Introduction to Java in 4 Hours

Andrew D. Gramigna

College of Information and Computer Sciences, University of Massachusetts, Amherst

May 10th, 2018

Abstract

At UMass Amherst, many students across all disciplines will at some point take the course CS121: Introduction to Problem Solving with Computers. In this course students learn how to program in Java. No prior programming experience is required or expected. Due to the steep learning curve and difficulty of material for those with no prior programming knowledge, it can be easy for students to fall behind in CS121. The researcher hypothesized that making weekly 15-20 minute interactive videos going over a specific programming topic on a high level would be beneficial to students' comprehension.

The researcher was able to test this idea through an independent study, by making 12 weekly videos which were a screencast of the researcher going over a hands-on example with Java code using the jGRASP IDE. The intention was for the students to follow and code along. The 12 videos covered a variety of topics that aligned with what the students were learning. The videos, and accompanying code designed prior to recording, were posted on the course page online. At the end of the semester a survey was conducted to determine if the videos were effective and helpful to the students who watched them. Based on 25 responses, 18 called the videos "Very Helpful", and 7 saying they were "Helpful". That is a perfect success rate of those who responded showing that the videos were indeed helpful to many. With these results in mind, it seems likely that these videos can be used to help a new batch of students in future semesters.

1 Introduction

1.1 Background

During his first two semesters as a graduate student at UMass Amherst, the researcher was placed as a Teacher's Assistant (TA) for the course CS121: Introduction to Problem Solving with Computers. During these semesters, he consistently noticed many students he taught had difficulty with grasping the material in this course. There are many potential reasons for this. For one, the course material is very difficult with someone with no prior programming background. Additionally, the class is a popular class not just for Math and Computer Science majors, but also students of other disciplines who are just looking to get some coding experience. Also, this class is taken predominantly by first year students, many of whom struggle initially with the hands-off nature that comes with college classes [3].

From his experience as a TA, the researcher felt like students would do a lot better if they were able to see practical examples where they are putting what they learned for the week of lecture into actual code. He also felt like a 15-20 minute video to hammer down concepts learned in lecture would be very effective for retaining knowledge, similar to how one works out to stay in shape. These videos could be a way for someone to train their programming by coding hands-on once a week so they do not forget what they have learned. And since the videos were so short, the hope was student's would be more motivated to watch and get a quick refresher on the material to reinforce key concepts. Moreover, the researcher also took this course in his first semester as an undergraduate at UMass Amherst, and wished there was something like this to aid in his learning.

1.2 Justification

Interactive videos provide time and location flexibility for students, give them unlimited access to electronic learning material, and foster a self-paced learning environment. In most college classes very little visual information is presented: students mainly listen to lectures and read textbooks to learn material. Most people are visual learners, which means that most students do not learn as much as they would if more visual presentation was used in class [2]. A study on the effect of interactive videos in learning showed that "students in the e-learning environment that provided interactive video achieved significantly better learning performance and a higher level of learner satisfaction [compared to traditional methods]" [1]. Another interesting finding from this study showed that students who learned with non-interactive videos actually did not achieve a better learning performance when compared with traditional methods.

2 Methodology

Each week the researcher would meet with a professor with whom he was doing an independent study with to go over plans for each video. This was helpful because both people could collaborate to make a plan for the video. The researcher would record the videos over the weekend and release them early in the week to align with the material the students were learning.

2.1 Methods of Evaluation

At the end of the semester, a survey would be released to the students in the course so they could give feedback on the effectiveness of the videos. The survey was 5 questions, the first two asking how many (out of 12) videos the students estimated they watched, and the second was rating the effectiveness of the videos on a scale of 1-5. The remaining three questions were short answer questions, asking the students if they had anything in particular they liked about the videos, any suggestions they had for the videos, and any final comments. At the end of the semester, 25 students responded to the survey.

2.2 Process Of Creating Videos

Prior to recording the videos, the researcher would make a small, and usually incomplete, Java project that he would code in during the video and encourage the students to follow along. A .zip file of the starter code was available for students to download on the course webpage so they would be at the exact same starting position as the researcher. They could follow the code as the video went on, simply pausing whenever they needed to take more time to understand what was being explained. For each video, the accompanying Java project was intended to be quite simple, because the researcher thought it was important that the videos were accessible to those who were having a tougher time with the course.

