

Translating Elementary Cellular Automata into Tangible Gameplay: Design of Shellmate

Sagnik Ghosh and Prasad Bokil

IDC School of Design, IIT Bombay, Mumbai, India

22b3631@iitb.ac.in

prasad.bokil@iitb.ac.in

Abstract: Computational thinking is increasingly being recognized as a critical cognitive skill beyond computer science education (Wing, 2006) (Shute et al., 2017). While game-based approaches to fostering computational thinking have been explored, much of the existing work focuses on instructional tools (Wu et al., 2025) (Wang et al., 2023). There is limited research on how abstract computational models can inform the design of non-digital strategy games. The aspects of Elementary Cellular Automata were analysed to identify transferable principles such as local rules, generational growth, and emergent pattern formation. This paper presents a case study of the design and development of Shellmate, a two-player strategy-based board game derived from the principles of Elementary Cellular Automata (ECA). The study documents how key ECA concepts, including neighbourhood interactions, rule-based state transitions, and generational progression, were translated into game elements through an iterative design process. Design decisions were guided by repeated prototyping and playtesting, allowing mechanics to be refined while preserving the underlying computational principles. Rule combinations and edge cases were explored through repeated rapid prototyping before arriving at the final design. The final playtesting was conducted with 10 participants in pairs over two rounds each, where participants showed progressive improvement in rule card selection and strategic planning, with blocking and diagonal anticipation emerging as dominant behaviours. A central finding from the design process is that strict adherence to ECA simulation rules proved incompatible with playable game design. Leaving some neighbourhood combinations undefined within each round, while departing from what a complete ECA would require, was a necessary decision that introduced uncertainty and created scope for player agency. Playtesting showed that this structure naturally prompted computational thinking behaviours such as pattern anticipation, forward planning, and adaptive reasoning, without explicitly instructing players in the underlying model. Participants gradually shifted to anticipating longer-term consequences of their actions, suggesting that the evolving game state encouraged strategic reasoning through repeated play. The paper demonstrates that ECA as a computational model can serve as a structural foundation for game design, and that the translation process explored here could be helpful for gamifying simple computational models. The resulting design process highlights how abstraction and selective implementation can preserve the underlying logic of a computational model while leaving space for player agency and playability. This approach has potential implications for computational thinking education, where physical game systems can engage learners with computational concepts without direct instruction. Notably, leaving rule combinations partially undefined proved not to be a design compromise but a mechanism that actively drives engagement and reasoning.

Keywords: Game design, Computational games, Board game, Serious games, Elementary cellular automata

1. Introduction

Using computational thinking (CT) as a basis for designing games is a direction that has been explored in previous research. However, much of this work focuses on educational activities and guided learning experiences intended to introduce programming logic and algorithmic thinking (Wing, 2006) (Shute et al., 2017). While CT based games are common in digital format, less work has been done around designing tangible tabletop games using computational thinking and computational models. Unlike digital tools, tabletop games offer a hands-on environment where players engage with computational logic through physical interaction (Poole et al., 2021) (Salen & Zimmerman, 2003).

Elementary Cellular Automata (ECA) presents a compelling computational model in this context. ECA operates by applying simple evolution rules across multiple cells in a generation. Its minimal rule structure and generative behaviour make it particularly suitable for translation into a playable system. ECA can produce multiple intricate patterns, and understanding ECA involves dealing with several key aspects of CT including pattern recognition and decomposition, rule based reasoning and emergent complexity (Wu et al., 2025) (Wang et al., 2023). The structural and evolutionary principles of ECA form the core of Shellmate, a 2-player strategy-based board game. Shellmate does not directly simulate an ECA or teach the model explicitly. Instead, it requires players to make decisions based on past moves, while trying to anticipate future turns, within a rule-based environment inspired by ECA. The game adapts concepts such as generational progression, neighbourhood interactions and rule sets into a tangible playable artifact.

The playtesting aimed to examine how players engaged with the system, particularly in terms of strategy formation and pattern anticipation within rule-based constraints. This paper makes the following contributions:

It presents a case study of the design and development process of Shellmate and demonstrates a process for translating computational models into tangible tabletop games.

This work demonstrates a replicable design process for grounding game mechanics in computational structures, offering a practical reference for educators and designers seeking to embed computational concepts into tangible play experiences without relying on explicit instruction or simulations.

