Liu, Z., & Moon, J. (2023). A Framework for Applying Sequential Data Analytics to Design Personalized Digital Game-Based Learning for Computing Education. *Educational Technology & Society*, *26*(2), 181-197. https://doi.org/10.30191/ETS.202304_26(2).0013

A Framework for Applying Sequential Data Analytics to Design Personalized Digital Game-Based Learning for Computing Education

Zhichun Liu[1*] and Jewoong Moon[2]

[1]Human Communication, Development, and Information Sciences, The University of Hong Kong, Hong Kong SAR, China // [2]Department of Department of Educational Leadership, Policy, & Technology Studies, University of Alabama, Tuscaloosa, AL, USA // liulukas91@gmail.com // jmoon19@ua.edu

[*]Corresponding author

ABSTRACT: In this study, we have proposed and implemented a sequential data analytics (SDA)-driven methodological framework to design adaptivity for digital game-based learning (DGBL). The goal of this framework is to facilitate children's personalized learning experiences for K–5 computing education. Although DGBL experiences can be beneficial, young children need personalized learning support because they are likely to experience cognitive challenges in computational thinking (CT) development and learning transfer. We implemented the educational game Penguin Go to test our methodological framework to detect children's optimal learning interaction patterns. Specifically, using SDA, we identified children's diverse gameplay patterns and inferred their learning states related to CT. To better understand children's gameplay performance and CT development in context, we used qualitative data as triangulation. We discuss adaptivity design based on the children's gameplay challenges indicated by their gameplay sequence patterns. This study shows that SDA can inform what in-game support is necessary to foster student learning and when to deliver such support in gameplay. The study findings suggest design guidelines regarding the integration of the proposed SDA framework.

Keywords: Digital game-based learning, Computational thinking, Sequential data analytics, Adaptivity, Personalized learning

1. Introduction

A major goal of recent computing education is to enhance children's computational thinking (CT). CT is a way of thinking that involves representing solutions via computational practices (Grover & Pea, 2013). Research has shown a concern that young children are likely to face cognitive challenges in developing CT due to its complexity (Lye & Koh, 2014). CT-related learning tasks are likely to overwhelm children and then undermine motivation and learning engagement (Zhao & Shute, 2019). It hence necessitates engaging and effective ways to support CT development for young children. Correspondingly, recent research has called for digital game-based learning (DGBL) as a means that promotes children's problem-solving and hands-on experiences — resulting in the development of concrete cognitive footings for abstract knowledge (Zhao & Shute, 2019). Previous works have demonstrated purposeful DGBL design that facilitates children's CT skills development through playful learning (e.g., Asbell-Clarke et al., 2020; Bers, 2020; Israel-Fishelson & Hershkovitz, 2020). Children as players, are guided to explore a variety of game missions where CT skills are necessary. Through playing, students are expected to initiate hypotheses and then come up with creative solutions derived from appropriate CT skills and concepts through multiple rounds of game trials. Despite the emergence of DGBL in computing education, skepticism exists on whether and how DGBL supports students with different knowledge levels and backgrounds.

Despite increasing DGBL research on computing education, there is a lack fo studies that discussed how DGBL supports children's personalized learning experiences (Hooshyar et al., 2021). Whereas DGBL enhances engagement and motivation, research reports that young children may undergo cognitive distractions and in-game frustration easily (Lye & Koh, 2014; Bers, 2020). To guide children's attention and mindful gameplay in DGBL, it is essential to help them keep engaged and focused in gameplay through personalized support. However, there is little systematic guide for designing the content of the support, the timing of support delivery, and the format of the support (Liu et al., 2020). Since DGBL with evidence-centered design (ECD) supposes observable game actions that represent children's learning states, it is essential to seek ways to grasp and analyze the nature of children's in-game behaviors aligned with CT. Whereas researchers used various data analytics to investigate learners' in-game behaviors in DGBL, existing data-driven approaches are limited in identifying children's needs under the gameplay nature (Moon & Liu, 2019). In this study, we propose, implement, and test sequential data analytics (SDA)-driven methodological framework to investigate young children's (K-5) gameplay patterns in the educational game Penguin Go. Furthermore, we discuss how this SDA-driven approach helps to conduct data-driven decisions for developing adaptive DGBL for young children.

ISSN 1436-4522 (online) and 1176-3647 (print). DOI 10.30191/ETS.

2. Literature review

2.1. Computing education and GBL

The field of computing education highlights CT, which is an analytical ability to decompose complicated problems, identify their patterns, and execute tailored solutions by computational means (Lye & Koh, 2014). Shute et al. (2017) identified the main competencies of CT as follows: (1) decomposition; (2) algorithm thinking; (3) abstraction; (4) debugging; (5) iteration; (6) generalization. However, due to children's inexperience entering computing education, they tend to undergo cognitive challenges that may result in low engagement and high frustration. Therefore, it is essential to provide children with motivating environments to boost their learning engagement.

A current CT movement has focused on enabling all learners to engage in computing education (Weintrop et al., 2016). There are two pivotal design rationales of DGBL in computing education. First, a major assumption of DGBL in computing education is implicit learning (Rowe et al., 2021) from everyday play behavior that does not explicitly appear. A game is a good platform that allows learners to demonstrate a particular pattern of behavior through play. Individuals' gaming actions and their consequences in game tasks are linked with the implicit CT learning. In this sense, many researchers sought to create a game mechanic that purposefully fosters learners' CT-related behaviors from play. Second, another lens of DGBL for computing education is constructionism. Weintrop et al. (2016) stated three design principles of an educational game: (1) personally meaningful artifact design, (2) exploration and discovery in play, and (3) engaging with powerful ideas to be advanced. They underscored that a game needs to present challenges that allow learners to initiate and test their conceptions from simple to complex. While building a pile of codes with iterations, learners can build and elaborate design rationales and internalize their programming logics through a series of game tasks. Game challenges and failure experiences help them to detect misconceptions, analyze consequences, and debug execution codes during multiple rounds of play. In this sense, DGBL has been useful to introduce computing education to young children. The key to incorporating DGBL into computing education is to make computer programming practices more engaging to young learners (Hsu et al., 2018). Previous research indicated that DGBL benefits learners' CT development by enhancing their engagement via gameplay (Israel-Fishelson & Hershkovitz, 2020; Turchi et al., 2019; Zhao & Shute, 2019). Moreover, during gameplay, learners can build and test their problem-solving solutions (Grover et al., 2017). Such problem-solving processes during gameplay seamlessly facilitate learners' iterations of hypothesis testing and solution executions, which in turn contribute to their development of CT skills. Asbell-Clarke et al. (2020), for example, created and implemented *Zoombinis*, a 2D learning game teaching CT to young children. Using data-driven automatic detectors of student gameplay (i.e., classification algorithms), they reported that children who demonstrated evidence of active problem solving in the game (e.g., change one variable while holding others constant) were more proficient in CT skills compared to those who were still learning the game mechanics (e.g., repeatedly using the same but ineffective solution in one puzzle).

2.2. Challenges in children's gameplay and adaptive game design

Although DGBL engages young learners in computing education, research has suggested that young children are likely to face cognitive challenges in CT-related problem-solving in gameplay. Young children tend to demonstrate inefficient solution implementations and unsystematic debugging (e.g., trial-and-error) caused by random, non-strategic, or sometimes unproductively wheel-spinning. Such inefficient solutions often involve step-by-step execution, testing with random combinations, or debugging without meaningful subgoals (Fessakis et al., 2013; Liu et al., 2017). Although iterative trial-and-error may help to solve game problems, such patterns do not always lead to meaningful learning (Owen et al., 2019). Multiple trials and errors without further improvement rather give rise to frustration and disengagement. This behavior pattern is largely attributed to children's limited cognitive and meta-cognitive resources. In a highly interactive environment such as DGBL, children are exposed to high cognitive load (Azevedo & Aleven, 2013; Morrison et al., 2015), which poses challenges for higher-order CT skills—such as loop and conditional statement development (Ching et al., 2018).

