A Spaced, Interleaved Retrieval Practice Tool that is Motivating and Effective

Iman YeckehZaare
University of Michigan
School of Information
Ann Arbor, MI
oneweb@umich.edu

Paul Resnick
University of Michigan
School of Information
Ann Arbor, MI
presnick@umich.edu

Barbara Ericson
University of Michigan
School of Information
Ann Arbor, MI
barbarer@umich.edu

ABSTRACT

Retrieval practice, spacing, and interleaving are known to enhance long-term learning and transfer, but reduce short-term performance. It can be difficult to get both students and instructors to use these techniques since they perceive them as impeding initial student learning. We leveraged user experience design and research techniques, including survey and participant observation, to improve the design of a practice tool during a semester of use in a large introductory Python programming course. In this paper, we describe the design features that made the tool effective for learning as well as motivating. These include requiring spacing by giving credit for each day that a student answered a minimum number of questions, adapting a spaced repetition algorithm to schedule topics rather than specific questions, providing a visual representation of the evolving schedule in order to support meta-cognition, and providing several gameful design elements. To assess effectiveness, we estimated a regression model: each hour spent using the practice tool over the course of a semester was associated with an increase in final exam grades of 1.04%, even after controlling for many potential confounds. To assess motivation, we report on the amount of practice tool use: 62 of the 193 students (32%) voluntarily used the tool more than the required 45 days. This provides evidence that the design of the tool successfully overcame the typically negative perceptions of retrieval practice, spacing, and interleaving.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Human-centered computing** → *Interactive systems and tools.*

KEYWORDS

desirable difficulties, spacing, interleaving, retrieval practice, procrastination, speed, gameful design, introductory programming

ACM Reference Format:
Iman YeckehZaare, Paul Resnick, and Barbara Ericson. 2019. A Spaced, Interleaved Retrieval Practice Tool that is Motivating and Effective. In *International Computing Education Research Conference (ICER '19), August 12–14, 2019, Toronto, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3291279.3339411

ICER '19, August 12–14, 2019, Toronto, ON, Canada

ACM ISBN 978-1-4503-6185-9/19/08…$15.00
https://doi.org/10.1145/3291279.3339411

1 INTRODUCTION

Our beliefs regarding the ways that we learn are often flawed. They are informed by subjective impressions rather than learning studies. For example, even though 90% of college students performed better after spaced practice (practicing over several days) than massed practice (studying the night before an exam), 72% of the students reported that massed practice was more effective [19]. Bjork and Soderstrom found that individuals' incorrect beliefs lead them to manage their learning or instruct others in less than ideal ways [8, 9, 26]. They have conducted numerous experimental studies that demonstrate that the conditions of instruction that accelerate initial learning performance often fail to support long-term learning and transfer. Conversely, some interventions that appear to create difficulties for the learner during initial knowledge acquisition, often optimize long-term learning and transfer [18]. Bork has labeled these "desirable difficulties" [8]. Soderstrom and Bjork [4, 8, 9, 26] have identified the following techniques that improve long-term learning, but negatively impact initial learning performance:

(1) Retrieval practice (vs. passive review of previous content)
(2) Spacing (vs. massing) study/practice
(3) Interleaving (vs. blocking) distinct topics to study/practice
(4) Generating (vs. being exposed to) the learning content
(5) Varying (vs. keeping steady) the environmental context

We created a learner-centered retrieval practice tool for an introductory Python programming course that spaces practice and interleaves topics. We used the tool in a semester-long course with 193 undergraduate students. This allowed us to observe variations in the tool usage and how it correlated with final exam grades. We followed a design-based research approach, iteratively improving the tool during the semester in response to observing students using the tool, anonymous feedback, and usage data.

We designed the tool to provide motivators to reduce the typical negative student reaction to desirable difficulties. We also devised a grading scheme to motivate students to space their practice out over the semester, while still giving students significant autonomy in when and how much they used the practice tool.

We hypothesized that the grading incentives and the features designed to motivate students might lead students to use the tool throughout the semester, not just to cram for exams. Our first research question is:

- **RQ1**: Do students space or mass their use of the tool?

Because the course and design included both intrinsic and extrinsic motivators, we cannot fully tease apart their separate effects. However, any usage that did not help students earn points towards course grades is an indication of intrinsic motivation. Thus, our second research questions is:

- **RQ2**: Do students use the practice tool more than they were required to?

Finally, we analyzed the overall effectiveness of the practice tool with respect to performance on the final exam:

- **RQ3**: Does the practice tool usage correlate with higher final exam grades?

The paper begins with a summary of the theories of desirable difficulties, self-determination theory, and gameful design that inspired the development of our tool, as well as prior research using desirable difficulties in computing education, research on other practice systems for introductory programming, and research on procrastination. Next, we describe the setting of the study and the specific features of the tool and the grading system, including the evolution of some of the features over the course of a semester-long deployment. Then, we describe the data sources and analysis methods for our quantitative analysis of the research questions and report the results of those analyses. Finally, we discuss the limitations of this study and our conclusions.

2 RELATED WORK

This work is inspired by work on desirable difficulties, self-determination theory, gameful design, practice tools, and procrastination.

2.1 Desirable Difficulties

Many experiments show that desirable difficulties result in superior long-term learning, but may negatively affect short-term learning performance. Soderstrom and Bjork [26] provide a comprehensive review of these experiments. In this study, we report the design of our practice tool around three of these techniques: spacing, interleaving, and retrieval practice.

2.1.1 Spaced Practice. Spreading out practice sessions over a period of time, usually over many days, is called spaced practice. It can be contrasted with massed practice, which is "cramming" just before an exam. Massed practice is successful in the short-term, but it leads to very rapid forgetting. In contrast, spaced practice negatively affects short-term performance, but significantly improves long-term learning [6, 10, 11, 14, 20, 24, 26].

