8-14-2017

Dual Modality Code Explanations for Novices: Unexpected Results

Briana B. Morrison

Dual Modality Code Explanations for Novices: Unexpected Results

Briana B. Morrison University of Nebraska at Omaha 6001 Dodge Street Omaha, NE 68182 bbmorrison@unomaha.edu

ABSTRACT

The research in both cognitive load theory and multimedia principles for learning indicates presenting information using both diagrams and accompanying audio explanations yields better learning performance than using diagrams with text explanations. While this is a common practice in introductory programming courses, often called “live coding,” it has yet to be empirically tested. This paper reports on an experiment to determine if auditory explanations of code result in improved learning performance over written explanations. Students were shown videos explaining short code segments one of three ways: text only explanations, auditory only explanations, or both text and auditory explanations, thus replicating experiments from other domains. The results from this study do not support the findings from other disciplines and we offer explanations for why this may be the case.

CCS CONCEPTS • Social and professional topics→K-12 education; Computer science education;

KEYWORDS cognitive load; modality; live coding

1 INTRODUCTION

Cognitive load theory (CLT) describes the role of the learner’s memory during the learning process. The central problem identified by CLT is that learning is impaired when the total amount of processing requirements exceeds the limited capacity of working memory [43]. By minimizing undesirable loads within the instructional materials, the learner’s memory can hold more relevant information, thereby improving the effectiveness of the learning process.

CLT has been used to explain key research findings. One example of this is the *split-attention effect.* Some worked examples have been found to be ineffective for improving learning. This occurs when learners have to split their attention between at least two sources of information, each of which is necessary for learning the material. This can occur when the information for learning is split into different pieces which are separated spatially or temporally. The information in the pieces is required to learn the material and each piece would make no sense alone. Because the learner must integrate all the disparate sources of information, the cognitive load to do so is unnecessarily high when the sources are separated by space or time. Having to switch focus and attention between two or more sources requires information to be maintained in working memory while searching and processing interacting elements in the linked source. Sweller states that “cognitive load theory does not distinguish between text and diagrams, text and text, or diagrams and diagrams” as contributors to a split-attention effect. [50, p.98]

Another recognized research finding has been labelled the *modality effect*. Modality is a particular form of sensory perception. In this context it pertains to how the information within the instructional materials is delivered to the learner. The modality effect occurs when the information to be learned is being delivered through multiple sensory channels –namely the auditory and visual channels. While the modality effect has been well documented as effective in other disciplines, it has yet to be empirically tested within the domain of computer science and introductory programming. Instructors presenting code in class often utilize dual modality: while code is displayed via a shared display, the instructor explains the code verbally. The students may concentrate on looking at code while the instructor explains the code. This removes the split-attention problem of a textbook. Intuitively we may believe this is an effective pedagogical technique and aids in student learning by reducing cognitive load, but what evidence do we have?

This paper reports on a study designed to empirically test the modality effect in computer science, specifically introductory programming. Given trends toward active learning classrooms and research for multimedia instructional design, we developed instructional material in an online format. All available recommendations for designing the instructional materials were used to create videos explaining small code examples in one of three formats: 1) auditory only explanations, 2) text only explanations, or 3) auditory and text explanations. This experiment design is a replication of studies conducted in other disciplines which confirmed the modality effect. We sought to answer the research question: **Does altering the modality (text, oral, both) of code explanations improve student learning as measured by retention and transfer questions?** Before the study we had two hypotheses: 1) Students receiving oral explanations will demonstrate better retention of the material, and 2) Students receiving both oral and text explanations will demonstrate the worst retention of the material. These hypotheses correspond with previous research findings. First we present the background literature in which this study is grounded. We then present the study method followed by the data analysis and results. We conclude with a discussion of the results and the implications for the research community and instructors.

2 BACKGROUND

To understand the basis for the study, we present a brief overview of Cognitive Load Theory followed by measurement techniques of cognitive load. Then previous research on the modality effect is discussed. We conclude with a summary of the necessary conditions in order to produce the modality effect.

2.1 Cognitive Load Theory

According to the original definition of CLT [51, 56, 63], instruction can impose three different types of cognitive load on a student's working memory: intrinsic load, extraneous load, and germane load. Intrinsic load (IL) is defined as a combination of the innate difficulty of the material being learned as well as the learner's characteristics [28]. A topic is considered to have a high IL if the material being learned is interconnected; that is, learning requires processing several elements simultaneously to understand their relations and interactions [55]. Intrinsic load can also vary with the domain expertise and previous knowledge of the learner [53] in that learners with a higher level of previous knowledge may chunk the material differently than novices [8], allowing them to hold more information in working memory. Extraneous load (EL) is the load placed on working memory that does not contribute directly toward the learning of the material –for example, the resources consumed while understanding poorly written text or diagrams without sufficient clarity [28]. The IL and EL are the factors that can be controlled through instructional design. The final category is that of germane load (GL) which are the instructional features that are necessary for learning the material [28]. One of the assumptions of original CLT is that these three components are additive [44]. If the extraneous load is using most of the capacity in working memory, little can be devoted to the germane load. However, the more recent consensus among researchers is that the three components –intrinsic, extraneous, and germane –are not additive to an overall sum [24]. Researchers now consider that cognitive load consists of the use of resources, both germane and extraneous. It is now believed that instructional material can help to reduce the extraneous load and minimize the intrinsic load thus leaving the remaining working memory free for learning [52].

2.2 Measuring Cognitive Load

Since the identification of CLT, researchers have searched for a means to measure cognitive load. To date, this has been accomplished through indirect, subjective, and direct measures. Indirect measures of cognitive load use production system models [4, 49] or learner performance indicators [11, 12] including error rates [4, 5]. Subjective measures of cognitive load include survey instruments that ask users to assess their mental effort [38, 40, 42, p.429]. The subjective rating scale has been shown to be the most sensitive measure available to differentiate the cognitive load imposed by different instructional methods [53] and have been consistent in matching performance data predicted by CLT [33]. The subjective scale has been used extensively to measure the relative cognitive load of different instructional methods with over 25 studies having used it between 1992 and 2002 [37]. Another subjective measurement is an efficiency measure for cognitive load [39], which combines both mental effort with task performance indicators. Over 30 cognitive load theory related studies have used this efficiency measure [61]. Two basic means of measuring cognitive load through direct measures have been used in research: using a dual task [9, 13, 60] and physiological measurements such as measurements of heart rates [41], pupillary response [59], EEGs [2], and eye tracking [58, 62].

Several researchers have attempted to distinguish between and measure the different types of cognitive load. Ayres attempted to keep the extraneous cognitive load (EL) constant between treatments thus attributing the differences to a change in intrinsic load (IL) [3]. DeLeeuw and Mayer used a mixed approach (both subjective measures and a secondary task method) to investigate if different instruments could measure the three loads separately [15]. The results indicated that different measures do tap into different processes and show varying sensitivities. In an attempt to measure the different cognitive load categories, Gerjets, Scheiter, and Catrambone selected three items from the NASA-TLX [21] associated with task demands [16, 17]. The researchers argued that the three items selected (mental and physical activity required, effort to understand the contents, and navigational demands of the learning environment) could be mapped to the intrinsic, germane, and extraneous loads, respectively. The test manipulated the complexity of worked examples. While the groups with the highest learning outcomes reported the lowest cognitive load, there was no corroborating evidence that the three measures corresponded to the different types of cognitive load as proposed.