In addition, research has reported learning transfer as a significant issue after the gameplay: Children seem to enjoy and excel within the game, but they did not perform well on the knowledge test outside of the game (Arena & Schwartz, 2014; Mason et al., 2011). When children are asked to perform the learned skills in a different context (often referred as far transfer), they need to first understand the similarity between the original learning context and then apply the learned cognitive processes into a new context (Taatgen, 2013). Both steps require a large amount of cognitive and meta-cognitive resources; hence, it is less likely that they can perform well on transfer tasks after simply playing games (Liu & Jeong, 2022). In a highly interactive environment such as games, children should pay mindful attention requiring cognitive and meta-cognitive resources under diverse gaming trajectories (Ke & Abras, 2013). Therefore, it is essential for DGBL researchers to identify the cognitive or meta-cognitive needs and design personalized support to help children to acquire transferrable skills through games.

Children's cognitive challenges augment the importance of personalized learning experiences. Personalized learning is a learning design that adjusts either learning modules and instructional strategies tailored to children's learning states or interests (Walkington, 2013). To perform personalized learning, identifying children's learning trajectories and dynamic problem-solving processes in advance is crucial (Lin et al., 2013). In DGBL, to systematically support children's personalized learning, emerging research has incorporated adaptivity in games (Vanbecelaere et al., 2020). Here, adaptivity refers to the systematic and dynamic delivery of game-based instructional activities through ongoing and in-situ learner analyses (Liu et al., 2020). Furthermore, to determine either level or format of adaptive learning support best suited to individuals, DGBL systems need to collect and analyze learner profiles and present appropriate support to them. A recent study by Hooshyar et al. (2021) showed how to provide personalized CT learning experiences via gameplay. They introduced *AutoThinking*, which is a 2D agent-based computer programming game. This game allowed players to use a collection of icons to control a game character's movement in a maze environment. They adopted Bayesian networks algorithm to decide the adaptivity level of students' gameplay. A game system automatically assessed players' CT skills and presented different types of game character movement patterns (i.e., random, provocative, aggressive, and lenient). Despite a promising view of adaptivity implementation in DGBL for computing education, limited research has demonstrated how to orchestrate systematic and data-driven decision-making with adaptive DGBL design. Specifically, few studies discussed how to implement data analytics to drive the design of adaptivity in DGBL.

2.3. Evidence-centered design and data analytics

For learner analysis and corresponding adaptive support in DGBL, research has suggested implementing stealth assessment. Stealth assessment is designed to collect students' competency states in an unobtrusive way (Moore & Shute, 2017). Evidence-centered design (ECD) provides rationales for the implementation of stealth assessments (Shute & Kim, 2014). ECD is a framework with which to design learning assessments to measure students' knowledge, skills, and attitudes. To detect student learning states through stealth assessment, research used various data analytics that model learners' competency (e.g., Akram et al., 2018; Min et al., 2019). However, existing competency models typically focus on evaluating the entire learning history, but they are limited in collecting and analyzing in-situ data indicating individuals' learning trajectories in real time. In research of DGBL,previous predictive modeling approaches tend to compute cumulative performance levels instead of their chronological development of gameplay learning experiences. For instance, previous featured DGBL studies with ECD frameworks (Shute & Moore, 2017; Ke & Shute, 2015; Levy, 2019) used Bayesian networks to compute the conditional probability to operate the adaptivity during gameplay. To determine game adaptivity levels, they discretized a granular level of game log data by accumulations. However, this approach has limited success in understanding learners' behavior from a chronological perspective and projecting individuals' gameplay sequences that function as a proxy of their way of thinking during gameplay.

To better capture student learning trajectories in gameplay, emerging research has introduced SDA in DGBL (Moon & Liu, 2019; Tlili et al., 2021). Given a pronounced concern of existing prediction models above, SDA is advantageous to better capturing and delineating learners' temporal and salient sequences of gameplay behaviors representing individuals' "learning paths." Because students' gameplay patterns are likely to expose their knowledge paths in learning tasks, SDA enables researchers to better understand whether and how students face learning challenges in gameplay. Gameplay patterns indicate children's understanding of given game rules and clues. If a child goes to wrong paths and actions related to a game task, it indicates students' game challenges. Under this analytics assumption, DGBL research increasingly tends to use SDA to measure students' patterns of self-regulated learning (Kinnebrew et al., 2015) and scientific reasoning (Taub et al., 2018). Given that SDA is particularly useful to visualize individuals' way of thinking amid a collection of gameplay event data, it is useful to be implemented in DGBL for computing education. Since analytics in computing education requires researchers to identify students' stepwise compilation of blocks to execute their codes with success, SDA can be useful to gather relevant evidence effectively.

2.4. Research gap

Despite aforementioned challenges, limited research has implemented data analytics to better capture, model, and understand children's learning states during related to CT development. Existing data analytic approaches in DGBL rarely analyzed how students learn and what challenges occur aligned with game contexts. Corresponding to such problems, this study proposes and implement an SDA-driven framework to provide evidence of designing personalized learning experiences of CT in DGBL. Aligned with this study's goal, we propose research questions as follows.

(1) What are the emerging gameplay patterns among children who played Penguin Go?

(2) What are the differences in gameplay patterns between children in different game conditions (i.e., with or without additional cognitive support)?

(3) What are the design implications of the highlighted gameplay patterns in terms of promoting personalized learning experiences and the development of transferrable CT skills?

To answer the research questions, we have implemented three steps: (1) implementing an educational game (Penguin Go) for CT development; (2) building SDA-driven assessment framework DGBL for adaptivity design; and (3) implementing a case study to explore the relationships among children's gameplay patterns, CT skill development, and learning transfer as the evidence for adaptivity design.

3. Method

3.1. Penguin Go and computational thinking skills

Penguin Go is an educational game teaching block-based programming language for both elementary and middle school students' CT development developed by the research team (Liu & Jeong, 2022; Zhao & Shute, 2019). This game provides various game tasks to children in the context of the breeding behaviors of emperor penguins. The game's goal is to move the penguin to the destination (i.e., the footprint) using different combinations of code blocks (Figure 1). The game has 18 levels in total. Players need to plan the path of the penguin strategically based on the level terrain. For example, the penguin can waddle on snow (i.e., the light blue blocks) but will slip on the ice (i.e., the deep blue blocks) and has to travel with a toboggan. Table 1 demonstrates the relationships between CT competencies and the concepts covered in the game.

Figure 1. Level "*Which Way?*" in *Penguin Go* and a possible solution

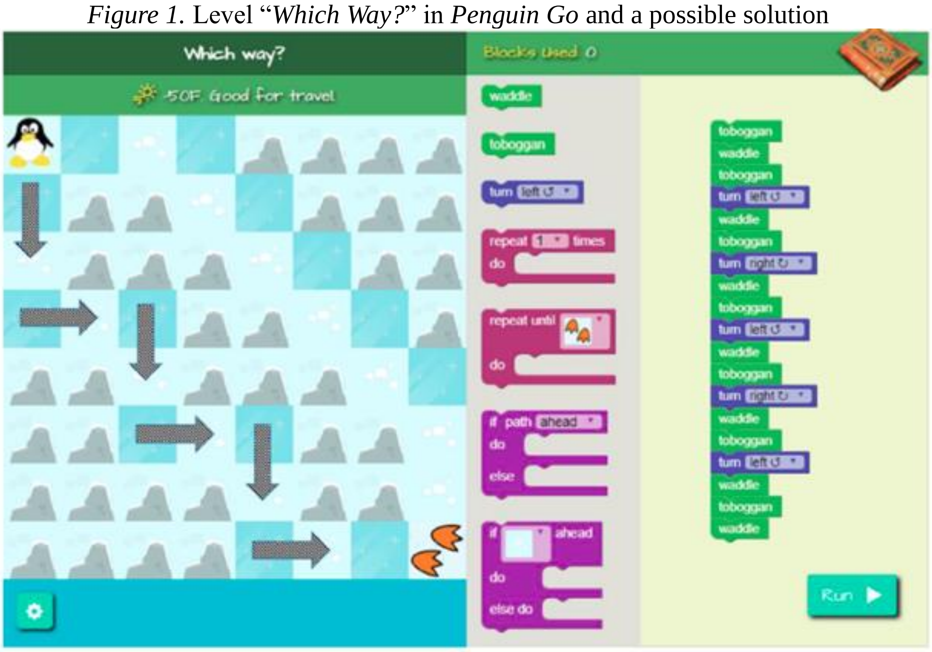


Table 1. The relationships between CT competencies and CT concepts covered

CT competencies	Sequence structure	Conditional structure	Loop structure	Description
Decomposition	X	X	X	Identify the goal of each level, the potential pathways, constraints, and patterns in a solution.
Algorithm thinking	X	X		Translate the solution into a sequence of blocks that guide the penguin through the maze.
Abstraction		X	X	Use as few blocks as possible in the solution. Successful implementation of the conditional structure and loop structure can increase the abstraction level of the solution.
Debugging and iteration	X	X	X	Identify the problems and improve the solutions iteratively if the coding blocks do not work as desired