2.1.2 Interleaving. Intermixing different topics in a practice session is called interleaving [14]. In contrast, blocking means practicing the same topic. Students often prefer blocked practice because it provides a sense of fluency while interleaving leaves a sense of confusion [30]. Blocked practice is also pervasive. Textbooks only cover one topic at a time [18] and the end of chapter exercises focus on that topic. Most curricula are designed with problem sets covering only one topic, with no questions from earlier chapters.

2.1.3 Retrieval practice (testing). Practice which forces the learner to retrieve the information from memory is called retrieval practice or testing. An example would be flashcards where the student sees a question on one side and has to recall the answer on the other side. It can be contrasted with studying by rereading a section of a textbook. Retrieval practice helps modify memory which makes the information more recallable. Testing also provides better feedback as to what has or has not been learned than rereading material [26].

2.2 Self-determination Theory (SDT)

SDT characterizes two main types of motivation: intrinsic and extrinsic [25]. Intrinsic motivation is driven by internal rewards (enjoyment of a subject) rather than external (e.g., grades). SDT identifies autonomy, or letting someone choose how and when to perform a task, as the most important factor for increasing intrinsic motivation. SDT advises providing students with more autonomy rather than external rewards [25]. In our design, students decide when, where, and how they use the practice tool, which may support their intrinsic motivation.

2.3 Gameful Design

Gameful design is the process of redesigning core elements of a learning environment to support intrinsic motivation [1]. Providing students with extrinsic motivation by earning grades or rewards can reduce their intrinsic motivation [22]. Through three studies, Aguilar et al. [1] found a positive correlation between gameful design and students spending more time studying and feeling more in control. Gameful design "requires simultaneously increasing the opportunities for students to have autonomy and mitigating the impact of failure, such that learners are empowered to exert effort in spaces that they might otherwise have avoided" [1].

2.4 Prior Work on Practice Tools

Several practice tools have been created for helping students learn to program. CodeLab is a commercial tool that instructors can use for programming assignments [7]. Epplets [21] is a tool for solving mixed-up code (Parsons) problems [23]. Problem Roulette asks random questions from previous exams [15], but students choose the topics. None of these tools provide automated support for spaced and interleaved retrieval practice.

Intelligent Tutoring Systems (ITS) provide personalized practice [2]. These systems provide feedback either after the student has solved a problem or while the student is solving a problem [13]. Some systems select the next task or problem based on a student model that indicates if the student has mastered the current topic [13]. ITS have been used successfully in programming [3]. However, they typically support initial learning and blocked practice and do not include support for spaced or interleaved practice.

2.5 Procrastination

Multiple studies [16, 27, 28] have reported a significant negative correlation between procrastination and course grades. Kazerouni et al. [17] analyzed a dataset of 6.3 million program edits and software tests from a programming course and found a significant negative correlation between procrastination and programming project grades. In a retrieval based practice tool, Problem Roulette, students primarily used the tool just before exams as shown in Figure 4, which indicates procrastination.

3 SYSTEM DESIGN AND DEVELOPMENT

We designed a practice tool for an introductory Python programming course for informatics and non-CS majors. Most, but not all, of these students have no prior programming experience. The course serves as a pre-requisite for an upper-division major that involves additional programming courses, but much less programming than is typical for computer science majors.

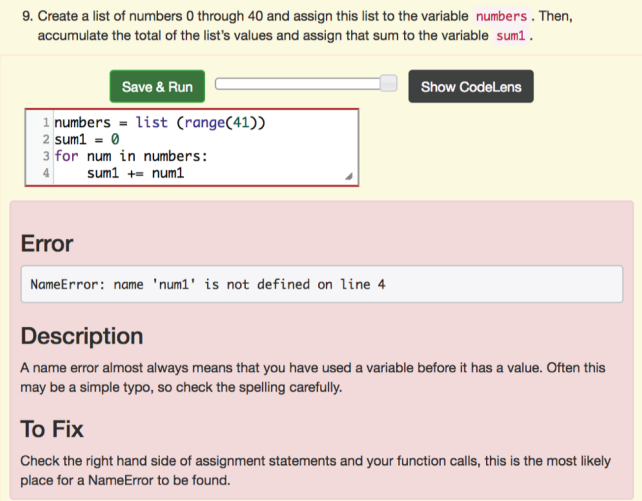
The course pedagogy is based on active learning. Students are expected to study specific sections of an interactive e-book prior to attending lectures. Both lectures and discussions involve frequent opportunities for students to answer questions and interact with each other. Furthermore, the e-book itself provides many interaction opportunities, as described in subsection 3.1. Grades are assigned on an absolute scale rather than on a curve. Students accumulate points throughout the semester from problem sets and other activities. The final exam was worth 25% of the total course points. The practice tool was worth 5%; how students earned those points is described in subsection 3.3 and 3.6.

3.1 Underlying Platform

The practice tool was incorporated into Runestone Interactive[1], which is an open-source platform that enables authors to create e-books. Instructors can assemble assignments from the interactive elements. An e-book page can include text, images, videos, and external links. Pages can also include interactive elements, such as:

- **Multiple-choice** questions, appropriate for quick practice;
- **Fill-in-the-blank** questions;
- **Parsons** (mixed-up) code questions, where students have to rearrange code blocks in the correct order and with the correct indentation [23];
- **Codelens** questions, which allow step-by-step code execution and visualization of variables;
- **Active-code** questions, where students can write programs and edit code (Figure 1).

