

K(o)mplexity: Design de um *Serious Game* com Integração Intrínseca para Ensino de Complexidade Algorítmica

Title: K(o)mplexity: Design of a Serious Game with Intrinsic Integration for Teaching Algorithmic Complexity

Jader Louis de Souza Gonçalves¹, Lucas Marques de Cunha¹

¹Departamento de Computação
Faculdade de Engenharia, Tecnologia e Inovação (FAETI)
Universidade Federal de Rondônia (UNIR)
Campus José Ribeiro Filho, BR-364, Km 9,5
CEP 76801-059, Porto Velho, RO, Brasil

jaderlouis@usp.br, lucas.marques@unir.br

Abstract. Introduction: Algorithmic complexity is central to Computer Science education, but is often learned only symbolically. **Objective:** This paper presents K(o)mplexity, an Android serious game that represents computational cost through energy consumption. **Steps:** The design is examined through the Learning Mechanics and Game Mechanics (LM-GM) framework [Arnab et al. 2015], intrinsic integration [Habgood e Ainsworth 2011], pedagogical progression, and three gameplay scenarios. **Expected Results:** The analysis maps concepts to game and learning mechanics and specifies objectives and evidence for each level. These artifacts are expected to support subsequent empirical evaluation of learning and engagement.

Keywords serious game, algorithmic complexity, intrinsic integration, formative feedback, game-based learning.

Resumo. Introdução: A complexidade algorítmica é central na formação em Computação, mas costuma ser aprendida apenas de modo simbólico. **Objetivo:** Apresenta-se o K(o)mplexity, um serious game Android que representa o custo computacional por consumo de energia. **Etapas:** O design é analisado pelo framework Learning Mechanics and Game Mechanics (LM-GM) [Arnab et al. 2015], pela integração intrínseca [Habgood e Ainsworth 2011], pela progressão pedagógica e por três cenários de gameplay. **Resultados Esperados:** A análise relaciona conceitos a mecânicas de jogo e de aprendizagem e especifica objetivos e evidências por fase. Os artefatos produzidos deverão subsidiar a avaliação empírica posterior da aprendizagem e do engajamento.

Palavras-Chave serious game, complexidade algorítmica, integração intrínseca, feedback formativo, game-based learning.

1. Introdução

A complexidade algorítmica (Big-O) é central na formação em Computação, mas sua notação pode ser memorizada sem compreensão das implicações no desempenho [Chuechote et al. 2020]. Jogos para Estruturas de Dados e Algoritmos (EDA) tendem

a focar operações pontuais [Genesio et al. 2024, Battistella et al. 2017, Su et al. 2021, Yohannis e Prabowo 2015]; jogos com código real [Kroustalli e Xinogalos 2021, Choi e Choi 2024] não adotam Big-O como mecânica central, e o *feedback* por modelos de linguagem ainda é recente [Petula et al. 2026].

Nesse contexto, propõe-se o design do **K(o)mplexity**, um *serious game*¹ para Android, no qual cada instrução escrita pelo jogador em uma linguagem específica de domínio (*domain-specific language*, DSL) consome energia conforme o custo definido no modelo de execução. Como consequência, algoritmos ineficientes esgotam os recursos disponíveis e levam à derrota.

A proposta é inspirada em *Factorio* [Reid et al. 2021] e *The Farmer Was Replaced* [Metaroot 2024], nos quais decisões algorítmicas produzem efeitos mensuráveis, embora não sejam exploradas explicitamente como objetivo educacional.

Formula-se a pergunta: *como mecânicas de jogo podem tornar observável a complexidade algorítmica por integração intrínseca?* A resposta mobiliza o *framework Learning Mechanics and Game Mechanics* (LM-GM) [Arnab et al. 2015, Lameris et al. 2017] e a integração intrínseca [Habgood e Ainsworth 2011, Cutting e Iacovides 2022]. A contribuição é um design educacional pré-empírico que combina código real, Big-O central e *feedback* por modelo de linguagem de grande porte (*large language model*, LLM), combinação não identificada no corpus analisado. Não se reivindica eficácia pedagógica.