In 2013, Leppink et al.[28] developed an instrument specifically for measuring different types of cognitive load which consists of a ten question subjective survey. This study revealed that none of the existing survey tools adequately separated the three types of cognitive load in that each had significant cross-loading between factors. However the newly developed survey yielded results that were consistent with outcomes based on CLT. Leppink et al. [29] extended their 2013 work by adapting the survey instrument to another domain, that of learning languages, and replicated their analyses. These new findings reinforce the strong support for the survey measuring both intrinsic and extraneous load, but found less support for the direct measure of germane load. In 2014, Morrison et al. [35] adapted the Leppink survey instrument to the programming domain and provided initial validation. This is the instrument used in this study.

2.3 Modality Effect Research

Research in both cognitive load theory [54] and multimedia principles for learning [31] indicates presenting information using both diagrams with audio explanations yields better learning performance than using diagrams with text explanations. According to the available models of multimedia learning [32, 45], cognitive processing of related text and pictures involves the selection and organization of the relevant elements of visual and auditory information, resulting in a coherent unified representation. All this is processed in the learner’s working memory. CLT argues that limited working memory can be effectively expanded by using more than one presentation modality.

Working memory consists of three subsystems: a phonological loop, a visuospatial sketchpad, and a central executive [6]. The phonological loop processes auditory information while the visuospatial sketchpad processes pictorial or written information. Because these are separate processes, we can assume that each have capacity and duration limitations. In some situations we can effectively increase the capacity of working memory by utilizing both processors.

In [36], the authors found that a visually presented geometry diagram, combined with aurally presented statements, enhanced learning compared to a conventional, visual-only presentation. In a split-attention situation, increasing effective working memory by using more than one modality produced a positive effect on learning. In [57], the authors used elementary electrical engineering instructions to show that an audio/visual diagram format was superior to purely visual instructions. The cognitive load measurement tool [42] was used to support the suggestion that the effect can be attributed to cognitive load factors. In [26] the authors confirmed that a dual-mode presentation of instructional information is a viable alternative to physical integration of all written materials (eliminating split-attention) within an elementary electrical engineering domain. In addition, Mayer [32] presents evidence of several studies done in the multimedia medium with animated videos and spoken explanations and reveals findings which indicate that the spoken explanation was only effective when done simultaneously rather than sequentially with the visually presented information. Kalyuga [25] provides an overview of all modality studies alongside instructional implications.

One modality study in computer science [48] involved students debugging introductory programs with a modified development environment that used auditory cues. Students were assigned to one specific modality interface (text only, auditory only, or both) and asked to complete comprehension questions and debugging tasks. No statistical significance between the modalities existed for the comprehension questions, but the auditory only interface was statistically worse for the debugging tasks. This study did not involve learning new material but only performance on tasks previously learned. There are limitations as to when the modality effect will affect learning. Textual information presented in spoken form will not generate a modality effect if it merely re-describes a diagram. The information presented in the diagram and the textual information must be unintelligible by themselves. If a diagram and text are being used, both must contain information that requires learners to refer to the other source in order to enable comprehension [54].

Ginns conducted a meta-analysis of modality effects based on 43 different experiments [18]. The meta-analysis generally supported the positive effects of dual-modality presentations. However, two major moderators were found: the level of element interactivity and the pacing of the presentation. Generally, only problems with a high level of element interactivity will benefit from a dual-modality presentation. The benefit of a dual-modality presentation can be lost, however, if the interactivity is excessively high. Strong effects of a dual-modality presentation were found only under system-paced conditions or fixed timings.

2.4 Summary

Given that research shows the modality effect is present in some instances while not in others, we sought to design the instructional materials for this study to result in a modality effect. In other words, we attempted to follow all recommendations to cause the modality effect to occur. In essence, this study is a replication of [26, 36, 57] within the computer programming domain. Using the known limitations of when the modality effect occurs, examples were specifically selected and instructional materials developed with the expectation that the experiment would show that the modality effect holds with novice programmers.

Sweller et al. [54] list the following conditions required to obtain the modality effect:

- Diagrammatic and textual information must refer to each other and be unintelligible unless they are processed together.

- Element interactivity must be high, but not excessive.

- Auditory text should be limited. Any lengthy, complex text should be written, not spoken.

- If the diagrams are complex, cuing or signaling may be required so that learners can focus on the appropriate portion of the diagram and not be forced to search for the relevant piece.

In this experiment, the code and the explanations are separate pieces of information for the learner that need to be processed together to be understood. Learners for this experiment are complete novices, with many having never seen any code before. It is expected that seeing only the source code or hearing only the explanation would not be enough information to acquire the desired knowledge. Source code elements, or tokens, are highly interactive as they depend upon each other to be understood and interpreted. The explanations were designed to be simple and limited. Signaling was utilized during the explanations to illustrate the line of code being explained. Explanations were not strictly line by line or top-down, but done in chunks of program execution.

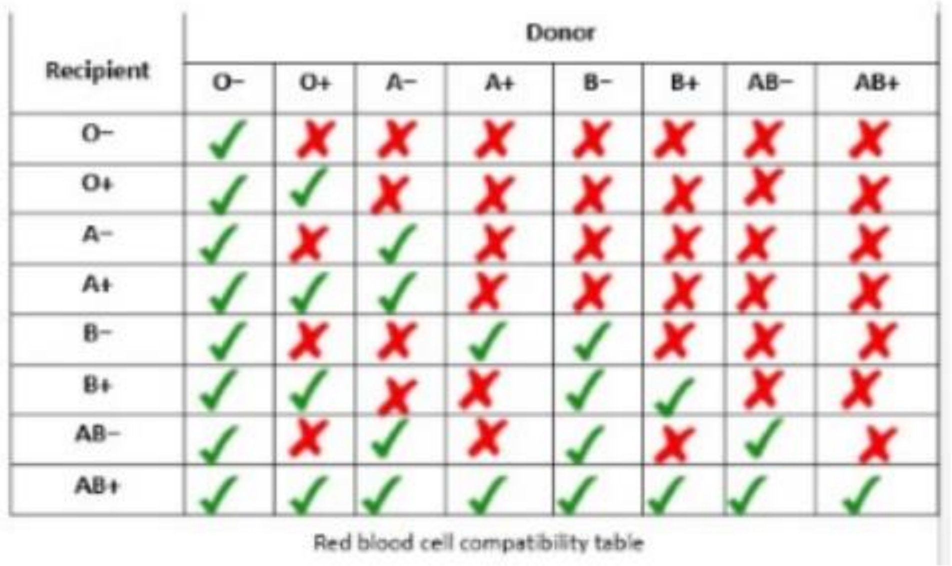
We have limited evidence that people see code segments more as a diagram rather than text to be read [20]. We know that expert programmers do not read a program line by line to understand it [10]. Instead, they group the lines of code into ’chunks’ which represent a purpose. This is similar to what chess grand masters do when they see a chess board [19], or what physics experts do when examining a diagram by classifying the problem by the components within the diagram ([7, 54]). Jeffries suggests that novices may begin by reading code segments as text [22], however eye-tracking data suggests that novices transition to an expert style of viewing code as they learn [34].

3 STUDY METHOD

We designed a series of videos, each with the purpose of explaining a single segment of code written in Python. Three different introductory programming topics were addressed: assignment with mathematical operators, nested selection (if) statements, and finding an element within a collection (using a for loop). In addition, an appropriate context or real life scenario was derived to motivate the problem. In video 1, the problem is summing lines of an invoice, calculating the tax due and then the final total for the invoice. In the second video, the code determines whether or not a donor and recipient have compatible blood types. The final video presents how to find the next possible movie time from a list of movie times.