Figure 1: Runestone Interactive Sample Active Code



[1]https://runestone.academy/runestone/static/overview/overview.html

The interactive elements provide students with immediate feedback. For example, the active-code questions can include unit tests which test submitted solutions. Assignments can be created using these interactive elements. The practice tool reuses Runestone's database of questions. Students could access the practice tool through a direct link from Canvas, the learning management system used to organize the course materials, or through a menu link on each Runestone textbook page.

3.2 Basic Design: Retrieval Practice

The practice tool presents a web page with a single question on it, either multiple-choice, fill-in-the-blank, Parsons, or ActiveCode, as illustrated in Figure 2. When a question is presented in the practice tool, the student's previous answers are hidden, unlike presentations of those same questions elsewhere in the textbook, where students can access their past history of answers.

3.3 Interleaving Algorithm

Each time the practice page was loaded, the system automatically chose which question to display. We modified the SuperMemo 2 algorithm [29], which automatically interleaves topics, repeating topics less and less frequently as the student demonstrates mastery of them. It optimizes learning by reducing forgetting and overlearning (restudying what you have already mastered). Supermemo 2 defines three factors that get updated for question y when student x answers it. These factors are:

- $i\_interval_{xy}$: demonstrates the number of days remaining to ask question y from student x;
- e_factor_{xy}: indicates how easy question y is for student x.
- q_{xy}: measures the correctness of student x's answer to question y. Supermemo 2 updates the other two factors based on the value measured for q_{xy}, which is an integer from 1 to 5.

We customized Supermemo 2 in two ways to make it appropriate for use in our practice tool:

- SuperMemo 2 was designed for language learning, in which it makes sense to space repetitions of retrieving the meaning of each word. However, when it comes to STEM courses, it becomes boring to repeatedly answer the same question about a formula or a programming algorithm. We wanted students to learn generalizable skills, rather than just memorizing the answer to a specific question. Therefore, instead of applying the algorithm to individual questions, we applied it to topics, with each topic potentially having many questions. When the algorithm suggests answering a question from any specific topic, the practice tool asks one of the questions for that topic using Round-robin scheduling [5].
- While Supermemo 2 sets q_{xy} based only on whether a student gets the question right, we use multiple factors:
  - d_{xy}: demonstrates the duration of the time student x spent answering question y, measured in minutes.
  - t_{xy}: indicates the number of attempts student x made before submitting their final answer for question y.
  - v_{xy}: shows the number of other textbook pages viewed while answering question y; more page views indicate that student x needed to look up information rather than recalling it.

Over the first two weeks of the semester, we monitored each of these measures based on the students' answers to practice questions and iteratively modified our algorithm. Eventually, we settled on the following:

(1) $q_{xy} = 5$ if $v_{xy} == 0$, $t_{xy} <= 1$, and $d_{xy} <= 2$;
(2) $q_{xy} = 4$ if $t_{xy} <= 2$ and $d_{xy} <= 2$;
(3) $q_{xy} = 3$ if $t_{xy} <= 3$ and $d_{xy} <= 3$;
(4) $q_{xy} = 2$ if $t_{xy} <= 4$ and $d_{xy} <= 4$;
(5) $q_{xy} = 1$ if correct, but other conditions were not met;
(6) $q_{xy} = 0$ if the final answer was incorrect.

3.4 Grading System: per Day, not per Question

A student could earn one point for each day of completing a set of practice questions. Beyond that, they could keep answering questions, but would not earn any additional points that day. A student could earn a point for a maximum of 45 days during the semester; beyond that, they could continue to use the tool, but would not earn additional course points. Students started using the tool about three weeks into the semester, so they needed to complete a practice set about four days per week in order to earn all of the available points. This system was intended to create an incentive for students to space their use of the practice tool out over the semester, rather than only to cram for exams.

On each question, a student could make unlimited attempts, getting automated feedback each time. If a student submitted a wrong answer, the system asked the corresponding question again until they submitted a correct answer. However, during participant observations with students early in the semester, we discovered that this feature could leave a student "stuck" on a question that they could not answer, which was frustrating for students. In response to that, we introduced an escape valve, the option to postpone a question to the next day (by clicking the red button on the bottom part of Figure 2). That allowed them to ask their peers or instructors about the question before it was asked again. Postponed questions were asked again the next day that the students used the tool, though the students could postpone them again if they wanted.

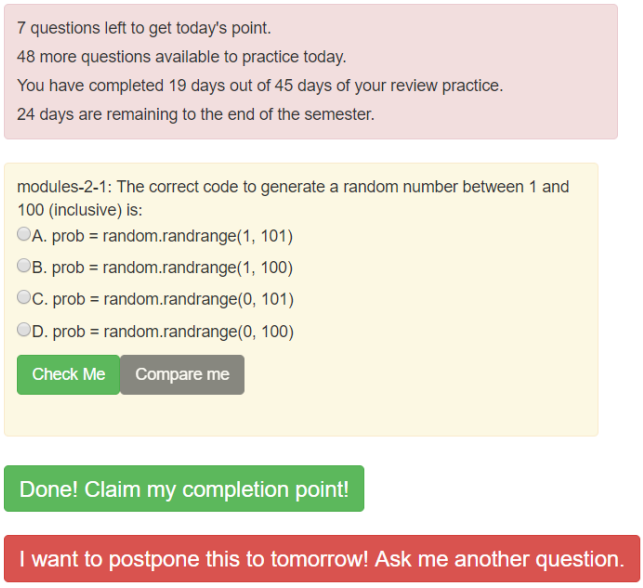
Initially, students were required to answer all of the available questions in order to earn that day's point. However, we found that if a student got behind in using the practice tool, they might have dozens of questions to answer, and the task seemed insurmountable. We changed the daily goal to be the minimum of either ten questions or all of the questions that were available that day.