2. Fundamentação Teórica

A aprendizagem de complexidade algorítmica exige relacionar notações assintóticas ao crescimento da entrada. A memorização de classes, porém, não assegura uma interpretação operacional [Chuechote et al. 2020]. Por esse motivo, o design educacional deve tornar perceptíveis as consequências das escolhas algorítmicas e permitir a comparação e a revisão de soluções.

Nesse sentido, a integração intrínseca estabelece que o conteúdo seja exercitado pela mecânica central [Habgood e Ainsworth 2011, Cutting e Iacovides 2022]. Para operacionalizar esse princípio, o LM-GM relaciona a ação do jogador ao processo cognitivo pretendido [Arnab et al. 2015, Lameris et al. 2017]. Desse modo, a taxonomia de Bedwell et al. [Bedwell et al. 2012] e a Teoria da Autodeterminação [Proulx et al. 2017] complementam o *framework* ao classificar atributos do jogo e explicar condições motivacionais; porém, nenhuma delas, isoladamente, demonstra o vínculo entre mecânica e aprendizagem.

Para que esse alinhamento seja cognitivamente acessível, a Teoria da Carga Cognitiva (*Cognitive Load Theory*, CLT) fundamenta a restrição de construções da DSL alheias ao objetivo educacional [Sweller 1988, Sweller et al. 2011], em consonância com acesso inicial simples [Resnick et al. 2009] e com a transição entre blocos e texto [Weintrop e Wilensky 2017]. Por sua vez, a Teoria Cognitiva da Aprendizagem Multimídia (*Cognitive Theory of Multimedia Learning*, CTML) orienta a contiguidade entre código, animação e energia e a remoção de sinais redundantes [Mayer 2020]. Portanto, a

¹Jogo digital cujo propósito primário excede o entretenimento [Arnab et al. 2015].

CLT regula a complexidade da tarefa, enquanto a CTML organiza sua apresentação verbal e visual.

Além de estruturar a tarefa e sua apresentação, o jogo deve orientar a revisão das soluções. Nesse nível, o *feedback* opera de duas formas: a energia informa o desempenho da solução, e Ada orienta o processo. Essa distinção deriva de Hattie e Timperley [Hattie e Timperley 2007] e dos princípios de especificidade e não revelação direta [Shute 2008, Tsai et al. 2015]. Desse modo, a LLM atua como mediadora da reflexão, com respostas ancoradas no estado do jogo e sujeitas a avaliação pedagógica.

3. Trabalhos Relacionados

Jogos com código real (*CodeCombat* [Kroustalli e Xinogalos 2021, Choi e Choi 2024], *Human Resource Machine* (HRM) [Tomorrow Corporation 2015], *Screeps* [Screeps LLC 2017], *Robocode* [Hartness 2004, O’Kelly e Gibson 2006]) ensinam sintaxe ou introduzem eficiência por orçamento de ciclos, mas classes assintóticas permanecem implícitas. Jogos para EDA cobrem operações pontuais (ordenação [Yohannis e Prabowo 2015, Battistella et al. 2017, Chuechote et al. 2020], visualização [Su et al. 2021], programação dinâmica [Zhu 2024], otimização [Martins et al. 2025]); mapeamento sistemático recente [Genesio et al. 2024] confirma o uso predominante de *quiz/drag-and-drop*. A produção brasileira recente cobre pilhas/filas (*Iron Ears* [da Rosa et al. 2020]) e busca em grafos (*Graph Defender* [Pavani et al. 2023]); retrospectiva recente [von Wangenheim et al. 2025] oferece análise do *Model for the Evaluation of Educational Games* (MEEGA+). No corpus analisado, não se identificou trabalho que combine escrita de código real, complexidade como mecânica central e *feedback* por LLM em um único jogo. A comparação sustenta a delimitação da contribuição de design.

4. Design do K(o)mplexity

O ciclo de jogo sintetiza seis decisões de design (Figura 1).