3.1 Instructional Materials

Within each video, the problem was presented followed by an explanation of the code. Each explanation presented the overall solution outline followed by an explanation of each line of code, much like an instructor would do in class. Each video then concluded with an explanation of one or more traces of execution of the code with sample values.



Recipient	Donor							
	O−	O+	A−	A+	B−	B+	AB−	AB+
O−	✓	✗	✗	✗	✗	✗	✗	✗
O+	✓	✓	✗	✗	✗	✗	✗	✗
A−	✓	✗	✓	✗	✗	✗	✗	✗
A+	✓	✓	✓	✗	✗	✗	✗	✗
B−	✓	✗	✗	✓	✓	✗	✗	✗
B+	✓	✓	✗	✗	✓	✓	✗	✗
AB−	✓	✗	✓	✗	✓	✗	✓	✗
AB+	✓	✓	✓	✓	✓	✓	✓	✓

Red blood cell compatibility table

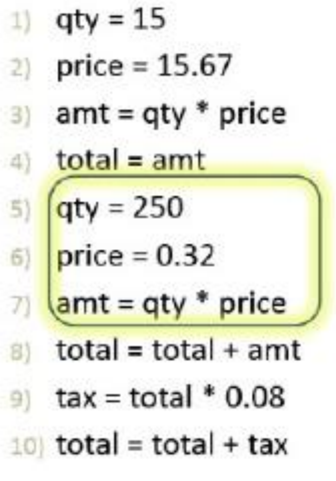
Figure 1: Red Blood Cell Compatibility Table

After each video the participant was asked a series of questions. The first question always asked the purpose of the code segment for verification of understanding. This was followed by one or more recall questions concerning the purpose of variables or interpretation of a given line of code. One or more application questions were then presented, asking the participant to predict the output for a segment of code from the original example. The last questions were transfer questions, asking the learner to apply their new knowledge to novel problems. All transfer questions were taken in their original form or adapted from [1]. A final question allowed the participant to indicate if there were any technical issues with the video. This added up to a total of eight questions after each video.

3.1.1 Video 1. The first example is straightforward using mathematical operators and the assignment statement to compute the total price from quantities and prices in an invoice. After the video there were four recall questions concerning the purpose of the code, number of invoice lines processed, and the purpose of variables. There was one application question concerning the ability of a variable to appear on both sides of an assignment type. This was followed by two transfer questions involving assignment statements only. The final question asked about technical difficulties.

3.1.2 Video 2. The second example involves nested selection statements and determining if a donor's and recipient's blood types are compatible, including the Rh factor. Participants were shown the problem definition along with a chart indicating blood type compatibility (Figure 1). Two examples were described on how to read and interpret the table. The code was then explained, followed with two examples tracing through the code, one for a compatibility match and one for an incompatible match. Recall questions covered the purpose of the code, possible values for a boolean variable, the name of the function, and the second thing checked for blood compatibility. The one application question involved tracing a portion of the nested selection statements. Two transfer questions were then asked, both with nested selections statements. The final question asked about technical difficulties.

This second example was written to include a "main" program along with a function call and function definition. This was done purposefully to allow for easy changing of the values of the variables for compatibility testing.



```
qty = 15
price = 15.67
amt = qty * price
total = amt
qty = 250
price = 0.32
amt = qty * price
total = total + amt
tax = total * 0.08
total = total + tax
```

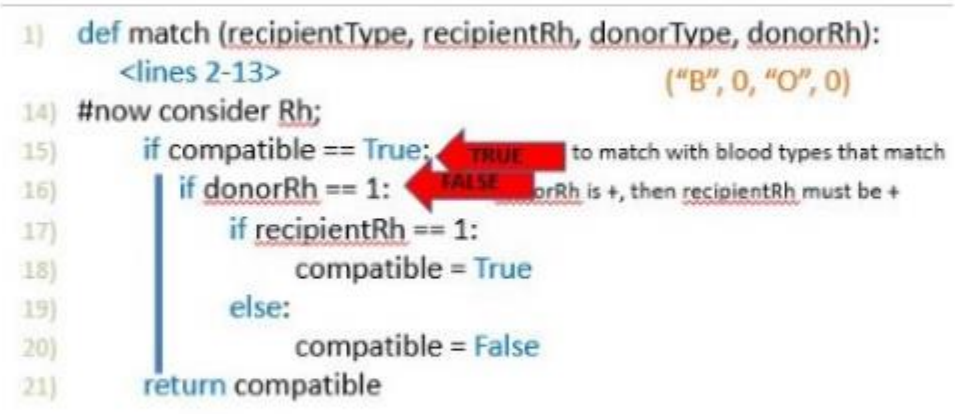
Figure 2: Signaling Example

3.1.3 Video 3. The third video describes finding the next possible movie time from a list of non-sequential movie times and involved a loop. The participant was given the problem definition and an outline of the solution approach followed by an explanation of the code. The video concluded by tracing through two executions with sample values. The recall questions for this video were the purpose of the code, understanding the solution, and representation of the data. The application question asked the user to determine what would happen if all the movie times had already passed for the day. There were the usual three transfer questions and the - final question on technical difficulties. All loops in the example and questions used the for loop format.

3.1.4 Design Considerations. It should be noted that the videos were designed with the purpose of minimizing cognitive load. During the explanation of the code, there was signaling indicating which line or lines were being discussed (Figure 2). When examples were being traced, the variable values and results of comparisons were integrated into the code diagram (Figure 3). There were no code comments for the first video, minimal (2 lines) of comments for the second video, and only a few lines of comments for the third and most complex video.

For each problem solved, the exercise was presented and followed by an algorithm, then the initial code. After the explanation of the code, a sample trace using test values was explained. The second example given in each video allowed a pause for participants to attempt to trace through the code on their own before the solution was presented.

A script for the code explanation was written for each problem. For each video, three different versions were created: one with an audio only code explanation (no text explanation other than the code), one with a text only explanation (no sound), and one that combined both the text explanation and the audio explanation simultaneously. (See Figure 4 to illustrate the text only condition.) The line(s) of code currently being explained was highlighted in all three treatments. At the conclusion of each video, a summary page was presented.



```
1)  def match (recipientType, recipientRh, donorType, donorRh):
        <lines 2-13>                                    ("B", 0, "O", 0)
14) #now consider Rh;
15)         if compatible == True:                to match with blood types that match
16)             if donorRh == 1:                    orRh is +, then recipientRh must be +
17)                 if recipientRh == 1:
18)                         compatible = True
19)                 else:
20)                         compatible = False
21)         return compatible
```

Figure 3: Code Tracing Example

The time, with one exception, was controlled for within each treatment and participants were instructed not to pause the videos except where instructed. The time spent on each screen was constant for all three versions. It was determined how long it took to read the current text on the screen using the reading time of an average 17 year old (plus a slight delay), and the audio was controlled to match that time. The one exception was when the participant was asked to trace through the code for the second example in each video. The instructions asked the user to pause the video while the code was showing and walk through the example. They were asked to continue the video when they knew the expected output.