3.5 Progress Tracking and Celebration

In the initial deployment, students did not receive any feedback about their progress towards the daily goal. Only after completing the daily goal did the page confirm that they had completed it. So, students asked us to add information to the practice tool including the number of questions they needed to answer to gain that day's point, the number of questions that were available to answer which is the same or more than the required number, the number of days that they needed to practice to gain the maximum points, and the number of days remaining in the semester. In response, we added a section to the top of the practice page, as shown in Figure 2.

We also implemented an animated fireworks feature to celebrate students' completion of the daily goal. Students who finished all the available questions for a day were shown extended fireworks. Following SDT, these features were intended to maintain students' motivation and autonomy while building their sense of competence, as they could see their progress toward the daily goal and anticipated a celebration when they finished. In group office hours soon after introducing the fireworks, we saw students' faces light up. They reported that it was very motivating. However, this enthusiasm seemed to wane as the novelty wore off later in the semester.

Figure 2: The Practice Tool Sample Interface



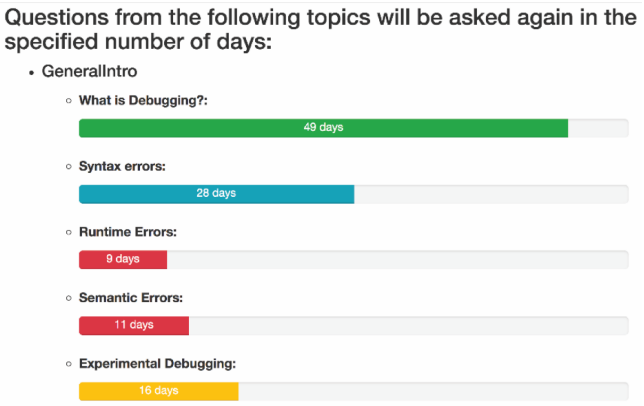
3.6 Schedule Information to aid Metacognition

From the beginning of the semester to spring break, some students only used the practice tool a couple of times and some students did not use it at all. To investigate the reason, we interviewed some of the students who used it only a few times. They explained that they thought the practice tool asked them random questions. One explicitly told us: "I don't like to answer random questions. I prefer to go through the pages of the textbook and only practice those questions that I think I may have difficulty with." To solve this issue, we developed a visualization of the practice tool's predictions of the number of days remaining before they would be asked a question about each topic in the syllabus. We added the visualization below the display of the question in the practice tool interface (Figure 3). In addition, in one of the lectures the instructor explained this visualization and gave an overview of how the algorithm predicts the student's optimal review day for each topic. As shown in Figure 5, the total number of questions practiced on every single day shows a decreasing trend from the beginning of the semester to spring break. However, introducing the schedule visualization after spring break, we see the trend becomes more level, which suggests a positive effect of the visualization on students' practice.

4 ANALYSIS AND RESULTS

We deployed the practice tool in a course running from January to April 2018. To answer the three research questions, we assembled

Figure 3: Schedule visualization example, only showing the first five topics. For each topic, there is an indicator of how many days until a question will be asked on that topic. The color indicates the student's mastery of the topic. Green indicates the highest mastery and red is the lowest.



usage data from the practice tool and other parts of the interactive e-book used in the course, students' final exam scores, and information about the students provided by the registrar's office, such as GPA and demographics.[2]

4.1 Student Demographics

- **Gender**: 85 (44.04%) male; 108 (55.96%) female.
- **Year**: 75 frosh; 52 sophomore; 28 junior; 38 senior.
- **Native language**: 150 (77.77%) English; 43 (22.28%) other.
- **Race/Ethnicity**: 120 (62.18%) White, 48 (24.87%) Asian, and 25 (12.95%) other.
- **Math score**: university-wide test (only 184 students) Mean (18.84), SD (4.41), Range [5.00, 25.00], Quartiles {16.00, 19.00, 23.00}.
- **Pass/Fail**: 168 (87.05%) took the course for a grade and 25 (12.95%) took it for pass/fail.
- **GPA**: cumulative GPA excluding this course: Mean (3.42), SD (0.36), Range [2.36, 4.01], Quartiles {3.22, 3.45, 3.68}.

4.2 RQ1: Do students space or mass their use of the tool?

Previous studies reported that students massed their use of practice tools. Figure 4 shows the activity log reported by "Problem roulette" [15], which indicates a low usage rate over the semester and spikes on the days before each of the four exams. In contrast, our usage log showed usage throughout the semester (Figure 5).

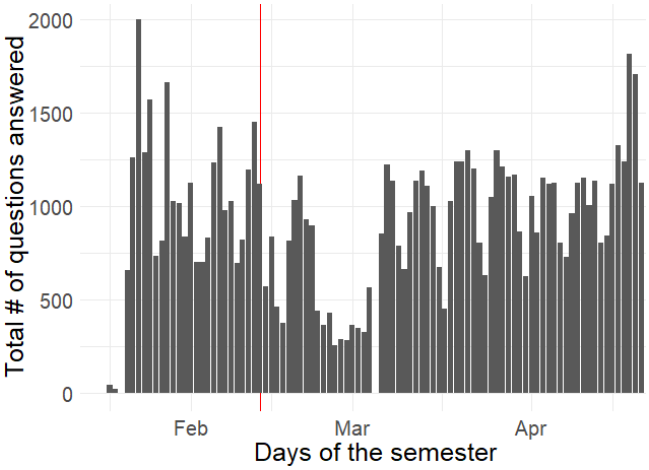
4.3 RQ2: Do students use the practice tool more than they were required to?

For RQ2, our first analysis examines if students kept answering questions on individual days even after earning the point for that day. Students had to practice only a total of 45 out of 81 days to earn the maximum points. So, we consider only days when students viewed the practice tool, and classify each of those practice days as "incomplete" (student did not earn the point), "stopped" (the student

[2]This study is exempted under IRB HUM00144387.

