Performance comparison of programming learning curve between the use of Written text with Video digital media

Anas Alnayef (117260)
University of Bauhaus
Weimar, Germany
anas.alnayef@uni-weimar.de

He Ren (119181)
University of Bauhaus
Weimar, Germany
he.ren@uni-weimar.de

***Abstract*—This comparative study report of time-series analysis shows the performance of the two resources i.e. the use of written material and digital media in the form of video to calculate the time complexity to reach the result and the learning curve. To conclude the hypothesis, comparison analysis with test subjects is done with the written text resources and the digital media as video resources. With this study we want to analyze the probing factors to improve the learning rate with high satisfaction. The competitive study is conducted with the test subjects with less or no programming experience, and the results are computed with Mann-Whitney-U test. The programming setup was chosen which delivers user friendly learning and development environment with minimalistic options to ease the test subjects. Results concluded that the learning curve of the test subject is inclined towards the learning from the video-based media compared to the written text material. We adopted Living lab methodology to operationalize the research and quantitative data analysis to foster the research model.**

***Index Terms*—Digital media, resource, Mann-Whitney-U test, video, living lab, evaluation**

1. INTRODUCTION

Nowadays, there are an increasing number of businesses relying on computers and automation, which, as a result, adds a great importance to programming skills. For that reason, more novice people are interested in gaining theses skills. For a long time, books and written tutorials were the only resources available for that purpose. However, with the easier access to the internet we have these days, video and media tutorials have become one of the most popular online resources for e-learning. These relatively- new type of resources fill some gaps compared with written and text tutorials. Through video, one can engage multiple senses, which contributes to better and easy understanding of the materials being taught. Moreover, in the context of programming tutorials, watching a video about learning a certain task usually would include a demo of how the output would look like when executing the code; this gives the chance for the viewer to compare the output they get with the one in the video to make sure they are on the right track. On the other hand, it is arguable that the written tutorials have some advantages over the video ones, for instance: being able to search for keywords, copying and pasting the code, etc. Our expectation is that learning programming through video, especially for novices, can be faster and more helpful than the traditional books and written tutorials. Therefore, we wanted to conduct this study to see if there are enough notable differences to support the superiority of learning programming through video.

2. OBJECTIVE AND HYPOTHESIS

The goal of this study is to empirically explore the efficiency of using programming tutorials videos compared to the use of text and written resources. This includes presenting the research question, generating hypotheses, and conducting experiments that evaluate the hypotheses.

2.1. Research question:

The question that is asked in our study is: Is learning programming for novices more efficient using video tutorials compared with written ones?

2.2. Hypotheses:

From the research question, we generated the following hypotheses:

H0: Video tutorials are resources with the same level of efficiency or less compared with written ones to learn programming for novices

H1: Video tutorials are more efficient resources than written ones to learn programming for novices

2.3. Variables: Independent and dependent variables:

With the hypotheses generated, we want to test it by comparing the performance of two types of tutorial resources, which is hard to measure. Therefore, we had to think of variables that can help us in the comparison and they are operationalized as Table I shows.

Independent	Dependent
Using video tutorials/using written tutorials	Completion time, Error rate

TABLE 1
Independent and dependent variables of the study

The experiments were conducted in two stages (video, text) for the same programming task by two different groups of participants. During the experiments, we measured the task completion time and the error rate (number of thrown exceptions and times at which the participants had to ask us for help during the task).

2.4. Confounding factors:

Like any field experiment, there are some confounding factors that can affect the outcome of the dependent variables. And since we have a small sample size, which means these factors will have bigger effect, we had to think of the most important factors in the context of this study. The following are the confounding factors that might arise with the suitable ways of controlling each one:

- Level of programming experience: participants having varied experiences in programming, in any language, would certainly play a role in their performance regardless of what type of resources they use (video or text). To control this factor, we made sure that all of the participants have never coded before, i.e. students from academic backgrounds not related to computer science.
- Participant performance might be affected if they feel nervous by the experimenter: even though, as experimenters, we share the same room with the participant (to make sure they are only using one type of resources i.e. only videos in the video experiment), but we made sure that we keep a fair distance and that the participant doesn't notice that we are running a timer or taking notes of their progress.
- Typewriting speed: the participants' speed of typewriting must not be below average so they can be eligible for the experiment. For that, we conducted a small typewriting speed test before each experiment using an online tool
- Workstation's OS: some participants might find it difficult to operate on an OS different from what they're used to (UI, keyboard layout), which might also affect their performance negatively. Therefore, we provided each participant a workstation with an OS of their choice. Thus, they will be comfortable using the UI and the suitable keyboard layout with ease, i.e. for apple users; apple keyboard layout, pc users: pc keyboard layout.