3.2. SDA-driven assessment framework of DGBL for adaptivity design

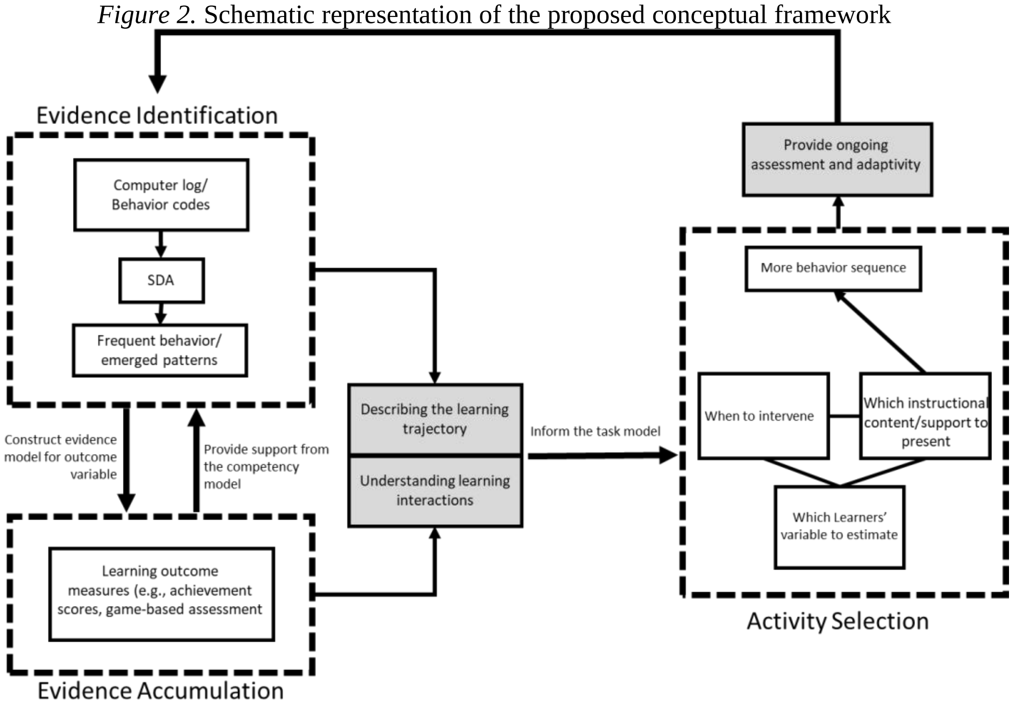
Previous research using *Penguin Go* suggested that children tend to undergo difficulty developing abstract thinking (Zhao & Shute, 2019). Abstract thinking is one of the hard-to-achieve but a core CT competency for K–5 children (Lye & Koh, 2014; Wing, 2008; Zhang & Nouri, 2019). In this study, we aim to design a personalized support mechanism that promotes children's transferrable CT across various contexts. Empirical evidence has also shown, however, that mandatory instructional activities might reduce autonomy, which hinders motivation and engagement (Clark et al., 2011; Zhao & Shute, 2019). Therefore, personalized learning supports should be delivered to the children during their in-game problem solving. With personalized learning supports, children are more likely to engage in gameplay instead of receiving instructions passively.

We propose an SDA-driven framework to assess young children's gameplay that evidence of designing adaptivity in DGBL. Here, we aim at identifying meaningful gameplay patterns related to children's either CT development and game challenges. We then focus on exploring how to inform the design of adaptivity based on gameplay results extracted from SDA, putting forth the methodological framework to guide the adaptivity design integrated with SDA.

Figure 2 presents our methodological framework. This framework consists of three major phases based on both the ECD approach (Mislevy et al., 2003) and the four-process adaptive cycle (Shute & Zapata-Rivera, 2012): (1) evidence identification; (2) evidence accumulation; and (3) activity selection. In comparison to the existing adaptive cycle, the proposed framework specifies what kinds of data the system capture in DGBL (e.g., frequent play patterns). Whereas the architecture of the original adaptive cycle poses a general adaptivity design, the proposed model better contextualizes data collection and analyses aligned with SDA. For example, in evidence identification, this framework particularly collects data that orderly arranges a chain of multiple behavior states. Such a collection of behavior states represents students' gameplay patterns that imply decision-making processes. If a sequence of specific game actions is frequent, it is defined as an emerging pattern of gameplay. Whereas existing frameworks tend to emphasize the macro level of adaptivity design and implementation, the proposed framework particularly aims at capturing the in-situ data containing children's gameplay patterns in the adaptive system cycle.

The framework depicts how best to guide children's personalized learning and design adaptivity in DGBL. *Evidence identification* refers to the phase of collecting children's behavioral data through computer logs and/or qualitatively annotated behavior codes and using SDA techniques to identify frequently occurring behaviors or emerging sequence patterns. The identified evidence describes children's gaming sequences and serves as the empirical evidence for the later phases. The purpose of *evidence accumulation* is to interpret existing input data (*evidence identification)* via external measures because identifying the noticeable pattern may not necessarily be self-explanatory. In this phase, we can understand the identified emerging patterns and behaviors in context. For example, we can determine whether a substantial behavioral difference between high performers and low performers is present. As a result, evidence can be accumulated to infer children's competency and identify the potential challenges children are facing, which, in turn, inform the design of the task models. The *activity selection* phase adjusts the instructional activity based on the evidence identified and accumulated (i.e., adaptivity). The goal of this adjustment is to match the appropriate support to children and elicit further behaviors that feedback to the *evidence identification* phase. Researchers need to select which learner variables to estimate (e.g., cognitive competency, problem-solving states, affective states), when to intervene, and which instructional content or support to present.

Figure 2. Schematic representation of the proposed conceptual framework



3.3. Study procedure

We conducted a case study with an experimental design at two large K–8 schools with a diverse student population in the southeast of the United States. The population was selected because (a) the game was designed for elementary school students, and (b) computational thinking and programming learning opportunities have often been reserved for more advantageous groups (Lachney et al., 2021). The goal of this case study is to understand children's gameplay data and discuss what learning supports are appropriate based on the collected data under the proposed methodological framework. In total, 85 students enrolled in the study, and six students dropped out because of various reasons, including lack of interest or not finishing the posttest. The sample consisted of 79 children (43 self-reported to be female and 27 self-reported to be male; ages ranged from 9 to 11 years old with a median of 10). About half of the sample was from underrepresented ethnic groups (i.e., 22 Black or African American students, 7 Hispanic students, and 2 American Indian or Alaska Native students). We randomly assigned all participants to one of two conditions prior to the experiment: control or treatment. The control group (n = 39) only accessed the *Game Mechanism Support* (GMS) during the gameplay voluntarily. Besides the GMS, the treatment group (n = 40) voluntarily interacted with additional cognitive support in the form of information prompts and partial worked examples (i.e., *Concept-Specific Support* and *Level-Specific Tips*, Table 2) in addition to experiencing *GMS*. We used this treatment design to validate the efficacy of cognitive supports on children's CT development. Here, aligned with the scope of this study, we focus on reflecting the design implications from the experiment not investigating the treatment effect. The study participants joined five 50-minute class sessions and yielded a total of 135 minutes of gameplay. We assessed children's CT development at the pretest, near transfer, and far transfer levels.