2. Elementary Cellular Automata (ECA)

Elementary Cellular Automata (ECA) are simple, one-dimensional computational models in which cells exist in binary states (0 or 1) and evolve over discrete time steps based on local rules (Wolfram, 2002). ECAs have been used to model a range of generative and natural pattern-forming systems such as in genetics, shell growth, coral formation, and animal skin patterns, making them an effective abstraction tool for studying decentralized, generative behaviour (Redeker et al., 2013).

2.1 ECA Structure

The row-by-row, generational growth in an ECA creates patterns that are often highly intricate (Meunier, 2016). Figure 1 shows the structure of an ECA and its generational evolution. The key aspects of an ECA are the following:

- Each cell exists in one of two states: 0 or 1.
- The next state of a cell is determined by its own current state and the states of its immediate left and right cells.
- A uniform rule determines how all cells evolve.
- All cells in a generation update their states simultaneously.



Figure 1: Abstraction of ECA Structure

Associated Terms

Cells: Binary unit (0/1) in the system. **Neighbours:** Cells to the immediate left and right of one cell. **Neighbourhood:** A cell along with its neighbours. **State:** The current value of the cell (0 or 1). **Generation:** A row of cells created by applying evolution rules to the previous row, representing a single step in the automaton's evolution. **Rule Sets:** Evolution rules mapping from neighbourhood configuration to next state.

2.2 Rule Sets and Generational Growth in an ECA

In an ECA, each cell's next state depends on itself and its two immediate neighbours (3 cells in total), defined by ECA rules. With three binary inputs, there are 8 possible neighbourhood configurations (e.g., 111, 110, 000). Each configuration maps to an output of either 0 or 1. A rule set defines one output for each of these 8 configurations, and since each configuration can have 2 possible outputs, there are 256 (2^8) possible rule sets in total (Borriello & Walker, 2017; Wolfram, 2002).

The evolution of an ECA happens row by row, with each generation forming the basis for the next. Over time, this cumulative stacking of rows based on the active rule set allows development of complex patterns on a two-dimensional plane.

3. Translating ECA to Game Mechanics

This section describes how the principles of Elementary Cellular Automata (ECA) were translated into the mechanics of Shellmate. Selected structural aspects such as neighbourhood-based interactions, rule-driven state changes, and stepwise evolution were adapted into a playable system. The aim was to retain the underlying logic of ECA while making it accessible through tangible gameplay.

3.1 Abstraction of Computational Principles

ECA operates through simple local rules, where each cell updates its state based on its immediate neighbours, producing patterns that evolve over successive generations.

In designing Shellmate, these principles were adapted to support player interaction instead of direct simulation. The design focuses on three key aspects: local dependency, where moves are influenced by neighbourhoods, rule consistency where the same logic applies across the game and generational progression where the game evolves step by step. Player actions replace automatic updates while still operating within these rule-based constraints.

3.2 Rule Cards

In an ECA, state transitions over generations happen automatically since every neighbourhood configuration maps to a fixed output. This logic was translated into a set of 16 cards, each showing one of the 8 possible neighbourhood configurations paired with one of its two possible outcomes. Each permutation was assigned a number (1–8), and the two possible outcomes were labelled ‘a’ and ‘b’ (e.g., 1a, 1b). The card with permutation 1 and outcome a was labelled 1a, and so on. To introduce player agency, only 4 out of 8 possible configurations were to be defined by rules at once. The chosen rule cards would remain unchanged for 1 round of the game. Figure 2 shows all the 16 rule cards designed initially.

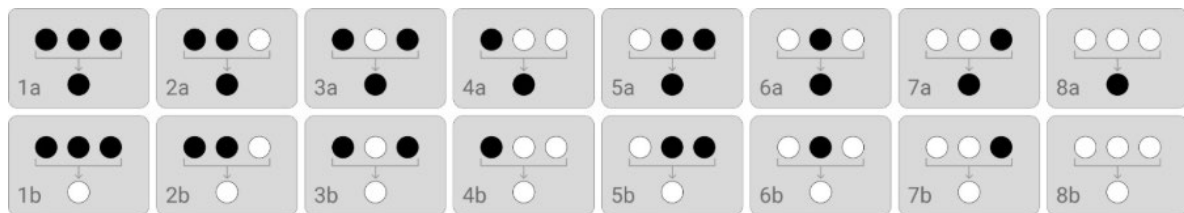


Figure 2: Breakdown of Rule Cards

3.3 Mapping ECA to Game Elements

Table 1 summarises how each element of ECA was mapped to a game element. The game elements are grouped into three types: game components (tangible elements), Game mechanics (Rules that govern the gameplay) and Outcomes (results produced by the players as a result of abiding by the game rules).