Each video was made with an XLR microphone piped through a stereo mixer for optimal quality. This allowed for mic volume to be controlled at all times. The videos were one continuous take done through the screencast feature on the Mac app QuickTime Player. Each video started with a 3 slide PowerPoint presentation that outlined what would be talked about in the video. The researcher thought this would be helpful if students wanted to rewatch an old video to find something specific they were forgetting, they could just check the opening PowerPoint to see if it was the right video.

After the PowerPoint, the researcher would overview the starter code and then work though the Java project, detailing his thought process and giving tips along the way. Occasionally a portion would be left unfinished for the viewer to apply the concepts learned in the video on their own. Putting this all together, the intention was if a student wanted to watch a video, they should follow these simple steps:

1. Download the starter code and click on the video link on the course Moodle page.
2. Code along during the video, pausing when necessary to make sure they are not getting lost.
3. By the end of the video the student has seen a hands-on example of something they learned in lecture, and were able to code along themselves.

Week	Topic	Length
1	Installing Java JDK and jGRASP	18:35
2	Variables, Print Statements, jGRASP Features[1]	21:34
3	Objects and the Debugger	23:13
4	Scanner and if/else Statements	21:14
5	Methods	17:45
6	Looping	18:27
7	Arrays	21:51
8	Recursion	16:02
9	Inheritance	20:40
10	File I/O and Exceptions	19:00
11	Interfaces and Abstract Classes	22:09
12	Graphics	20:40

Table 1: The twelve videos created and the length of each

2.3 Videos

Table 1 details the 12 videos created and how long each video is. In total the 12 videos are 241 minutes and 50 seconds, or about 4 hours and 2 minutes. Each video averaged to be a bit over 20 minutes, which is slightly higher than the goal of 15-20 minutes, but sometimes it was necessary for to spend longer on a video to make sure certain concepts were reinforced well. Also, 20 minutes a week of extra content is certainly not a huge burden for a student.

3 Results

The results of the survey are shown below, beginning with the range of videos each student estimated they watched, as shown in Figure 1. This pie chart shows that almost 2/3 of the respondents watched 4 or more videos. The videos were formatted in a way that they do not reference previous videos, in order to give the students freedom to watch any amount. The next question rated the overall helpfulness of the videos, as shown in Figure 2 and Figure 3.

