Journal of Pacific Rim Psychology
Volume 15: 1–18

DOI: 10.1177/18344909211008264
journals.sagepub.com/home/pac
SAGE

Evaluation of a Practice System Supporting Distributed Practice for Novice Programming Students

Baoping Li[1,2], Fangjing Ning[3], Lifeng Zhang[1], Bo Yang[2] and Lishan Zhang[4]

Abstract

Programming is an important skill in the 21st century, but it is difficult for novices to learn. To help students practice and learn efficiently, the authors developed a mobile platform called Daily Quiz, which incorporated distributed practice theory. To evaluate the impact of distributed practice in programming learning facilitated by Daily Quiz, the authors conducted a between-subject experiment with 200 freshmen divided into two groups. Both groups received the same number of multiple-choice questions via Daily Quiz. However, the control group was encouraged to practice every 7 days, whereas the experimental group was encouraged to practice every 3 days. The results showed that this simple manipulation significantly improved the experimental group's performance on final exams. Further analysis revealed that the experimental group of students achieved a higher rate of first-check correctness and tended to be more engaged in academic social interaction. Finally, a behavioral sequence analysis was adopted to compare the behavioral patterns of the two groups to investigate how distributed practice helped the students learn more efficiently.

Keywords

learning analytics, programming learning, distributed practice, massed practice, knowledge retrieval

Received 29 February 2020; accepted 15 March 2021

Introduction

In our information society, programming is an essential skill for data analysis in domains such as economics, chemistry, biology, and social science. Mastering a programming language has become a requirement for many college students (Reardon & Tangney, 2015).

Programming contains many knowledge components, and its application scenarios feature great comprehensiveness and abstraction. Students sometimes fail to understand the most fundamental concepts and are unable to produce the most basic programs (Eckerdal, 2009). Therefore, students often complain that programming is hard to become accustomed to, which often leads to high classroom dropout rates (Yadin, 2011). Many learning strategies have been applied to help students learn programming. For example, peer collaboration, timely feedback, and reflective learning have been used to promote students' learning efficacy (Tom, 2015; Vihavainen et al., 2014), and a visualized programming environment has been developed to improve students' learning motivation (Tsai, 2019; Yukselturk & Altiok, 2016). As a result, the average failure rate of introductory programming courses has decreased from 33% to 25%, and, compared with algebra and science, technology, engineering, and mathematics courses, the failure rate of introductory programming courses does not seem alarmingly high (Bennedsen & Caspersen, 2007, 2019; Simon et al., 2019; Watson & Li, 2014). This

[1]Faculty of Education, Beijing Normal University
[2]Advanced Innovation Center of Future Education, Beijing Normal University
[3]Elementary School, Peking University
[4]National Engineering Research Center for E-Learning, Central China Normal University

Corresponding author:
Lishan Zhang, Science Hall, Central China Normal University, 152 Luoyu Road, Wuhan, Hubei 430079, P. R. China.
Email: lishan.zhang@mail.ccnu.edu.cn

achievement gives us confidence that programming is a skill that can be taught to everyone instead of a special group of students (Spieler et al., 2019). Therefore, it is worthwhile to study how non-engineering students can be supported efficiently when learning programming.

Optimizing both learning content and practice-time allocation is important for effective learning (Dunlosky & Rawson, 2015). Different from most of the existing studies that focus on prior factors, this study explored how to help non-engineering students better allocate practice time. Non-engineering students often lack opportunities to practice programming in their coursework, so the effective use of their limited time for programming learning should be considered (Luxton-Reilly, 2016). In cognitive psychology, scholars have found a distributed practice effect (lag effect or the effect of spacing) in learning. For learning tasks with the same content, learned knowledge is retained longer when practice is distributed rather than massed (Dempster, 1988; Litman & Davachi, 2008). In this study, we translated distributed practice theory into technology-enhanced learning environments and evaluated its effectiveness with learning analytics and artificial intelligence technologies.

Related Work on Distributed Practice

DeCecco and Crawford (1974) define massed practice as the learning of tasks concentrated into one time period and distributed practice as the learning of tasks spread over several time periods alternating with periods of rest. Due to the lack of strict and exact criteria, massed practice and distributed practice are relative concepts (Moss, 1995), so distributed practice often co-occurs with massed practice (Budé et al., 2011).

Psychological experiments in word learning, reading, and memorizing tasks have supported the distributed practice effect and found that distributed practice enhanced learners' memorizing and understanding of knowledge, thus facilitating the reduction of mistakes, the retention of knowledge, and the transfer of knowledge (Bjotk & Bjork, 2011; Karpicke & Bauernschmidt, 2011; Soderstrom et al., 2016). In recent years, the development and application of functional magnetic resonance imaging has further provided neurological evidence for the distributed practice effect (Xue et al., 2013).

In addition to the preceding laboratory experiments, authentic teaching experiments have confirmed the distributed practice effect. For example, Budé et al. (2011) asked students to learn statistics in a distributed way, and the students grasped the related knowledge better and more deeply. Rohrer and Taylor (2006) divided students into two groups; one group worked on 10 mathematics questions in a single session in 1 week (massed practice), while the other worked on the same 10 questions in two sessions held over 2 weeks, five each week (distributed practice). A test conducted 4 weeks after the students completed their practice showed that the students engaged in distributed practice obtained better learning results. However, the spanned time was extended in the distributed practice, in addition to the time allocation.

The effectiveness of distributed practice has been explained from different perspectives. Sobel et al. (2011) believe that the distributed effect occurs because of a memory advantage when people learn material on several occasions. Challis (1993) found that material is preactivated in the storage area of the brain when it is repeatedly presented in a massed manner, so the learner does not need to activate it by further processing. However, if the material is presented at intervals, the learner needs to retrieve the material more often, lengthening memory.

Shimoni (2013) claims that distributed practice not only consolidates memory, but also refines students' understanding of knowledge by allowing time for them to forget over the interval between successive presentations. Gerbier and Toppino (2015) claim that the spacing effect can be described by the deficient processing hypothesis: spatial repetition can result in more efficient encoding and better memory in the brain than immediate repetition.

In summary, existing studies have shown the advantage of distributed over massed practice in several domains. The practice strategy helps students both memorize and understand the related knowledge. However, the effectiveness of distributed practice has not been extensively studied in programming education.

Related Work on Programming Education

Appropriately designed computer learning environments can provide learners with great and unique opportunities for learning (Kordaki, 2010). For example, Hoffman et al. (2011) used an application (app) called C-doku to improve students' code-reading skills. C-doku provides a large number of code snippets to students and allows them to complete the input and output by reading the code. When a student completes the code, C-doku automatically checks the solution and provides immediate feedback. Zingaro et al. (2013) developed the Python Classroom Response System, which enables students to submit their answers, and the system then automatically provides immediate feedback on the correctness. Teachers can view the answers and adjust their teaching plans accordingly. Hovemeyer and Spacco (2013) introduced

CloudCoder, a web-based platform for programming practice. On the platform, a typical practice exercise asks the student to write a complete function or complete an incomplete program. Na et al. (2017) developed a web-based Java programming learning assistant system. It provides fill-in-the-blank problems for novices to study grammar and basic programming skills through code reading. The existing systems provide students with plenty of contexts for programming exercises and give them appropriate, immediate feedback on their answers. In addition to cognitive feedback, some of the platforms provide metacognitive instructions such as self-regulation guidance. However, strategies regarding time management have not yet received enough attention in such system designs (Prather et al., 2020).

Quizit is one of the very few exceptions. The web-based system helps students manage their practice time and enables students to practice a little each time (Alzaid et al., 2017). Zhang et al. (2020) used Quizit to facilitate students in learning programming and found that web-based features impeded students from accessing the system. Procrastination and related problems with managing programming are viewed as primary causes of student attrition (Martin et al., 2015). Therefore, we developed a mobile app named Daily Quiz (DQuiz), which also supports students in performing bite-sized practice every day and extends Quizit's functions by improving the feedback mechanism, practicing commenting functions, and adding more data visualization features. With the help of the logger in DQuiz, students' time management and learning behaviors can be comprehensively evaluated.

Research Questions and Contributions

Although most of the studies have shown that distributed practice is favored over massed practice, there are some contradictory conclusions regarding the distributed effect in real classroom learning. A 9-week experiment of mathematics learning in Grade 2 and Grade 4 students found that the massed practice group learned better than the distributed group (Moss, 1995). According to Suzuki and DeKeyser (2015), massed practice seems to be at least as effective as distributed practice in procedural knowledge learning. Due to complications in the real learning context, little is known about the typical distribution of practice that occurs in the classroom (Dempster, 1988). A new research approach is needed to explore the distributed effect in real classroom environments.

