

Design of a Programming Trainer Based on Narrative Theory

Huitong Zeng

Zhejiang University of Technology, Hangzhou 310000, Zhejiang, China

*Corresponding author: Huitong Zeng, zht00530@163.com

Copyright: © 2026 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

Abstract: Addressing the high cognitive barriers and abstract nature of early programming for children aged 5–8, this study integrates narrative theory into the design of Tangible User Interfaces (TUIs). We developed a five-dimensional narrative model encompassing themes, characters, actions, scenes, and props to mitigate learners' cognitive load through contextualized representation. A one-week comparative experiment demonstrated that children in the narrative tangible programming group significantly outperformed those in traditional computer programming and abstract tangible programming groups in terms of core concept comprehension, task efficiency, and self-correction proficiency. The findings suggest that narrative design achieves the “de-abstraction” of programming logic by embedding it into concrete storylines, fostering deep logical understanding in autonomous learning environments. This research provides valuable insights and design pathways for the development of early programming educational tools.

Keywords: Narrative theory; Tangible user interfaces; Programming trainer; Children aged 5–8

Online publication: May 21, 2026

1. Introduction

In recent years, the global focus of programming education has gradually shifted from higher education toward the K-12 stage^[1]. Against the backdrop of the digital era, early exposure to programming education is widely regarded as beneficial for enhancing children's future career competitiveness and life prospects. Furthermore, there is a broad academic consensus that systematic programming education can effectively cultivate and refine children's computational thinking and logical reasoning. For young learners, particularly those in home-based settings or lacking professional instruction, developing scientifically robust and effective pedagogical frameworks is of paramount importance^[2].

Early programming tools for children primarily relied on text-based or Graphical User Interfaces (GUIs), requiring the use of a mouse and keyboard to assemble code blocks. In response, many scholars have developed dedicated pedagogical models tailored to programming education. Game-Based Learning (GBL) emerging as a dominant method that plays a pivotal role in fostering students' programming skills^[3]. The immersive potential of

gamification has garnered significant attention. For instance, “Small World” programming games, grounded in computational thinking frameworks, transform abstract programming logic into concrete game mechanics. By utilizing meta-games for deep understanding and multi-round iterative optimization, these tools have effectively enhanced primary students’ comprehension of conditional statements, algorithm design, and testing^[4].

In a similar vein, Chen *et al.* leveraged unplugged board games to drive high-order computational thinking. By using card-based instructions and competitive play, these games encourage students to instinctively employ algorithmic logic and decomposition to secure a win^[5]. Lee developed a role-playing game (RPG) that reframes “programming” as “teaching a virtual character how to play” guiding children to iteratively transform game concepts into abstract algorithmic rules^[6]. While existing GBL models successfully engage children through intrinsic motivation and improve their logical decision-making, a significant challenge remains: due to their early stage of cognitive development, young children still operate within the concrete operational stage. Without the support of physical objects or vivid imagery, they often struggle to achieve a profound understanding of abstract learning concepts.

With the advancement of interaction technologies, the emergence of Tangible User Interfaces (TUIs) has enabled the materialization of programming commands by effectively lowering cognitive barriers for children through embodied interaction^[7]. Young children typically lack mature literacy, numeracy, and abstract reasoning skills. Without physical representations, they struggle to fully grasp abstract concepts and complex logic^[8]. The embodied cognition perspective posits that humans acquire conceptual knowledge through sensorimotor experiences derived from interacting with their environment^[9]. This framework facilitates early childhood learning by emphasizing how learners leverage bodily experiences to understand abstract concepts, asserting that cognition is deeply grounded in the physical environment^[10]. Consequently, this perspective aligns with young children’s developmental reliance on physical contexts^[11]. Furthermore, given that concept-based kinesthetic interaction can stimulate conscious reflection, researchers consider embodied approaches a valuable means of teaching abstract and symbolic systems, such as programming and mathematics, to beginners^[12]. Research by Sung *et al.* on kindergarten and first-grade children found that integrating computational perspectives with embodied practices significantly bolstered both mathematical understanding and programming skills. However, for children whose cognitive levels are still maturing, simply shifting from virtual media to tactile physical blocks is insufficient for achieving a complete understanding of abstract programming content.

