.ch007>

Turchi, T., Fogli, D. and Malizia, A. (2019). Fostering computational thinking through collaborative game-based learning. *Multimed Tools Appl* 78, 13649–13673. <https://doi.org/10.1007/s11042-019-7229-9>

Utesch, M.C., Faizan, N.D., Kremer, H. and Heininger, R. (2020). Pic2Program—an educational android application teaching computational thinking. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*, pp 1493–1502.

Videnovik, M., Vold, T., Kionig, L., Madevska Bogdanova, A. and Trajkovik, V. (2023). Game-based learning in computer science education: a scoping literature review. *International Journal of STEM Education*. 10. 10.1186/s40594-023-00447-2.

Zaharija, G., Mladenović, S., and Boljat, I. (2013). Introducing basic Programming Concepts to Elementary School Children. *Procedia - Social and Behavioral Sciences*, 106, 1576–1584. <https://doi.org/10.1016/j.sbspro.2013.12.178>

Wassila, D., and Tahar, B. (2012). Using serious game to simplify algorithm learning. *International Conference on Education and E-Learning Innovations*, 1–5. [10.1109/ICEELI.2012.6360569](https://doi.org/10.1109/ICEELI.2012.6360569)

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. and Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *J. Sci. Educ. Technol.* 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2011). *Research notebook: Computational thinking—what and why*. The Link Magazine, Pittsburg, PA: Carnegie Mellon University, School of Computer Science. Retrieved from

XIII Congresso Brasileiro de Informática na Educação (CBIE 2024)
XXXV Simpósio Brasileiro de Informática na Educação (SBIE 2024)

<https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>.
Accessed 29 July 2024.