Educational data mining and learning analytics promise a better understanding of student behavior and knowledge, and can explore new information on the tacit factors affecting students' learning (Ihantola et al., 2015). Researchers have started to use students' learning data collected from a learning management system to predict whether a student improves in their programming skills (Spacco et al., 2015). In the area of distributed learning, Alzaid et al. (2017) analyzed study data to measure the effects of bite-sized practice for programming novices from a statistical perspective. By analyzing students' behavior in an introductory programming course, Leppänen et al. (2016) found that students tended to space out their work over multiple days each week and, while working on course assignments, pauses of only a few seconds correlated positively with examination scores, while pauses of a few minutes correlated negatively with examination scores; however, the study found that "student pausing behaviours are poorly explained by self-regulation"(Leppänen et al., 2016, p. 41). Our pilot study found that those who practiced in a distributed manner gained significant improvements in programming learning (Zhang et al., 2020). However, it is unknown whether encouraging students to conduct distributed practice can help them engage in an effective learning pattern to promote programming learning.

To further understand how and why distributed practice affects programming learning, our study used learning analytic techniques to analyze the log files recorded by DQuiz and explore the following three research questions:

1. Does distributed practice with multiple-choice questions help students achieve better academic performance?
2. Does distributed practice affect students' performance in the practice system?
3. Does distributed practice affect students' behavioral patterns in the practice system?

By answering the research questions, this article makes three contributions:

1. It describes the design and implementation of the mobile app DQuiz, which facilitates students to conduct distributed practice in programming learning by using multiple-choice questions.
2. It shows that simple human encouragement of students' distributed practice behaviors, combined with DQuiz, could significantly improve students' learning outcomes.
3. It uses learning analytic techniques on the system log files to evaluate comprehensively how distributed practice affects students' learning.

The remainder of the article first describes the system supporting distributed practice on multiple-choice questions. It then reports on the evaluation of

the distributed practice strategy by conducting a between-subject experiment. Finally, the results are discussed.

Description of the System Supporting Distributed Practice

A practice system on the mobile platform DQuiz was implemented to support students in conducting distributed practice conveniently. DQuiz used only multiple-choice questions for students' practice because this question type can keep each practice session short, provide students with immediate feedback with ease, and prevent students from becoming overwhelmed. Multiple-choice questions are also easy to fit on a mobile screen. E-medals and discussion boards were embedded in the system to encourage students' self-regulation and interactions with each other, potentially promoting student learning (Chi et al., 2018; Denner et al., 2014; Dillenbourg, 2005). The system assigned students a few practice questions every day (usually two or three) and used a colorful calendar to show them when they had unfinished daily practice to complete. Following a calendar to practice is considered distributed practice because intervals are inserted between consecutive sets of daily practice.

The rest of this section first details the implementation of DQuiz and then describes how students used it for practice and how teachers could authorize daily practice with the system.

Implementation of DQuiz

The architecture of the DQuiz system can be divided into three main layers: presentation, business logic, and data (see Figure 1). The presentation layer was the user interface between the user and the system. The system supported two user roles: student and teacher. Students accessed the system to practice through mobile devices and teachers accessed it to manage practice questions through computers. The specific functions are detailed in the following section.

HTML5 was used to build the student mobile app so that it would be compatible with both iOS and Android. The business logic layer included management and interaction modules. The management module helped to manage student login information and practice questions. The interaction module processed and logged student behaviors and postings on the discussion board. The data layer consisted of four databases that stored user information, practice questions, forum postings, and interactive behaviors, so that the data analysis could be easily performed.

Functions of DQuiz

This section describes how the students could practice and how the teachers could manage practice questions in the system.

Student App. The mobile device app enabled students to practice and interact on the discussion board. When a student logged into the system, they would see the

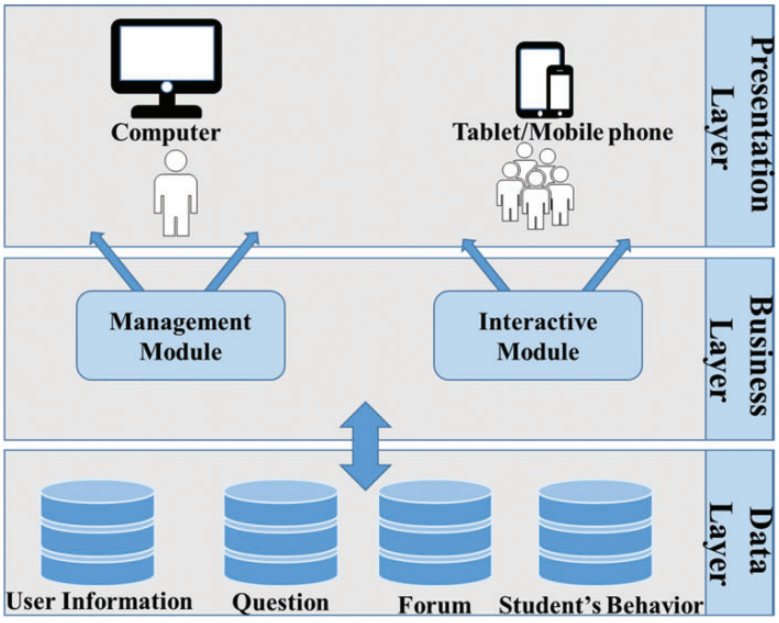

Figure 1. Architecture of DQuiz.

practice calendar (Figure 2). The calendar showed the status of all daily practice in the current month. The status of the daily practice was denoted with different colors: yellow represented today's practice; green represented practice that had been finished; red represented an unfinished practice; and gray meant no practice was available. E-medals were shown under the calendar to encourage the students to practice continually, improve practice correctness, post comments on the discussion board, and obtain more thumbs-up for their comments. The meaning of each type of e-medal is described in Figure 2. This visualization could help students realize quickly how many unfinished practices they needed to complete and how well they were performing in their studies.

Students could choose a date for that day's practice. After a student clicked on the date, they saw a multiple-choice question (see Figure 3). The questions were intended to help the students enhance their retention of the material learned (Butler et al., 2008).

After a student selected an alternative and clicked on the "Submit" button, they received minimal feedback indicating the correctness of their answer. The student could continue trying until they made the correct selection or they could click on the "See the correct answer" button. Clicking on the "See the correct answer" button also meant the student had given up on finding the correct answer. Providing appropriate explanations for each alternative on multiple-choice questions can enhance students' learning (Butler & Roediger, 2008; Yang et al., 2015). Therefore, an explanation was added for each question. Students could click on the "See the explanation" button to further understand how the correct answer had been reached. On completion of the current practice, the students could continue to finish other practice questions by clicking on the buttons showing different question identifications, or they could return to the calendar through the "Navigation" button to choose another date on which to practice.