Therefore, the introduction of narrative theory as a pedagogical support is of paramount importance. Its goal is to foster a deep understanding of core programming concepts through contextualized representation, particularly within autonomous learning environments. In the design of children’s products, especially toys, narrativity provides a framework where the child remains the protagonist throughout the experience, maintaining agency over the story’s progression and its conclusion. This fulfills the primary objective of narrative design: enhancing children’s engagement and interaction.

Within this narrative paradigm, the child acts as the “narratee”, simultaneously the user of the physical product and the hero of the story, ensuring constant immersion and proactive play^[13]. By embedding abstract programming logic into concrete storylines, characters and scenes, narrative design facilitates the “de-abstraction” of logical problems, effectively mitigating the cognitive load for young learners during self-directed study.

2. Methods

Narrative design involves reconfiguring and redefining the relationships of purpose and object through a narrative lens. The construction of a narrative design model begins with the identification of a central theme, which is subsequently decomposed into several sub-elements. Following this, localized design models are developed around these sub-elements and eventually integrated organically to form the final, holistic narrative design model.

In the construction of the sub-elements model, the primary elements include as follows:

- (1) Characters, comprising both the user and the virtual personas within the story;
- (2) Actions, defined as user behaviors designed to fulfill product functions while aligning with the narrative theme;
- (3) Scenes, the environmental backgrounds that correspond to the theme and plot development;
- (4) Props, the settings of relevant product elements required for the narrative.

These four components are interconnected and interdependent (**Figure 1**). The narrative theme dictates the fundamental content of characters, actions, scenes, and props. Specifically, characters and actions are closely intertwined, while scenes and props complement each other. Furthermore, characters perceive and interpret scenes in distinct ways, while the scene provides the contextual backdrop for the character. Additionally, actions act as a catalyst, facilitating transitions within the scene space.

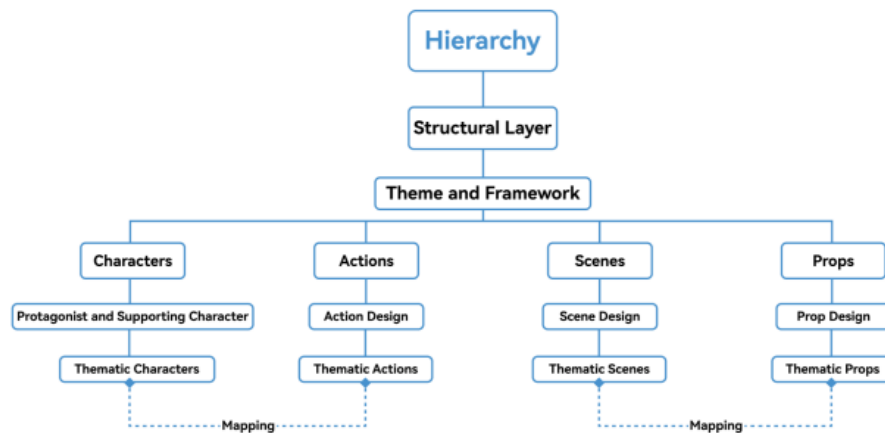


Figure 1. Hierarchical layering of narrative structure.

Integrating the core principles of narratology and experience design, narrative design can be categorized into five distinct dimensions: the Structural Layer, Character Layer, Action Layer, Scene Layer, and Prop Layer ^[14].

Based on this hierarchical framework, the design and development phase for the narrative experimental group was initiated. First, in-depth interviews were conducted with 10 parents of children aged 5–8 to preliminarily define the gaming preferences and core aesthetic characteristics of this target group. Subsequently, a questionnaire was developed based on the key elements extracted from these interviews, and statistical analysis was performed on 377 valid responses (**Figure 2**).

Children's Programming Game Content Design Questionnaire

Total Responses

377

Figure 2. Total Responses.