Table 1: Mapping of ECA Concepts to Game Elements

ECA Concept	Roll in System	Game Element	Type
Cell Grid	Spatial structure for growth across generations	8×8 square grid	Component
Binary Cell State (0 or 1)	Fundamental unit representing system state	Bi-colour tokens	Component
Update Rules (256 ECA Rules)	Definite range of possible system behaviours	Rule Cards	Component
3 Cell Neighbourhood	Defines local context for computing next state	Pattern configurations used in rule cards	Mechanics
Local Rule Update	Determines next state based on current neighbourhood	Rule cards and “reserved spaces”	Mechanics
Generational Evolution	System evolves step by step over discrete time steps	Row-wise progression of gameplay	Mechanics
Emergent Pattern Formation	Complex global behaviour from local interactions	Strategic formations emerging during gameplay	Outcome
Generation 0	Basis of formation of an ECA	Initial board setup	Outcome

4. Design of ShellMate

Shellmate is a two-player strategy-based board game designed around rule-based interactions derived from ECA principles. The game is played on a 2D grid where players take turns placing tokens while navigating constraints defined by rule cards. The system is structured to allow patterns to emerge over time through turn-by-turn token placement.

4.1 Game Components

The game consists of a common board, 64 tokens, and 16 rule cards. The board is an 8×8 grid that serves as the playing surface. Gameplay progresses row by row, with players filling each row sequentially. The tokens are bi-coloured and represent the two players. Players place these tokens on the board during their turns. The rule cards define constraints on where tokens can be placed. Each card specifies a pattern condition that determines valid or restricted positions during gameplay.

4.2 Core Mechanics

Gameplay proceeds through turn-based placement of tokens, with players alternating turns to fill the grid row by row. Each move is constrained by the active rule cards. The grid evolves sequentially, with each completed row influencing the next. This introduces a dependency between moves, requiring players to consider both current placements and their future implications. While ECA defines all possible neighbourhood configurations, Shellmate uses a subset of rule cards in each round. This allows players choice and makes the system playable besides introducing variability and allowing different rule combinations to shape each game. In addition to placement constraints, the game incorporates competitive elements such as blocking and capturing, which introduce interaction between players and affect the evolving board state. During the design process, different rule combinations were explored through repeated prototyping and playtesting, allowing gameplay balance and edge cases to be refined before arriving at the final design.

4.3 Design Iterations

The final design of Shellmate emerged through multiple iterations, each addressing specific gameplay and balance challenges. This section outlines the progression of the game through successive prototypes and the design decisions that shaped its current form.

Iteration 1: The initial prototype of Shellmate was played on an 8×8 grid using two types of tokens and a set of rule cards. A subset of 4 neighbourhood rules was active in each round. Players selected rule cards at the beginning of the game, with mutually exclusive pairs ensuring that contradictory rules were not used together. Figure 3 shows the components of the game.

Gameplay proceeded through row-by-row placement of tokens under rule-based constraints, with additional mechanics such as filler placement used when no valid moves were available. The game concluded once the grid was filled, with scoring based on token count. Player with the most tokens on board wins. With an average playing time of 20-25 mins, the gameplay had the following design strengths and limitations.

Design strengths: The initial prototype successfully established a system of row-wise progression and rule-based constraints, allowing players to engage with generational growth and local interactions through gameplay.

Design issues: Players who placed "fillers" often gained an unfair advantage.

Victory was heavily influenced by the initial rule card selection.

Limited scope for strategizing, since the primary focus was only gaining more spaces.

Some rule cards were disproportionately powerful, disrupting balance.