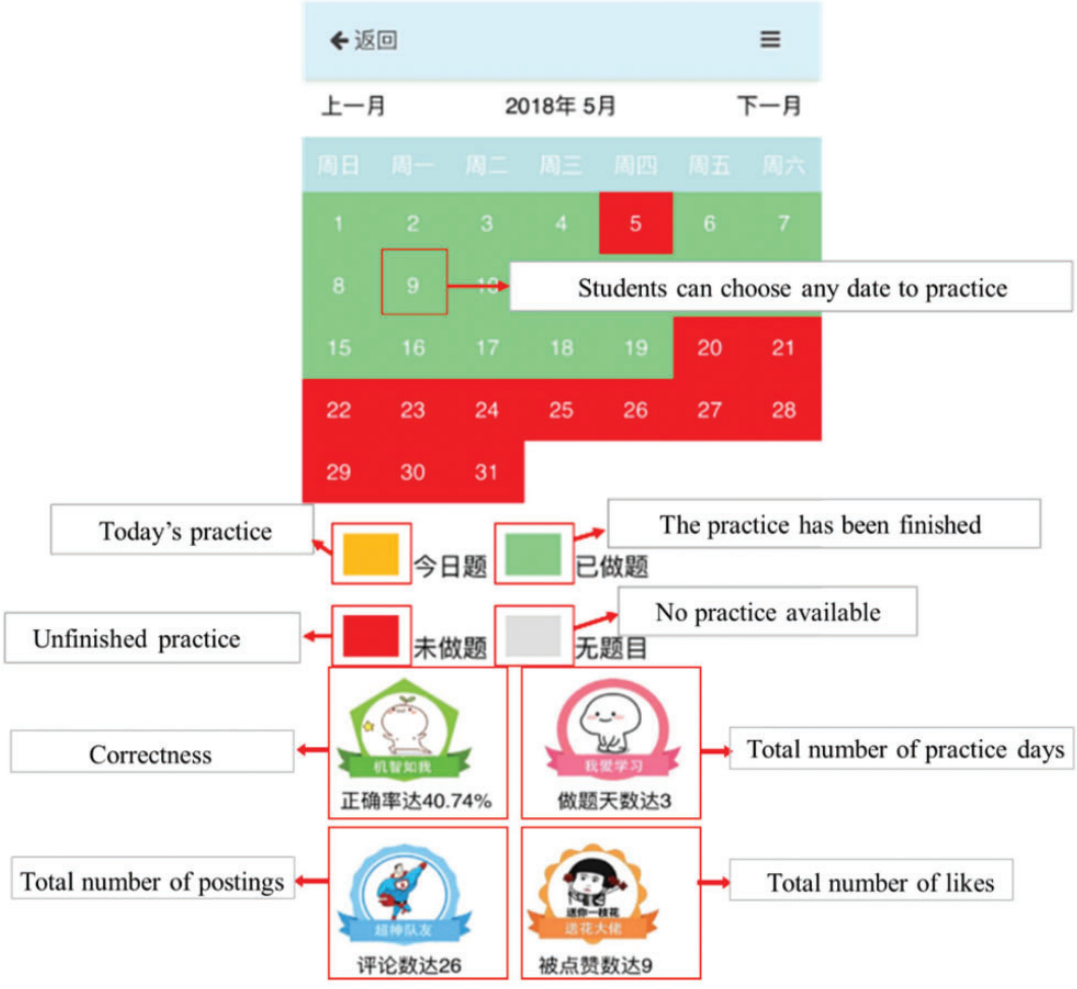

Figure 2. Practice calendar view.

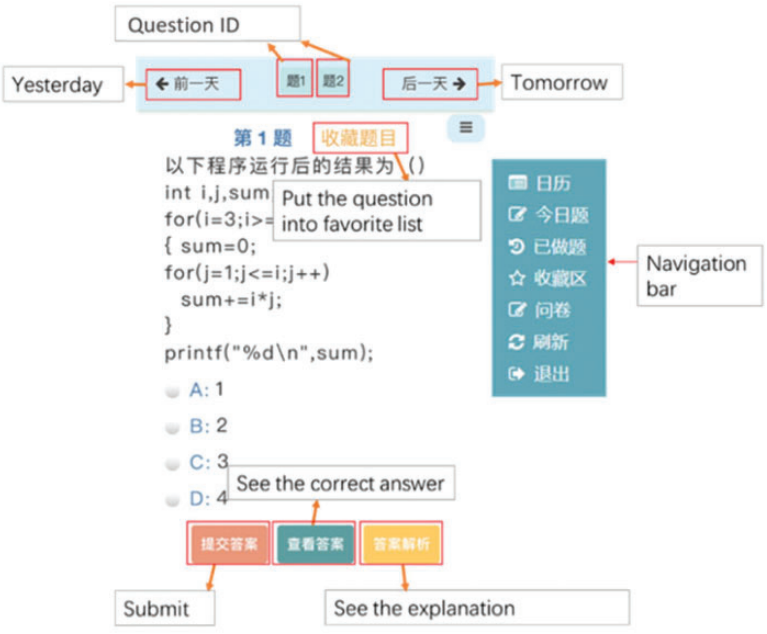

Figure 3. Daily practice.

Figure 4. Discussion Board.

A discussion board was attached below each practice question (see Figure 4), where the students could interact with each other. The discussion board enabled the students to post, reply, and give a thumbs-up to existing posts.

The students could click on the "Navigation" button to be redirected to the review panel (see Figure 5), where they could see the list of all the practice questions that had been completed. The questions were grouped by the knowledge component, defined by the

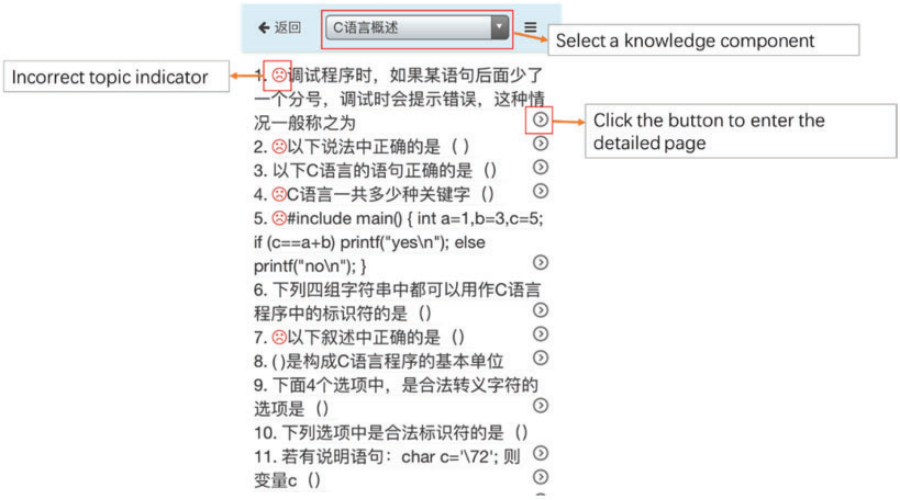

Figure 5. A Page of Finished Questions.

instructor. A special indicator was used to mark the questions that students had answered incorrectly.

Teacher Module on Web Browser. A web-based management platform was designed for the teachers (see Figure 6). The teachers could authorize questions and monitor the practice progress of the students. With this function, the teachers could identify the students who needed to be reminded. Additionally, the teachers could check the percentage of correctness by question or by student so that they could adjust the difficulty of the practice in the future.

Method

Participants

Two hundred freshmen participated in the experiment: 61 males and 139 females with an average age of 18.82 ($SD = 0.85$); they were all non-engineering majors.

Instrument

The C programming language was chosen as the teaching domain because it is usually the first programming language students learn in China. The practice questions were authorized by the lecturer and teaching assistant, and included four types: predicting the output of a given program; spotting the error in a buggy program; concept retention; and completing an incomplete program. A sample practice question is given for each type below.

Predicting the Output of a Given Program. The result of the following program is: _____

```
int x = 117, i = 0;
char a[5];
do{
switch (x%16)
{
case 10: a[i] = 'a'; break;
case 11: a[i] = 'b'; break;
case 12: a[i] = 'c'; break;
default: a[i] = 'f'; break;
}
i++;
x = x/16;
}
```

A. F A
B. F
C. A B
D. A C

Spotting the Error in a Buggy Program. The function implemented by the following program is to output array c in reverse order, but this function cannot be implemented. Please analyze the error.

```
int i = 5, j = 0;
char c[5] = {'a', 'b', 'c', 'd', 'e'}, s[6];
do{
s[j] = c[i - 1];
j++;
printf(%c, s[j]);
}while(i > 0);
```

A. Initialization error of array c[5].
B. The length of the array s[6] is inconsistent with the number of elements of the output.
C. The position of the statement of j++; is wrong.
D. The judgement statement of while is wrong.

Concept Retention. The legal identifier among the following options is:

A. _A

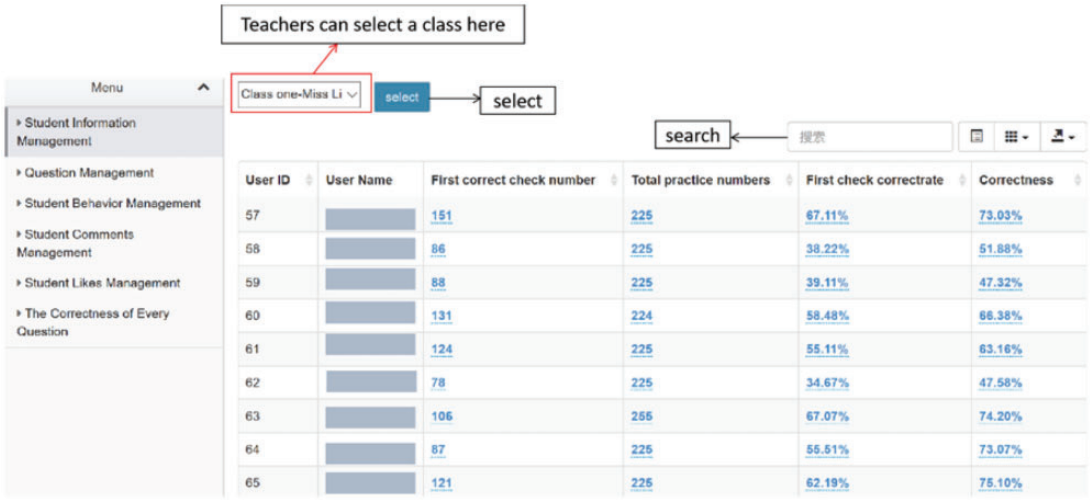

Figure 6. Interface of the Management Platform.