Table 2. Supports in Penguin Go

Support	Description
Game Mechanism Support	Static explanations and examples of the programming concepts
Concept-Specific Support	Interactive prompt that introduces the new block
Level-Specific Tips	A partial worked example that (1) encourages the use of a minimum number of blocks, (2) presents the target block, and (3) presents other blocks that nest inside the loop

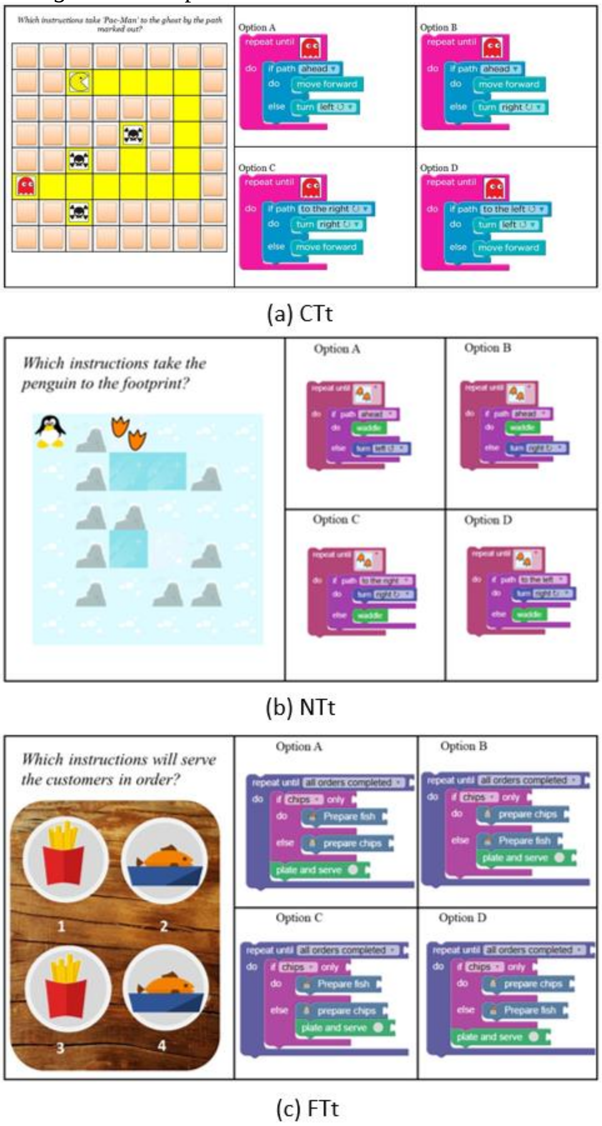
Note. The game mechanism support can be accessed by both groups voluntarily. Only treatment group could access Concept-Specific Support and level-specific Tips.

3.4. Instruments

3.4.1. CT tests

We developed and implemented three tests to assess children's CT skills. All tests were designed based on the Computational Thinking Test (CTt; Román-González et al., 2017). The pretest was a simplified version of the CTt (17 items). Based on the pretest, we also developed the near transfer test (NTt) that presents the problems in the context of *Penguin Go* while sharing the identical solutions of CTt. Finally, the far transfer test (FTt) mirrored the pretest in terms of the solutions but presented the problems beyond navigating through mazes. All three tests were isomorphic to each other regarding the CT competencies and concepts involved (Figure 3).

Figure 3. Sample items of matched CT instruments



3.4.2. Gameplay data

We collected gameplay logs to identify children's game interactions. All game interactions are logged. Gameplay logs included the data of (a) starting/ending the level; (b) creating/deleting a new block in the solution; (c) changing an existing block; (d) running coding blocks; (e) resetting the position of the penguin; and (f) accessing support. The log data also contained the game ID, action, level, code, and timestamp (an example is presented in Table 3). For data analysis, we removed the time gap between study sessions and aggregated each individual child's gameplay as one unit of analysis. Table 4 shows the descriptive data of each behavior. However, the raw descriptive data only did not indicate how children solve problems in *Penguin Go*. Therefore, we implemented SDA for further analyses.

Table 3. Sample gameplay data

User ID	Verb	Object	Level	Timestamp
tsms009	start	level	0-5	18:26:15
tsms009	create	blocks	0-5	18:26:40
tsms009	create	blocks	0-5	18:27:05
tsms009	run	blocks	0-5	18:27:07
tsms009	change	blocks	0-5	18:27:21
tsms009	reset	blocks	0-5	18:27:37
……				
tsms009	access	support	0-5	18:30:02
……				
tsms009	run	blocks	0-5	18:31:29
tsms009	end	level	0-5	18:31:37

Table 4. Descriptive game interaction data

	Treatment		Control		Total	
	Mean	*SD*	Mean	*SD*	Mean	*SD*
Start level	19.40	5.986	14.87	4.354	17.16	5.687
End level	15.15	4.481	14.64	4.094	14.90	4.275
Create blocks	182.75	72.875	184.87	60.270	183.80	66.530
Change blocks	27.83	14.595	28.77	14.377	28.29	14.403
Delete blocks	32.25	16.295	34.23	11.966	33.23	14.266
Reset blocks	50.98	23.818	56.95	22.797	53.92	23.364
Run blocks	66.10	25.129	71.49	23.124	68.76	24.158
Access support	17.53	15.563	5.03	5.747	11.35	13.295
Total	411.98	140.544	410.85	108.160	411.42	124.804

3.5. Sequential data analytics

As a technique of SDA, we conducted sequential pattern mining (SPM) with a *cSPADE* algorithm to understand children's gameplay patterns (Zaki, 2001). The purpose of sequential pattern mining here was to identify emerging gameplay patterns that most likely to occur. Each sequence refers to the gameplay data of one level completed by one student, and the chain of multiple sequences pattern consisted of several gameplay events that orderly occurred. We preset the sequence gap to be 2 (i.e., *max_gap* = 2, where the next event in the identified pattern should appear within two steps of the prior event but are not necessarily consecutive). The minimum support of a sequence was preset to be .5 (i.e., *min_sup* = .5; only displaying the frequent sequence patterns that occur over 50% of the time across all children's gameplay). If the support of a particular sequence was detected to be .6, it indicates that 60% of children's gameplay demonstrates such sequence.

3.6. Qualitative observations and field notes

In addition to the quantitative data collection (i.e., group comparison of CT tests and sequential pattern mining), we also conducted qualitative data analysis through behavior observations from facilitators. Four facilitators managed the gameplay sessions and then took notes on children's in-game problem solving and gameplay challenges. Specifically, the observation and field notes focused on (a) the gameplay experiences, (b) problem solving approaches, (c) attitudinal reactions, and (d) study logistics. At the end of each session, the facilitators debriefed their observations. We compiled and analyzed all the qualitative data through multiple rounds of open coding. The analysis focused on identifying children's particular gameplay behaviors and notable problem-solving patterns during the experiment. The qualitative data is used as secondary data to provide triangulation and contextual information to the quantitative findings.

4. Results

In the following sections, we present our study findings in accordance with our research questions and the proposed conceptual framework (i.e., evidence identification, evidence accumulation, and activity selection).

4.1. RQ1: Sequence pattern emerged (evidence identification)

We first modeled all the children's in-game behaviors across all levels with sequential pattern mining. The probability of behavioral transition is shown in Figure 4. We identified 28 sequence patterns containing five unique behaviors based on the threshold (i.e., *min_sup* = .5 and *max_gap* = 2). Among the identified patterns, the most frequent behavior was *Create Blocks*, which appeared in 26 sequence patterns. *Run Blocks* appeared in 16 patterns, and *Reset Blocks* was present in 13 patterns. The least frequent behavior patterns were *Delete Block*s and *Change Blocks*, which appeared in only five of the patterns and one of the patterns, respectively. *Access Support* did not appear in any of the patterns. This result suggests that children relied more on solution implementation (i.e., *Create Blocks* and *Run Blocks*) rather than refining solutions (i.e., *Reset*, *Delete*, and *Change Blocks*). The average support value for the identified sequential patterns was .67. We examined the top 10 gameplay sequences with the highest support values to identify emerging gameplay patterns among all children (Table 5). The support values ranged from .65 to .97.