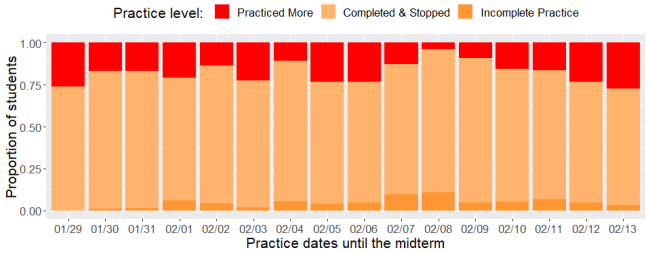
Figure 4: Number of visitors to the Problem roulette website per day over Fall 2012, which indicates low usage over the semester, but spikes on the days before each exam [15].

Figure 5: Total number of questions answered by the students each day. The red line indicates the midterm. At the end of February and early March, there was a week-long school holiday. On March 5th our server went down and the students were not able to access the practice tool.



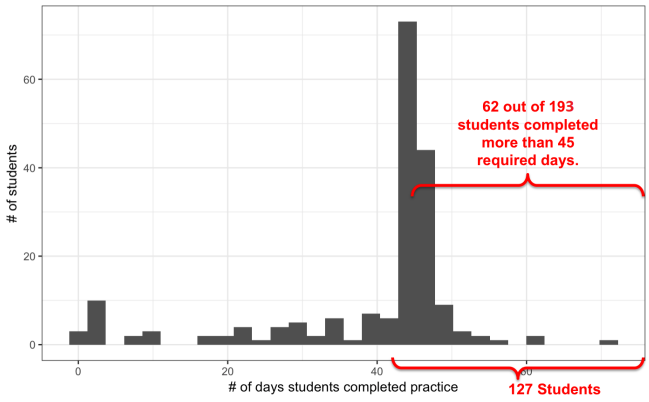
answered exactly the number needed to gain the point for that day) or "more" (the student answered at least one more question after earning the point for the day). Figure 6 shows the frequency of each of those outcomes for the two weeks leading up to the midterm. Over the entire semester, 5.16% of all student practice days were incomplete, 86.99% were stopped immediately after completing the required number of questions, and 7.86% concluded with the students voluntarily completing extra practice questions.

Figure 6: In two weeks before the midterm, among those students who viewed the practice tool each day, the proportion who had incomplete practice, stopped after earning the day's point, or practiced more after earning the point.



Our second analysis examines whether students completed more than 45 days of practice, the maximum that they could earn points

Figure 7: Histogram of the number of days each student completed practicing the required number of questions.



for. Figure 7 depicts a histogram showing the number of students who completed different numbers of days of practice. In this class, 127 (65.8%) out of 193 students completed at least 45 days of practice and received the maximum points. Among those 127 students, nearly half of them, 62, completed at least one additional day even though they did not receive any extra points for it. That means nearly one-third of the class used the practice tool more than the required number of days.

4.4 RQ3: Does the practice tool usage correlate with higher final exam grades?

Ideally, we would like to estimate the causal effect of using the practice tool on exam grades. It is not possible, however, to do so with confidence given the within-classroom design of this study, with students self-selecting how much to use the practice tool. Even if there is an observed correlation between the two, there could be many unmeasured confounders. In particular, it is likely that students with better general study habits would both be likely to use the practice tool more and likely to do better on the final exam even if the practice tool were unavailable.

The best that we can do is control for the many potential confounds that we *can* measure. We conducted an Ordinary Least Squares (OLS) regression analysis with the final exam score as the dependent measure. In addition to the number of hours of practice tool use, we included independent variables for several potential confounds, including some that may be proxies for general study habits, such as GPA, hours spent using the textbook outside of the practice feature, and a measure of students' tendency to procrastinate on assignments.

Some of the variables come from the student demographics provided by the registrar's office, as described in subsection 4.1. Others come from usage logs, as described below.

Hours of Using the Practice Tool. We collected the timestamps when the student was presented each practice question and when they submitted a final answer. We recorded the duration of practicing that question as the maximum of the elapsed time or ten minutes, and also treated the session as ten minutes long if the student never submitted a final answer. This leads to an overestimate of the length of sessions when the student loaded the page and then closed it, or took a long break and then answered the question quickly; it underestimates the length of sessions when the student genuinely worked on a question for more than ten minutes. We calculated the total number of hours the student spent using the practice tool by summing the duration of all of these practice sessions over the semester. The distribution of this measure was: Mean (8.68), SD (5.21), Range [0.00, 23.35], Quartiles {5.05, 8.07, 12.10}.

Hours of Studying the E-book. We started the timer for an e-book "session" with each page load. We marked the end of a session when another page was loaded, or after five minutes without a logged interaction on the page. Thus, our measure of session length probably slightly overestimates the length. We then added up the hours of studying the e-book for all of the sessions for each student over the semester, excluding the pages for the problem sets. The distribution of this measure was: Mean (37.49), SD (16.20), Range [3.00, 91.12], Quartiles {26.68, 35.20, 46.82}.

Speed. To get a measure of prior skill, we measured the students' speed at working on problem sets early in the course. The first five weekly problem sets were completed in the e-book, where we were able to measure the amount of time students spent. We define a student's speed as the number of points earned per hour of working on problem sets. The distribution of speed was: Mean (13.32), SD (6.18), Range [4.13, 43.19], Quartiles {8.82, 11.80, 15.93}.