B. b−a
C. goto
D. int

Completing an Incomplete Program. The function of the following program is to find the maximum value in a one-dimensional array. Which choice can be filled in 1 and 2?

```
int a[5]={3,4,7,2,5};
int i, max;
max=a[0];
for(i=1;i<5;i++)
if(1)
2;
Printf(max=%d\n, max);
A. a[i]<max, max=a[i]
B. a[i]>max, max=a[i]
C. a[i]<max, max=a[i-1]
D. a[i]>max, max=a[i-1]
```

Midterm and final exams were used to measure the students' academic performance. The types of questions in the two exams were the same. Each contained 20 multiple-choice questions (total score 40 points), 4 fill-in-the-blank questions (15 points), 4 debug questions (15 points), and 2 coding questions (30 points). The students were expected to finish the test within 100 minutes, and the total score of the exam was 100. All of the questions were randomly selected from the test-question bank. The students completed the exam in class on computers.

Experimental Design

Our main hypothesis was that students who followed the distributed practice strategy should be able to achieve better academic performance than those who did not.

Because DQuiz was unable to force the students to follow the practice strategy, even though it embedded the utility to facilitate it, the students were encouraged by a teaching assistant to practice periodically. Our pilot study found that 3 days were a reasonable time for consecutive practice sessions in this case. An interval between practice that was too short could overwhelm the students. Moreover, the students tended to practice every week or at even longer intervals when no intervention was enforced (Zhang et al., 2019). Therefore, teacher guidance was needed to help the students practice in a distributed fashion. By conducting a between-subject study and the related data analysis, we evaluated the effect of distributed practice on multiple-choice questions in the domain of programming learning.

The course lasted for one semester. Practice questions were assigned each day until a week before the final exam. The practice questions numbered 228. Among them were three questions for the students in the first 14 days and two questions for the students in the subsequent 93 days (to increase practice difficulty). The students were required to come to class once a week. The classes lasted 210 minutes, with a 10-minute break every 45 minutes. The students were randomly divided into two classes with 100 students in each, and both classes were directed by the same lecturer. The classes were then randomly chosen to be either the distributed practice group (i.e., the experimental group) or the massed practice group (i.e., the control group). Both groups studied in class and used DQuiz to practice after class as their daily practice. Both groups also wrote or debugged source code every week as their course assignments, which were also noted as

homework. The students reported the amount of study time spent on the assignments every week as an indicator of other study time (see Table 2).

At the beginning of the class, the lecturer introduced DQuiz to the students and told them to use the app voluntarily. The lecturer suggested that the students finish all the questions on the app as their daily practice by the end of the semester. As an encouragement to use the app, the students could obtain scores from 0 to 100 based on the number of practices they finished. This score constituted 5% of their course grades, which meant that a student who finished more than 95% of the practice would receive 5 and those who finished 60% would receive 3, so the students would not feel forced to use it. The remaining 95% of the course grades comprised class participation (5%), homework (15%), the midterm exam (15%), and final exam (60%). The lecturer advised all of the students to follow the practice calendar to achieve a good academic performance and informed them that some of them might receive different manipulations. However, the students did not know what the manipulations would be. Both groups were advised to complete their unfinished practice on DQuiz weekly during class. The only manipulation was that the experimental group was also encouraged every 3 days by the teaching assistant through instant messaging. This meant that the experimental group was advised to practice less but at a higher frequency than the control group. This intervention caused the experimental group to practice in a distributed manner and the control group to practice in a massed manner.

In the first class, all of the students were asked to answer five questions about the C programming language as the pretest. All of the students were also required to report their study time weekly by an e-questionnaire issued in class, which indicated the amount of effort they were putting into the class. The students' academic performance was measured by their midterm and final exams. The experimental procedure is illustrated in Figure 7.

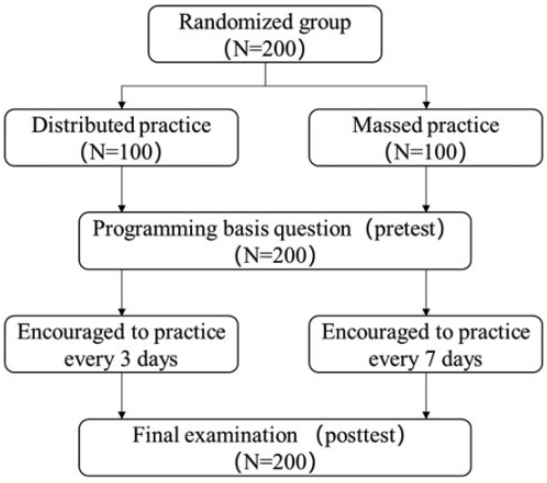

Figure 7. Experiment procedure.

Data Collection and Analysis

Indicators. We designed nine indicators to measure the students' behaviors on DQuiz and their off-system status. The indicators and the corresponding calculation methods are listed in Table 1. The indicators can help us to understand how the two groups of students behaved differently in the system. The correlations between the indicators and the students' final exam grades were calculated to potentially disclose how the students finished with different academic performances.

Lag Sequential Analysis. Lag sequential analysis was used to disclose the students' behavioral patterns. It was first proposed by Bernstein (1978). The method is mainly used to test whether statistical significance occurs when another behavior follows the occurrence of a certain behavior. Usually, the frequency of the occurrence of each behavioral sequence must be counted, and the z score of the frequency must be calculated. When the z score is greater than 1.96, the frequency of the corresponding behavioral sequence reaches a statistically significant level ($p < .05$). Based on the recorded log files, a practice session was coded into a list of actions according to the rules described in Table 2. A lag sequential analysis was performed by using the Generalized Sequential Querier (GSEQ) package to find significant transitions between two coded actions. By comparing the transitional graphs of the two groups, the difference in the behavioral patterns could be identified. This difference could provide explanations for any difference in academic performance.

Results

Students in the Distributed Practice Group Outperformed Those in the Massed Practice Group

The students in the distributed and massed practice groups scored 85.67 ($SD = 10.40$) and 77.40 ($SD = 13.42$), respectively, on the final exam. The difference is significant ($t = 4.76$, $p < .001$). However, we found that many students did not finish enough practice questions by the end of the semester, and most of these students were from the massed practice group. As a result, many of the students in the control group completed significantly fewer total practices than those in the experimental group. Therefore, the

Table 1. Indicators for Profiling the Students.

	Indicator name	Description
DQuiz measures	Practice frequency	Average number of days between two consecutive practice sessions
	Total number of postings	Total number of postings made by a student
	Time of each practice	The average amount of time a student spent on each question
	Online time	The overall amount of time a student spent on DQuiz
	First-check correctness	The percentage of a student's correct responses on their first submission
	Correctness	The percentage of a student's correct responses on their entire submission
	Total practice numbers	The number of questions each student completed
Off-system measures	Other study time	Other study time students spent per week on C programming learning after class (e.g., doing homework) other than using DQuiz
	Programming basis	The result of the programming basis question

Table 2. Coded Actions.

Action	Description	Code
Finish a new practice	Students start to work on a new practice question and submit the answer	NP
See the correct answer	Students click the button to see the correct answer	CA
Click on the explanation	Students click the button to check the answer explanation	CE
Favorite the question	Students click the button to put the question into a favorites list	CQ
Post a comment	Students post a comment on the forum	CM
Like	Students like a question or an answer	LK
Redo a practice	Students practice the question they have already done	RE

difference between the two groups may be due to the amount of practice rather than the practice distribution. To ensure that the two groups completed a similar number of practices in total, students who completed less than 70% of all questions were not included in the data analysis. The threshold was decided subjectively to retain a reasonable amount of the participant data. The reported results were conducted based on 93 students in the distributed practice group and 77 in the massed practice group.