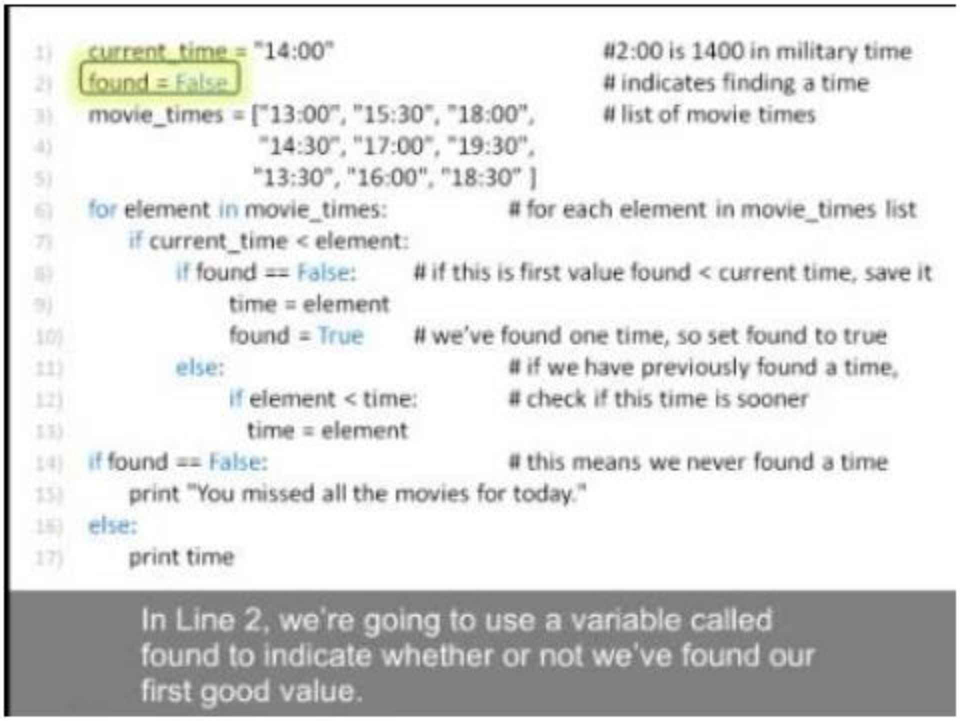
3.2 Participants

Participants were recruited from introductory programming courses or breadth-first (CS0) courses at multiple universities in the southeast United States via the internet or class announcements. Having read and given consent, participants were given a pre-test in order to eliminate those that had too much programming knowledge. Based upon the day of their birth, they were assigned to one of the three study conditions (audio only, text only, or both audio and text). After viewing each video they were asked to complete the CS cognitive load questionnaire [35], followed by a series of questions designed to determine how much information they recalled, how much they could apply, and how well they could transfer the knowledge. The only compensation participants received was an entry into a raffle for an Amazon gift card. At the conclusion of watching the videos, they were asked a series of demographic questions. The demographic questions were moved to the end to prevent stereotype threat [46, 47]. For the participants that answered the demographic questions, the average age of participants whose data was analyzed was 21.04, with a minimum of 18 and a maximum of 42. The median age was 19. In terms of their native language, 39 of the participants spoke English, 6 spoke Dutch, 2 spoke Korean, and 8 spoke other languages.

4 DATA ANALYSIS AND RESULTS

Data was collected from August 13, 2014 to November 29, 2014. See Table 1 for participant numbers and Table 2 for participants disaggregated by treatment.

Many participants did not view all three videos (Figure 5). Participants were given the option to quit the study after answering questions for each video. When piloting this study, the most common complaint was the overall length of the study, which caused participants to exit before completion. To improve participation, the incentive for the participants was changed to one raffle entry for each video and set of questions completed, and an extra raffle entry if all three videos were completed. If the participant chose to quit after answering questions for a video they were routed to the demographic questions in an effort to collect that information from all participants. Generally, if the participant continued on to the second video, they watched all three videos.



```
1)  current_time = "14:00"                          #2:00 is 1400 in military time
2)  found = False                                   # indicates finding a time
3)  movie_times = ["13:00", "15:30", "18:00",       # list of movie times
4)                 "14:30", "17:00", "19:30",
5)                "13:30", "16:00", "18:30" ]
6)  for element in movie_times:             # for each element in movie_times list
7)      if current_time < element:
8)          if found == False:      # if this is first value found < current time, save it
9)              time = element
10)             found = True        # we've found one time, so set found to true
11)         else:                           # if we have previously found a time,
12)             if element < time:          # check if this time is sooner
13)               time = element
14) if found == False:                          # this means we never found a time
15)     print "You missed all the movies for today."
16) else:
17)     print time
```

In Line 2, we're going to use a variable called found to indicate whether or not we've found our first good value.

Figure 4: Modality Study Example

Day of the month was used to randomly assign participants to a treatment because the survey package used, SurveyMonkey, did not support random assignment of participants to treatments. This resulted in an unequal assignment of participants to treatments. 36 participants were assigned to the text group, 30 were assigned to the audio group, but only 22 were assigned to the "both” group. Thus for the text only group, 62% of the participants assigned to the treatment went on to watch the first video and have their data analyzed. For the audio only group, 80% of the participants completed the first video and had data analyzed. However for the "both” treatment, only 50% of the participants assigned to the treatment had their data analyzed. This may be because these participants found having both video and audio explanations confusing or experienced cognitive overload and thus chose to end their participation in the experiment early.

The average and standard deviation were calculated for the cognitive load components for each video per treatment (Table 3). In all three groups the germane load (GL) was perceived as the highest component and the extraneous load (EL) was perceived as the lowest component. In addition video 2 consistently had the highest, intrinsic load (IL) measure. A statistical analysis was performed to determine if correlations existed between the cognitive load factors and treatment and all results were statistically insignificant (IL, p = .375, EL, p =.715, GL, p = .628).

Table 1: Participant Numbers

	N	Comment
Consented	141	
Non blank answers	99	Participants with all blanks removed
Novices (pre-test)	88	Those with a score of 67% or better (6 out of 9 possible points) were eliminated for having too much knowledge
Assigned treatment	77	Birth date question answered
Answered post video	61	Answered all video questions
Demographic data	55	Answered demographic data

