The Effect of Implicit Tutorials on User Learning Rate: Aiding User Acceptance of New Applications

EMILIE BECK*, JEREMIAH GEISTERFER*, and JOY PUTNAM*, Colorado State University, USA

CCS Concepts: • **Human-centered computing → Graphical user interfaces**; **HCI theory, concepts and models**; **User interface programming**; **Interaction design theory, concepts and paradigms**.

Additional Key Words and Phrases: implicit tutorial, assisted learning, 3D modelling, Unity, UI, UX, unfamiliar technology

ACM Reference Format:
Emilie Beck, Jeremiah Geisterfer, and Joy Putnam. 2024. The Effect of Implicit Tutorials on User Learning Rate: Aiding User Acceptance of New Applications. *J. ACM* 37, 4, Article 111 (August 2024), 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 ABSTRACT

This paper aims to answer the question of whether replacing standard video tutorials for learning new programs with a recommendation system of highlighted options can more efficiently teach users a new program. To perform this experiment, we created three versions of the same program: one to be accompanied by video tutorials to teach users where to click to complete tasks, one with the highlighting recommendation system, and one with no learning assistance. Our results supported the null hypothesis, as there was no statistical significance between any of the groups, though those with the standard tutorial performed significantly fewer button clicks per trial, meaning that while the learning rate was not improved for the standard tutorial, there was less menu navigation.

2 INTRODUCTION

In an ever-shifting technological world, companies are constantly made to adapt to new changes and upgrades. Similarly, for every change in service companies make, there is a new adaptation and learning curve required of their employees. Given this ever-changing landscape, Human Computer Interaction research has dedicated significant focus to usability and learning rate of software [5]. In some cases, this transition can be fairly smooth with simplistic applications and simplistic tasks that suit a user-friendly UI quite well. However, for more complicated applications like AutoCAD or the Adobe Suite, the sheer versatility of these programs often comes at the cost of simplicity, leading to steep learning curves and frustration for new users. Often, with more versatile programs, the programs themselves offer walk-through tutorials with a specific example that does not often allow its users to use the application the way they actually would for a task they want to complete. Despite the popularity of video tutorials as a means for software training there was determined to be a difference in how well tasks were completed based on being directly taught or tasks that the user

*All authors contributed equally to this research.

Authors' address: Emilie Beck, Emilie.beck@colostate.edu; Jeremiah Geisterfer, (jeremiah.geisterfer@colostate.edu; Joy Putnam, jocelyn.putnam@colostate.edu, Colorado State University, Fort Collins, Colorado, USA, 80523.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2024 Association for Computing Machinery.
Manuscript submitted to ACM

had to learn [23]. This means that many users must simply learn through "trial by fire," which, while effective, also costs users a lot of time when figuring out how to complete tasks for the first time.

Our research aimed to answer the question: Is there a way to design a tutorial that is so embedded into a program that users can learn while having the freedom to start by working on personal projects? We believe that research involving embedded tutorials and user action suggestions could provide a better understanding of user application fluency and acceptance when learning a new application. The goal was to create an environment where the user could explore the application freely while still being assisted to improve the learning rate and aid in user retention.

A "trial by fire" form of learning often leads to frustration in users, but forcing users to sit through long tutorials before they are allowed to try the features of an application is no much better. In a study where users were supplied with text predictions while entering data, speed and accuracy gains were both recorded [12]. The use of tutorials that have been well made and teach the software correctly has also been shown to increase usability [25]. With our approach, we modified the step-by-step guides of many application tutorials and replaced them with a predictive model that suggests possible next actions, given common user patterns, which guide a user through a program without forcing them to make any particular decisions.

This could be extremely helpful for companies aiming to switch systems for work, as well as for individuals who would like to learn a program that may improve their workflow. If a tool like this proves to be successful, it could also help increase technological fluency in most users in the future. This will also increase the amount of research on the topic of tutorials and user acceptance and learning rate, as the current research is quite limited.

3 RELATED WORK

There is currently limited research that directly relates tutorials and user acceptance rate, though there is some research relating to game tutorials and learning through application play/exploration. Multiple studies specifically looked into tutorials for computer science students learning to use Eclipse or develop applications for Android products, both that emphasizing the importance of an interactive element to the tutorials [1, 29], whereas another study was done about interface instructions that discusses utilizing legacy bias rather than trying to avoid it, although this makes the tutorial less applicable to users unfamiliar with the previous software [19].

Most tutorial research, however, is currently in game design. One study found that immersion and “perceived usefulness” were important to both user immersion and general enjoyment of the application. Game tutorials typically prevent the player from beginning play within the game, which is a potential detriment to player retention, but increases user acceptance when utilized properly [30]. Another study comparing different styles of tutorials found that text was more effective than the usage of NPCs or some other “person” when it came to example comprehension, while no tutorial at all had varying results in users’ abilities to learn mechanics [16]. While there is not as much research in non-game tutorials, there have been some studies on the ability of users to adapt to newer technology when there are constant changes being made. In a study of medical students learning with and without tutorials, the students that received tutorials had better long-term retention of the material and higher test scores [17]. An experiment was run on the Blackboard system, a web-based information system. This study highlights the importance of enjoyment, self-efficacy, and goal-oriented learning play in user acceptance [28].

Seeing as a previous study emphasized the importance of efficiency and self-efficacy for the user when building user acceptance of a system, it was clear that we had to include this research in our tutorial plans. Similarly, another study found that segmenting tutorials, in the specific case of the study video tutorials, was crucial to improving the user’s sense of usability of a software [14]. Thus, the teaching and tasks of users learning a new system must be segmented in such a way as to allow for time for absorption and to prevent overwhelm.