Table 5. Most frequent sequence patterns identified across conditions

Rank	Sequence	Support	Category
1	{create blocks} →{run blocks}	0.971	SI
2	{create blocks} →{create blocks}	0.943	CI
3	{create blocks} →{create blocks} →{run blocks}	0.909	SI
4	{create blocks}→{create blocks}→{create blocks}	0.841	CI
5	{create blocks}→{create blocks}→ {create blocks}→{run blocks}	0.800	SI
6	{create blocks}→{create blocks}→{create blocks}→{create blocks}	0.756	CI
7	{create blocks}→{create blocks}→{create blocks}→{create blocks}→{run blocks}	0.703	SI
8	{reset blocks}→{run blocks}	0.690	SE
9	{create blocks}→{create blocks}→{create blocks}→{create blocks}→{create blocks}	0.661	CI
10	{create blocks}→{reset blocks}	0.656	SE

Note. See Table 6 for details about solution implementation with execution (SI), consecutive solution implementation (CI), and solution evaluation (SE).

We classified gameplay patterns into three categories: (a) solution implementation with execution (SI, Pattern 1, 3, 5, 7), (b) consecutive solution implementation (CI, Pattern 2, 4, 6, 9), and (c) solution evaluation (SE, Pattern 8 and 10). SI patterns start with block creation and end with running the blocks, and CI patterns only consist of consecutive block creation. SE patterns involve *Reset Blocks* compared to SI and CI. *Reset Blocks* refers to resetting penguin position in the game, which does not appear until the blocks begin to run. *Reset Blocks* happens only when someone would like to interrupt the execution of the algorithm. Table 6 summarizes the characteristics and implications of each sequence pattern.

Table 6. Categories of sequence patterns

Category	Pattern description	Implications
Solution implementation with execution (SI)	Start with a series of *Create Blocks* and end with *Run Blocks*.	Implements and executes a solution with a clear algorithm in mind. The frequent occurrence of the SI behavior indicates the trial-and-error problem-solving heuristic, which is often inefficient.
Consecutive solution implementation (CI)	Only contains consecutive *Create Blocks* with no *Run Blocks*.	Does not have a clear plan of the algorithm, which could indicate unsystematic exploration or sometimes random block creation.
Solution evaluation (SE)	Contains *Reset Blocks* in combination with *Run Blocks* or *Reset Blocks*.	Interrupts the solution execution. Involves prediction of where the penguin is moving and the evaluation of the solution. Often associate with debugging.

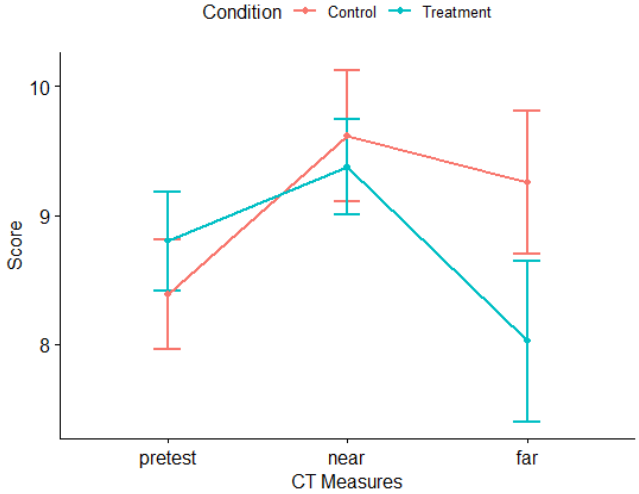
Based on the descriptive results of gameplay sequences and each game behavior, we also infer children's problem-solving patterns. First, we suggest that the children tended to undergo inefficient problem-solving heuristics—such as (a) the frequent occurrence of CI patterns because many levels (e.g., loop levels) can be solved with just a few blocks and frequent block creation could indicate hesitation and trial-and-error, (b) multiple trials on one level (e.g., 4.83 runs per level completion), and (c) infrequent change of blocks (e.g., 1.65 changes per level start). Second, we found less frequent prediction- and evaluation-related gameplay patterns, indicating children's lack of systematic problem solving. Third, the absence of accessing learning support in the gameplay patterns suggests that children used few learning supports and appeared less mindful in problem solving. Such findings highlight that children should have experienced more personalized supports, guiding their in-game problem solving. Overall, these findings help a DGBL system to tentatively identify the noticeable gameplay patterns that can be used for evidence accumulation.

4.2 RQ2: Understanding interaction pattern in situ (evidence accumulation)

4.2.1. Performance data

We first examined the performance difference between the two experimental conditions (Figure 4). The regression analysis results suggested that when controlling the pretest, both groups performed equally well on near transfer ($t(76) = -.62$, $p = .54$) and the control group outperformed the treatment group at the far transfer level ($t(76) = -2.69$, $p = .009$).

Figure 4. CT measures per condition



4.2.2. Behavioral data

We then investigated the difference between the two conditions regarding the sequence patterns. The same threshold (*min_sup* = .5, *max_gap* = 2) was used to keep consistent with the previous analysis. Table 7 shows a summary of the top 10 frequent gameplay patterns we identified.

Based on the classification, both conditions demonstrated similar patterns in terms of the most frequent behaviors. More than 70% of children's gameplay demonstrated similar SI and CI behavioral patterns in the treatment and control group based on the support value. In addition, SE patterns were relatively less frequent, and the support access was minimal. However, the children in the control condition demonstrated more frequent SE patterns than those in the treatment condition.

Children's sequence patterns demonstrate a high-level summary of their gameplay. As a result, we can infer that the similarity in general behavior patterns between the two groups could potentially explain why children in both two conditions performed equally well at the near transfer. However, the difference in engagement of SE could possibly contribute to the performance difference at the far transfer level.

Table 7. Most frequent sequence patterns identified by condition

	Treatment group			Control group		
#	Sequence	Support	Category	Sequence	Support	Category
1	{create blocks}→{run blocks}	0.961	SI	{create blocks}→{run blocks}	0.981	SI
2	{create blocks}→{create blocks}	0.923	CI	{create blocks}→{create blocks}	0.964	CI
3	{create blocks}→{create blocks}→{run blocks}	0.881	SI	{create blocks}→{create blocks}→{run blocks}	0.940	SI
4	{create blocks}→{create	0.822	CI	{create blocks}→{create	0.862	CI

	blocks}→{create blocks}			blocks}→{create blocks}		
5	{create blocks}→{create blocks}→{create blocks}→{run blocks}	0.774	SI	{create blocks}→{create blocks}→{create blocks}→{run blocks}	0.827	SI
6	{create blocks}→{create blocks}→{create blocks}→{create blocks}	0.729	CI	{create blocks}→{create blocks}→{create blocks}→{create blocks}	0.786	CI
7	{create blocks}→{create blocks}→{create blocks}→{create blocks}→{run blocks}	0.666	SI	{create blocks}→{create blocks}→{create blocks}→{create blocks}→{run blocks}	0.743	SI
8	{reset blocks}→{run blocks}	0.665	SE	{reset blocks}→{run blocks}	0.717	SE
9	{create blocks}→{create blocks}→{create blocks}→{create blocks}→{create blocks}	0.639	CI	{create blocks}→{reset blocks}	0.689	SE
10	{create blocks}→{reset blocks}	0.626	SE	{run blocks}→{reset blocks}	0.668	SE

Note. See Table 6 for details about solution implementation with execution (SI), consecutive solution implementation (CI), and solution evaluation (SE).

4.2.3. Qualitative data: Data triangulation

To further understand the difference in children's performance and gameplay patterns, we then triangulated SPM results with behavior observations from facilitators' field notes and debriefing results. The qualitative data included primarily four categories: (a) the gameplay experiences (e.g., number of levels played, challenges students had, notable game interactions such as accessing learning resources), (b) problem solving approaches (e.g., trial-and-error, pause-and-think, disengagement), (c) attitudinal reactions (e.g., excitement, confusion, boredom), and (d) study logistics (e.g., technological issues). In this study, we aim to use qualitative as the secondary data to ensure the consistency and trustworthiness of the quantitative findings. Specifically, we identified three notable themes through the qualitative data regarding children's gameplay (i.e., RQ1 and RQ2). First, the field notes in behavior observations reported that children relied on inefficient problem-solving approaches such as trial-and-error. Facilitators observed that some children were frequently moving back and forth between creating blocks and running blocks and built a solution incrementally. One facilitator noted that some children did not spend time reading the pre-level prompts when a new block was introduced.

