

# An Analysis of Interactive Feature Use in Two Ebooks

Barbara J. Ericson<sup>[0000-0001-6881-8341]</sup>

University of Michigan [barbarer@umich.edu](mailto:barbarer@umich.edu)  
<https://www.si.umich.edu/people/barbara-ericson>

**Abstract.** The ITICSE 2013 Working Group Report predicted that paper Computer Science (CS) books would be replaced by interactive electronic books (ebooks). CS ebooks provide the ability to run, edit, and write code directly in a browser window. CS ebooks also include interactive practice problems with immediate feedback. Several free interactive ebooks have already been created for introductory computing courses. Research has shown that using interactive ebooks leads to an increase in use, motivation, and learning gains versus static ebooks. Teachers who utilize more of the interactive features report more confidence in their ability to teach the material than those who do not. This paper reports on a log file analysis of two free interactive ebooks: one for the Advanced Placement (AP) Computer Science Principles (CSP) course and one for the AP Computer Science A (CSA) course. This paper examines usage of the ebooks across the chapters of the ebooks and does an analysis of usage of several of the interactive activities: short-answer, clickable-area, and mixed-up-code (Parsons) problems. This type of analysis can provide valuable information to both ebook authors and ebook platform designers.

**Keywords:** log file analysis · ebooks · active learning · Parsons problems

## 1 Introduction

Early electronic books were simply digital copies of paper books. Still people appreciated their portability and the ability to search this type of ebook [22]. Another early type of ebook, called a SuperBook, featured full text search, a table of contents, and annotations [18]. More recently, interactive ebooks for introductory computing courses have been developed, such as the OpenDSA ebook [19] and several free ebooks on the Runestone open-source platform [2][14].

The ITICSE 2013 Working Group Report surveyed 75 SIGCSE members [11]. Most of the respondents (75%) agreed or strongly agreed that they planned on teaching with an ebook in the next two years. This shows the growing importance of ebooks. The working group also recommended that ebooks include: 1) user control over pacing 2) practice problems with immediate feedback 3) learning activities like interactive visualizations and the ability to write and edit code, and 4) the ability for an instructor to customize the ebook modules.

In a study of undergraduate students using an early version of an interactive ebook, *How to Think Like a Computer Scientist*, on the Runestone platform, over 90% of the undergraduate students said that they would like to use an interactive textbook in another computing course [15]. Students particularly liked the ability to run and modify the code directly in the ebook.

In a recent study comparing use of a static PDF textbook to an interactive ebook in a large ( $n > 500$ ) required CS1 course for engineering students, there was an increase in use, motivation, and learning gains for the interactive ebook [17]. The text for the two books was nearly identical, but the interactive ebook also contained interactive program visualization and annotated code examples.

The author was part of a research team that developed an interactive ebook for high school teachers to help them learn to teach the Advanced Placement (AP) Computer Science Principles (CSP) course. AP CSP is intended to be equivalent to a first course for non-majors at the college level (CS0). This course covers programming fundamentals: variables, conditionals, loops, and functions. The ebook for AP CSP included text, pictures, videos, executable and editable Python code, audio tours of code, multiple-choice questions, fill-in-the-blank questions, and Parsons problems. In a Parsons problem the learner places mixed-up code blocks in the correct order as shown in Figure 1. Learners drag the blocks from the left to the right to assemble a solution.

**Fig. 1.** An example Parsons problem in Python. The mixed-up blocks are shown on the left and the correct solution is shown on the right.

The team found that teachers and students executed the code, answered multiple-choice questions, and attempted Parsons problems [6]. Few teachers or students watched the videos or listened to audio tours of the code. Students attempted more of the interactive activities at the beginning of the chapter and less by the end of the chapter. Teachers have found the ebook user interface to be to be legible, easy to navigate, and usable [2]. In a small study ( $n = 10$ ) teachers who used more of the interactive features in the teacher ebook did better on the post-tests and reported higher confidence in their ability to teach the material [2].