Table 2: Participants by Treatment

Treatment	N
Audio	24
Text	26
Both	11

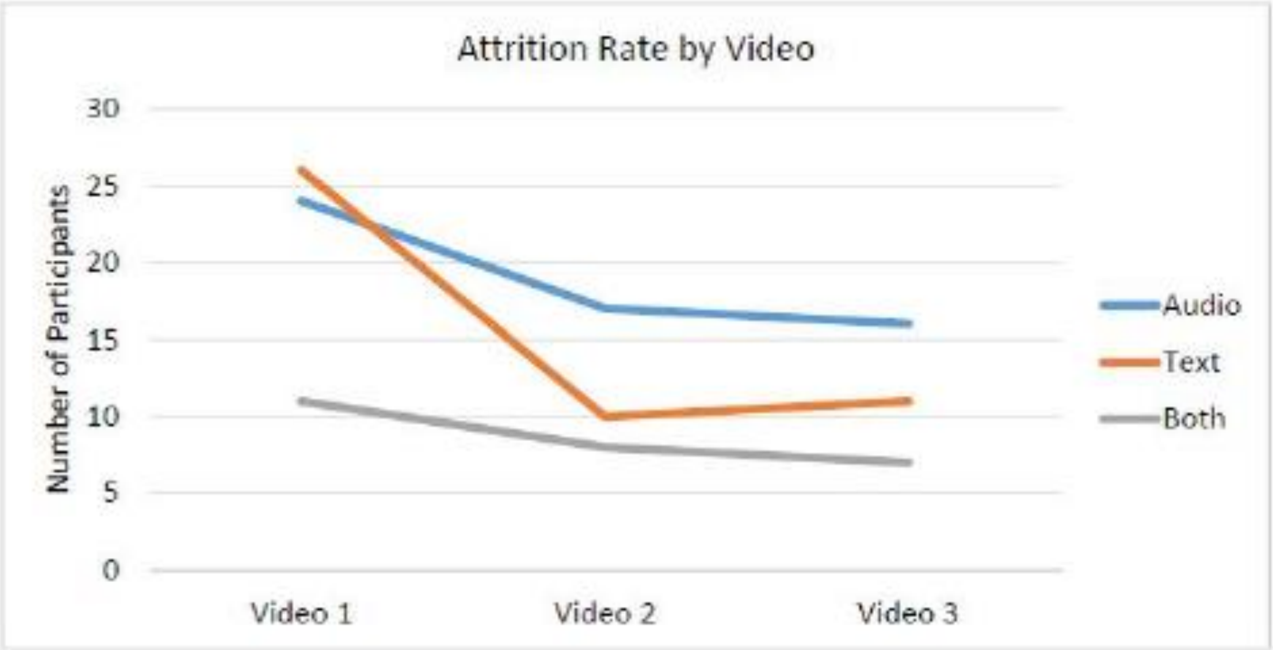


Figure 5: Participant Attrition by Video

Table 3: Cognitive Load Components by Video / Treatment

Treatment		Video 1	Video 2	Video 3
Audio	N	24	17	16
	IL avg (stddev)	3.35 (3.12)	4.69 (3.50)	3.92 (2.70)
	EL avg (stddev)	1.75 (2.05)	1.85 (2.38)	2.23 (2.78)
	GL avg (stddev)	6.10 (2.77)	5.18 (3.36)	6.09 (2.93)
Text	N	26	10	11
	IL avg (stddev)	3.31 (3.05)	4.27 (2.32)	4.27 (2.54)
	EL avg (stddev)	2.94 (2.88)	3.17 (2.55)	2.85 (2.44)
	GL avg (stddev)	6.04 (2.88)	5.5 (2.36)	5.34 (2.56)
Both	N	11	8	7
	IL avg (stddev)	2.55 (3.08)	3.92 (3.02)	2.62 (2.27)
	EL avg (stddev)	1.70 (2.04)	1.96 (2.06)	2.10 (2.76)
	GL avg (stddev)	6.14 (2.51)	6.34 (2.06)	6.46 (2.3)

Using learning performance as an indirect measure of cognitive load, we looked at the post-test results. The post-test questions for each video were scored for correctness by the author. For the open-ended purpose question, a rubric was developed and answers were scored for correctness out of 4 points. For multi-select multiple-choice questions, the number of incorrect choices was subtracted from the number of correct choices to get a final score. The results for each video by treatment can be seen in Table 4. For the purpose question, the average score (out of 4) is given along with the percentage of answers that were completely correct. The remaining questions were multiple choice questions and the percentage of correct answers are given.

A statistical analysis was done to determine if any correlations existed between treatment and participant performance. All results were statistically insignificant. There was no main effect for treatment, F (2, 52) = 0.178, MSE = 1.145, p = .837.

We grouped the results by question type to explore if there was a difference in performance based on treatment and question type. Recall that based on prior research, we would expect the audio only condition to perform the best in general and the group that received explanations in both audio and text to perform the worst.

Table 4: Post Test Results of Modality Study

		Treatment		
		Audio	Text	Both
Video 1	Purpose (% correct)	3.13 (54)	2.57 (42)	2.46 (27)
	first question (% correct)	46	23	18
	second question	83	46	64
	third question	50	46	45
	fourth question	42	23	45
	fifth question	33	38	36
	sixth question	17	46	36
Video 2	Purpose (% correct)	3.06 (65)	3.4 (80)	3.5 (88)
	first question (% correct)	47	60	50
	second question	47	20	75
	third question	82	100	75
	fourth question	47	90	50
	fifth question	6	40	25
	sixth question	41	60	38
Video 3	Purpose (% correct)	3.29 (56)	2.91 (64)	2.29 (29)
	first question (% correct)	56	73	57
	second question	19	9	0
	third question	75	73	43
	fourth question	44	55	43
	fifth question	19	18	29
	sixth question	19	27	29

As can be seen in Figure 6 which examines only the purpose question, video 2 had the best performance even though it was the longest video. While video 1 displays the expected performance based on treatment, the other videos do not. In fact, video 2 shows the exact opposite performance of what would be expected from prior research. In looking at the questions that asked learners to recall learned information (Figure 7), the average performance across all three videos is the expected result. However, in looking at each individual video, the results do not match prior research. It is interesting to note that video 3, arguably considered the most difficult concept (loops), had the best performance for both the audio and text groups. The final category of questions asked learners to transfer their newly acquired knowledge to novel problems and can be seen in Figure 8. None of the groups performed particularly well on the transfer problems, with a maximum of 50%. Interestingly, the text only group performed the best on the transfer questions.

The results from this study do not match what was found in the original studies and what we expected to find in this study. In the auditory only group, the code explanations were given aurally only along with color coded signaling. This was the group that was expected to have the best performance, especially with recall questions. If the modality principle was to hold in computer science, the audio treatment participants should have scored significantly higher statistically than the other two groups. The group that received both written text and an auditory explanation was expected to perform the worst on the learning performance tasks. While we expected these predictions to hold for all three videos, it should have held for at least the rst and most simple of the videos, but it did not.