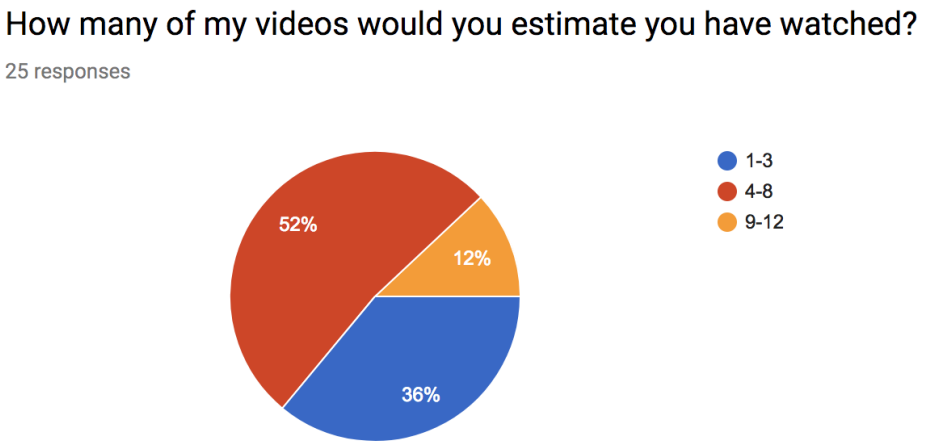

Figure 1: Pie chart of distribution of videos watched

[1]Features in this video include the jGRASP canvas and Interactions pane

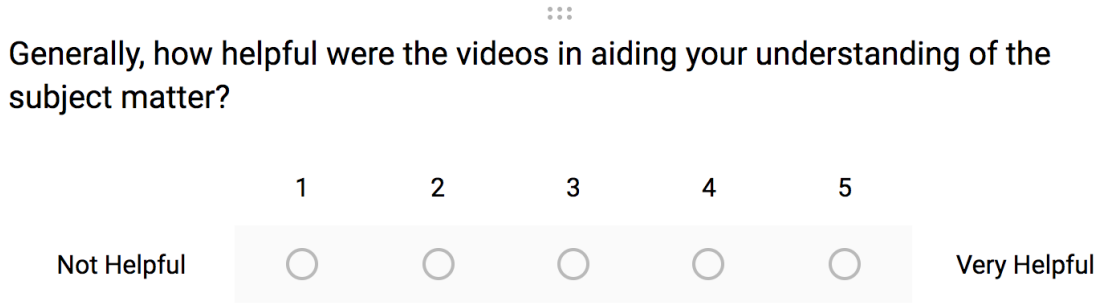

Figure 2: Question to determine the effectiveness of the videos.

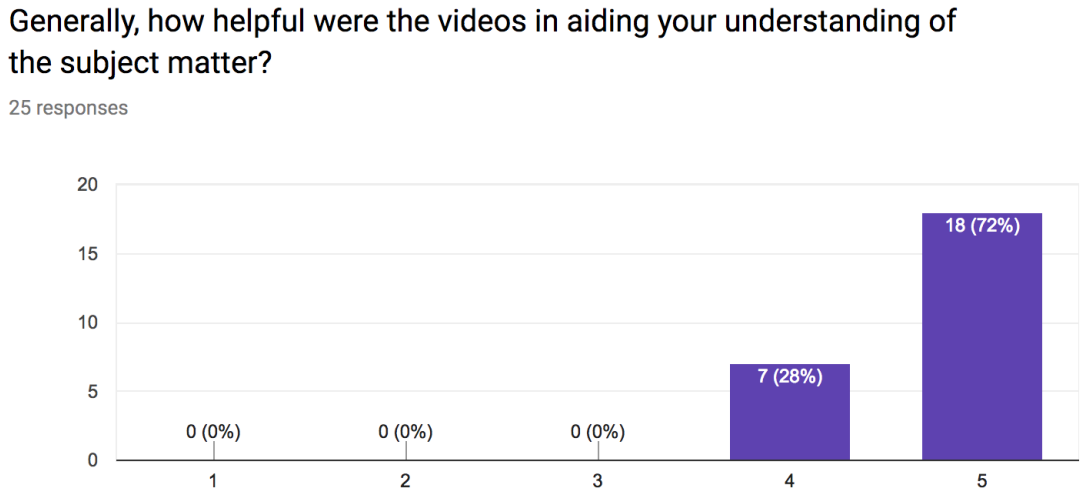

Figure 3: Bar Graph displaying the results of the question in Figure 2. Every respondent found the videos helpful.

The big takeaway from this independent study is found in the chart in Figure 3, is that every student who responded found the videos helpful, with 18/25 saying they were very helpful. This shows that the videos were able to positively impact all 25 of the respondents grasp on the material.

4 Analysis of Open Ended Questions

For the next part of the survey, the students were asked to give a couple sentences of feedback for particular qualities they liked, suggestions they had, and any final comments.

4.1 Positive Comments

Below are selected some responses that highlighted some of the best qualities of the videos.

> *"I liked that he went through the code slowly, line by line so I could actually understand what was going on. Also it was just helpful to see the ideas in class put into a real code.".*

The researcher's biggest goal when making these videos was to reinforce the material in class by providing a hands-on example. He wanted to go at a slow pace to appeal to any audience.

"I really enjoyed how he gave tips on how to write code while he was doing it... if I needed more help I would definitely watch [the videos], but as it was I didn't feel as though I needed them. For difficult material though they were very helpful!".

This comment shows that the videos were helpful even for students that were more proficient in this course. This student did not necessarily need the videos as supplemental help, but still found them effective at explaining difficult material.

4.2 Suggestions for Improvement

Additionally, below are selected responses that highlighted some suggestions for the videos.

"I would have liked it if Andrew worked on harder examples in the video.".

During the videos the researcher wanted to make them simple enough for anyone to follow, so while it would have been nice for some if the videos were harder, the main priority of the videos was for simple examples that anyone could use to reinforce basic skills. Though perhaps the difficulty could have been increased and the results would have been similar.