The author has been using the Runestone ebook platform to test the efficiency and effectiveness of learning from solving Parsons problems versus fixing code or writing code [7]. Some Parsons problems, called *two-dimensional Parsons problems*, require the learner to indent the code correctly. Some Parsons problems include *distractor* blocks which are not needed in a correct solution. The author invented two types of adaptation for Parsons problems: intra-problem and inter-problem. In intra-problem adaption if the user fails to solve the current Parsons problem, the problem can be dynamically made easier by disabling distractors, providing indentation, or combining blocks. In inter-problem adaptation the difficulty of the next problem is modified based on the learner’s performance on the previous problem. Studies have provided evidence that Parsons problems are a more efficient and effective form of practice than fixing code with errors or than writing the equivalent code [7][5][10].

The author also created an ebook for the AP CSA course. This course is offered in high school and is intended to be equivalent to a first course for computer science majors at the college level. It covers programming fundamentals (variables, strings, conditionals, loops), 1D and 2D arrays, lists, object-oriented programming, recursion, searching, and sorting. This ebook has been used to help underrepresented students succeed in AP CSA [1][3][8][4]. However, usage of this ebook had not been analyzed.

This study investigates how usage changed across the two ebooks (CSP and CSA), how usage of some of the new interactive features (short-answer and clickable-area) compared with other interactive features, and if the log file data could be used to identify common errors in Parsons problems.

## 2 Ebook Features

The two ebooks used in this study include instructional material (text, video, example code, audio tours, and images) as well as automated assessments (multiple-choice questions, fill-in-the-blank questions, Parsons problems, etc) with immediate feedback.

In a fill-in-the-blank question the learner enters an answer in response to a question as shown in Figure 2. Regular expressions are used to give feedback.

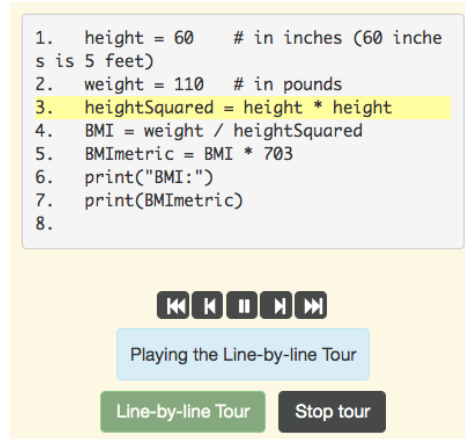
csp-5-2-1: What letter (like A, B, C, D, etc) will the program below draw in block style when you click on the Run button?

Check Me Compare Me

Correct!

**Fig. 2.** A fill in the blank question that asks the learner to predict what code will do when it executes.

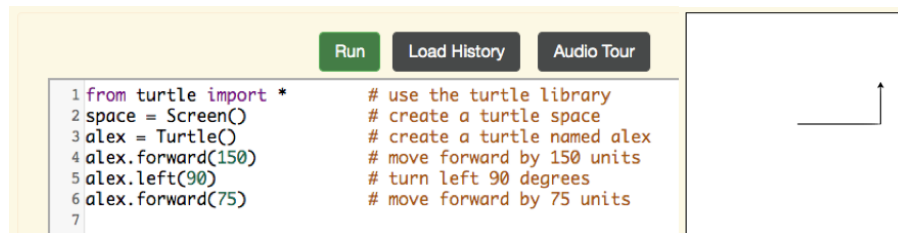
In an audio tour, one or more lines of code are highlighted as the audio plays as shown in Figure 3. This feature takes advantage of the fact that humans can process visual and audio information at the same time [12][13]. While few teachers or students play the audio tours, teachers have reported finding them useful since they demonstrate how to describe code [9].