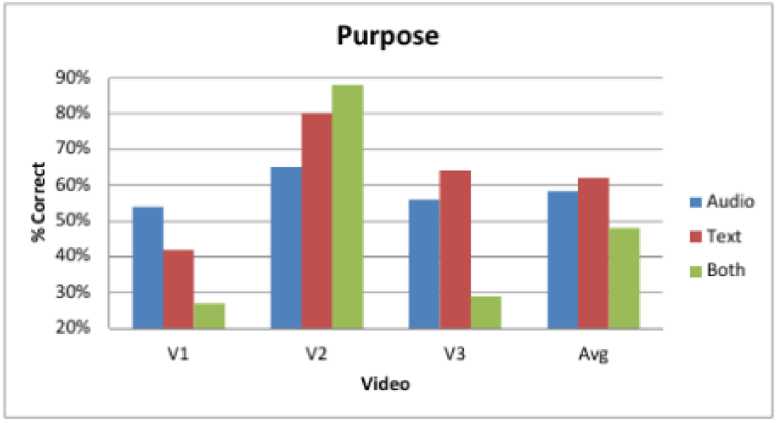


Figure 6: Performance on Purpose Question by Treatment

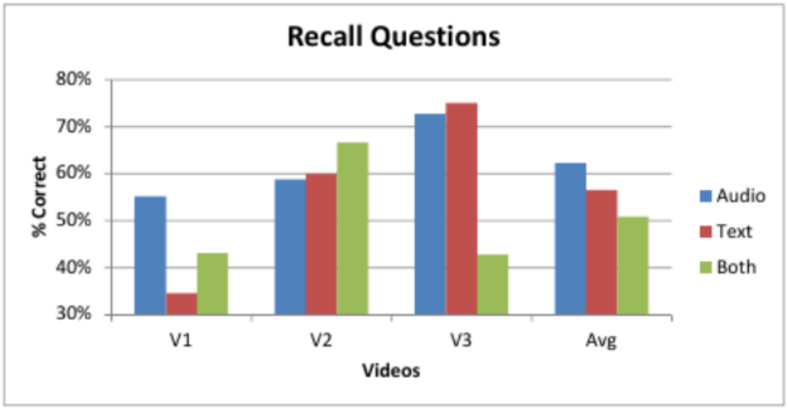


Figure 7: Performance on Recall Questions by Treatment

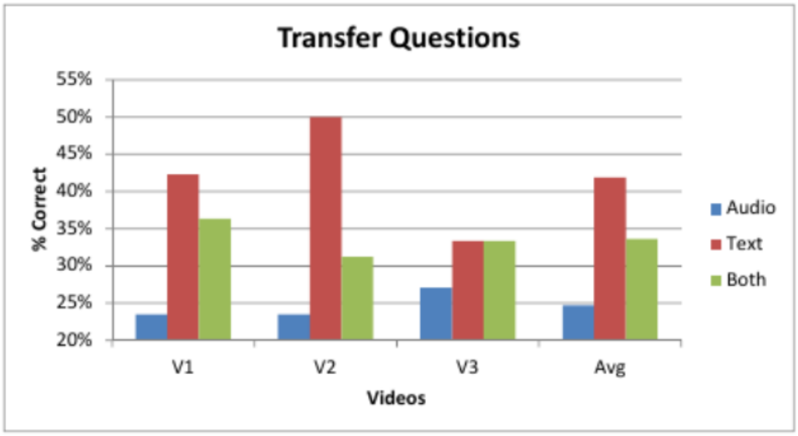


Figure 8: Performance on Transfer Questions by Treatment

To answer the research question, **we find no evidence to support either hypothesis: H1: Students receiving oral explanations will demonstrate better retention. H2: Students receiving both oral and text explanations will demonstrate the worst retention**. We find no evidence that altering the modality (text, oral, both) of code explanations improves student learning as measured by retention and transfer questions.

5 DISCUSSION

Seeing that the anticipated results were not obtained, we must examine if it is because the modality principle does not hold within the programming domain or if there are other explanations. It is possible the unanticipated results were obtained due to low and uneven participant rates. It may be possible to obtain the predicted results with additional and more evenly distributed participants across the treatment conditions. While care was taken to design the instructional material to produce the modality effect, it is possible that one of the recommendations may have been violated. We will examine each one in turn.

5.1 Separate Information

The recommendation to create a modality effect is that diagrammatic and textual information refer to each other and be unintelligible unless they are processed together. Kalyuga [23] posits two conditions when the modality effect may not be found: (1) when equivalent auditory and visual explanations are presented concurrently, and (2) when the instructional format is not matched to learner experience. While we attempted to ensure that the audio and text explanations were synchronized, there were two comments indicating that the participants' native language was not English and they were unable to keep up with the speed of the information presented. One of the participants was in the audio only group and the other was in the text only group. We do not believe this to be a likely explanation of the results.

5.2 Element Interactivity

It is recommended that element interactivity must be high, but not excessive. Certainly the elements of program code rely upon each other indicating interactivity between the elements. However, examples were chosen to limit the number of elements having to be retained in memory at a single time. The least amount of interactivity occurred in video 1, which had the worst overall performance on the recall questions (Figure 7). Therefore we do not believe this to be a likely explanation of the results.

5.3 Limit Auditory Length

The recommendation is that auditory text should be limited and that any lengthy, complex text should be written, not spoken. This recommendation is because of the transient effect [27, 64], or the amount of time auditory information can be retained. The participants in the study were novices with minimal computing knowledge (only those that failed the pre-test had data analyzed). It is possible that even the easiest and shortest of the videos overloaded their cognitive mental processing abilities. Knowing that we can only hold information in memory for no longer than 20 seconds [14], the videos may have been too long for the participants to comprehend and understand all the material asked of them. Video 1 was 5 minutes long; the second video was almost 23 minutes long and video 3 was 12 minutes long.

However in [27] the modality effect occurred when the videos were longer (going from 605 seconds to 867 seconds) but with fewer words (668 words to 576 words). In Leahy and Sweller's second experiment the explanation was simplified into smaller segments and less complex sentences. For this experiment, video 1 contained 528 words in 14 slides, video 2 contained 3167 words in 63 slides, and video 3 contained 1901 words in 33 slides. While videos 2 and 3 were lengthy in both time and words, video 1 was completely in line with those used in previous studies which reported these values and thus should have produced the modality effect.

Video 2 produced the most correct answers for the purpose question, even though it was the longest of the three videos. This would seem to contradict the transient effect. However, it may be due to the length of the video that the purpose was cemented in the learner's memory (more time on task). The text only explanation group performed the best on the transfer questions (Figure 8) which lends support that complex material should be written for better learning concerning transfer. It is unclear the role that auditory length plays in producing the modality effect in programming examples.

5.4 Complex Diagrams

If diagrams are complex, cuing or signaling may be required. This was accounted for within the videos through both cuing and signaling (Figures 2 and 3). It is also possible that no modality effect was found due to the complexity of the material. Kalyuga [25] provides an excellent overview of the modality effect and studies which find and do not find the modality effect in an effort to identify factors in-influencing its existence. The one consistent finding is to not use spoken explanations for any material which is highly complex. In this study, it could be reasoned that learning programming is so complex that no modality effect was found. However, if the material is truly inherently complex then the text only group should have performed better. This also did not occur, lending support that complexity of the diagrams or examples did not cause the unanticipated results.

5.5 Other Possible Explanations

Each video was designed to be a replacement for an in class lecture for that content topic. It is possible that each example was overreaching in its goal. Instead of trying to instruct an entire topic in a single video, it may be more effective to concentrate on a small piece of a topic per video. In other words, rather than covering every possible aspect of a selection statement, it may be more effective to have a shorter video that only explains what happens when a conditional expression evaluates to true in a selection statement. Then, a separate video describes when it evaluates to false, and yet another video for a nested selection statement. This would make each video much shorter and allow for progressive building of the information. This approach would support limiting the length of the auditory information.

Another possibility may be that novice programmers do not view code as a diagram. As Morgan et. al state, “Little is known about the progression of the process involved as novices become experts.”[34, p. 15]. The code most likely to be viewed as a diagram was within video 2 as it had a method/function call (or two separate sections) and would require just-in-time inquiries about the code, as predicted by [30], though questions related to video 2 did not yield a learning performance any better than the others.

To summarize, there are several possible explanations why the results from this experiment did not confirm the results from other disciplines. However, we have no definitive answer on whether there was a confounding variable causing the unanticipated results or if the modality principle does not hold for programming. This leads to the need for additional research.

6 FUTURE STUDIES

Below are a few different possible next steps to attempt to determine when, if, and how the modality principle applies to introductory programming:

- Think Aloud Study: Recruit a minimum of three participants for each treatment and have them watch the instructional videos used in this study. Instead of controlling for time, the researcher could pause the video at different times and ask the participant to describe aloud about what they have learned, what they can remember, and probe about possible transient effects. As the participants are answering the post video questions, they would be prompted to think aloud their thought process on how they are arriving at their answers. By capturing information during the learning and assessment process we may be able to more accurately determine what is occurring and explain the unanticipated results.

- Restructure the Videos: It may be possible to rework one of the three videos into separate and more distinct content topics as described previously. Another set of of participants could be recruited to determine if shortening the videos and scaling back the content makes a difference in the overall results.