After all the unqualifying students were removed, the distributed and massed practice groups scored 84.05 ($SD = 10.62$) and 79.58 ($SD = 12.55$), respectively, on the midterm exam, and 85.93 ($SD = 10.59$) and 79.76 ($SD = 12.68$), respectively, on the final exam. The results showed that the two groups had similar programming experiences and weekly study times, which could have affected their final exam grades, and no significant difference was observed according to the t test (see Table 3). Then, an analysis of covariance (ANCOVA) was performed to test the significance of the difference between the two groups on their final exams, where programming experience and personal study time were used as the covariates. The difference was significant ($p = .002$, $F = 6.52$). Therefore, distributed practice on multiple-choice questions improved students' academic performance in programming.

Impact of Distributed Practice on DQuiz Usage Outcomes

The distributed practice group might have performed better than the massed practice group simply because the students received encouragement more frequently and then spent more time practicing. Therefore, it was important to explore how distributed practice affected the usage of the practice system. The students' usage of the practice system was quantified roughly with the indicators introduced below (for details, see Table 4). The results suggested that the distributed practice group was more engaged in the discussion and had significantly higher rates of first-check correctness than the massed practice group. The students' practice frequencies were consistent with the frequency of encouragement. The distributed practice group spent a longer time on DQuiz than the massed practice group, but the difference was not significant. The difference was probably because the distributed practice group took longer to answer each practice question.

Table 3. Difference in Exam Scores and Study Time Between the Experimental and Control Groups.

	Distributed group ($N = 93$)					Massed group ($N = 77$)					
	M	*SD*	1/4	2/4	3/4	*M*	*SD*	1/4	2/4	3/4	*t* value
Midterm exam	84.05	10.62	80.25	86.00	90.75	79.58	12.55	72.00	81.50	90.00	2.50*
Final exam	85.93	10.59	80.65	88.70	92.75	79.76	12.68	70.75	81.30	90.00	3.46***
Pretest	3.42	1.50	1.96	4.17	5.00	3.08	1.40	1.76	3.33	4.17	1.48
Other study time (hours)	1.95	3.27	0.00	0.00	3.33	1.92	3.10	0.00	0.00	3.42	0.06

*$p < .05$. ***$p < .001$.

Table 4. Difference of Process Features Between the Experimental and Control Groups.

	Distributed group ($N = 93$)					Massed group ($N = 77$)					
	M	*SD*	1/4	2/4	3/4	*M*	*SD*	1/4	2/4	3/4	*t* value
Practice frequency	3.28	1.38	2.02	3.45	4.22	9.46	5.11	6.33	7.13	10.88	−11.18***
Total number of postings	3.72	6.46	0.00	0.00	5.50	1.01	2.06	0.00	0.00	1.00	4.12***
Correctness on the first check	0.62	0.10	0.55	0.62	0.69	0.56	0.16	0.45	0.58	0.68	2.95**
Correctness	0.66	0.08	0.60	0.67	0.72	0.62	0.11	0.55	0.63	0.71	2.58**
Time of each question (minutes)	3.15	1.40	2.02	3.01	4.07	2.85	1.69	1.91	2.47	3.41	1.23
Total online time (hours)	15.48	8.24	9.37	14.41	19.29	13.09	7.88	8.88	11.70	15.17	1.93
Total new practice number	227.20	5.16	225	230	230	227.21	3.92	225	229	230	−0.05
Total practice numbers	290.80	53.00	273	283	283	274.45	29.21	256	272	283	2.54*

*$p < .05$. **$p < .01$. ***$p < .001$.

Correctness on the first check was also found to be significantly correlated with final exam scores ($r = 0.57$, $p < .001$). Consequently, correctness on the first check could be treated as a strong predictor of students' final academic performance. This indicator could be used to monitor students' academic performance and was calculated throughout the entire experiment. To understand the difference in the first-check correctness between the two groups, a new measure—the difference in first-check correctness—was calculated. For each question, the difference in first-check correctness = (first-check correctness of the experimental group- − first-check correctness of the control group)/ correctness of the control group. Because two students could perform different sets of practice questions within a given time range and first-check correctness rates on different practice questions were not comparable, it was reasonable to monitor how the difference in first-check correctness varied using question identifications instead of time stamps. In addition, the questions were ordered by time stamps, so question identifications reflected time sequence information. Therefore, a graph was constructed to describe how the difference in the first-check correctness between the two groups of students varied with question identification (see Figure 8). The difference in the first-check correctness was averaged every 10 practice questions to smooth the graph. The graph roughly describes how the difference in the first-check correctness between the two groups of students changed over time. This finding was also consistent with the theoretical hypothesis of the distributed practice effect: each new knowledge component obtained a memory advantage over the time interval, and the memory advantage gradually accumulated and appeared in the subsequent learning process. The visual trend was clear, and a linear regression was performed to use the order of the questions to predict the difference in the first-check correctness ($R^2 = 0.257$). This result suggested that the difference in the first-check correctness of the students was gradually formed, which resulted in a significant difference in their final exam.

Impact of Distributed Practice on Behavioral Patterns

Table 2 lists the seven coded practice actions We defined the transition from one action to another as a transitional pair (a behavioral sequence). The seven coded actions formed 7*7 = 49 transitional pairs. Each transition can also be treated as a bigram action. Then, GSEQ was used to calculate the frequency and probability of the transitional pair of student actions, and a larger *z* score meant a greater transitional probability.

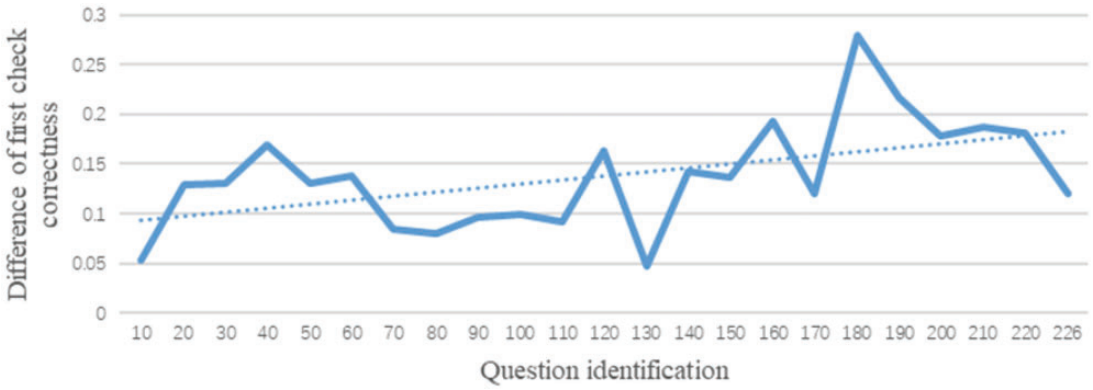

Figure 8. The trend of how the difference of the first-check correctness between the experimental and control groups varied with the question identification.
Note. The solid line is the difference of the first-check correctness between the two groups (the first-check correctness of the distributed group minus the first-check correctness of the massed group) in every question. The dashed line is the trend of the linear regression.

The results of the lag sequential analysis for the distributed and massed practice groups are described in Tables 5 and 6, respectively. To compare the differences visually, transitions with significance ($z > 1.96$, $p < .05$) were selected to form transitional diagrams (see Figure 9). In Figure 9, each node represents a coded action, and the arrow stands for the transitional direction, with the number indicating the transitional probability for that pair.

By comparing the two transitional diagrams, we found that the two groups shared the following three basic behavioral patterns:

1. Finishing new practices consecutively: the NP → NP pair means students doing new practices consecutively, and its probability in the distributed practice group is greater than that in the massed practice group. This result is because DQuiz provided feedback for the wrong answer; therefore, students could perform actions other than doing a new question. Our analysis shows that the distributed practice group had a higher first-check correctness rate, and this rate ensured that the students could complete several new questions consecutively.
2. Doing new practices and revising: two behavioral sequences, NP → CA → CM → LK and NP → CA → CM → CE → CQ, show this pattern.
3. Redoing the practice: two behavioral sequences, RE → RE and RE → CE → CQ → RE, show this pattern. Similar to doing new questions, doing completed questions consecutively was probably because the students made few mistakes while doing completed questions out of order, usually meaning that they still tended to make mistakes and therefore had to study the related knowledge further. It can also be inferred that the higher correctness rate of the distributed practice group ensured that they could redo the questions consecutively.