**Fig. 3.** An audio tour highlights a line of code as the audio explains that line.

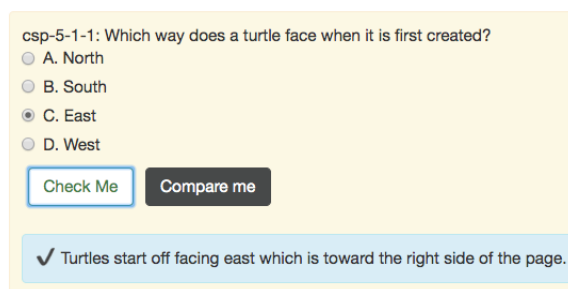
## 2.1 Ebook Design

The ebooks were designed using a worked example plus interleaved practice problem approach based on research findings from educational psychology [20][21]. A worked example is an expert's solution to a problem. The ebooks provide code examples with comments that explain the code as shown in Figure 4. Learners can run and modify the code. This example uses a LOGO-style turtle that draws with a pen as it moves as shown in Figure 4 [16].



**Fig. 4.** A worked example in Python using a LOGO-style turtle with comments explaining the code.

Each worked example is typically followed by one or more practice problems to encourage the learner to study the worked example and learn from it. The example code shown in Figure 4 was followed by the multiple-choice question shown in Figure 5 and the Parsons problem shown in Figure 6.



csp-5-1-1: Which way does a turtle face when it is first created?

A. North

B. South

C. East

D. West

Check Me Compare me

✓ Turtles start off facing east which is toward the right side of the page.

Fig. 5. A multiple-choice question with the correct answer selected.

### 3 Log File Study

We requested an anonymous log file from the Runestone platform. This file had all user and institution information replaced with unique identifiers. The log file contained data from the use of the two ebooks (CSP and CSA) from 8/13/2017 to 10/31/2018.

#### 3.1 Usage Across the Chapters of the Ebooks

In a previous study there was a steady drop in the number of unique users who attempted each interactive activity across a chapter of the *How to Think Like a Computer Scientist* ebook [6]. We wondered if we would see this same pattern across the CSA and CSP ebooks. We counted the number of unique users who attempted any of the interactive items in each subchapter of each ebook.

**AP CSA Ebook** As shown in Figure 7 not as many unique users attempted the pretest in chapter one of the AP CSA ebook as used an interactive feature in chapter two. The first chapter was labelled preface, which may have led to some users skipping it. In general there was a downward trend in the number of unique users who used any of the interactive features from chapter two to chapter 13. Over 3,500 users did something in chapter two, but by chapter 13 it was about 1,000 users. One exception to this pattern is chapter six, which is the Magpie Chatbot lab. It had a much lower number of users than the surrounding chapters. The AP CSA course includes three exemplar labs, but they are optional. This

**Mixed up programs**

csp-5-1-2: The following program uses a turtle to draw a capital L as shown to the left, but the lines are mixed up. The program should do all necessary set-up: import the turtle module, get the space to draw on, and create the turtle. The turtle should turn to face south, draw a line that is 150 pixels long, then turn to face east, and draw a line that is 75 pixels long. We have added a compass to the picture to indicate the directions north, south, west, and east.

Drag the needed blocks of statements from the left column to the right column and put them in the right order. There may be additional blocks that are not needed in a correct solution. Then click on *Check Me* to see if you are right. You will be told if any of the lines are in the wrong order or are the wrong blocks.