Second, children were less engaged in problem decomposition and debugging in the gameplay. The children appeared impatient because they tended to construct a solution and immediately delete blocks back after the penguin failed to move to the destination. Given that children's solutions comprise simple sequence structures, this result suggests that the children did not demonstrate a high level of abstraction during the in-game problem solving. They tended to choose simple solutions, which involve fewer cognitive resources.

Finally, the behavior observation also indicated that children did not access the learning support very often. Some children in the treatment group even used the in-level tips as cognitive shortcuts to plan simple solutions. The tips ended up being a "cheat sheet" to them and did not guide them to plan or evaluate their solutions.

These findings further explain the patterns in the context of CT development and transfer. The results address the potential challenges to children's gameplay and learning. They indicate which helps to inform the activity selection phase in designing adaptivity for DGBL. The triangulation from the qualitative data provides further support to the previous SDA findings, which are the basis of the design adaptivity.

4.3. RQ3: Design implications of personalization (activity selection)

4.3.1. Using SDA to understand in-game problem solving

One of the challenges of the current version of *Penguin Go* is that children demonstrated inefficient problem-solving heuristics and did not interact with the cognitive supports under the voluntary condition. Based on the game challenges, we found evidence of designing adaptivity from a competency-driven approach, emphasizing children's problem solving. With SDA implemented, the game can (a) infer children's general problem-solving competency (i.e., game performance history and pattern recognized); (b) monitor the noticeable sequence patterns; and (c) infer the stage of in-game problem solving.

4.3.2. Adaptive game challenges

Adaptive game challenges can guide children to focus on target skill acquisition and abstraction on knowledge. Based on the previous analyses, we concluded that children tended to demonstrate mostly SI and CI rather than CI, which can be inefficient. If such gameplay patterns emerge continuously, this continuous occurrence of the patterns indicated that children do not mindfully engage in problem solving particularly related to abstract thinking. Therefore, imposing constraints on the number of blocks (e.g., Zhao & Shute, 2019) can guide children to mindfully plan their solutions because of the limited resources. Moreover, based on the student gameplay proficiency (e.g., level completion time), constraints can be adjusted accordingly. In the context of the current study, one indicator that we can use is the support value of CI patterns being consistently higher than 90% across multiple levels, given that the population demonstrate such pattern more than 90% of the time on average. However, this baseline might vary across different populations with different proficiency levels.

4.3.3. Adaptive cognitive supports

While constraints provide personalized challenges, adaptive cognitive supports provide personalized support. For example, when CI patterns frequently occurred within one level (an indication of unsystematic problem-solving), the game delivered cognitive supports that helped children understand the content knowledge. When repetitive SI emerged, cognitive supports—such as worked examples—were delivered to help children refine solutions. SDA can help to identify these gameplay patterns by setting the minimum support value: if the algorithm detects a frequent pattern (e.g., *min_sup* > .5), the game will trigger the relevant support.

4.3.4. Adaptive meta-cognitive supports

Children's unsystematic problem solving was related not only to inefficient uses of cognitive resources but also to the limited access to meta-cognitive resources (Azevedo & Aleven, 2013). Such unsystematic problem-solving pattern is supported by children's infrequent SE pattern, and even the control group outperformed the treatment group at the far transfer level. SDA is viable to identify what type of meta-cognitive support should be presented and when to intervene within the game level. For instance, once the cumulative gameplay sequences of a child indicate the infrequent SE pattern during gameplay, a game needs to deliver meta-cognitive supports (e.g., analysis prompts, evaluation guides, or reflection activities) upon individuals' diverse paths. Furthermore, children's gameplay action transitions (e.g., consecutive block creation, resetting, or deleting blocks) indicate various problem solving stages (e.g., wheel-spinning or solution refinement). Based on the identified gameplay pattern results, we can then match the appropriate meta-cognitive supports to the individuals' play to support systematic gameplay related to CT development.

5. Discussion

This study implemented SDA into DGBL—performing an assessment to inform evidence of adaptivity design to promote young children's CT development. Based on our analysis findings, we discuss how each phase of the proposed framework helped to design children's personalized DGBL learning experiences by adaptivity design.

5.1. Using SDA to facilitate the evidence identification

SDA benefits researchers in collecting and identify the evidence of children's gameplay behaviors for design-based research in a game environment. The results of gameplay patterns in this study demonstrated young children's challenges overall when the supports were not tailored to individuals' diverse learning trajectories. Specifically, the children experienced difficulty in building a correct solution throughout in-game tasks without personalized support. Such patterns also represent students' challenges, including inefficient gaming performance and low understandings of CT during gameplay. These results are aligned with previous research that young children had difficulty mastering the concept of loops and conditional statements to build a complete solution (Ching et al., 2018). Children's such challenges augment the significance of guidance in experiential and interactive learning environments—considering young learners' cognitive capability (Ke et al., 2019; Kirschner et al., 2006; Mayer, 2004). In other words, the data in the *evidence identification* phase shows preliminary evidence of when and how to provide adaptive supports to guide children's problem-solving and promote solution design based on their current learning states.

When it comes to designing adaptivity for DGBL, SDA distilled students' gameplay data (a chain of sequences) and then examined children's frequent play paths as quantitative and contextual evidence. Given that an adaptive game system collects, assembles the evidence, and makes empirically data-driven decisions, at this stage, SDA illustrates what kinds of gameplay pattern data emerged and estimate children's states of game successes and challenges by estimating the frequency of certain gameplay pattern data. The information is essential to build different predictive supervised or semi-supervised algorithms for the purpose of learner modeling in designing adaptive DGBL systems (e.g., Almond et al., 2020; Basu et al., 2017; Rowe et al., 2021).

5.2. Validating and triangulating the evidence accumulation

The *evidence accumulation* phase in this study helped researchers to ensure the validity of data collected from SDA. For example, the group comparison of the interaction pattern and how the learning transfer performance was related to patterns highlighting the importance of children's SE patterns and the inefficiency of CI patterns in DGBL. Using triangulation, we further corroborated these findings. The children in the treatment group, with additional cognitive supports, tended to misuse the supports. The supports helped students at the near transfer level but not necessarily at the far transfer level. In comparison, the children in the control group, without cognitive supports, were more likely to engage in SE. Such pattern was not related to the near transfer performance, but it possibly contributed to children's transferrable knowledge and skill development evidenced by the study finding that the control group outperformed the treatment group at the far transfer level.

SDA is an exploratory approach that does not make a priori assumption (Sanderson & Fisher, 1994). The evidence identified, therefore, might not fully reflect students' learning needs. Consequently, researchers need to use external measures (e.g., learning measures, performance measures, or observations) to validate the meaning of the collected evidence. This step helps researchers and practitioners to identify emerging learning interaction patterns in context and further understand the learners' needs and challenges. This is consistent with the call of adding expandability to exploratory approaches of educational data mining (Lim et al., 2021; Shibani et al., 2020). With data triangulation, we identified how specific gaming actions and interactions fostered children's CT development at a fine-grain level. Subsequently, based on the study findings, we can suggest more robust instructional design decisions.

5.3. Designing personalized learning experiences with activity selection

Based on our understanding of children's learning interactions and challenges from the previous phases, we yielded decisions of the adaptivity design in *Penguin Go*. Basically, researchers need to answer three questions in response to designing adaptivity in DGBL: which of the learners' variables to adapt, when to intervene, and which instructional content or support to present (Shute & Zapata-River, 2012). With the help of SDA, we systematically approached these questions using data-driven systems grounded throughout students' gameplay. First, we identified children's needs during gameplay. The SDA findings revealed children's inefficient problem solving. SDA enabled researchers to either monitor noticeable play patterns or estimate the levels of competency in problem solving. Subsequently, the collected data from SDA supported the design decisions as to when and how to intervene children's play (e.g., a behavioral trigger based on observed play patterns or a threshold based on the baseline competency level). Finally, we explored children's interactions with the embedded instructional supports—adaptive game challenges, adaptive cognitive supports, and adaptive meta-cognitive supports based on the children's needs we identified. SDA-driven data collection and decision helped researchers to understand children's interactions with given supports and this data informs which types of supports can be useful across individuals' learning profiles. Through this process, we aim to propose a systematic framework to approach instructional design for DGBL environment driven by learning analytics (c.f. Ifenthaler, 2017). This approach also provides a viable way to design adaptive learning experiences through real-time assessments (e.g., Roll et al., 2011; Rowe et al., 2021; Shute et al., 2020).