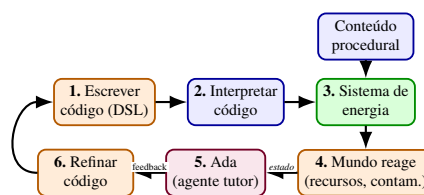


Figura 1. Ciclo de jogo do K(o)mplexity. Laranja: ações do jogador; verde: núcleo pedagógico (energia como *proxy* de Big-O); azul: sistemas internos; roxo: Ada, agente tutor.

Energia como integração intrínseca de Big-O. Em um mundo procedural e finito, o jogador coleta e processa recursos, e cada operação possui custo energético. O consumo varia com o algoritmo e a entrada; sob orçamento fixo, taxas de crescimento distintas geram consequências observáveis [Plass et al. 2013, Shah et al. 2012].

DSL, estruturas de dados e progressão. A DSL preserva sequência, seleção, repetição e operações sobre lista, fila e pilha, cujos custos orientam a escolha sob restrição de energia. A geração procedural varia entrada e recursos [Hooshyar et al. 2018]. A Tabela 1 sintetiza objetivos, desafios e evidências por fase.

Tabela 1. Progressão pedagógica e *level design* do K(o)mplexity.

Fase	Objetivo de aprendizagem	Desafio e mecânica	Evidência e <i>feedback</i>
1. Explorar	Relacionar instruções ao custo e distinguir $O(1)$ de $O(n)$	Coleta sequencial com energia limitada e entrada crescente	Conclusão no orçamento; barra associa execução e consumo
2. Escolher	Comparar custos de lista, fila e pilha	Seleção da estrutura para armazenar e recuperar recursos	Comparação entre tentativas; Ada solicita justificativa da escolha
3. Otimizar	Identificar repetição aninhada e crescimento quadrático	Reorganização da cadeia produtiva em mundo finito	Falha de $O(n^2)$ e sucesso de alternativa de menor crescimento
4. Transferir	Aplicar a estratégia a novas entradas	Variações procedurais de n , recursos e topologia	Estabilidade do desempenho; Ada promove reflexão entre cenários

A progressão antecede comparação de estruturas e otimização com custo por operação. Cada fase exige ação observável e justificativa; interface, tutorial e geração procedural permanecem componentes de suporte.

Ada: *feedback* formativo e *game feel*. Ada usa código, traço e estado para formular pistas sem fornecer soluções [Shute 2008, Tsai et al. 2015]. O contexto é ancorado por geração aumentada por recuperação (*retrieval-augmented generation*, RAG) em regras e exemplos validados [Lewis et al. 2020]; severidade, modelos de mensagem e pistas determinísticas reduzem respostas incorretas, dependência e variação, mas não as eliminam. Energia e sinais visuais são o retorno primário, contíguo à execução [Mayer 2020]; Ada exerce função reflexiva.

5. Análise de Design e Demonstração

Integração intrínseca. Habgood e Ainsworth [Habgood e Ainsworth 2011] requerem que o conteúdo seja entregue pela mecânica central (C1) e corporificado no mundo (C2). Em C1, escrever e executar a DSL constitui a interação principal; em C2, a energia representa o custo e conecta decisões a consequências [Shah et al. 2012, Plass et al. 2013, Reid et al. 2021]. O alinhamento oferece evidência analítica de coerência, não de aprendizagem.

Mapeamento LM-GM. A Tabela 2, baseada no LM-GM [Arnab et al. 2015] e em Bedwell et al. [Bedwell et al. 2012], separa conceitos, mecânicas e evidências para não classificar interface ou sistemas internos como mecânicas. O protótipo demonstra $O(1)$, $O(n)$ e $O(n^2)$; outras classes, espaço e análise amortizada permanecem fora do recorte.

Demonstração em cenários de *gameplay*. Na Figura 2, **C1** contrasta uma solução $O(n^2)$, que esgota a energia, com uma $O(n)$. Em **C2**, indexação $O(1)$ amortizada substitui busca $O(n)$; em **C3**, Ada aponta repetições sem revelar a solução. Os cenários demonstram comportamentos implementados, mas não mensuram compreensão.