3. EXPERIMENTAL DESIGN

As we have one independent variable (Learning programming) with two levels (Video and text), so the appropriate design for our experiments would be *between subjects*: two different groups, one performs the task using video, the other using text. Table II

Group 1	Experiment with video
Group 2	Experiment with text

TABLE 2
Between subject using two independent groups

This will insure that each group of participants does the task for the first time, thus, allowing us to obtain data that accurately reflect the performance of each participant, since this lowers the chances of participants suffering boredom after repeating the test twice (after all, it's a programming task) or, alternatively, becoming more accomplished through practice and experience, thus, skewing the results.

3.1. Research method:

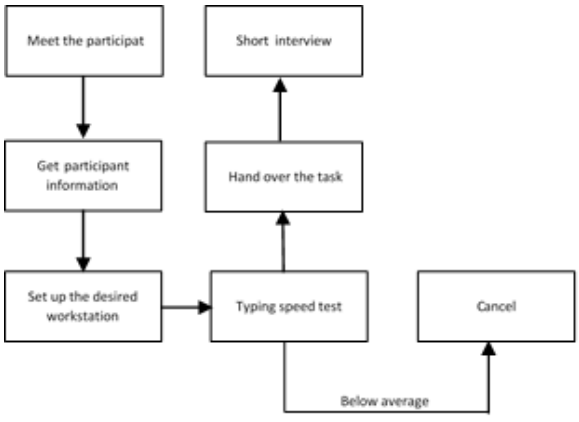

FIG 1
Flowchart diagram shooing the steps of the study

ID	Group	Academic background	Platform familiarity	Task completion time	Error rate	Typing speed avr(30 WPM)
P1	Text	Architecture	Mac os	10.08 min	1	43 WPM
P2	Text	Public Art	Windows	20.61 min	3	31 WPM
P3	Text	Major in music	Mac os	18.32 min	1	35 WPM
P4	Text	Public Art	Mac os	14.53 min	2	30.37 WPM
P5	Text	Urban planning	Mac os	12.08 min	3	44 WPM
P6	Vidoe	Public Art	Windows	9.15 min	1	37 WPM
P7	Vidoe	Urban planning	Windows	7.61 min	2	32 WPM
P8	Vidoe	Urban planning	Mac	9.78 min	1	39 WPM
P9	Vidoe	Civil engineering	Windows	9.05 min	0	31 WPM
P10	Vidoe	Civil engineering	Windows	12.16 min	2	46 WPM

TABEL 3

3.1.2. Preparation:

In order to collect the required data that would help us measure the variables of this study, we made one-time observation sessions with each participant, in which we give the participant a simple programming task, then a brief interview to get their impressions and feedback to understand certain events that happened during the observation. Before starting each session, we ask the participants for their names, academic background and whether or not they've ever coded before. Then, we briefly explain the goal of our study to the participants, and we emphasize on the fact that it's an exploratory study in which we do not expect them to complete the given task quickly and perfectly without any errors or having to ask for help along the way. Moreover, we promised them to protect their anonymity and confidentiality.

Since we settled on between objects design, we made sure that the first group of participants would perform the task using only video resources, while the second group would use any resource as long as it's not video, i.e. text. Afterwards, we would ask them which type of OS they are used to and we set up the suitable workstation (whether it's Apple of PC). When the workstation is up and running, we ask them to conduct a short typing speed test in order to measure the eligibility for the study using an online tool [1]: Below average⇒ not eligible.

To make record of the observed behavior and performance of the participants, we used a timer which we would start whenever the participants have understood the given task and the technical environment is ready, i.e. setting up the IDE, stable internet connection.

While the task was being done, we took care not to interrupt the participant to give them full concentration. We did the experiments in a matter of one week, some of them were scheduled, some were not. The language we've chosen was Python, for its simplicity in learning for the first time and how easy it is to set up the required IDE. The IDE was Jupyter which is part of a Python platform suit called Anaconda. We opted for this IDE for the fact that it's lightweight, since we only need a console-styled output, and easy to set up since it can be run from a browser. Moreover, the input and the output are shown on the same page on top of each other. This will give the participant the chance to tweak their input code and see the output without having to switch between windows, which will the make the whole process faster.