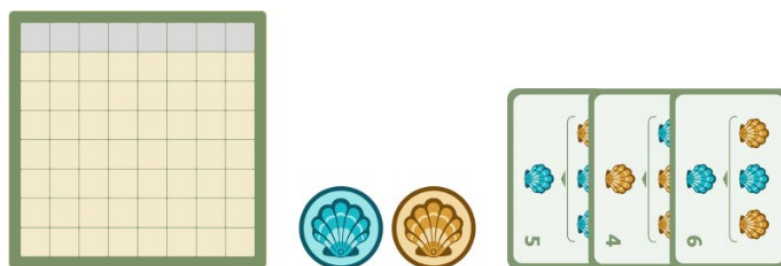


Figure 3: Final Game Components

Iteration 2: To address these issues, the winning condition was modified to emphasise diagonal formation, encouraging more focused strategic play and reducing reliance on filler placement. The rule cards 1a and 8b were removed to reduce imbalance. The filler mechanic was modified: if a player placed fillers to complete a row, their opponent would begin the next row. The average playing time was reduced to 15 mins.

Design issues: Despite these changes, the game remained unbalanced. First player advantage persisted, and the revised winning condition proved too easy to achieve.

Final Iteration: The final iteration focused on improving balance and increasing strategic depth by extending gameplay and refining scoring conditions. Victory was now determined by the number and length of diagonals formed by each player's shells after the entire 8×8 grid has been filled- the longer the diagonal, the higher the points. This ensured that both players had nearly equal opportunities to fill the grid, minimizing first-player advantage. To add a final layer of conflict and surprise, a capturing mechanism was introduced, which is explained in detail in Figure 4. To simplify gameplay while retaining the essence of ECA, each round used two rule cards in play that were to be chosen at the beginning of the game. Additional rule cards were removed to simplify gameplay and ensure better balance.

The rule cards 3a and 6b were removed since they made capturing too easy, and cards 1a and 8b were re-introduced since the winning goal was no longer to have most pieces on board. The average playing time was around 30-35 mins.

In the final prototype, the tokens were made bi-coloured. The suffixes 'a' and 'b' were dropped from the cards. Cards with the same input combinations but opposite outputs have the same number on them and are called 'contradictory cards'.

4.4 Gameplay

The final gameplay in Shellmate involves a process of grid formation followed by a resolution phase.

Once the grid is filled, the game transitions into a capturing phase, where players convert opponent tokens based on positional conditions. This is followed by a scoring phase, where resulting patterns on the board are evaluated.

This progression from local decision-making based on active rule cards allows the system to evolve through player interaction rather than predefined computation.

Initial Setup: The players start with laying out the game board and selecting 1 colour each, yellow or blue. They shuffle eight 1-sided tiles (4 of each colour) into a bag and draw them out without looking at them to fill the first row of the board from left to right. Thereafter, they select 4 non-contradictory rule cards that govern the gameplay. Contradictory rule cards are denoted by the same number. e.g. referring to Figure 2, if a player has chosen card 2a, 2b cannot be chosen in the same game.

Turn Actions: Once the first row has been drafted and the rule cards have been chosen, players take turn placing their tokens on the topmost empty row (active row). They cannot however place tokens on spaces "reserved" for their opponents by the rule cards. In case a player cannot place anymore token on the active row, the other player fills the remaining spots on the active row with his tokens. This is called adding fillers. If a player ends a row by adding fillers, the other player starts the new row. If a player ends the row by placing just 1 token, they start the next row. Figure 4 summarizes the gameplay structure.