Earliness. To get a measure of general study habits, we evaluate students' tendency to procrastinate, or rather its inverse, their earliness in working on problem sets. We encouraged students to submit by a soft deadline of Friday afternoons but offered a two-day grace period. We calculated the average number of hours from every interaction with any of the problem set questions to the corresponding problem set deadline. Students who procrastinate will have a low score on this measure, whereas students who complete the majority of the work early will have a high score. This measure was computed only for the first five problem sets, where actions were recorded in Runestone. After this period, the course switched to using Jupyter notebook for problem sets and the log files were not available. The distribution of earliness was: Mean (58.20), SD (30.93), Range [2.27, 175.42], Quartiles {36.92, 50.93, 73.38}.

We released each problem set approximately a week before the deadline. The mean of 58.20 was a little more than two days before the hard deadline and ten hours before the soft deadline; note that this was the average timestamp for all work, not for the last work that each student completed on each problem set (the turn-in time).

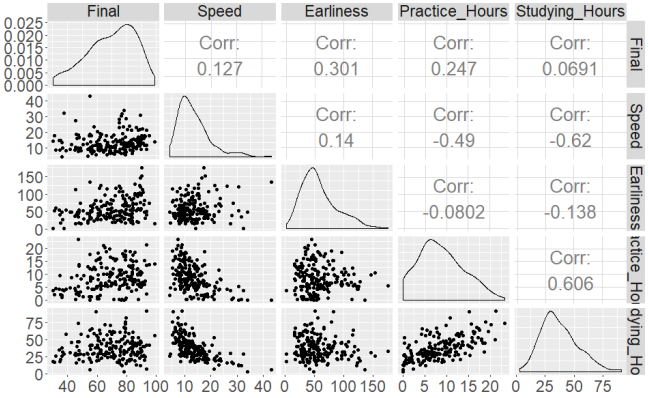
Final exam grade. The final was a pencil and paper exam, with multiple choice, short answer, and code writing questions. The distribution of exam grades was: Mean (69.65%), SD (15.87%), Range [30.20%, 98.80%], Quartiles {59.50%, 71.60%, 82.80%}.

4.5 Distributions and Correlations

Figure 8 presents distributions for the key variables and their correlations. The diagonal entries show the distributions for individual variables. The upper right cells summarize pairwise correlations as Pearson correlation coefficients, while corresponding cells on the lower left side provide scatterplots. Some of the interesting correlations include:

- Answering problem sets faster (higher speed) significantly correlated with less studying (r=-.62) and practicing (r=-.49), but not higher final exam grades (r=.127).
- Earliness (less procrastination) correlated with final exam grades (r=.301) but not speed (r=.14) or studying hours (r=-.138).
- More studying correlated with more practicing (r=0.606).

Figure 8: Distributions of the key variables and their correlations. The rows and columns are labeled by the corresponding variable names.



4.5.1 Regression Analysis. Table 1 shows the coefficients of the Ordinary Least Squares (OLS) regression model, with the final exam percentage as the dependent variable. It shows the results for 181 students. Twelve of the students were excluded from this model because we did not have all of their data. Several independent variables were categorical: the base (left out) or comparison categories were White, female, native English speakers, and taking the course for a grade. The results show that, after controlling for other factors:

- Each extra hour using the practice tool was associated with a 1.04% increase in the final exam score.
- Each extra hour using the e-book, excluding the practice tool usage, had no significant residual correlation with exam scores.
- As expected, measures of prior preparation (GPA, Math score) were positively correlated with exam score, as was earliness.
- The effect of speed on early problem sets was non-significant after controlling for other covariates.

We note that the positive coefficient for gender may be misleading. Overall, exam scores did not differ significantly by gender. However, when controlling for all the other covariates, males performed better. In another study [31], we identified three pathways correlated with success in this course: studying more, speed in completing the problem sets, and not procrastinating. We also found that male and female students achieved similar final exam grades, but female students got higher grades through studying more, and male students through completing problem sets faster. No gender difference in procrastination on problem sets was observed.

4.6 End of Course Survey

To examine students' perceptions about the practice tool and its features, we asked them to fill out a survey near the end of the course. To encourage honest feedback, there were no incentives for completing it and responses were anonymous. Eighty-three (43%) of the 193 students filled out the survey. The survey contained 20 questions on what the students found helpful, what could be improved, and student experiences with other practice tools.

	Effect on Final Grade	Std. Error
(Intercept)	−10.30	(11.17)
Practice Hours	1.04***	(0.25)
Studying Hours	0.05	(0.09)
Speed	0.36	(0.22)
Earliness	0.11***	(0.03)
GPA	10.34**	(3.18)
Math Score	0.91***	(0.24)
Pass/Fail (vs. Graded)	−7.93	(4.27)
Male (vs. Female)	5.17*	(2.17)
Junior (vs. Freshman)	9.11**	(3.00)
Senior (vs. Freshman)	10.28**	(3.72)
Sophomore (vs. Freshman)	5.87*	(2.41)
Asian (vs. White)	3.79	(2.38)
Other Ethn. (vs. White)	0.58	(2.95)
Non-native English	−2.04	(2.56)
Adj. R^2	0.39	
Num. obs.	181	
RMSE	12.60	

$^{***}p < 0.001, ^{**}p < 0.01, ^{*}p < 0.05$

Table 1: OLS regression results.

4.6.1 Learning. Students were positive about the practice tool and its effect on their learning. One student wrote, "Overall, the practice tool had helped me tremendously and I wouldn't have done this well in the course if there was no such tool." Another one wrote, "I am definitely stronger in the things I learned on the practice feature." They recognized the importance of practice. "I think classes like these require a lot of practice and then the concepts become easy, but most of the time there aren't enough resources to do so. This practice tool equipped us with ample amounts of practice!"

4.6.2 Spacing Algorithm. Student responses suggest that students found the spacing algorithm useful. "I also like that it waits a certain number of days before asking you questions about specific concepts after you complete them the first time." Students also recognized the usefulness of revisiting previous topics. "The repetitive structure of the practice tool allowed me to reinforce older topics."