Drag from here

- ella.right(90)
- ella.turn(90)
- from turtle import \*
- from turtle import \*
- space = Screen()
- space = screen()
- ella.forward(75)
- ella.go(75)
- ella = Turtle()
- ella.left(90)
- ella.forward(150)

Drop blocks here

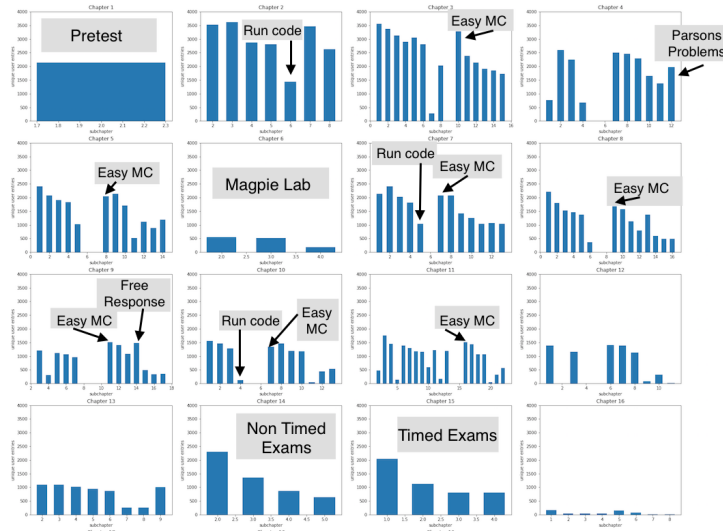
Correct Solution

Drop blocks here

- from turtle import \*
- space = Screen()
- ella = Turtle()
- ella.right(90)
- ella.forward(150)
- ella.left(90)
- ella.forward(75)

Check Me
Reset

**Fig. 6.** A Parson problem with the code mixed-up on the left (with additional unneeded code blocks) and the correct solution on the right.



**Fig. 7.** The number of unique users that used an interactive feature in each subchapter of the AP CSA ebook.

may be one reason why a lower number of users attempted this lab. Another possibility is that teachers prefer to use the provided external lab materials.

There was an large increase in the number of unique users in chapter 14 and 15 as shown in Figure 7. This is not surprising since chapter 14 has non-timed practice multiple-choice exams and chapter 15 has timed practice multiple-choice exams. While there was a big jump in the number of users who tried the first non-timed practice exam, it is still less than the number of users in chapter two. This indicates that the ebook is being used for more than just taking practice exams. It is also interesting to note that the number of unique users typically decreases across the subchapters, until the easy multiple-choice questions, where it increases again. One of the interesting findings is a common dip in the number of unique users for subchapters that only contain run code and no practice activities. This shows that users are more likely to attempt the interactive practice activities, than run the example code. This is consistent with our previous findings. Teachers told us that they did not feel that they had to run the code if they thought they understood it [6]. They checked their understanding by answering the interactive questions and if they were correct, they did not run the code.

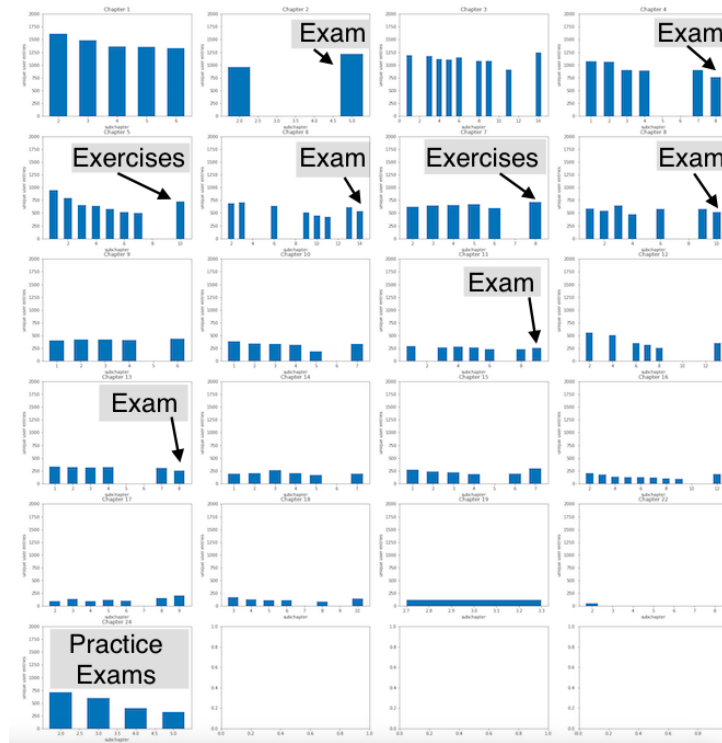
Most chapters contained a summary of new concepts that included the ability to drag definitions to concepts. The second to last subchapter was usually code practice. The last subchapter was mixed-up code (Parsons) problem practice. As you can see in Figure 7 the number of users who attempted the Parsons problems at the end of each chapter was similar to or in some cases larger than the number who attempted the code problems.