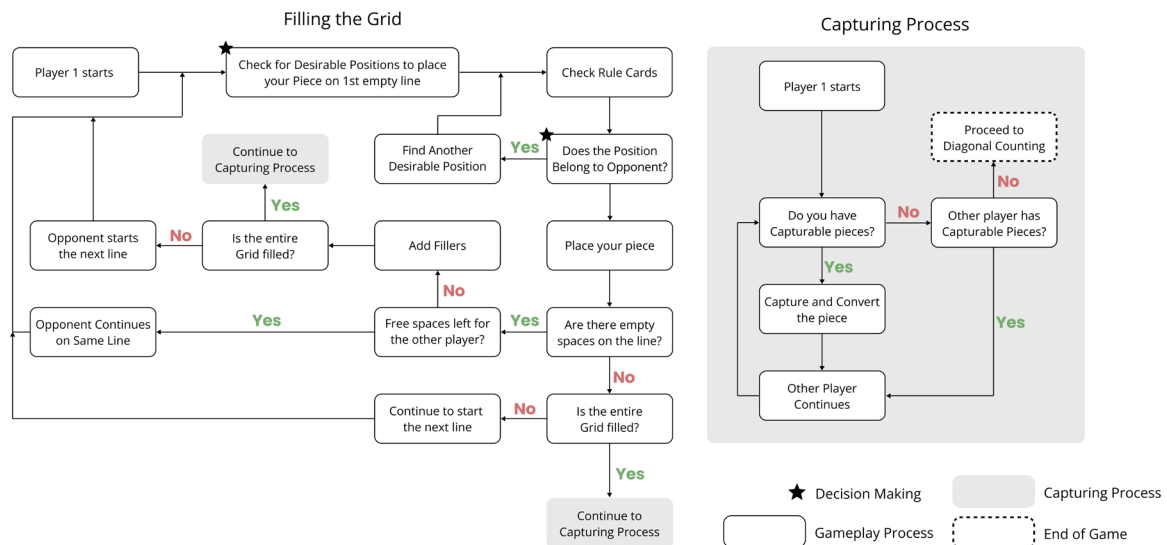


Figure 4: Flowchart illustrating the game mechanism

Capturing: Once the entire grid has been filled, players may take turns to capture their opponent's tokens trapped in the configuration shown in Figure 5 and convert it to their own colour. If a player is unable to capture any more pieces, the other player may keep capturing till no more pieces can be captured.

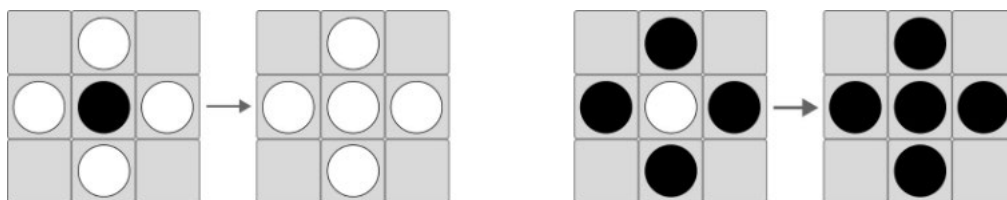


Figure 5: Capturing phase breakdown

End Scoring: After the capturing process, players count their diagonals. Each diagonal scores players a sum of $(n-2)$ where n is the length of the diagonal. Eg. A diagonal of 4 tokens would score 2 points.

5. Playtesting and Observations

Playtesting was conducted with 10 participants in pairs over two rounds each. Sessions lasted approximately 30 to 40 minutes. Gameplay was observed, and brief post-game interviews were conducted to capture player reflections on decision-making, strategy, and overall experience. Players took time to understand rule card logic in the first round, with several making poor initial selections. However, most showed clear improvement by the second round. One participant noted: "I messed up my first card pick, but in the second game I knew what to choose looking at the initial configuration." Another reflected: "Should have chosen the rule cards more wisely, I made a mistake, but I somehow worked around it." This suggests the game has a learning curve that rewards repeated play and making an initial mistake doesn't make recovery impossible.

Strategic planning developed progressively across the session. Players reported planning multiple moves ahead, with one comparing it to chess: "At least 2 lines ahead I was trying to think of." Blocking emerged as a dominant early strategy, "First I blocked, then later I focused on building." One player identified a trade-off cost to over-blocking: "When I realised I had spent time blocking; it left me with bad deals like a tetris piece." Players also demonstrated pattern anticipation based directly on rule constraints: "Eventually I knew it would form a straight line, unless I placed a piece differently." This reflects pattern recognition tied to ECA behaviour, where players began reading generational outcomes before they occurred.

Two friction points were consistently reported. Turn flow after row completion caused confusion: "There was confusion about who moves next after a row fills." The scoring phase was visually demanding: "Reading the diagonals at the end was mentally taxing." Both point toward areas where digital support or clearer visual design could reduce unnecessary cognitive load.

However, it is important to acknowledge that the playtest evidence and analysis remain highly qualitative as of now, relying heavily on qualitative data. Creating a clearer and better replicable structure might prove helpful here.