4.6.3 Spacing Feature. More importantly, as opposed to previous studies that report students' dislike of the desirable difficulties, many students in our study found our design of spacing helpful. They were particularly interested in how they are incentivized to space their practice. "Having exposure to python every day is key to learning how to code. It's not like other things where you can just do work once a week." The other student wrote: "after I started getting used to doing the questions everyday I really wanted to keep up with it." Other students noticed how it helps them get more fluent in programming: "I like that it's a way for us to make sure we do a little work every day and keep concepts fresh in our minds. I find that helpful to keep me on top of things." The other one wrote:

"I enjoyed the coding questions and having to use it around every other day, it helped a lot with fluency and problem solving."

4.6.4 Interleaving Feature. Most studies on interleaving have notably reported that students prefer blocking. The way we modified SuperMemo 2 algorithm based on topics made interleaving enjoyable for students. They especially liked "the variety of different questions, because it helps remind me of/relearn past topics" Another student wrote, "The rotation of questions was extremely helpful." The other one noted, "I enjoyed the range of topics covered by the practice feature throughout the semester, especially those that would have potentially could have been forgotten if not practiced." They also mentioned, "It also kept me thinking about the smaller topics we learned at the beginning of the semester."

4.6.5 Schedule Visualization. Several students found the schedule visualization shown in Figure 3 useful. "I found the progress meters for each topic to be helpful, indicating areas which I needed to review or practice more." However, some students did not recall seeing this feature. One student, who received an A in the course, still didn't use the practice tool much since that student still thought that the questions were selected at random. This indicates that this visualization could be improved so that more students notice it and understand that the spacing algorithm is not just selecting questions at random.

4.6.6 Progress Tracker. Students found the progress tracker shown in Figure 2 useful. "I like that it gives you a countdown of how many days are left, and tells you how many questions you've answered so far that day." Students also found the progress tracker motivating. "The number of questions on the screen that I have to complete - it motivates me to keep practicing."

4.6.7 Other Practice Tools. Multiple students preferred our practice tool to Problem Roulett. One of them wrote: "I started using the "Problem Roulett" practice tool from Stats 250 after I realized the impact that this practice tool had on my knowledge of python. This tool had many more questions than the statistics tool and the encouragement for students to complete just 10 questions a day motivated students to use the tool more, rather than just leading up to the exam (which is mostly when I used the statistics tool)."

5 DISCUSSION

RQ1: The practice tool and grading scheme were designed to encourage spaced (rather than massed) practice. To gain the maximum possible points, students had to use the tool on at least 45 days during the semester. It was possible that students still mass their practice in a block of days before the midterm and final, but as shown in Figure 5 practice was spread out over the entire semester. This provides evidence of the effectiveness of the tool and grading scheme for encouraging spaced practice over massed practice.

RQ2: Of the 193 students, 62 (32%) used the practice tool for more than the required number of days. The end-of-semester survey provided evidence that students found the practice tool useful, perceived that it improved their understanding, and helped them track their progress. These findings provide evidence that the practice tool design and grading system which incorporated elements from self-determination theory and gameful design overcome students' typically negative reaction to desirable difficulties.

RQ3: The regression analysis showed that every hour of using the practice tool correlated with on average 1.04% increase in their final grades, even after controlling for potential confounds. Besides, there was no significant correlation between the number of hours the students used the e-book outside of the practice tool and their final grades. This suggests that not all time on task is equivalent; there appears to be something special about the practice tool. Since the time using the e-book outside of the practice tool may include preparation for lecture, review after lecture, and student-directed practice, we can not make strong conclusions about exactly what was better about using the practice tool. However, the theory of desirable difficulties predicts that spaced, interleaved, and retrieval practice should improve long-term retention of concepts.

6 LIMITATIONS

We conducted this study on undergraduate students in an introductory programming course at the University of Michigan. The results may not be generalizable to other courses, universities, or countries. More studies should be done to verify these findings in a wide variety of contexts.

Students who used the practice tool may have also practiced outside of the e-book environment, for example, with flashcards or in study groups. More studies should be done to verify that the correlation with exam grades is not at least partially due to unmeasured external practice.

It is well known that practice can improve learning [12]. This study does not separate the effect of more practice of all kinds from the effect of the particular kind of practice embodied in this tool: spaced, interleaved, retrieval practice. Additional studies should be done to isolate the effect from this type of practice.

7 CONCLUSION

Desirable difficulties impede short-term learning but significantly enhance long-term learning and transfer. These techniques include spacing practice, interleaving topics during practice, and retrieval practice (testing). However, students tend to hold negative perceptions about these techniques.

We improved SuperMemo 2 algorithm and used ideas from gameful design to create a practice tool that was integrated into an e-book platform. The tool automatically reused the interactive exercises in the e-book to provide personalized spaced and interleaved retrieval practice. Students earned points per day of use, rather than per question, to encourage spacing activity out over the entire semester. We replaced questions with topics of questions and refined the measure of assessment in SuperMemo 2. Gameful features included unlimited attempts at answering each question with no penalty, feedback about progress toward the daily and semester-long goals, and celebration of completion of daily goals.

The tool successfully encouraged spaced practice, a high percentage (32%) of the students used the tool more than required, and use of the tool correlated with higher exam grades. This tool is already integrated into an open-source e-book platform, Runestone, which has over 25,000 users a day. We encourage others to use and test this tool in their programming courses.

REFERENCES

[1] Stephen J Aguilar, Caitlin Holman, and Barry J Fishman. 2018. Game-inspired design: Empirical evidence in support of gameful learning environments. *Games and Culture* 13, 1 (2018), 44–70.