**Ebook for AP CSP** As shown in Figure 8 over 1,500 unique users did some interactive activity in chapter one of the AP CSP ebook, but by chapter five it was less than 1,000 and by chapter 9 it was less than 500. As with the AP CSA ebook, we see an increase again in chapter 24 with the practice exams. We also see a decline across some chapters like chapter five, but then an increase in the exercises at the end of the chapter.

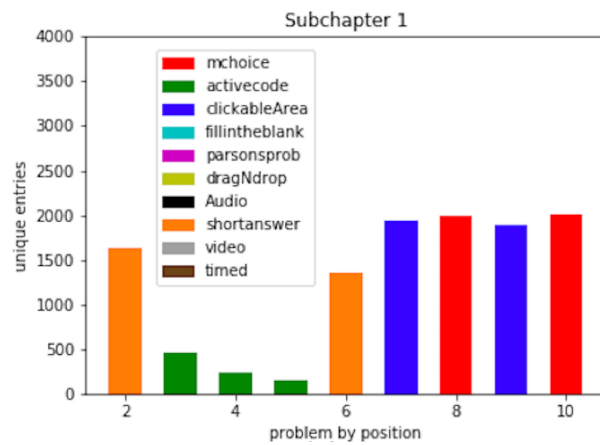
## 4 Analysis of Short-Answer Questions

In a short answer question the user can enter one or more sentences in response to a question. For example, in the AP CSA ebook we included text and images that compared an array to a row of lockers and then asked the users if they could think of another example of something that is like an array (like a row of lockers). We were surprised to find that over 1,500 unique users had answered this question as shown in Figure 9. In comparison only 500 users ran the code that followed this question.

We have only skimmed the answers so far and plan to code the answers into categories. We have noticed that some answers were off task ("potato"), some were correct ("Donuts in a box"), and some were wrong ("an Array is



**Fig. 8.** The number of unique users that used an interactive feature in each subchapter of the AP CSP ebook.



**Fig. 9.** The number of unique users that attempted each interactive activity (problem) in chapter eight subchapter one about arrays.



like a bag”). Short-answer questions may be helpful for discovering common misconceptions.

The AP CSA ebook also included a short-answer question that asked the user what questions they had about arrays. We are currently examining the answers to this question and plan to revise the ebook to include the answers to at least some of these questions. One way to automatically evolve an ebook over time would be to allow users to ask and answer questions in the ebook.

## 5 Analysis of Clickable Area Questions

This interactive feature has two types of questions: clickable code and clickable table elements.

### 5.1 Clickable Code

In clickable code the user clicks on one or more lines of code to answer a question. Each line that the user clicks is highlighted in yellow. The user clicks the *Check Me* button to submit an answer. The correct lines that were clicked remain highlighted in yellow and any incorrect lines that were clicked are boxed in red as shown in Figure 10. Textual feedback is displayed under the question. It tells the user how many of the correct lines were clicked and how many of the incorrect lines were clicked.