Before we start the test, we give the participant a list of links for online tutorials, whether they're video or written ones, which we think would be helpful for the task. Although, they were free to search for other tutorials as long they're only video or only text according to which group they belong.

The whole experiment steps are illustrated in Fig 1

3.3. The Task:

Since we made sure the participants would have no experience in programming, the task we gave them was simple: create a function that finds the sum of two numbers. We did not explain the task any further because we wanted to test how much the online resources are going to be helpful in making the participant understand how the task should be conducted.

3.4. The participants:

The study was conducted, with a total of 10 participants, 5 in each group. As mentioned before, and for convenience of our study, we wanted the participants to have no previous experience in programming. Therefore, we made sure that all of them come from academic backgrounds that are not related to computer science. Neither age nor gender in our study would play a big role as confounding factors, therefore, they were not considered when choosing participants. The Table 3 shows an overview of the

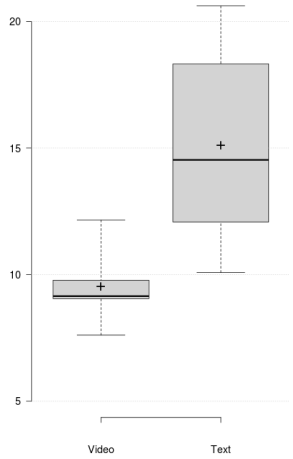

FIG 2
Box plot showing the task completion time for both tested samples

Participant ID	Time in min	Rank (Low to high)	Group
P2	20.61	1	Text
P3	18.32	2	Text
P4	14.53	3	Text
P10	12.16	4	Video
P5	12.08	5	Text
P1	10.08	6	Text
P8	9.78	7	Video
P6	9.15	8	Video
P9	9.05	9	Video
P7	7.16	10	Video

TABEL 4

participants along with the observations of each. Based on the observation data in the table, and since we have a small sample size, we can see that there is a weak correlation between the error rate and task completion time. Therefore, we decided to exclude the error rate from the analysis.

4. ANALYSIS

4.1. Descriptive statistics:

Looking at Fig 2, it can be easily judged, visually, that there is a significant difference between the use of the two resources in our test. We can see that with a smaller mean for the video tests, 9.55 min of task completion time (represented as a small cross in the graph), the participants accomplished the task faster when using video tutorials compared to a mean of 15.12 min using text tutorials.

4.2. Significance test:

However, from a statistical point of view, it is not objectively practical to base our judgment of significance solely on the difference in means or the visualization. And since our sample size is too small, it is far from being considered as normally distributed. Therefore, we had to opt for a non-parametric test, which is the *Mann-Whitney-U* [2]. This test can be done with non-normal distributed metric data, like the one in our study. Moreover, it compares 2 independent samples by a rank test. That means we have to order our entire data, including the 2 groups together, in ranks.

This is done in Table 4 where the data are rearranged so that they are in rank order from slowest to fastest (slow being more time, and fast being less time).

Next, the sum of ranks of each group has to be calculated. In this case the sum of ranks was 17 for the text group participants and 38 for the video group participants. We would then use this information in calculating the U statistic, which is the degree of overlap, in ranks, between the two groups; it gives us a measure of how many data points in one group have a higher or lower rank than data points in the other group. The diagram in Fig 3 illustrates how the two groups overlap in ranks.

This U statistic value will determine how big the difference between the groups is. The smaller it is, the bigger the difference, the bigger it is, the smaller the difference. The first step in calculating the U value is to identify the groups with smaller sum of ranks, which in this case, the text group has a smaller sum:17 compared to 38 for the video group. Then, for each data point in that group, we add up how many data points in the other group are smaller in rank. That means for each participant in text group, we have to count and add how many participants in the video group have a lower rank than them. This is done as follows:

Looking at fig we can observe that ⇒

- There are no participants in video group that are ranked lower than P2, P3 and P4 in text group. These 3 participants are to be represented as 0's three times respectively in the U formula.
- There is one participant in video group who is ranked lower than P5 and P1 in text group. These 2 participants are to be represented as 1's two times respectively in the U formula.