- Use other measurement techniques to determine the cognitive load. We may be able to use eye-tracking to determine exactly where the learners are looking on the screen to determine how they view code segments. We may also be able to determine exactly when and where the cognitive processing is occurring by examining the frequency and duration of gaze at specific areas of interest within the code segment.

7 CONCLUSION

It is possible that the modality effect does not hold within learning programming, but we cannot conclude that based on this study alone. This initial evidence indicates that simply replicating existing studies is not enough. Instead, more information is needed on several topics. First, the transient effect within programming should be studied to determine at what point the learner begins to lose information. Exactly how much content can be explained in a single audio explanation should be evaluated as well. It may be possible to cover only the most simple, straightforward case for a control structure without deviating to the exceptions or even including a trace of code. Additional studies using other measurement techniques, such as eye-tracking, may be utilized to more precisely pinpoint when and where the cognitive processing occurs.

Knowing whether or not the modality effect holds in programming is important to all those teaching and developing instructional materials for introductory programming. In traditional classrooms many instructors do “live coding,” assuming that presenting the code on the overhead display and explaining the code is an effective pedagogical technique. Instructional programming videos consistently show the code with verbal explanations and expect students to learn and retain the information. Yet as this study illustrates, we still have no empirical evidence that this dual modality teaching technique is effective for learners of introductory programming. More studies should be conducted.

ACKNOWLEDGMENTS

The author thanks the students who participated in the study and their instructors who helped to recruit them. Thanks also to my advisor, Mark Guzdial, under whose guidance this study was conducted. Many thanks to the reviewers whose suggestions made this a better paper.

REFERENCES

[1] Alireza Ahadi and Raymond Lister. 2013. Geek genes, prior knowledge, stumbling points and learning edge momentum: parts of the one elephant?. In Proceedings of the ninth annual international ACM conference on International computing education research. ACM, 123–128. http://dl.acm.org/citation.cfm?id=2493416

[2] Pavlo D Antonenko and Dale S Niederhauser. 2010. The in-uence of leads on cognitive load and learning in a hypertext environment. Computers in Human Behavior 26, 2 (2010), 140–150.

[3] Paul Ayres. 2006. Using subjective measures to detect variations of intrinsic cognitive load within problems. Learning and Instruction 16, 5 (2006), 389 – 400.

[4] Paul Ayres and John Sweller. 1990. Locus of difficulty in multistage mathematics problems. The American Journal of Psychology (1990).

[5] Paul L Ayres. 2001. Systematic mathematical errors and cognitive load. Contemporary Educational Psychology 26, 2 (2001), 227–248.

[6] Alan Baddeley. 1986. Working Memory. Oxford University Press.

[7] J. Bransford. 2000. How people learn: Brain, mind, experience, and school. National Academies Press. http://books.google.com/books?hl=en&lr=lang_en&id=WaCW7i92lYkC&oi=fnd&pg=PA1&dq=How+people+learn&ots=JsRMAzrP8r&sig=q_klm0JlS2ATIqrFP-CnnXCsE64

[8] John D. Bransford, Ann L. Brown, and Rodney R. Cocking. 2000. How People Learn: Brain, Mind, Experience, and School (expanded ed.). National Academy Press, Washington, D.C.

[9] Roland Brunken, Jan L Plass, and Detlev Leutner. 2003. Direct measurement of cognitive load in multimedia learning. Educational Psychologist 38, 1 (2003), 53–61.

[10] S. N. Cant, D. Ross Jeery, and Brian Henderson-Sellers. 1995. A conceptual model of cognitive complexity of elements of the programming process. Information and Software Technology 37, 7 (1995), 351–362. http://www.sciencedirect.com/science/article/pii/095058499591491H

[11] Paul Chandler and John Sweller. 1991. Cognitive load theory and the format of instruction. Cognition and instruction 8, 4 (1991), 293–332.

[12] Paul Chandler and John Sweller. 1992. The split-attention eect as a factor in the design of instruction. British Journal of Educational Psychology 62, 2 (1992), 233–246.

[13] Paul Chandler and John Sweller. 1996. Cognitive load while learning to use a computer program. Applied cognitive psychology 10, 2 (1996), 151–170.

[14] N. Cowan. 2010. The magical mystery four how is working memory capacity limited, and why? Current directions in psychological science 19, 1 (2010), 51–57.

[15] Krista E DeLeeuw and Richard E Mayer. 2008. A comparison of three measures of cognitive load: Evidence for separable measures of intrinsic, extraneous, and germane load. J. of Educational Psychology 100, 1 (2008), 223.

[16] Peter Gerjets, Katharina Scheiter, and Richard Catrambone. 2004. Designing instructional examples to reduce intrinsic cognitive load: Molar versus modular presentation of solution procedures. Instructional Science 32, 1-2 (2004), 33–58.

[17] Peter Gerjets, Katharina Scheiter, and Richard Catrambone. 2006. Can learning from molar and modular worked examples be enhanced by providing instructional explanations and prompting self-explanations? Learning and Instruction 16, 2 (2006), 104–121.

[18] Paul Ginns. 2005. Meta-analysis of the modality effect. Learning and Instruction 15, 4 (2005), 313–331. http://www.sciencedirect.com/science/article/pii/S0959475205000459

[19] Fernand Gobet and Herbert A. Simon. 1998. Expert chess memory: Revisiting the chunking hypothesis. Memory 6, 3 (1998), 225–255. http://www.tandfonline.com/doi/abs/10.1080/741942359

[20] M. E. Hansen, A. Lumsdaine, and R. L. Goldstone. 2013. An experiment on the cognitive complexity of code. In Proceedings of the Thirty-Fifth Annual Conference of the Cognitive Science Society.

[21] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. Advances in psychology 52 (1988), 139–183.

[22] Robin Jeries. 1982. A comparison of the debugging behavior of expert and novice programmers. In Proceedings of AERA annual meeting.

[23] Slava Kalyuga. 2000. When using sound with a text or picture is not beneficial for learning. Australasian Journal of Educational Technology 16, 2 (2000). http://ascilite.org.au/ajet/submission/index.php/AJET/article/view/1829

[24] Slava Kalyuga. 2011. Cognitive load theory: How many types of load does it really need? Educational Psychology Review 23, 1 (2011), 1–19. http://link.springer.com/article/10.1007/s10648-010-9150-7

[25] Slava Kalyuga. 2012. Instructional benefits of spoken words: A review of cognitive load factors. Educational Research Review 7, 2 (2012), 145–159. http://www.sciencedirect.com/science/article/pii/S1747938X11000546

[26] Slava Kalyuga, Paul Chandler, and John Sweller. 1999. Managing split-attention and redundancy in multimedia instruction. Applied cognitive psychology 13, 4 (1999), 351–371. http://www.researchgate.net/prole/Paul_Chandler3/ publication/238680916_Managing_split-attention_and_redundancy_in_ multimedia_instruction/links/0a85e5300f4b3b95d6000000.pdf

[27] Wayne Leahy and John Sweller. 2011. Cognitive load theory, modality of presentation and the transient information effect. Applied Cognitive Psychology 25, 6 (2011), 943–951. http://onlinelibrary.wiley.com/doi/10.1002/acp.1787/pdf

[28] Jimmie Leppink, Fred Paas, Cees PM Van der Vleuten, Tamara Van Gog, and Jeroen JG Van Merriënboer. 2013. Development of an instrument for measuring different types of cognitive load. Behavior research methods 45, 4 (2013), 1058– 1072.