"Maybe give a set of practice problems with solutions at the end of the videos.".

A couple of students mentioned the idea of separate problems for them to try on their own. The researcher would have liked to do this, but found the workload of making the videos enough of a burden that there was simply not enough time to make any supplemental material.

4.3 Analysis of Final Comments

Finally, below are selected final comments the students had on the videos or the researcher's teaching

"I really appreciate all the effort put into the videos. They really helped me through some more difficult topics and its clear how much you wanted them to be helpful and well put together."

"Thank you! I have definitely downloaded one of the videos in particular so that I can look back on it for further reference. Private viewing only I promise! Thank you!"

A lot of effort went into the videos and to see that the students could see this and appreciated the effort is rewarding. Moreover, the fact that one student chose to download a video to look back on in the future truly further reinforces the hypothesis the the videos and the accompanying code were indeed helpful in aiding the students' learning.

5 Discussion

The results showed positive feedback. Additionally, most students chose to watch at least 4 videos, which shows they saw them as an effective study tool. Though only 25 students responded, there is not enough information to estimate the helpfulness of the videos to the rest of the class. Perhaps the videos were too difficult or too simple for the majorty of the class. However, The videos were helpful to at least 25 students, and hopefully will continue to be helpful to students in future semesters.

5.1 Issues

The biggest issue the researcher faced when making these videos was the amount of other classwork he had to cope with on a weekly basis. Sometimes, completing his own schoolwork was a priority over the videos, and occasionally this caused the videos to be released after their scheduled release date. However, this did not seem to be an issue with the finished product.

Another issue the researcher had to deal with when making these videos was the way they were recorded. The screencasts must be completed in one take to avoid looking unprofessional or unintentionally making a disparity between the researcher's code and the student's code. A couple times the researcher would make it far into a video and then be faced with an unexpected compilation error that he was not able to figure out quickly. This delay would force the video to be scrapped and another video must be made from scratch. Oftentimes, the researcher found that on his second attempt of the video would be a lot cleaner in general, so it was not a big deal to redo them if the issue arose. If he were to do this again, the researcher would not change his recording strategy even though it was frustrating at times.

5.2 Future Work

The researcher has no current plans for Future work with these videos. If another researcher would like to expand on this work, they could try to use the researcher's video style, which was shown to be effective, to go over more difficult Java concepts such as Stacks, Queues, and Linked Lists. Also, they could try to recreate these videos with practice problems at the end, or more difficult examples, as some of the students suggested.

6 Acknowledgments

I would like to thank Professor Neena Thota for sponsoring me to do these videos with an independent study. I was more or less given me complete control of the format of the videos and the lesson plan for the specific topic by Professor Thota. Her trust in me allowed me to create each video with my own vision, and the results showed that this payed off in a major way. I would also like to thank the 25 respondents of the final survey. These responses gave me the confidence that my teaching and my videos were helpful, and constructive suggestions were helpful for seeing what I could do to improve on these videos in the future. I am truly grateful to those who took the time to give me feedback on my work.

A Where to access the videos

As of the release of this paper, I do not feel comfortable making my videos accessible to the public. If you would like to see the videos, you may contact me directly at *adgramigna1@gmail.com* and I will share them with you, or contact Professor Neena Thota at UMass Amherst.

As long as you have a Umass student ID, and my email does not get taken down, my videos can be accessed here.

1 Installing Java JDK and jGRASP
2 Variables, Print Statements, jGRASP Features
3 Objects and the Debugger
4 Scanner and if/else Statements
5 Methods
6 Looping
7 Arrays
8 Recursion
9 Inheritance
10 File I/O and Exceptions
11 Interfaces and Abstract Classes
12 Graphics

References

[1] D. Zhang, L. Zhou, R. Briggs, J. Nunamaker Jr., *Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness*, 2005.
https://www.sciencedirect.com/science/article/abs/pii/S0378720605000170

[2] R. Felder and B. Soloman, *Learning Styles and Strategies*, North Carolina State University, 2000.
http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/styles.pdf

[3] T. Jenkins, *On the Difficulty of Learning to Program*, Univsersity of Leeds, 2002.
http://www.psy.gla.ac.uk/ steve/localed/jenkins.html