6. Discussion

In this work, ECA functions as a design grammar, where its principles are abstracted and selectively implemented to form an interactive artifact. A key design decision was to leave some neighbourhood combinations undefined within any single round. In a complete ECA simulation, all 8 rule configurations are fixed and the system evolves without ambiguity. Breaking this completeness was a deliberate gamification choice. Undefined combinations introduce uncertainty, and uncertainty creates space for player intent. Players reason under incomplete information, weigh risks, and commit to decisions they cannot fully control. Players have to plan, adapt, and reflect, which are the same processes that underlie computational thinking.

Observations from playtesting show that players engaged with behaviours closely tied to ECA properties, even without being told about the model. Players anticipated how current configurations would evolve. Others shifted strategies mid-game, moving from blocking to building as the board developed. These responses reflect pattern recognition, local rule reasoning, and adaptive planning, all of which are core aspects of computational thinking (Berland & Duncan, 2016). The game did not instruct players in these skills but created conditions where they were needed naturally.

Rapid prototyping of game components allowed faster iteration during the design process, which involved experimenting with rule combinations to identify balance issues and edge cases. More broadly, this case study suggests that computational models can offer a starting point for game design. The underlying logic of a model, its rules, states, and generational structure, can directly shape how a game is played and how players think within it. Future work could explore whether this approach generalises to other computational models, and whether a more structured framework can be built for this kind of translation.

7. Conclusion

This paper presented the design and iterative development of Shellmate, a tangible strategy game inspired by the structural principles of Elementary Cellular Automata. Deliberately leaving some neighbourhood combinations undefined, although departed from what a complete ECA simulation would require, proved necessary to make the system playable. The design process demonstrated that translating a computational model into gameplay can require preserving selected structural principles while adapting others to balance computational fidelity with strategic playability.

Compared to prior work on computational thinking games, which largely rely on explicit instruction or digital interfaces (Wu et al., 2025) (Wang et al., 2023), Shellmate demonstrates that computational logic can be embedded into physical gameplay without direct teaching. Players showed pattern recognition and adaptive planning behaviours that align with CT skills, emerging naturally through play rather than instruction. This distinction is the primary contribution of this work.

Limitations include the small qualitative sample and the absence of a structured evaluation instrument, which constrain generalisability. A digital or hybrid version could address the scoring complexity and rule card tracking friction identified during playtesting. More broadly, the approach taken here shows the potential to generalise to other computational models with similar local rule structures, and developing a reusable framework for this kind of translation remains a meaningful direction for future work.

Ethics Declaration: Due consent was taken from the participants before playtesting. Playtesting sessions lasted approximately one hour, before which participants were informed about the game concept and their approval to participate in the playtest was taken.

AI Declaration: No AI tools were used for generating any content for this paper. However, ChatGPT was used for checking grammatical and spelling errors after the contents of the paper had been put together.

References

- Berland, M., & Duncan, S. (2016). Computational thinking in the wild: Uncovering complex collaborative thinking through gameplay. *Educational Technology*, 29–35.
- Borriello, E., & Walker, S. I. (2017). An information-based classification of elementary cellular automata. *Complexity*, 2017, 1280351.

- Chen, C.-H., & others. (2023). Effectiveness of AR board game on computational thinking and programming skills for elementary school students. *Systems*, 11(1), 25. <https://doi.org/10.3390/systems11010025>
- Meunier, P.-É. (2016). Unraveling simplicity in elementary cellular automata. *Theoretical Computer Science*, 641, 2–10.
- Poole, F. J., Clarke-Midura, J., & Lee, V. R. (2021). Tabletop games designed to promote computational thinking. *Computer Science Education*. <https://doi.org/10.1080/08993408.2021.1947642>
- Redeker, M., Adamatzky, A., & Martínez, G. J. (2013). Expressiveness of elementary cellular automata. *International Journal of Modern Physics C*, 24(03), 1350010.
- Salen, K., & Zimmerman, E. (2003). *Rules of Play: Game Design Fundamentals*. MIT Press.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158.
- Wang, X., Cheng, M., & Li, X. (2023). Teaching and learning computational thinking through game-based learning: a systematic review. *Simulation & Gaming*. <https://doi.org/10.1177/07356331231180951>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media.
- Wu, C.-H., Chien, Y.-C., Chou, M.-T., & Huang, Y.-M. (2025). Integrating computational thinking, game design, and design thinking: a scoping review on trends, applications, and implications for education. *Humanities and Social Sciences Communications*, 12(1), 1–12.