[29] Jimmie Leppink, Fred Paas, Tamara van Gog, Cees PM van der Vleuten, and Jeroen JG van Merriënboer. 2014. Effects of pairs of problems and examples on task performance and different types of cognitive load. Learning and Instruction 30 (2014), 32–42.

[30] Stanley Letovsky, Jeannine Pinto, Robin Lampert, and Elliot Soloway. 1987. A cognitive analysis of a code inspection. In Empirical studies of programmers: second

workshop. Ablex Publishing Corp., 231–247. [31] R. E. Mayer. 2002. Multimedia learning. Psychology of Learning and Motivation 41 (2002), 85–139. http://www.sciencedirect.com/science/article/pii/ S0079742102800056

[32] Richard E. Mayer. 2009. Multi-Media Learning (2nd ed.). Cambridge Univ Press.

[33] Roxana Moreno. 2004. Decreasing cognitive load for novice students: Effects of explanatory versus corrective feedback in discovery-based multimedia. Instructional science 32, 1-2 (2004), 99–113.

[34] Andrew Morgan, Bonita Sharif, and Martha E Crosby. 2015. Understanding a novice programmerâĂŹs progression of reading and summarizing source code. Eye Movements in Programming Education II: Analyzing the NoviceâĂŹs Gaze (2015), 13.

[35] Briana B. Morrison, Brian Dorn, and Mark Guzdial. 2014. Measuring cognitive load in introductory CS: adaptation of an instrument. In Proceedings of the tenth annual conference on International computing education research. ACM, 131–138. http://dl.acm.org/citation.cfm?id=2632348

[36] Seyed Yaghoub Mousavi, Renae Low, and John Sweller. 1995. Reducing cognitive load by mixing auditory and visual presentation modes. Journal of educational psychology 87, 2 (1995), 319. http://psycnet.apa.org/journals/edu/87/2/319/

[37] Fred Paas, Juhani E Tuovinen, Huib Tabbers, and Pascal WM Van Gerven. 2003. Cognitive load measurement as a means to advance cognitive load theory. Educational psychologist 38, 1 (2003), 63–71.

[38] Fred G Paas. 1992. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. J. of educational psychology 84, 4 (1992), 429.

[39] Fred GWC Paas and Jeroen JG Van Merriënboer. 1993. The eciency of instructional conditions: An approach to combine mental eort and performance measures. Human Factors: The Journal of the Human Factors and Ergonomics Society 35, 4 (1993), 737–743.

[40] Fred GWC Paas and Jeroen JG Van Merriënboer. 1994. Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive-load approach. Journal of educational psychology 86, 1 (1994), 122.

[41] Fred GWC Paas and Jeroen JG Van Merriënboer. 1994. Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive-load approach. Journal of educational psychology 86, 1 (1994), 122.

[42] Fred GWC Paas, Jeroen JG Van Merriënboer, and Jos J Adam. 1994. Measurement of cognitive load in instructional research. Perceptual and motor skills 79, 1 (1994), 419–430.

[43] Jan L Plass, Roxana Moreno, and Roland Brünken. 2010. Cognitive load theory. Cambridge University Press.

[44] Jan L Plass, Roxana Moreno, and Roland Brünken. 2010. Cognitive load theory. Cambridge University Press.

[45] Wolfgang Schnotz. 2005. An integrated model of text and picture comprehension. The Cambridge handbook of multimedia learning (2005), 49–69. https://books.google.com/books?hl=en&lr=&id=Cvw6BAAAQBAJ&oi=fnd&pg=PA72&dq=schnotz+2005&ots=HuUbWczY_Q&sig=keMb_F17xn6jwCMemW8k7LHNt0M

[46] Claude M. Steele and Joshua Aronson. 1995. Stereotype threat and the intellectual test performance of African Americans. Journal of personality and social psychology 69, 5 (1995), 797. http://psycnet.apa.org/journals/psp/69/5/797/

[47] Claude M. Steele, Steven J. Spencer, and Joshua Aronson. 2002. Contending with group image: The psychology of stereotype and social identity threat. Advances in experimental social psychology 34 (2002), 379–440. http://www.sciencedirect.com/science/article/pii/S0065260102800090

[48] Andreas Stek and Ed Gellenbeck. 2009. Using spoken text to aid debugging: An empirical study. In Program Comprehension, 2009. ICPC’09. IEEE 17th International Conference on. IEEE, 110–119.

[49] John Sweller. 1988. Cognitive load during problem solving: Eects on learning. Cognitive science 12, 2 (1988), 257–285.

[50] John Sweller. 1999. Instructional Design in Technical Areas. Australian Education Review, No. 43. ERIC. http://eric.ed.gov/?id=ED431763

[51] John Sweller. 2010. Element interactivity and intrinsic, extraneous, and germane cognitive load. Educational Psychology Review 22, 2 (2010), 123–138.

[52] John Sweller. 2010. Element interactivity and intrinsic, extraneous, and germane cognitive load. Educational psychology review 22, 2 (2010), 123–138. http://link.springer.com/article/10.1007/s10648-010-9128-5

[53] John Sweller, Paul Ayres, and Slava Kalyuga. 2011. Cognitive load theory. Vol. 1. Springer.

[54] John Sweller, Paul Ayres, and Slava Kalyuga. 2011. Cognitive load theory. Vol. 1. Springer.

[55] John Sweller and Paul Chandler. 1994. Why some material is dicult to learn. Cognition and instruction 12, 3 (1994), 185–233.

[56] John Sweller, Jeroen JG Van Merriënboer, and Fred GWC Paas. 1998. Cognitive architecture and instructional design. Educational psychology review 10, 3 (1998), 251–296.

[57] Sharon Tindall-Ford, Paul Chandler, and John Sweller. 1997. When two sensory modes are better than one. Journal of experimental psychology: Applied 3, 4 (1997), 257. http://psycnet.apa.org/journals/xap/3/4/257/

[58] Georey Underwood, Lorraine Jebbett, and Katharine Roberts. 2004. Inspecting pictures for information to verify a sentence: Eye movements in general encoding and in focused search. Quarterly Journal of Experimental Psychology Section A 57, 1 (2004), 165–182.

[59] Pascal WM Van Gerven, Fred Paas, Jeroen JG Van Merriënboer, and Henk G Schmidt. 2004. Memory load and the cognitive pupillary response in aging. Psychophysiology 41, 2 (2004), 167–174.

[60] Pascal WM van Gerven, Fred Paas, Jeroen JG van Merriënboer, and Henk G Schmidt. 2006. Modality and variability as factors in training the elderly. Applied cognitive psychology 20, 3 (2006), 311–320.

[61] Tamara Van Gog and Fred Paas. 2008. Instructional eciency: Revisiting the original construct in educational research. Educational Psychologist 43, 1 (2008), 16–26.

[62] Tamara van Gog and Katharina Scheiter. 2010. Eye tracking as a tool to study and enhance multimedia learning. Learning and Instruction 20, 2 (2010), 95–99.

[63] Jeroen JG Van Merriënboer and John Sweller. 2005. Cognitive load theory and complex learning: Recent developments and future directions. Educational psychology review 17, 2 (2005), 147–177.

[64] Anna Wong, Wayne Leahy, Nadine Marcus, and John Sweller. 2012. Cognitive load theory, the transient information effect and e-learning. Learning and Instruction 22, 6 (2012), 449–457. http://www.sciencedirect.com/science/article/pii/S0959475212000369