3-2-1: Click on all of the variable declarations in the following code.

```
public class Test2
{
    public static void main(String[] args)
    {
        int numLives;
        numLives = 0;
        System.out.println(numLives);
        double health;
        health = 8.5;
        System.out.println(health);
        boolean powerUp;
        powerUp = true;
        System.out.println(powerUp);
    }
}
```

Check Me

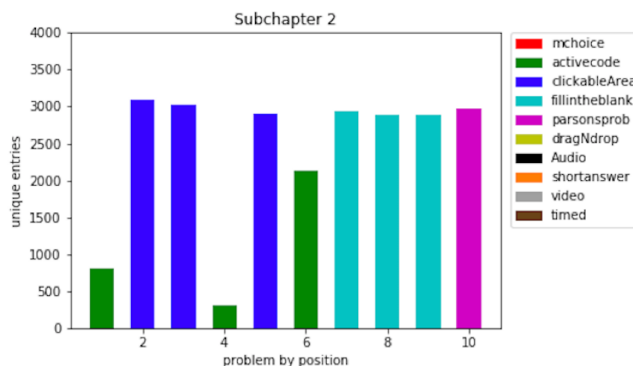
Incorrect. You clicked on 3 of the 3 correct elements and 3 of the 12 incorrect elements. Variable declarations start with a type and then a name.

**Fig. 10.** A clickable code question that shows the correct answers still highlighted in yellow and the incorrect answers boxed in red.

The author added clickable code problems to the ebook for AP CSA. One of the difficulties in learning to program is understanding the terminology. Novices

often do not understand the difference between declaring a variable and initializing a variable. The question in Figure 10 tries to help the user learn to spot variable declarations.

As you can see from Figure 11 a higher number of users answered the clickable-area questions than the other interactive activities in the same subchapter. This is encouraging and means that we should further study the effectiveness of this type of practice problem.



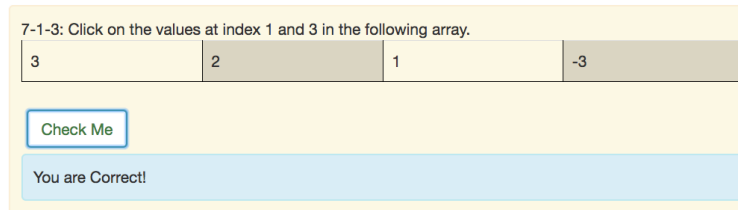
**Fig. 11.** The number of unique users that attempted each interactive activity in the chapter on variables (chapter 3 subchapter 2)

## 5.2 Clickable Table Item

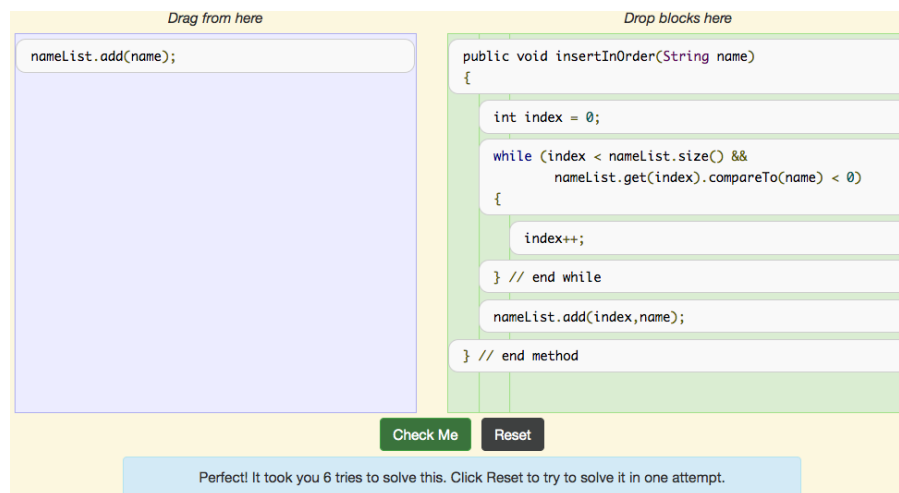
A common error with arrays is thinking that the first element is at index one instead of zero. Clickable area problems display a question and table of values and the user clicks on the table items to answer the question. Figure 12 shows the user correctly identifying the values at index one and three. We found that the number of users who attempted this type of activity in chapter 8 was lower than, but similar to, the number who attempted nearly multiple-choice questions as shown in Figure 9.