Then we could calculate the U value by adding up all of the resulted values from the previous steps:

U= 0+0+0+1+1= **2**

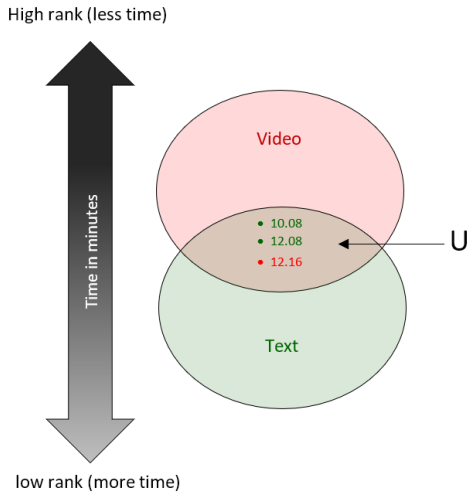

FIG 3

	Mean	Median	Rank sum	Mann-Whitney-U
Completion time using video	9.55	9.15	38	2
Completion time using text	15.12	14.53	17	

TABEL 5

The final step was to compare the resulted U value to the U distribution table [3] to estimate the significance level. We find the critical U value in the table by looking at where the group sample sizes meet in the table and the alpha value, which is conventionally accepted at 0.05 [4]. In our case, with the size of 5 for group1 and group2, the critical value for an alpha value of 0.05 in the table is: **2.** In *Mann-Whitney-U* test, the calculated U value must be equal to or lower than the U value in the table to consider the difference as significant, which in this case, they are equal. Therefore, $p < 0.05$, indicating that there is significant difference between the two tested samples. Table 5 shows the results of the test.

5. INTERPRETATION

Since the p-value for the Mann-Whitney-U test is < 0.05, the null hypothesis can be rejected, i.e. video tutorials are more efficient resources than written ones to learn programming for novices. Moreover, with the comparison of the rank sums of the two samples (Table 5), it is concluded that the task completion time using video resources is significantly lower than when using written ones. However, the result of this test cannot be used to completely accept the alternative hypothesis. One important reason for that is the small sample size we have in the study, which certainly played a negative role in the outcome of this test, i.e. more prone to outliers, the non-normality in the distribution leading to large standard deviation which means less accurate results.

6. THREATS TO VALIDITY

It is necessary to think about all the factors that potentially threaten the validity of a study. In our case, there could be several limitations to the internal and external validity of our results.

For the internal validity, we are aware that the task we gave to the participant was too simple to actually test how powerful and efficient each type of resources is. As a result, for this simplicity, the participants' chance to find the right material for the given task is high regardless of what the type of resources are being used.

Another threat to internal validity, is that the participants' performance and behavior might be affected negatively since they were observed. This threat cannot be removed completely due to the nature of our study. However, as mentioned before, we tried to overcome this by assuring the participants total anonymity and confidentiality.

Since we did not test a random sample that completely represents the target population of novice programming learners, our findings are difficult to be generalized. A further point that needs to be emphasized is, that the study here addresses only pure programming time. Possible influences of readability or maintenance of software/hardware cannot be concluded from the experiment's results. Therefore, the last two points could be threats to the external validity of this study.

7. CONCLUSION

In this paper, we presented results of a study that compares the efficiency between the use of digital media in the form of video and the use of traditional written text for learning programming. We observed 10 participants that were split into 2 groups, one for measuring the performance of task completion over time using online video tutorials, and the other for the same purpose but with written text tutorials. Our findings confirmed what expected that the participants were faster performing the task with the help of video tutorials. Hence, we expected a rather commonly assumed fact and based on several studies mentioned in [5] that video is more engaging and insightful.

But it must not be forgotten that there are still some limitations to video, especially for more complex programming tasks. For example, one cannot search for

keywords from within a video, instead we have to seek through the video until we find the point in time at which mentions what is looked for. Another limitation is that contents and text in a video cannot be copied to be used when need, unlike, written text tutorials where it is possible. Another point to mention is that two of the participants, P4 and P7 made an interesting point that in real-life scenario, novices would normally incorporate the use of both resources when learning new skills.

All in all, in order to get a better insight into the connection between the two methods, further comprehensive experiments need to be conducted

8. RESOURCES

[1] Keyhero Typing speed test
https://www.keyhero.com/free-typing-test/

[2] Mann-Whitney-U test
https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test

[3] Critical values of the Mann-Whitney U
http://ocw.umb.edu/psychology/psych-270/other-materials/RelativeResourceManager.pdf

[4] Alpha value
http://www.statisticshowto.com/what-is-an-alpha-level/

[5] University of Queensland: Pedagogical benefits
http://www.uq.edu.au/teach/video-teach-learn/ped-benefits.html