Ultimately, the narrative theme of “Interstellar Exploration” was established. In this scenario, children assume the role of a “Magician” guided by an “Elf Helper” to defeat enemies and protect the “Space Magic Academy” (Figure 3). Within this narrative-driven context, children are required to apply core programming concepts, such as sequences, directions, loops, and conditions, to plan their action paths, thereby completing programming training tasks.

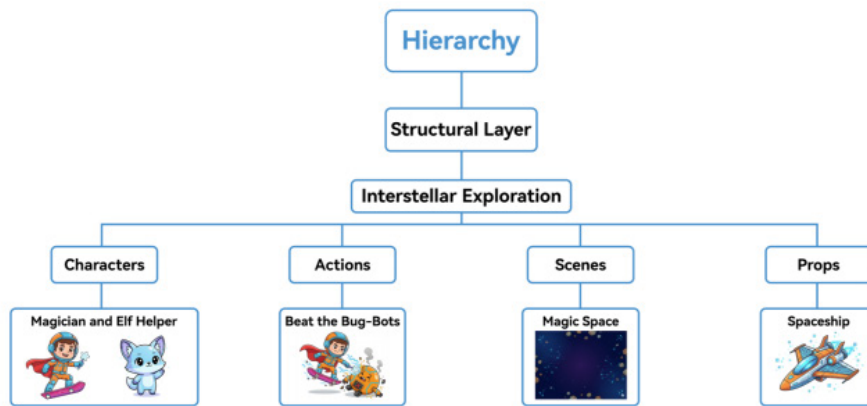


Figure 3. Interstellar exploration.

3. Experimental details

3.1. Participants

The participants in this study consisted of 30 senior-class children recruited from a local kindergarten, with a mean age of 6.6 years. The sample was composed of 60% boys and 40% girls.

3.2. Experimental design

This study adopts a single-factor controlled design, categorized into three experimental groups: the Computer-based Programming group (Group A), the Abstract Tangible Programming group (Group B), and the Narrative Tangible Programming group (Group C). The task logic of the programming training was strictly controlled to remain consistent across all groups. Interaction modality and content presentation format serves as the independent variables.

Specifically: Group A is based on a traditional GUI, interacting via keyboard and mouse. Groups B and C employ TUI with a higher level of embodiment, utilizing physical coding blocks integrated with programming commands (Figure 4). Regarding content presentation, both Groups A and B utilize abstract

problem prototypes based on arithmetic or symbols (**Figure 5**). In contrast, Group C integrates narrative theory by embedding these abstract problems into specific story backgrounds, characters, and scenes (**Figure 6**). This approach provides a concrete, gamified “wrapper” for the abstract tasks, thereby achieving “de-abstraction.”

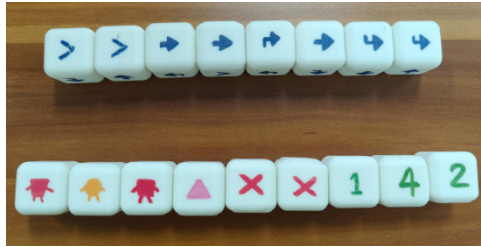


Figure 4. Physical coding blocks.

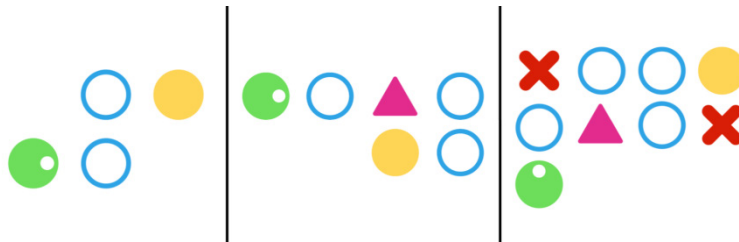


Figure 5. Abstract problem prototypes.