## 6 Analysis of Parsons problem

We counted the number of unique users who submitted the same incorrect solution to each Parsons problem. The goal was to look for common error states as a way to detect common student misconceptions. The problem with the most unique users who submitted the same incorrect error state was a problem that asked the user to insert a name in alphabetical order in a list as shown in Figure 13.



**Fig. 12.** A clickable area question that asks the user to click on the array values at index 1 and 3.



**Fig. 13.** A correct solution on the right side for a Parsons problem that asks the user to insert a name in alphabetical order in a list. There is a distractor on the left side.

Sixty percent of the users (467 of 778) checked an incorrect solution that moved the `nameList.add(index,name)` inside the loop before the increment of the index. Fifty-two percent (406 of 778) checked an incorrect solution with the `nameList.add` inside the loop, but after the index increment.

Users are used to seeing loops that add something to a variable each time through a loop, which may be why such a high percentage of users put the add inside of the loop, instead of after it. This is likely the first time that a user has seen a loop that only searches for the correct place to add an item. This implies that it might be useful to provide a similar example before asking users to solve this problem. A think-aloud study with this problem might also be helpful to determine why students are placing the add inside of the loop. This type of analysis can help uncover common problems and can be used to revise the ebook to improve instruction.

## 7 Discussion

This paper first investigated how usage changed across two ebooks for Advanced Placement (AP) Computer Science A (CSA) and Principles (CSP). In both ebooks there was a similar drop off in the number of unique users who attempted each interactive activity from the beginning of the ebook to the end. However, more users attempted the practice exams at the end of the ebooks. This analysis also showed that less users ran code in subchapters that only included a run code activity with no practice problems. This implies that subchapters should include more than just the ability to run code, such as adding practice multiple-choice questions or Parsons problems. The paper next investigated usage of a couple of newer interactive features: short-answer and clickable-area. It found that a high percentage of users used these new features. More questions should be added to the ebooks using these features and their effect on learning and motivation should be studied. The paper also explored using the answers to the short-answer questions to identify common misconceptions of arrays and common questions. However, more work needs to be done on this to classify the answers.

Finally the paper determined which Parsons problem had the highest number of unique users that all submitted the same incorrect answer. This type of analysis can be used to identify problems that should be further studied and used to improve the instruction in the ebook by identifying where more explanation and examples are needed.

Interactive ebooks can improve learning and motivation in computing and are likely to replace paper textbooks. It is important to understand how ebooks are actually used and what we can learn from the usage data. This information can be used to improve ebooks by making them more engaging, improving instruction, and addressing misconceptions. Interactive ebooks can be used in large introductory computing courses to provide immediate feedback and practice at scale.