[2] John R Anderson, C Franklin Boyle, and Brian J Reiser. 1985. Intelligent tutoring systems. *Science* 228, 4698 (1985), 456–462.

[3] J. R Anderson and E. Skwarecki. 1986. The Automated Tutoring of Introductory Computer Programming. *Commun. ACM* 29, 9 (Sept. 1986), 842–849. https://doi.org/10.1145/6592.6593

[4] Michael C Anderson, Robert A Bjork, and Elizabeth L Bjork. 1994. Remembering can cause forgetting: retrieval dynamics in long-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 20, 5 (1994), 1063.

[5] Remzi H Arpaci-Dusseau and Andrea C Arpaci-Dusseau. 2014. *Operating systems: Three easy pieces*. Vol. 151. Arpaci-Dusseau Books Wisconsin.

[6] Harry P Bahrick, Lorraine E Bahrick, Audrey S Bahrick, and Phyllis E Bahrick. 1993. Maintenance of foreign language vocabulary and the spacing effect. *Psychological Science* 4, 5 (1993), 316–321.

[7] Valerie Barr and Deborah Trytten. 2016. Using Turing's Craft Codelab to Support CS1 Students As They Learn to Program. *ACM Inroads* 7, 2 (May 2016), 67–75. https://doi.org/10.1145/2903724

[8] Elizabeth L Bjork, Robert A Bjork, et al. 2011. Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. *Psychology and the real world: Essays illustrating fundamental contributions to society* 2, 59-68 (2011).

[9] ROBERT BJORK. 2017. Creating Desirable Difficulties to Enhance Learning. *Progress* (2017).

[10] Robert A Bjork and Ted W Allen. 1970. The spacing effect: Consolidation or differential encoding? *Journal of Verbal Learning and Verbal Behavior* 9, 5 (1970), 567–572.

[11] Kristine C Bloom and Thomas J Shuell. 1981. Effects of massed and distributed practice on the learning and retention of second-language vocabulary. *The Journal of Educational Research* 74, 4 (1981), 245–248.

[12] John D Bransford, Ann L Brown, Rodney R Cocking, et al. 2000. *How people learn*. Vol. 11. Washington, DC: National academy press.

[13] Tyne Crow, Andrew Luxton-Reilly, and Burkhard Wuensche. 2018. Intelligent Tutoring Systems for Programming Education: A Systematic Review. In *Proceedings of the 20th Australasian Computing Education Conference (ACE '18)*. ACM, New York, NY, USA, 53–62. https://doi.org/10.1145/3160489.3160492

[14] Frank N Dempster. 1988. The spacing effect: A case study in the failure to apply the results of psychological research. *American Psychologist* 43, 8 (1988), 627.

[15] August E Evrard, Michael Mills, David Winn, Kathryn Jones, Jared Tritz, and Timothy A McKay. 2015. Problem roulette: Studying introductory physics in the cloud. *American Journal of Physics* 83, 1 (2015), 76–84.

[16] Joseph R Ferrari. 1994. Dysfunctional procrastination and its relationship with self-esteem, interpersonal dependency, and self-defeating behaviors. *Personality and Individual Differences* 17, 5 (1994), 673–679.

[17] Ayaan M Kazerouni, Stephen H Edwards, and Clifford A Shaffer. 2017. Quantifying incremental development practices and their relationship to procrastination. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*. ACM, 191–199.

[18] Nate Kornell and Robert A Bjork. 2008. Learning concepts and categories: Is spacing the "enemy of induction"? *Psychological science* 19, 6 (2008), 585–592.

[19] Nate Kornell and Robert A Bjork. 2009. A stability bias in human memory: Overestimating remembering and underestimating learning. *Journal of experimental psychology: General* 138, 4 (2009), 449.

[20] Robert N Kraft and James J Jenkins. 1981. The lag effect with aurally presented passages. *Bulletin of the Psychonomic Society* 17, 3 (1981), 132–134.

[21] Amruth N Kumar. 2018. Epplets: A Tool for Solving Parsons Puzzles. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, 527–532.

[22] Mark R Lepper, David Greene, and Richard E Nisbett. 1973. Undermining children's intrinsic interest with extrinsic reward: A test of the" overjustification" hypothesis. *Journal of Personality and social Psychology* 28, 1 (1973), 129.

[23] Dale Parsons and Patricia Haden. 2006. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*. Australian Computer Society, Inc., 157–163.

[24] James H Reynolds and Robert Glaser. 1964. Effects of repetition and spaced review upon retention of a complex learning task. *Journal of Educational Psychology* 55, 5 (1964), 297.

[25] Richard Michael Ryan and Edward Lewis Deci. 2016. Facilitating and hindering motivation, learning, and well-being in schools: Research and observations from self-determination theory. *Handbook of motivation at school* 96 (2016).

[26] Nicholas C Soderstrom and Robert A Bjork. 2015. Learning versus performance: An integrative review. *Perspectives on Psychological Science* 10, 2 (2015), 176–199.

[27] Piers Steel. 2007. The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure. *Psychological bulletin* 133, 1 (2007), 65.

[28] Christopher A Wolters. 2003. Understanding procrastination from a self-regulated learning perspective. *Journal of educational psychology* 95, 1 (2003), 179.

[29] PA Wozniak. 2004. SuperMemo: First experiments (1982-1985). https://www.supermemo.com/english/ol/beginning.htm

[30] Veronica X Yan, Elizabeth Ligon Bjork, and Robert A Bjork. 2016. On the difficulty of mending metacognitive illusions: A priori theories, fluency effects, and misattributions of the interleaving benefit. *Journal of Experimental Psychology: General* 145, 7 (2016), 918.

[31] Iman YeckehZaare and Paul Resnick. 2019. Speed and Studying: Gendered Pathways to Success. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM, 693–698.