6. Discussão e Considerações Finais

A análise indica coerência entre custo algorítmico, energia, comparação e refinamento. As contribuições são o mapeamento LM-GM (Tabela 2), a progressão por fases (Tabela 1), a análise de integração intrínseca [Habgood e Ainsworth 2011] e os cenários do protótipo (Figura 2). Elas sustentam consistência conceitual e implementação, sem permitir inferências sobre aprendizagem, motivação ou engajamento.

Tabela 2. Mapeamento entre conceito, mecânica de jogo, mecânica de aprendizagem e evidência no protótipo.

Conceito alvo	Mec. de jogo	Mec. de aprendizagem	Evidência no protótipo
Custo por operação	Orçamento de energia	Observação e <i>feedback</i> imediato	Drenagem associada ao traço de execução
Crescimento linear	Coleta sequencial	Experimentação e comparação	Consumo cresce com n no cenário C1
Crescimento quadrático	Mundo finito e energia	Experimentação e contraste de soluções	Esgotamento de $O(n^2)$ no cenário C1
Custo de estruturas	Armazenamento e recuperação	Modelagem e tomada de decisão	Escolha entre lista, fila e pilha em C2
Refinamento algorítmico	Ciclo executar e ajustar	Prática deliberada e reflexão	Tentativas comparadas por energia residual
Generalização	Variação procedural	Transferência entre problemas	Mudanças de n , recursos e topologia
Justificativa conceitual	Diálogo com Ada	Autorregulação e explicação	Pistas e perguntas vinculadas ao código
Progressão conceitual	Desbloqueio de fases	Instrução gradual e prática	Sequência definida na Tabela 1

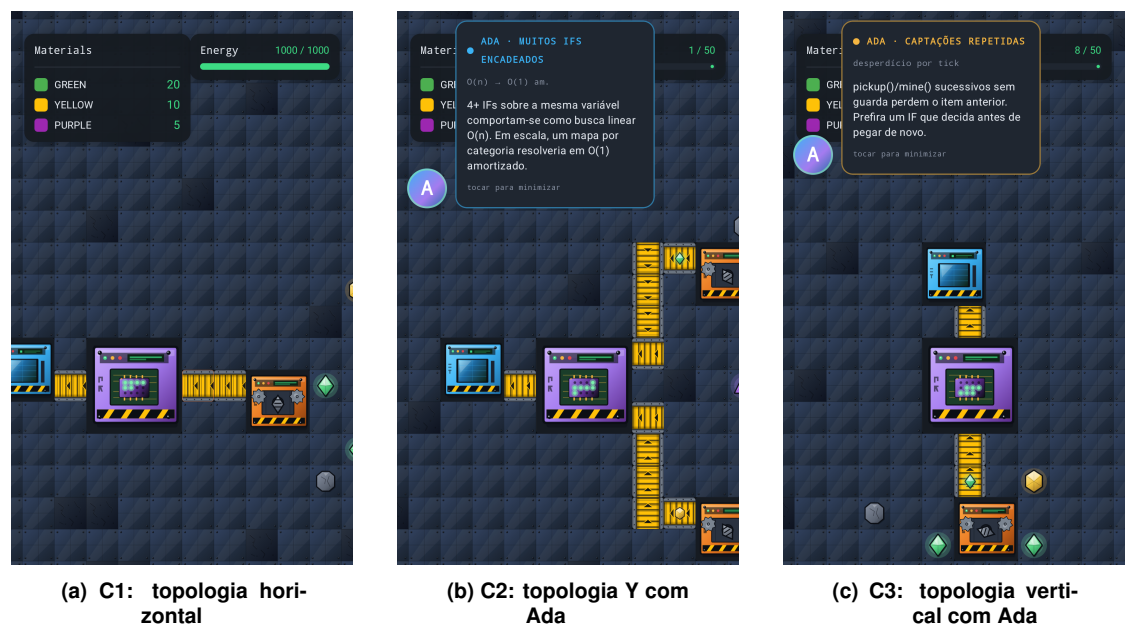


Figura 2. Cenários demonstrativos. Em (b,c), Ada apresenta pistas com severidade baixa e média, respectivamente.