## References

1. Ericson, B., Engelman, S., McKlin, T., Taylor, J.: Project rise up 4 cs: increasing the number of black students who pass advanced placement cs a. In: Proceedings of the 45th ACM technical symposium on Computer science education. pp. 439–444. ACM (2014)
2. Ericson, B., Guzdial, M., Morrison, B., Parker, M., Moldavan, M., Surasani, L.: An ebook for teachers learning cs principles. *ACM Inroads* **6**(4), 84–86 (2015)
3. Ericson, B., McKlin, T.: Helping african american students pass advanced placement computer science: A tale of two states. In: 2015 Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT). pp. 1–8. IEEE (2015)
4. Ericson, B., McKlin, T.: Helping underrepresented students succeed in ap csa and beyond. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education. pp. 356–361. ACM (2018)
5. Ericson, B.J., Foley, J.D., Rick, J.: Evaluating the efficiency and effectiveness of adaptive parsons problems. In: Proceedings of the 2018 ACM Conference on International Computing Education Research. pp. 60–68. ICER '18, ACM, New York, NY, USA (2018). <https://doi.org/10.1145/3230977.3231000>, <http://doi.acm.org/10.1145/3230977.3231000>
6. Ericson, B.J., Guzdial, M.J., Morrison, B.B.: Analysis of interactive features designed to enhance learning in an ebook. In: Proceedings of the eleventh annual International Conference on International Computing Education Research. pp. 169–178. ACM (2015)
7. Ericson, B.J., Margulieux, L.E., Rick, J.: Solving parsons problems versus fixing and writing code. In: Proceedings of the 17th Koli Calling Conference on Computing Education Research. pp. 20–29. ACM (2017)
8. Ericson, B.J., Parker, M.C., Engelman, S.: Sisters rise up 4 cs: Helping female students pass the advanced placement computer science a exam. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education. pp. 309–314. ACM (2016)
9. Ericson, B.J., Rogers, K., Parker, M., Morrison, B., Guzdial, M.: Identifying design principles for cs teacher ebooks through design-based research. In: Proceedings of the 2016 ACM Conference on International Computing Education Research. pp. 191–200. ACM (2016)
10. Ericson, B.J.: evaluating the effectiveness and efficiency of parsons problems and dynamically adaptive parsons problems as a type of low cognitive load practice problem. Ph.D. thesis, Georgia Institute of Technology (2018)
11. Korhonen, A., Naps, T., Boisvert, C., Crescenzi, P., Karavirta, V., Mannila, L., Miller, B., Morrison, B., Rodger, S.H., Ross, R., et al.: Requirements and design strategies for open source interactive computer science ebooks. In: Proceedings of the ITiCSE working group reports conference on Innovation and technology in computer science education-working group reports. pp. 53–72. ACM (2013)
12. Mayer, R.E.: Applying the science of learning: Evidence-based principles for the design of multimedia instruction. *American psychologist* **63**(8), 760 (2008)
13. Mayer, R.E., Moreno, R.: A split-attention effect in multimedia learning: Evidence for dual processing systems in working memory. *Journal of educational psychology* **90**(2), 312 (1998)
14. Miller, B., Ranum, D.: Runestone interactive: tools for creating interactive course materials. In: Proceedings of the first ACM conference on Learning@ scale conference. pp. 213–214. ACM (2014)

15. Miller, B.N., Ranum, D.L.: Beyond pdf and epub: toward an interactive textbook. In: Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education. pp. 150–155. ACM (2012)
16. Papert, S.: Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc. (1980)
17. Pollari-Malmi, K., Guerra, J., Brusilovsky, P., Malmi, L., Sirkiä, T.: On the value of using an interactive electronic textbook in an introductory programming course. In: Proceedings of the 17th Koli Calling International Conference on Computing Education Research. pp. 168–172. ACM (2017)
18. Remde, J.R., Gomez, L.M., Landauer, T.K.: Superbook: an automatic tool for information explorationhypertext? In: Proceedings of the ACM conference on Hypertext. pp. 175–188. ACM (1987)
19. Shaffer, C.A., Karavirta, V., Korhonen, A., Naps, T.L.: Opendsa: beginning a community active-ebook project. In: Proceedings of the 11th Koli Calling International Conference on Computing Education Research. pp. 112–117. ACM (2011)
20. Sweller, J.: Cognitive load during problem solving: Effects on learning. *Cognitive science* **12**(2), 257–285 (1988)
21. Trafton, J.G., Reiser, B.J.: The contributions of studying examples and solving problems to skill acquisition. Ph.D. thesis, Citeseer (1994)
22. Zhang, Y., Kudva, S.: Ebooks vs. print books: Readers’ choices and preferences across contexts. In: Proceedings of the 76th ASIS&T Annual Meeting: Beyond the Cloud: Rethinking Information Boundaries. p. 108. American Society for Information Science (2013)