5.4. Theoretical and practical implications

The study contributes to the previous instructional design research by proposing a framework for applying learning analytics techniques such as SDA in the learning design of adaptive DGBL experiences for computing education. DGBL environments engage children in complex and interactive problem solving, which often needs systematic guidance and facilitation (see Kirschner et al., 2006). Practically, the conceptual framework proposed by this study provides instructional designers with a feasible way to utilize learning analytics in supporting instructional design (Ifenthaler, 2017). Based on the conceptual framework, we provided empirical evidence of how to integrate SDA into DGBL and discussed how to approach the design systematically with multiple sources of data. Specifically, the current study presents a case for how to design personalized learning experiences based on identified learners' needs through SDA.

In addition, the empirical data highlighted children's gameplay patterns and challenges in learning. This further advances the field's knowledge of how children learn through playing and the role of problem-solving in DGBL (c.f. Taub et al., 2020). Both quantitative and qualitative data underscore the needs in children's CT learning and provide practical design recommendations (i.e., game challenges, cognitive supports, and metacognitive supports) about how to potentially address the needs through adaptive design.

5.5. Limitations and future directions

This study has a few limitations. First, we did not fully execute a personalized game system, including real-time prediction modeling and not testing the usability of the proposed adaptivity design in DGBL. The scope of the current study was to suggest a methodological framework using SDA that informs evidence of adaptivity design in DGBL. Therefore, future research should develop and contextualize a validated prediction model based on SDA data to measure children's either problem-solving phases or CT development states and examine the efficacy of adaptivity triggered by SDA. Second, we did not refine relevancy behavior codes that indicate how gameplay event transitions refer to specific problem-solving phases. Future studies should refine behavior codes to clearly show the different stages of problem solving. For instance, the data of the SDA appeared skewed because one type of event (e.g., *creating blocks*) dominantly emerged. This event occurred through children's gameplay across different contexts (e.g., consecutive block creations and support abuse that switched back and forth between block creation and support access), but we could not label them differently in this study.

6. Conclusions

In this study, we have presented our SDA-driven methodological framework that focuses on collecting evidence of adaptivity design in DGBL. Specifically, using the game Penguin Go, we implemented a case study and the study finding demonstrates how the proposed methodological framework and its implementation ran to detect children's game behavior patterns. Through the case study, SDA identified children's key gameplay patterns and highlighted the effect of solution evaluation on developing CT. Finally, this study has presented design implications based on SDA results in DGBL for computing education.

Acknowledgement

We wish to acknowledge the contributions of Dr. Weinan Zhao for initially creating Penguin Go. We thank our colleagues, Dr. Ginny Smith, Dr. Demetrius Rice, Chih-pu Dai, Curt Fulwider, and Renata Kuba for their tremendous help in participants recruitment and facilitation of the sessions. We would also like to express our appreciation to the anonymous reviewers for their insightful comments and suggestions. Finally, this research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

Akram, B., Min, W., Wiebe, E., Mott, B., Boyer, K. E., & Lester, J. (2018). Improving stealth assessment in game-based learning with LSTM-based analytics. In *Proceeding of the International Conference of Educational Data Mining* (pp. 208-218). https://par.nsf.gov/biblio/10100664.

Arena, D. A., & Schwartz, D. L. (2014). Experience and explanation: Using videogames to prepare students for formal instruction in statistics. *Journal of Science Education and Technology*, *23*, 538-548. https://doi.org/10.1007/s10956-013-9483-3

Almond, R. G., Shute, V. J., Tingir, S., & Rahimi, S. (2020). Identifying observable outcomes in game-based assessments. In R. Lissitz & H. Jiao (Eds.), *Innovative psychometric modeling and methods* (pp. 163-192). Information Age Publishing.

Asbell-Clarke, J., Rowe, E., Almeda, V., Edwards, T., Bardar, E., Gasca, S., Baker, R. S., & Scruggs, R. (2020). The Development of students' computational thinking practices in elementary-and middle-school classes using the learning game, Zoombinis. *Computers in Human Behavior*, 106587. https://doi.org/10.1016/j.chb.2020.106587

Azevedo, R., & Aleven, V. (2013). Metacognition and learning technologies: An Overview of current interdisciplinary research. In R. Azevedo & V. Aleven (Eds.), *International Handbook of Metacognition and Learning Technologies* (pp. 1–16). Springer. https://doi.org/10.1007/978-1-4419-5546-3_1

Basu, S., Biswas, G., & Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Modeling and User-Adapted Interaction*, *27*(1), 5-53. https://doi.org/10.1007/s11257-017-9187-0

Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.

Ching, Y. H., Hsu, Y. C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends*, *62*(6), 563-573. https://doi.org/10.1007/s11528-018-0292-7

Clark, D. B., Nelson, B. C., Chang, H. Y., Martinez-Garza, M., Slack, K., & D'Angelo, C. M. (2011). Exploring Newtonian mechanics in a conceptually-integrated digital game: Comparison of learning and affective outcomes for students in Taiwan and the United States. *Computers & Education*, *57*(3), 2178–2195. https://doi.org/10.1016/j.compedu.2011.05.007

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A Case study. *Computers & Education*, *63*, 87-97. https://doi.org/10.1016/j.compedu.2012.11.016

Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A Framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education, 17*(3), 1-25. https://doi.org/10.1145/3105910

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A Review of the state of the field. *Educational Researcher*, *42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Hooshyar, D., Pedaste, M., Yang, Y., Malva, L., Hwang, G. J., Wang, M., Lim, H., & Delev, D. (2021). From gaming to computational thinking: An Adaptive educational computer game-based learning approach. *Journal of Educational Computing Research*, *59*(3), 383-409. https://doi.org/10.1177/0735633120965919

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Ifenthaler, D. (2017). Designing effective digital learning environments: Toward learning analytics design. *Technology, Knowledge and Learning*, *22*(3), 401-404. https://doi.org/10.1007/s10758-017-9333-0

Israel-Fishelson, R., & Hershkovitz, A. (2020). Persistence in a game-based learning environment: The Case of elementary school students learning computational thinking. *Journal of Educational Computing Research*, *58*(5), 891-918. https://doi.org/10.1177/0735633119887187

Ke, F., & Abras, T. (2013). Games for engaged learning of middle school children with special learning needs. *British Journal of Educational Technology*, *44*(2), 225-242. https://doi.org/10.1111/j.1467-8535.2012.01326.x

Ke, F., Shute, V. J., Clark, K. M., & Erlebacher, G. (2019). Designing dynamic support for game-based learning. In *Interdisciplinary Design of Game-based Learning Platforms. Advances in Game-Based Learning* (pp. 119-140). https://doi.org/10.1007/978-3-030-04339-1_6

Ke, F., & Shute, V. (2015). Design of game-based stealth assessment and learning support. In C. Loh, Y. Sheng, D. Ifenthaler (Eds), *Serious games analytics* (pp. 301-318). https://doi.org/10.1007/978-3-319-05834-4_13

Kinnebrew, J. S., Segedy, J. R., & Biswas, G. (2015). Integrating model-driven and data-driven techniques for analyzing learning behaviors in open-ended learning environments. *IEEE Transactions on Learning Technologies*, *10*(2), 140-153. https://doi.org/ 10.1109/TLT.2015.2513387

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An Analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, *41*(2), 75-86. https://doi.org/10.1207/s15326985ep4102_1

Lachney, M., Ryoo, J., & Santo, R. (2021). Introduction to the special section on justice-centered computing education, Part 1. *ACM Transactions on Computing Education*, *21*(4), 1-15. https://doi.org/10.1145/3477981