A DSL restringe expressividade e pode ocultar custos de linguagens gerais, o que limita transferência. A LLM admite imprecisão, variabilidade, dependência e latência de 4 a 8 segundos. Não houve avaliação com estudantes, e o artigo não comporta o documento de design de jogo educacional (*educational game design document*, EGDD) completo. Trabalhos futuros devem expandir o EGDD; avaliar correção e utilidade de Ada; refinar a interface pela CTML [Mayer 2020]; e conduzir estudo pré e pós-intervenção, com registros e MEEGA+ [von Wangenheim et al. 2025].

Agradecimentos

Declaração de uso de inteligência artificial (IA) generativa: utilizou-se o OpenAI Co-dex no apoio à revisão linguística, editorial e de formatação. Os autores validaram as alterações e assumem integral responsabilidade pelo conteúdo.

Referências

- Arnab, S., Lim, T., Carvalho, M. B., Bellotti, F., de Freitas, S., Louchart, S., Suttie, N., Berta, R., e Gloria, A. D. (2015). Mapping learning and game mechanics for serious games analysis. *British Journal of Educational Technology*, 46(2):391–411.
- Battistella, P. E., von Wangenheim, C. G., von Wangenheim, A., e Martina, J. E. (2017). Design and large-scale evaluation of educational games for teaching sorting algorithms. *Informatics in Education*, 16(2):141–164.
- Bedwell, W. L., Pavlas, D., Heyne, K., Lazzara, E. H., e Salas, E. (2012). Toward a taxonomy linking game attributes to learning: An empirical study. *Simulation & Gaming*, 43(6):729–760.
- Choi, W. C. e Choi, I. C. (2024). The influence of CodeCombat on computational thinking in Python programming learning at primary school. In *Proceedings of the 2024 5th International Conference on Education Development and Studies (ICEDS)*, pages 26–32. ACM.
- Chuechote, S., Nokkaew, A., Phongsasithorn, A., e Laosinchai, P. (2020). A neo-piagetian analysis of algorithmic thinking development through the “sorted” digital game. *Contemporary Educational Technology*, 12(1):ep261.
- Cutting, J. e Iacovides, I. (2022). Learning by doing: Intrinsic integration directs attention to increase learning in games. *Proceedings of the ACM on Human-Computer Interaction*, 6(CHI PLAY).
- da Rosa, A., de Jesus, A., Igarashi, G. V., e Pereira, V. S. (2020). Iron ears: primeiras impressões de um jogo educativo para ensino de estrutura de dados lineares. In *Anais do XXVIII Workshop sobre Educação em Computação (WEI)*, pages 31–35, Cuiabá. Sociedade Brasileira de Computação.
- Genesio, N. O. S., de Oliveira, A. M., Chaves, L. J., Silva, W. A. F., e Valle, P. H. D. (2024). Um mapeamento sistemático sobre jogos educacionais digitais para o ensino-aprendizagem de estrutura de dados. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação (SBIE 2024)*, pages 819–837, Rio de Janeiro, RJ, Brasil. Sociedade Brasileira de Computação.
- Habgood, M. P. J. e Ainsworth, S. E. (2011). Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games. *Journal of the Learning Sciences*, 20(2):169–206.
- Hartness, K. (2004). Robocode: Using games to teach artificial intelligence. *Journal of Computing Sciences in Colleges*, 19(4):287–291.
- Hattie, J. e Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1):81–112.
- Hooshyar, D., Yousefi, M., Wang, M., e Lim, H. (2018). A data-driven procedural-content-generation approach for educational games. *Journal of Computer Assisted Learning*, 34(6):731–739.
- Kroustalli, C. e Xinogalos, S. (2021). Studying the effects of teaching programming to k-6 students through a serious game: A case study with codecombat. *Education and Information Technologies*, 26(4):4431–4451.