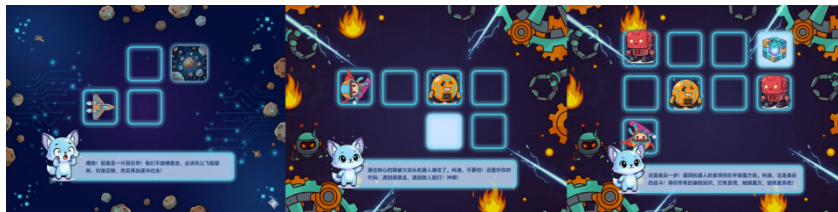


Figure 6. Group C integrates narrative theories.

3.3. Methodology

The 30 participants were randomly divided into three groups (n=10): Computer-based (Group A), Abstract Tangible (Group B), and Narrative Tangible (Group C). The study spanned one week, comprising 10 sessions of 45 minutes each (**Figure 7**). Data collection involved multimodal sources, including audio/video recordings and system logs.

Initially, participants filled out a questionnaire regarding their demographics and baseline programming knowledge. Following a one-time introductory session led by researchers, the children completed the remaining nine training sessions autonomously. This ensured that the primary intervention effects were derived from the children’s independent interaction with the respective systems.



Figure 7. Training sessions.

3.4. Performance metrics

Upon completion of the ten training sessions, participants were required to sequentially complete the Computational Thinking Test for Lower Primary (CTtLP, **Figure 8**) and a series of practical tasks using Tynker Junior levels (**Figure 9**)^[15]. Experimental data were collected through multimodal acquisition, including audio-visual recordings and system behavior logs. By cross-referencing these data sources, this study measured three core metrics to evaluate the children’s programming performance:

- (1) Task Completion Time: The duration from the initiation of a programming task to its successful execution;
- (2) Error Count: The total number of failed program executions recorded prior to successful task completion;
- (3) Debugging Capability: A measure of the participant’s ability to modify and rectify code following an execution error.



Figure 8. CTtLP.



Figure 9. Tynker junior levels.

To evaluate the debugging phase, we adopted the Carver and Klahr model and established a three-level ordinal variable ^[16]. Based on the analysis of audio-visual recordings and interaction logs, participants' reactions following program execution errors were categorized as follows:

- (a) Full Debugging: The participant successfully located and rectified the error;
- (b) Partial Debugging: The participant perceived an anomaly in the program's execution but was unable to independently locate or correct the error;
- (c) No Debugging: The participant failed to perceive the error and took no corrective action.

4. Results

4.1. CTtLP assessment results and analysis

In the Computational Thinking Test, the Narrative Tangible Programming group (Group C) achieved a mean score of 19.7 (out of a total CTtLP score of 33), significantly outperforming both the Abstract Tangible Programming group (Group B) and the Computer-based Programming group (Group A). The mean scores of the three groups exhibited a stepwise pattern (**Figure 10**). These results indicate that the tangible programming trainer designed based on narrative theory can effectively enhance 5–8-year-old children's understanding of core programming concepts, such as sequences, loops, and conditional statements. Compared to traditional abstract instruction or purely digital interaction, the narrative approach embeds abstract programming logic into vivid storylines, thereby reducing the cognitive load for young children.

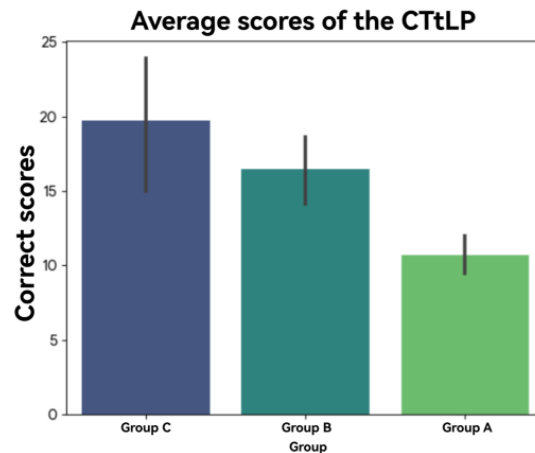


Figure 10. Average scores of the CTtLP.