Three additional significant transitions could be found for the massed practice group: the students tended to practice new questions (NP) after adding the current question to their favorites list (CQ); they tended to add the current question to their favorites list after making comments (CM → CQ); and they tended to view the correct answer—that is, a bottom-up hint—after submitting answers to the practice questions they had already done (RE → CA).

When a student adds new items to a favorites list, they probably believe that the corresponding concept of the question has not been learned well and that they need to practice it again in the future. Rather than starting to practice a new question, reflection is better at this moment. However, the massed practice group was probably in a hurry to complete all the unfinished questions. Doing so caused them to spare no time between CQ and NP. In addition, the frequent combination of CQ and NP could quickly increase the number of questions a student needed to redo. As a consequence, the massed practice group relied more on the check-answer function than the distributed practice group when they practiced again. This result explained why the transition from RE to CA was identified as significant.

During the daily lectures, we learned from the students that they liked putting questions that were beyond their understanding into the favorites list for subsequent review or turning to their classmates and the lecturer for help. The CQ → RE pair indicates that the massed practice group tended to seek help later for their mistakes, while the distributed practice group

Table 5. Frequency of the Behavioral Conversion (z Score) of the Distributed Group.

	NP	CA	CE	CQ	CM	LK	RE
NP	61.17*	12.06*	−12.59	−21.02	−17.34	−3.73	−62.53
CA	−2.51	−6.30	18.49*	−2.79	7.11*	−0.37	−9.24
CE	−18.73	−0.62	−5.38	35.83*	5.04*	−1.34	8.00*
CQ	−2.48	−8.41	−3.11	0.92	−0.99	−0.81	12.98*
CM	−12.79	−2.43	−2.43	1.05	41.68*	9.65*	−0.18
LK	−3.61	−0.42	−1.38	−0.86	6.35*	44.50*	0.41
RE	−57.88	−5.25	9.96*	0.15	−5.56	−1.04	78.43*

*$z > 1.96$.

Table 6. Frequency of the Behavioral Conversion (z Score) of the Massed Practice Group.

	NP	CA	CE	CQ	CM	LK	RE
NP	38.60*	6.39*	−5.38	−21.72	−17.24	−3.00	−43.75
CA	−11.75	−3.74	23.30*	−4.01	4.45*	0.83	−2.79
CE	−15.72	−3.79	−10.84	40.19*	9.59*	−0.83	8.70*
CQ	9.35*	−7.81	−7.40	−4.48	0.00	−0.49	4.55*
CM	−9.77	−2.02	−1.60	2.89*	33.77*	9.96*	1.72
LK	−2.04	−0.70	−0.85	−0.51	6.29*	27.91*	1.05
RE	−39.32	4.36*	4.33*	−3.75	−2.52	−0.66	65.69*

*$z > 1.96$.

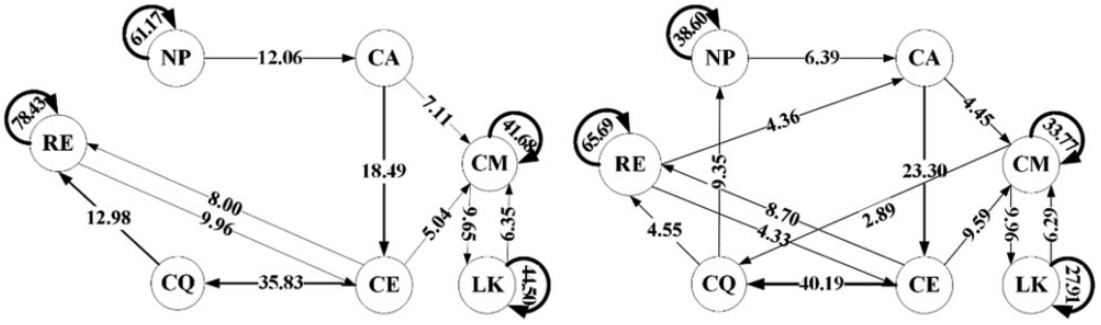

Figure 9. The behavior sequence diagrams of the distributed practice group (left) vs. the massed practice group (right).

liked to do questions again when they made mistakes, to deepen their understanding. For this reason, the two groups completed a different total number of questions.

The RE → CA pair shows that the students in the massed practice group tended to check their answers after completing the questions, and the lack of CE (checking the answer explanation) here indicates that they seldom tried to understand the question thoroughly since the answer explanation provided far more detailed information than the right answer.

Discussion

DQuiz was designed to facilitate novice programmers in engaging in distributed bite-sized practice sessions—that is, multiple-choice questions—to help them learn programming. To make the practice convenient, a mobile device app was developed so that the students could easily access the system to practice during suitable time slots. A practice calendar was implemented to help the students perform distributed practice. Additionally, a discussion board was implemented to promote student interactions. The results showed that the students were not engaged in posting on the discussion board, but the posting actions were highly correlated with their final academic performance. Similarly, the practice calendar was insufficient to convince the students to engage in distributed practice. They had to be encouraged by a teaching assistant. A comparison of the experimental and control groups showed that simply encouraging students to follow the distributed

practice strategy significantly improved their final exam grades. This finding is consistent with existing findings that distributed practice can improve students' learning outcomes (Benjamin & Tullis, 2010). However, readers should be aware of the experiment's context. All of the participating students were non-engineering novice programmers. Their major courses, such as Introduction to Psychology, were not as relevant to programming as those of engineering students. This status meant that their class schedules allowed them to spend only a very limited amount of time in programming-related practice. Therefore, practice on multiple-choice questions about basic syntax and algorithms was helpful for them. However, this kind of practice might not be as helpful for those who have already mastered one programming language or those who have plenty of opportunities to practice programming every week.

The results of this study are somewhat contrary to Moss's (1995) study, which found that massed practice could achieve better outcomes than distributed practice in mathematics learning. However, the ages of the participants in the two studies are different. Our research was aimed at college freshmen, while Moss's study was aimed at second- and fourth-grade students. Moss's study also found that the distributed practice group of fourth-graders performed better than the second-graders. As age increases, the distributed practice effect becomes increasingly obvious (Toppino & DeMesquita, 1984). Moreover, our experimental design was very different from that used in Moss's (1995) study. Moss spent 9 weeks while we spent 16 weeks practicing and examining students. Leppänen et al. (2016) discuss that the distributed effect might not be immediately applicable to the context of programming learning. Our results showed that the effect of distributed practice required some time to accumulate to be observed (see Figure 8 and Table 3). Therefore, 9 weeks may not have been long enough to make the distributed effect observable. Suzuki and DeKeyser (2015) found that the distributed effect was not clear in procedural knowledge learning. They attribute their findings to the complexity of practice but, at the same time, state that "a larger number of practice sessions may consolidate the memory more strongly over time particularly when the practice is distributed" (Suzuki & DeKeyser, 2015, p. 185). Given that, in our case, a significant difference in learning outcomes between the distributed and massed practice groups was observed after a large number of training sessions, their statement is somewhat confirmed by our study.

System log files were used to disclose the impact of distributed practice. The first important finding was that the distributed practice group had a significantly higher rate of first-check correctness on the practice questions than the massed practice group. This finding might be because distributed practice can help students eliminate mistakes (Underwood, 1961) and retain knowledge to give more correct responses (Tsao, 1951). Additionally, the distributed practice group spent more time on each practice question than the massed practice group, although this difference was insignificant. This result suggested that students tended to think more carefully when they had fewer practice items to do in one practice session.

One disappointment was that both of the groups generated a very limited number of postings after they finished answering the practice questions. The discussion board was designed to help the students create good interpersonal relationships and construct deep individual knowledge. Writing posts can cause students to reflect on their own learning, which can promote higher-order thinking processes such as analysis, synthesis, and evaluation (Newman et al., 1997). An explanation for the low level of usage was that most of the practice questions were not sufficiently challenging to elicit student interactions. However, the distributed practice group still posted significantly more than the massed practice group. This result suggested that the students in the distributed practice group were more engaged in the practice than the students in the massed practice group. Given that a significant correlation was found between the number of postings and the final exam grades, the difference in postings may have been another contributing factor.