Levy, R. (2019). Dynamic Bayesian network modeling of game-based diagnostic assessments. *Multivariate Behavioral Research*, *54*(6), 771-794. https://doi.org/10.1080/00273171.2019.1590794

Lin, C. F., Yeh, Y. C., Hung, Y. H., & Chang, R. I. (2013). Data mining for providing a personalized learning path in creativity: An application of decision trees. *Computers & Education*, *68*, 199-210. https://doi.org/10.1016/j.compedu.2013.05.009

Lim, L. A., Gasevic, D., Matcha, W., Ahmad Uzir, N. A., & Dawson, S. (2021). Impact of learning analytics feedback on self-regulated learning: Triangulating behavioural logs with students' recall. In M. Scheffel, N. Dowell, S. Joksimovic, & G. Siemens (Eds.), *11th international learning analytics and knowledge conference* (pp. 364-374). ACM. https://doi.org/10.1145/3448139.3448174

Liu, Z., & Jeong, A. C. (2022). Connecting learning and playing: The Effects of in-game cognitive supports on the development and transfer of computational thinking skills. *Educational Technology Research and Development, 70*, 1867-1891. https://doi.org/10.1007/s11423-022-10145-5

Liu, Z**.,** Moon, J., Kim, B., & Dai, C. (2020). Integrating adaptivity to educational games: A Combination of bibliometric analysis and meta-analysis review. *Educational Technology Research and Development*, *68*(4), 1931-1959. https://doi.org/10.1007/s11423-020-09791-4

Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education, 27*(1), 1-29. https://doi.org/10.1080/08993408.2017.1308651

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61. https://doi.org/10.1016/j.chb.2014.09.012

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? The Case for guided methods of instruction. *American Psychologist*, *59*(1), 14-19. https://doi.org/10.1037/0003-066X.59.1.14

McGill, M. M., & Decker, A. (2020). Tools, languages, and environments used in primary and secondary computing education. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 103-109). https://doi.org/10.1145/3341525.3387365

Min, W., Frankosky, M., Mott, B. W., Rowe, J., Smith, P. A. M., Wiebe, E., Boyer, K. E., & Lester, J. (2019). DeepStealth: Game-based learning stealth assessment with deep neural networks. *IEEE Transactions on Learning Technologies*, *13*(2), 312-325. https://doi.org/10.1109/TLT.2019.2922356

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). Focus article: On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives, 1*(1), 3-62. https://doi.org/10.1207/S15366359MEA0101_02

Moon, J., & Liu, Z. (2019). Rich representations for analyzing learning trajectories: Systematic review on sequential-data analytics in game-based learning research. In A. Tlili., & M. Chang. (Eds.), *Data Analytics Approaches in Educational Games and Gamification Systems* (pp. 27-53). Springer. https://doi.org/10.1007/978-981-32-9335-9_2

Moore, G. R., & Shute, V. J. (2017). Improving learning through stealth assessment of conscientiousness. In *Handbook on digital learning for K-12 schools* (pp. 355-368). Springer, Cham.

Morrison, J. R., Bol, L., Ross, S. M., & Watson, G. S. (2015). Paraphrasing and prediction with self-explanation as generative strategies for learning science principles in a simulation. *Educational Technology Research and Development*, *63*(6), 861–882. https://doi.org/10.1007/s11423-015-9397-2

Owen, V. E., Roy, M. H., Thai, K. P., Burnett, V., Jacobs, D., Keylor, E., & Baker, R. S. (2019). Detecting wheel-spinning and productive persistence in educational games. In C. F. Lynch, A. Merceron, M. Desmarais, & R. Nkambou (Eds.), *Proceedings of the 12th International Conference on Educational Data Mining* (pp. 378-383). International Educational Data Mining Society.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, *21*(2), 267-280. https://doi.org/10.1016/j.learninstruc.2010.07.004

Rowe, E., Almeda, M. V., Asbell-Clarke, J., Scruggs, R., Baker, R., Bardar, E., & Gasca, S. (2021). Assessing implicit computational thinking in Zoombinis puzzle gameplay. *Computers in Human Behavior, 120*, 106707. https://doi.org/10.1016/j.chb.2021.106707

Sanderson, P. M., & Fisher, C. (1994). Exploratory sequential data analysis: Foundations. *Human–Computer Interaction*, *9*(3-4), 251-317. https://doi.org/10.1207/s15327051hci0903%264_2

Shibani, A., Knight, S., & Shum, S. B. (2020). Educator perspectives on learning analytics in classroom practice. *The Internet and Higher Education*, *46*, 100730. https://doi.org/10.1016/j.iheduc.2020.100730

Shute, V. J., & Kim, Y. J. (2014). Formative and stealth assessment. In *Handbook of research on educational communications and technology* (pp. 311-321). Springer.

Shute, V. J., & Moore, G. R. (2017). Consistency and validity in game-based stealth assessment. In H. Jiao & R. W. Lissitz (Eds.), *Technology enhanced innovative assessment: Development, modeling, and scoring from an interdisciplinary perspective* (pp. 31-55). Information Age Publishing, Inc.

Shute, V. J., Rahimi, S., Smith, G., Ke, F., Almond, R., Dai, C-P, Kamikabeya, R., Liu, Z., Yang, X., & Sun, C. (2020). Maximizing learning without sacrificing the fun: Stealth assessment, adaptivity, and learning supports in educational games. *Journal of Computer-Assisted Learning*, *37*(1), 127-141. https://doi.org/10.1111/jcal.12473

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142-158. https://doi.org/10.1016/j.edurev.2017.09.003

Shute, V. J., & Zapata-Rivera, D. (2012). Adaptive educational systems. In P. Durlach (Ed.), *Adaptive technologies for training and education* (pp. 7-27). Cambridge University Press.

Taub, M., Azevedo, R., Bradbury, A. E., Millar, G. C., & Lester, J. (2018). Using sequence mining to reveal the efficiency in scientific reasoning during STEM learning with a game-based learning environment. *Learning and Instruction*, *54*, 93-103. https://doi.org/10.1016/j.learninstruc.2017.08.005

Taub, M., Sawyer, R., Smith, A., Rowe, J., Azevedo, R., & Lester, J. (2020). The Agency effect: The Impact of student agency on learning, emotions, and problem-solving behaviors in a game-based learning environment. *Computers & Education*, *147,* 103781. https://doi.org/10.1016/j.compedu.2019.103781

Tlili, A., Chang, M., Moon, J., Liu, Z., Burgos, Chen, N., & Kinshuk. (2021). Literature review of empirical studies on learning analytics in educational games: From 2010 to 2018. *International Journal of Interactive Multimedia and Artificial Intelligence*, *7*(2), 250-261. http://dx.doi.org/10.9781/ijimai.2021.03.003

Turchi, T., Fogli, D., & Malizia, A. (2019). Fostering computational thinking through collaborative game-based learning. *Multimedia Tools and Applications*, *78*(10), 13649-13673. https://doi.org/10.1007/s11042-019-7229-9

Vanbecelaere, S., Van den Berghe, K., Cornillie, F., Sasanguie, D., Reynvoet, B., & Depaepe, F. (2020). The Effectiveness of adaptive versus non-adaptive learning with digital educational games. *Journal of Computer Assisted Learning*, *36*(4), 502-513. https://doi.org/10.1111/jcal.12416

Walkington, C. A. (2013). Using adaptive learning technologies to personalize instruction to student interests: The Impact of relevant contexts on performance and learning outcomes. *Journal of Educational Psychology*, *105*(4), 932–945. https://doi.org/10.1037/a0031882

Weintrop, D., Holbert, N., Horn, M. S., & Wilensky, U. (2016). Computational thinking in constructionist video games. *International Journal of Game-Based Learning*, *6*(1), 1-17.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725. https://doi.org/10.1098/rsta.2008.0118

Zaki, M. J. (2001). SPADE: An Efficient algorithm for mining frequent sequences. *Machine learning*, *42*(1-2), 31-60. https://doi.org/10.1023/A:1007652502315

Zhang, L., & Nouri, J. (2019). A Systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *141*, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills? *Computers & Education*, *141*, 1-13. https://doi.org/10.1016/j.compedu.2019.1