4.2. Analysis of efficiency and accuracy in Tynker junior practical tasks

Data from the practical experiments further corroborate the superiority of the narrative approach. Regarding task completion time, the Narrative Group (Group C) demonstrated the highest efficiency (**Figure 11**), averaging only 28.4 seconds per level, while the Computer-based Group (Group A) required 63.8 seconds. In terms of error frequency, Group C averaged only 0.27 errors per task, which was significantly lower than Group B (0.50) and Group A (0.71).

These findings suggest that the narrative context provides clear situational cues, allowing children to anticipate the correct programming path through “narrative logic” before execution, thereby significantly reducing aimless trial-and-error behavior. The synergy between the tactile feedback of physical coding blocks and narrative guidance makes the programming process more intuitive and deterministic.

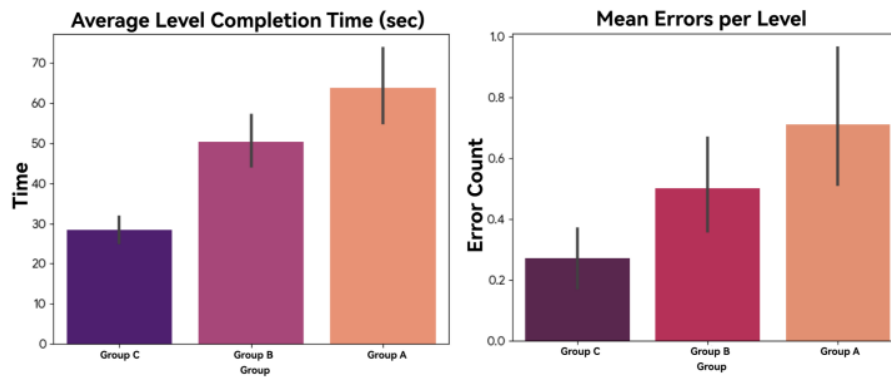


Figure 11. Efficiency and accuracy in Tynker junior.

4.3. Analysis of debugging ability and problem-solving behavior in Tynker junior practical tasks

The experimental results indicate that when program execution deviated from the expected outcome, the Narrative Group (Group C) demonstrated exceptional self-correction proficiency, with a debugging success rate of 94.4% (Figure 12), compared to only 70.2% in the Computer-based Group (Group A). The debugging score was calculated using a weighting scheme: 1 point for full debugging, 0.5 points for partial debugging, and 0 points for no debugging.

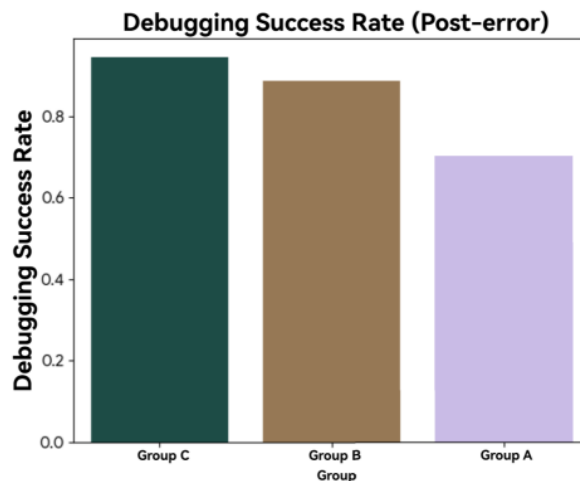


Figure 12. Debugging ability and problem-solving behavior in Tynker junior.

Qualitative analysis of the audio-visual transcriptions revealed that children in Group C were able to identify logical breakpoints by evaluating the fulfillment of story-driven goals (e.g., “the spaceship failing to reach the target planet”). By recontextualizing “programming errors” as “plot deviations,” narrative theory triggered a stronger correction motivation in children, helping them more effectively align their “intended goals” with “actual execution results.” In contrast, children in Groups A and B, lacking situational support, were more prone to frustration and engaged in mindless, mechanical debugging behaviors.