- Lameras, P., Arnab, S., Dunwell, I., Stewart, C., Clarke, S., e Petridis, P. (2017). Essential features of serious games design in higher education: Linking learning attributes to game mechanics. *British Journal of Educational Technology*, 48(4):972–994.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., e Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Martins, W., Honda, F., Pires, F., e Pessoa, M. (2025). OptimaCorporation: um jogo educacional interativo para redução da percepção de complexidade em problemas de otimização. In *Anais do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 1910–1921, Salvador, BA. SBC.
- Mayer, R. E. (2020). *Multimedia Learning*. Cambridge University Press, Cambridge, 3 edition.
- Metaroot (2024). The farmer was replaced. [videogame].
- O’Kelly, J. e Gibson, J. P. (2006). RoboCode & Problem-Based Learning: A Non-prescriptive Approach to Teaching Programming. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE ’06)*, pages 217–221, New York, NY, USA. ACM.
- Pavani, R. B., Junior, M. M. C., e Aylon, L. B. R. (2023). Busca em largura e profundidade aplicado a jogos educativos: estudo de caso no projeto do jogo Graph Defender. In *Anais Estendidos do XXII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*. SBC.
- Petula, W., Joshi, A., Tu, P., Somasundar, A., e Saha, S. (2026). DeliverC: Teaching pointers through GenAI-powered game-based learning. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education (SIGCSE TS 2026)*, pages 838–844. ACM. arXiv:2509.14496.
- Plass, J. L., O’Keefe, P. A., Homer, B. D., Case, J., Hayward, E. O., Stein, M., e Perlin, K. (2013). The impact of individual, competitive, and collaborative mathematics game play on learning, performance, and motivation. *Journal of Educational Psychology*, 105(4):1050–1066.
- Proulx, J.-N., Romero, M., e Arnab, S. (2017). Learning mechanics and game mechanics under the perspective of self-determination theory to foster motivation in digital game based learning. *Simulation & Gaming*, 48(1):81–97.
- Reid, K. N., Miralavy, I., Kelly, S., Banzhaf, W., e Gondro, C. (2021). The factory must grow: Automation in Factorio. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO ’21)*, pages 243–244, New York, NY, USA. Association for Computing Machinery. Also available as arXiv preprint arXiv:2102.04871.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., e Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11):60–67.
- Screeps LLC (2017). Screeps: World. [videogame].

- Shah, A. K., Mullainathan, S., e Shafir, E. (2012). Some consequences of having too little. *Science*, 338(6107):682–685.
- Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.
- Su, S., Zhang, E., Denny, P., e Giacaman, N. (2021). A game-based approach for teaching algorithms and data structures using visualizations. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*, pages 1128–1134, New York, NY, USA. ACM.
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285.
- Sweller, J., Ayres, P., e Kalyuga, S. (2011). *Cognitive Load Theory*, volume 1 of *Explorations in the Learning Sciences, Instructional Systems and Performance Technologies*. Springer, New York, NY.
- Tomorrow Corporation (2015). Human resource machine. [videogame].
- Tsai, F.-H., Tsai, C.-C., e Lin, K.-Y. (2015). The evaluation of different gaming modes and feedback types on game-based formative assessment in an online learning environment. *Computers & Education*, 81:259–269.
- von Wangenheim, C. G., Petri, G., e Borgatto, A. F. (2025). Using MEEGA/MEEGA+ for the evaluation of educational games: A retrospective. *Revista Brasileira de Informática na Educação*, 33:943–971.
- Weintrop, D. e Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education*, 18(1):1–25.
- Yohannis, A. R. e Prabowo, Y. D. (2015). Sort attack: Visualization and gamification of sorting algorithm learning. In *2015 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)*, pages 1–8, Skövde, Sweden. IEEE Computer Society.
- Zhu, Y. (2024). DPGame: Game-based learning for dynamic programming algorithms. In *Entertainment Computing – ICEC 2024*, volume 15192 of *Lecture Notes in Computer Science*, pages 276–285. Springer.