Lag sequential analysis was used to discover further differences in the behavioral sequence patterns of the two groups. The massed practice group tended to abandon finding the correct answer even for the questions they had previously answered, but this pattern was not found in the distributed practice group. Another unique pattern in the massed practice group was that the students tended to favor the current question and start work on a new practice question. However, the distributed practice group always went back to practice the questions they had already done. The students usually put a question into their favorites list because they felt it was challenging to answer. Thus, it was better at that moment to practice the related questions again to consolidate the corresponding concepts, instead of looking at new practice questions. However, the students in the control group were probably on their way to completing all the unfinished questions, so they had no time to consolidate the corresponding concepts

In summary, because the massed practice group accumulated more questions in each practice session, the control group was more inclined than the

experimental group to rush to complete the practice questions, without carefully choosing the answers or posting on the discussion board. This action probably resulted in the control group's lower rate of first-check correctness and their lower final exam grades. Gerbier and Toppino (2015) claim that multiple memory paths can be formed through continuously extracting practice, which makes information acquisition easier. Each information extraction is a refactoring of knowledge, so it is meaningful learning rather than simple memorization (Aibao et al., 2013). The students in the experimental group were forced to extract knowledge components in a more distributed fashion than the students in the control group, which probably helped them strengthen the acquisition patterns of the corresponding knowledge components. Additionally, because the students in the control group gave up more easily than the students in the experimental group while practicing again, they were more likely to extract information from system feedback than from their minds, which was less helpful for knowledge acquisition.

In addition to the quantitative analysis, we conducted face-to-face interviews to collect the students' feedback on the design of DQuiz. The students gave positive feedback on the convenience of using the system—for example: "It is helpful for improving the ability of code reading, and it is convenient for practicing anytime and anywhere." The students could easily practice when they wanted to: "I will practice in my spare time; it is appropriate to practice two to three questions every day" or "I am accustomed to practicing twice a week, gradually." Our results showed that more of the students in the distributed practice group (93) completed more than 70% of all the questions than the students in the massed practice group (77). We think that the convenience of DQuiz and humans encouraging one another constructed a learning environment that helped the students to self-regulate themselves to conduct distributed practice (Auvinen, 2015). The students also provided some suggestions for further improvement—for example: "I cannot receive feedback immediately when I post some question or comments in the discussion board." The lack of this function probably stopped some students from actively participating in the discussion board.

Given the current findings, our next step is to find a way to automatically make students follow the effective strategy, so that the system can be easily scaled up. The low usage of the discussion board left us with concerns about its current design. Perhaps the system should have a central discussion board for all practice questions instead of separate ones for each question.

Conclusion

This research has shown that distributed practice with multiple-choice questions can help novices in their programming learning. With limited human intervention (encouraging students to practice using the app), the mobile app DQuiz effectively supported students in adopting a distributed practice strategy without increasing their after-class learning time. According to the lag sequential analysis of the students' learning behaviors, the distributed practice strategy reduced the learning load for students each time and enabled them to give more time and focus to each practice question. Furthermore, the students' behavioral sequences were more coherent in both learning new knowledge and reviewing previous sections. Coherence was considered a quicker retrieval of procedural knowledge (Suzuki & DeKeyser, 2015). This research has provided long-term evidence of authentic teaching and learning for the effect of the distributed practice strategy on basic knowledge and high-level capability acquisition in programming learning. In fact, intervals that are too short or too long could change the effect of the distributed practice strategy, and we will continue to explore the comparatively optimal length of the practice interval by using artificial intelligence technology in the future.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: We thank for its financial support from Beijing Education Science Plan Office (grant number CHEA2020024).

ORCID iD

Bo Yang https://orcid.org/0000-0001-9600-4265

References

Aibao, Z., Xiaofeng, M. A., Jing, L., & Dan, C. U. I. (2013). The advantage effect of retrieval practice on memory retention and transfer: Based on explanation of cognitive load theory. *Acta Psychologica Sinica*, *45*(8), 849–859. https://doi.org/10.3724/SP.J.1041.2013.00849

Alzaid, M., Trivedi, D., & Hsiao, I.-H. (2017, October 18–21). *The effects of bite-size distributed practices for programming novices* [Paper presentation]. IEEE Frontiers in Education Conference, Indianapolis, IN, United States. https://doi.org/10.1109/FIE.2017.8190593

Auvinen, T. (2015). *Educational technologies for supporting self-regulated learning in online learning environments*

[Doctoral dissertation, Aalto University]. Aaltodoc. https://aaltodoc.aalto.fi:443/handle/123456789/17235

Benjamin, A. S., & Tullis, J. (2010). What makes distributed practice effective? *Cognitive Psychology*, *61*(3), 228–247. https://doi.org/10.1016/j.cogpsych.2010.05.004

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *SIGCSE Bulletin*, *39*, 32–36.

Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. *ACM Inroads*, *10*(2), 30–36.

Bernstein, N. R. (1978). Observing behavior, vol. 1: Theory and applications in mental retardation; Observing behavior, vol. 2: Data collection and analysis methods. *American Journal of Psychiatry*, *135*(9), 1130–1130. https://doi.org/10.1176/ajp.135.9.1130

Bjotk, E., & Bjork, R. (2011). Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. In M. A. Gernsbacher, R. W. Pew, L. M. Hough, & J. R. Pomerantz (Eds.), *Psychology and the real world: Essays illustrating fundamental contributions to society* (pp. 56–64).

Budé, L., Imbos, T., Wiel, M. W. van de, & Berger, M. P. (2011). The effect of distributed practice on students' conceptual understanding of statistics. *Higher Education*, *62*(1), 69–79. https://doi.org/10.1007/s10734-010-9366-y

Butler, A. C., Karpicke, J. D., & Roediger, H. L. (2008). Correcting a metacognitive error: Feedback increases retention of low-confidence correct responses. *Journal of Experimental Psychology*, *34*(4), 918–928. https://doi.org/10.1037/0278-7393.34.4.918

Butler, A. C., & Roediger, H. L. (2008). Feedback enhances the positive effects and reduces the negative effects of multiple-choice testing. *Memory and Cognition*, *36*(3), 604–616. https://doi.org/10.3758/MC.36.3.604

Challis, B. H. (1993). Spacing effects on cued-memory tests depend on level of processing. *Journal of Experimental Psychology*, *19*(2), 389–396. https://doi.org/10.1037/0278-7393.19.2.389

Chi, M., Adams, J., Bogusch, E. B., Bruchok, C., Kang, S., Lancaster, M., Levy, R., Li, N., McEldoon, K. L., Stump, G. S., Wylie, R., Xu, D., & Yaghmourian, D. L. (2018). Translating the ICAP theory of cognitive engagement into practice. *Cognitive Science*, *42*(6), 1777–1832. https://doi.org/10.1111/cogs.12626

De Cecco, J.P., and W.R. Crawford (1974) The Psychology of Learning and Instruction. Prentice Hall, Englewood Cliffs, NJ.

Dempster, F. N. (1988). The spacing effect: A case study in the failure to apply the results of psychological research. *American Psychologist*, *43*(8), 627–634. https://doi.org/10.1037/0003-066X.43.8.627

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, *46*(3), 277–296. https://doi.org/10.1080/15391523.2014.888272

Dillenbourg, P. (2005). Designing biases that augment socio-cognitive interactions. In R. Bromme, F. W. Hesse, & H. Spada (Eds.), *Barriers and biases in computer-mediated knowledge communication: And how they may be overcome* (pp. 243–264). Springer. https://doi.org/10.1007/0-387-24319-4_11

Dunlosky, J., & Rawson, K. (2015). Practice tests, spaced practice, and successive relearning: Tips for classroom use and for guiding students' learning. *Scholarship of Teaching and Learning in Psychology*, *1*(1), 72–78. https://doi.org/10.1037/stl0000024

Eckerdal, A. (2009). *Novice programming students' learning of concepts and practise* [Doctoral dissertation, Uppsala University]. Semantic Scholar. https://www.semanticscholar.org/paper/Novice-Programming-Students%27-Learning-of-Concepts-Eckerdal/bccc7fb6b4080d8c01aedf2b0f701989f7b9841d

Gerbier, E., & Toppino, T. C. (2015). The effect of distributed practice: Neuroscience, cognition, and education. *Trends in Neuroscience and Education*, *4*(3), 49–59. https://doi.org/10.1016/j.tine.2015.01.001

Hoffman, D. M., Lu, M., & Pelton, T. (2011). A web-based generation and delivery system for active code reading. In *SIGCSE '11: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (pp. 483–488)*. Association for Computing Machinery. https://doi.org/10.1145/1953163.1953301

Hovemeyer, D., & Spacco, J. (2013). CloudCoder: A web-based programming exercise system. *Journal of Computing Sciences in Colleges*, *28*(3), 30–30.