5. Conclusion

This study validated the effectiveness of narrative theory in the design of programming trainers for children through a comparative experiment. The results demonstrate that the narrative tangible programming mode, which embeds abstract programming logic into concrete story backgrounds (e.g., “Interstellar Exploration”), significantly enhances the programming proficiency of children aged 5–8. Data analysis revealed that the narrative group outperformed both the traditional GUI and abstract tangible interaction groups in terms of CTtLP scores, task efficiency, and error control, achieving a full debugging success rate as high as 94.4%.

By achieving the “de-abstraction” of programming problems, the contextualized representation within the narrative design effectively reduces the cognitive load of young children in autonomous learning environments. Furthermore, it guides them to leverage narrative logic to quickly locate and rectify program errors. In conclusion, the organic integration of narrativity and embodied interaction not only reinforces children’s deep understanding of core programming concepts but also provides a scientifically grounded development path for early programming educational products in home-based or unguided learning contexts.

Disclosure statement

The author declares no conflict of interest.

References

- [1] Review of Children Programming Education Based on Computational Thinking: The China Education Information Website ICTEDU, May 13, 2025.
- [2] Chen L, Xiao S, Chen Y, et al., 2024, ChatScratch: An AI-Augmented System Toward Autonomous Visual Programming Learning for Children Aged 6-12, Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, 1–19.
- [3] Xia J, 2024, A Study of the Impact of Mind Mapping-based Gamified Programming Instruction on High School Students’ Computational Thinking, thesis, Central China Normal University.
- [4] Jiang X, Ding N, Chen S, et al., 2022, The Design Model and Effect Verification of Children’s Programming Games. *Modern Educational Technology*, 32(12): 58–67.
- [5] Chen K, Chi H, 2022, Novice Young Board-Game Players’ Experience about Computational Thinking. *Interactive Learning Environments*, 30(8): 1375–1387.
- [6] Lee T, Mauriello M, Ahn J, et al., 2014, CTArcade: Computational Thinking with Games in School Age Children. *International Journal of Child-Computer Interaction*, 2(1): 26–33.
- [7] Sapounidis T, Demetriadis S, Stamelos I, 2015, Evaluating Children Performance with Graphical and Tangible Robot Programming Tools. *Personal and Ubiquitous Computing*, 19(1): 225–237.
- [8] Beilin H, 1971, Developmental Stages and Developmental Processes. In D. R. Green, M. F. Ford & G. B. Flamer (Eds.), *Measurement and Piaget*, 172–188.
- [9] Borghi A, Flumini A, Cimatti F, et al., 2011, Manipulating Objects and Telling Words: A Study on Concrete and Abstract Words Acquisition. *Frontiers in Psychology*, 2011(2): 15.
- [10] Kopcha T, Ocak C, Qian Y, 2021, Analyzing Children’s Computational Thinking through Embodied Interaction with Technology: A Multimodal Perspective. *Educational Technology Research and Development*, 69(4): 1987–2012.

- [11] Hu W, Huang R, Li Y, 2024, Young Children’s Experience in Unplugged Activities about Computational Thinking: From an Embodied Cognition Perspective. *Early Childhood Education Journal*, 52(4): 769–782.
- [12] Sung W, Ahn J, Black J, 2017, Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives through Embodiment in Mathematics Education. *Technology Knowledge and Learning*, 22(3): 443–463.
- [13] Zhang L, 2019, Application Research of Narrative Design Method in Children’s Toy Design, thesis, Changchun University of Science and Technology.
- [14] Zhang K, Gao Z, 2018, Study on the Design of Child Medical Products Based on the Narrative Design. *Art & Design*, 2018(1): 111–113.
- [15] Zhang S, Wong G, 2023, Development and Validation of a Computational Thinking Test for Lower Primary School Students. *Educational Technology Research and Development*, 71(4): 1595–1630.
- [16] Carver S, Klahr D, 1986, Assessing Children’s LOGO Debugging Skills with a Formal Model. *J Educ Comput Res*, 2(4): 487–525.

Publisher’s note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.