Ihantola, P., Hellas, A., Butler, M., Börstler, J., Edwards, S., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., Rubio, M., Sheard, J., Skupas, B., Spacco, J., Szabo, C., & Toll, D. (2015, July 4). *Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies*. https://doi.org/10.1145/2858796.2858798

Karpicke, J. D., & Bauernschmidt, A. (2011). Spaced retrieval: Absolute spacing enhances learning regardless of relative spacing. *Journal of Experimental Psychology*, *37*(5), 1250–1257. https://doi.org/10.1037/a0023436

Kordaki, M. (2010). A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation. *Computers and Education*, *54*(1), 69–87. https://doi.org/10.1016/j.compedu.2009.07.012

Leppänen, L., Leinonen, J., & Hellas, A. (2016). *Pauses and spacing in learning to program*, 41–50. https://doi.org/10.1145/2999541.2999549

Litman, L., & Davachi, L. (2008). Distributed learning enhances relational memory consolidation. *Learning and Memory*, *15*(9), 711–716. https://doi.org/10.1101/lm.1132008

Luxton-Reilly, A. (2016). Learning to program is easy. In ITiCSE '16: Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (pp. 284–289). Association for Computing Machinery. https://doi.org/10.1145/2899415.2899432

Martin, J., Edwards, S., & Shaffer, C. (2015). The effects of procrastination interventions on programming project success. In *ICER '15: Proceedings of the 11th Annual International Conference on International Computing Education Research* (pp. 3–11). Association for

Computing Machinery. https://doi.org/10.1145/2787622.2787730

Moss, V. D. (1995). *The efficacy of massed versus distributed practice as a function of desired learning outcomes and grade level of the student* [Doctoral dissertation, Utah State University]. Utah State University Digital Commons. https://digitalcommons.usu.edu/etd/3552

Na, T., Funabiki, N., Zaw, K. K., Ishihara, N., Matsumoto, S., & Kao, W.-C. (2017). A fill-in-blank problem workbook for Java programming learning assistant system. *International Journal of Web Information Systems*, *13*(2), 140–154. https://doi.org/10.1108/IJWIS-12-2016-0076

Newman, D. R., Johnson, C., Webb, B., & Cochrane, C. (1997). Evaluating the quality of learning in computer supported co-operative learning. *Journal of the American Society for Information Science*, *48*(6), 484–495.

Prather, J., Becker, B. A., Craig, M., Denny, P., Loksa, D., & Margulieux, L. (2020). What do we think we are doing? Metacognition and self-regulation in programming. In *ICER '20: Proceedings of the 2020 ACM Conference on International Computing Education Research* (pp. 2–13). Association for Computing Machinery. https://doi.org/10.1145/3372782.3406263

Reardon, S., & Tangney, B. (2015, December). Smartphones, studio-based learning, and scaffolding: Helping novices learn to program. *ACM Transactions on Computing Education*, *14*(4), Article 23. https://doi.org/10.1145/2677089

Rohrer, D., & Taylor, K. (2006). The effects of overlearning and distributed practise on the retention of mathematics knowledge. *Applied Cognitive Psychology*, *20*(9), 1209–1224.

Shimoni, R. (2013). Addressing the needs of diverse distributed students. *International Review of Research in Open and Distance Learning*, *14*(3), 134–157.

Simon, Luxton-Reilly, A., Ajanovski, V. V., Fouh, E., Gonsalvez, C., Leinonen, J., Parkinson, J., Poole, M., & Thota, N. (2019). Pass rates in introductory programming and in other STEM disciplines. In *ITiCSE-WGR '19: Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education* (pp. 53–71). Association for Computing Machinery. https://doi.org/10.1145/3344429.3372502

Sobel, H., Wiseheart, M., & Kapler, I. (2011). Spacing effects in real-world classroom vocabulary learning. *Applied Cognitive Psychology*, *25*(5), 763–767. https://doi.org/10.1002/acp.1747

Soderstrom, N. C., Kerr, T. K., & Bjork, R. A. (2016). The critical importance of retrieval—and spacing—for learning. *Psychological Science*, *27*(2), 223–230. https://doi.org/10.1177/0956797615617778

Spacco, J., Denny, P., Richards, B., Babcock, D., Hovemeyer, D., Moscola, J., & Duvall, R. (2015). Analyzing student work patterns using programming exercise data. In *SIGCSE '15: Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 18–23). Association for Computing Machinery. https://doi.org/10.1145/2676723.2677297

Spieler, B., Grandl, M., Ebner, M., & Slany, W. (2019). *"Computer science for all": Concepts to engage teenagers and non-CS students in technology*. Cornell University. http://arxiv.org/abs/1908.06637

Suzuki, Y., & DeKeyser, R. (2015). Effects of distributed practice on the proceduralization of morphology. *Language Teaching Research*, *21*(2), 166–188. https://doi.org/10.1177/1362168815617334

Tom, M. (2015). Five C framework: A student-centered approach for teaching programming courses to students with diverse disciplinary background. *Journal of Learning Design*, *8*(1), 21–37. https://doi.org/10.5204/jld.v8i1.193

Toppino, T. C., & DeMesquita, M. (1984). Effects of spacing repetitions on children's memory. *Journal of Experimental Child Psychology*, *37*(3), 637–648. https://doi.org/10.1016/0022-0965(84)90081-X

Tsai, C.-Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior*, *95*, 224–232. https://doi.org/10.1016/j.chb.2018.11.038

Tsao, J. C. (1951). An analysis of the achievements in spaced and massed practice: A study of the inhibition theory. *Journal of General Psychology*, *44*(2), 189–197. https://doi.org/10.1080/00221309.1951.9710070

Underwood, B. J. (1961). Ten years of massed practice on distributed practice. *Psychological Review*, *68*(4), 229–247. https://doi.org/10.1037/h0047516

Vihavainen, A., Airaksinen, J., & Watson, C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. In *ICER '14: Proceedings of the 10th Annual Conference on International Computing Education Research* (pp. 19–26). Association for Computing Machinery. https://doi.org/10.1145/2632320.2632349

Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. In *ITiCSE '14: Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education* (pp. 39–44). Association for Computing Machinery. https://doi.org/10.1145/2591708.2591749

Xue, G., Dong, Q., Chen, C., Lu, Z.-L., Mumford, J. A., & Poldrack, R. A. (2013). Complementary role of frontoparietal activity and cortical pattern similarity in successful episodic memory encoding. *Cerebral Cortex*, *23*(7), 1562–1571. https://doi.org/10.1093/cercor/bhs143

Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. *ACM Inroads*, *2(4)*. https://doi.org/10.1145/2038876.2038894

Yang, T.-C., Hwang, G.-J., Yang, S. J. H., & Hwang, G.-H. (2015). A two-tier test-based approach to improving students' computer-programming skills in a web-based learning environment. *Journal of Educational Technology and Society*, *18*(1), 198–210. https://www.jstor.org/stable/jeductechsoci.18.1.198

Yukselturk, E., & Altiok, S. (2016). An investigation of the effects of programming with scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming: Programming with scratch.

British Journal of Educational Technology, *48*(3), 789–801. https://doi.org/10.1111/bjet.12453

Zhang, L., Li, B., Zhang, Q., & Hsiao, I. H. (2020). Does a distributed practice strategy for multiple choice questions help novices learn programming? *International Journal of Emerging Technologies in Learning*, *15*(18), 234–250.

Zhang, L., Li, B., Zhou, Y., & Chen, L. (2019). Can fragmentation learning promote students' deep learning in C programming? In M. Chang, E. Popescu, Kinshuk, N.-S. Chen, M. Jemni, R. Huang, J. M. Spector, & D. G. Sampson (Eds.), *Foundations and trends in smart learning* (pp. 51–60). Springer. https://doi.org/10.1007/978-981-13-6908-7_7

Zingaro, D., Cherenkova, Y., Karpova, O., & Petersen, A. (2013). Facilitating code-writing in PI classes. In *SIGCSE '13: Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (pp. 585–590). Association for Computing Machinery. https://doi.org/10.1145/2445196.2445369