It is clear from these studies that making an enjoyable tutorial that emphasizes a sense of independence and direction is the key to best aiding user adaptation of the program. For our specific style of tutorial, we also decided to look into the usage of predictive text, as we believed a tutorial that tried to act one step ahead of the user's personal plans would be useful. In a study on predictive text, it was found that text prediction improved accuracy and efficiency and also prompted user responsiveness, thus increasing user efficacy [7, 12]. In contrast to the previous study where predictive text was found to improve accuracy and efficiency another found that word suggestions decreased typing speed and accuracy when using a keyboard on a mobile device. The study also found that the users that used the suggestions were the slower keyboard users and that use of the suggestions slowed them down even further [15]. They considered one reason for suggestions to increase time is that they may increase cognitive costs, and that using the suggestions broke the natural flow of typing. Suggestions have been shown to reduce the user typing speed as well as increase it and we must be aware of this effect as we try to create a recommendation system that works with the user while teaching the user and not increasing cognitive load.

Another implication of tutorials is a change in cognitive load placed on the user. Knowledge and experience play a crucial role in the cognitive load that certain tasks place on individuals. A study designed to analyze cognitive load and user retention found that when using a multimedia tutorial cognitive load decreased and allowed for greater retention. They had tasks that were designed around computer networking and had varying tutorial types. The study determined that multimedia tutorial, a mixture of voice and text, helped reduce cognitive load on the participant allowing for greater retention [18].

A large part of the tutorial's recommendation design is capturing the user's attention to direct their eyes to the suggested next actions. For example, directing their eyes to the workspace after clicking the select button to indicate that they should look there to make their selection. There have been numerous studies on the usage of color in human attention. These studies have shown that not only is color more important than location, but that changes in value, which is to say that higher contrast, are better for capturing user attention [6, 9, 10].

There have also been numerous studies on the importance of visuals in not only attention management, but also in learning. One study found that the presence of visual stimulus actually increases attention due to a brain chemical called norepinephrine [8]. Another study attempted to compare visuals and verbal explanation in the teaching of new and unfamiliar systems or subjects. The study found that while visual and verbal teaching were both useful, visuals allowed for better retention [3]. There have also been multiple studies on the use of visuals in class settings that have shown that students are typically comfortable with, and learn well from, visuals [20, 24]. Further studies found that the use of particularly 3D visuals in an interactive space helped students to better visualize and conceptualize the material, advocating for the use of 3D model simulations in more teaching environments [21, 27].

One very applicable related work in this area is a case study done by the Naval Postgraduate School where students learned 3D modeling software to create their own simulation of the same target object, a particular kelp exhibit at an aquarium. Half of the participants were already learning a 3D modeling software, and the other half had no experience with 3D graphics. Since all students were tasked with recreating the same environment, it was easier to see the difference in output between the two groups, and they found that the students who were shown examples performed better and faster [4]. The study also claimed that one of the main issues with teaching 3D modeling software is the lack of interface conventions, which directly ties into this project as we are altering the chances for prior understanding to influence the results.

4 METHODOLOGY

For our study, we conducted an experiment where users completed shape manipulation tasks in an application with either no tutorial, a predictive UI-based tutorial, or a step-by-step guided tutorial. Users were divided into three groups, with each participant receiving only one of these three versions of the application. All participants were asked to complete the same tasks with the application. This experiment intended to determine if a less structured tutorial would improve application retention and literacy similarly to that of a more traditional tutorial. Previous research has primarily centered around the use of just the step-by-step guided tutorials.

To measure the effects of predictive tutorials on users, participants were asked to complete a series of tasks within the application. We measured their speed in completing the tasks and tracked how many clicks the user made to complete a step and what they clicked. To account for variance in reading speed when time was measured, each task instruction appeared in a pop-up, and the timer did not begin until the user closed it manually. This also kept the phrasing of instructions consistent for all participants. The task remained available for them to read on a separate screen in case they needed a reminder at any point during the trial, as well as a visual example of what the completed task might look like.

The intention of making a predictive tutorial was to hopefully guide the user to a likely path without directly telling them what to do. This was done in order to avoid biasing their results and to account for the findings of a study done on the impact of predictive text on the user's output [2]. According to the study, if people are given suggestions, they will often follow those suggestions in a predictable manner and with less variation. Although the idea is to help the user find the best path through the program to complete the task, there has to be a learning curve where their own decisions lead to their discoveries about the software.

There were three rounds of tasks per participant, with the first two rounds consisting of 3 tasks each, and the final round consisting of a singular task: build a house, which requires the subjects to complete a minimum set of tasks to build their "house" to better gauge the final results of their learning. The House needed to have a roof, a chimney, a door, and a window, all of different colors.

All three rounds of tasks were rather simple and all similar to each other, but each built on the last one. For example, the very first task was to "create a pink cube" which is two instructions, create the cube and color the cube. Then the second and third tasks kept the progress from the previous ones and had the user create more objects, color more objects, move these objects, and lastly rotate them. This pattern continued with the user completing a simple task and then expanding on what they've already learned and applying the same principles to more complex operations. As mentioned previously, the final task was to build a house that meets a certain set of minimum criteria. This allowed participants to have the chance to use their knowledge of the first two rounds and apply it to a larger task.

Participants all completed a post-experiment survey to gather information about their prior experience with similar software and their experience with the program and whatever tutorial type they received.

4.1 Experiment Design

For our experiment, there were three basic prototypes of the same program. The program, in structure, vaguely resembles something like Adobe Photoshop or AutoCAD, two programs with known steep learning curves, an array of menus, sub-menus, and buttons. This application was designed to have complicated menu navigation and strictly use icons for essential buttons to lower the chances of application fluency from affecting participant performance.

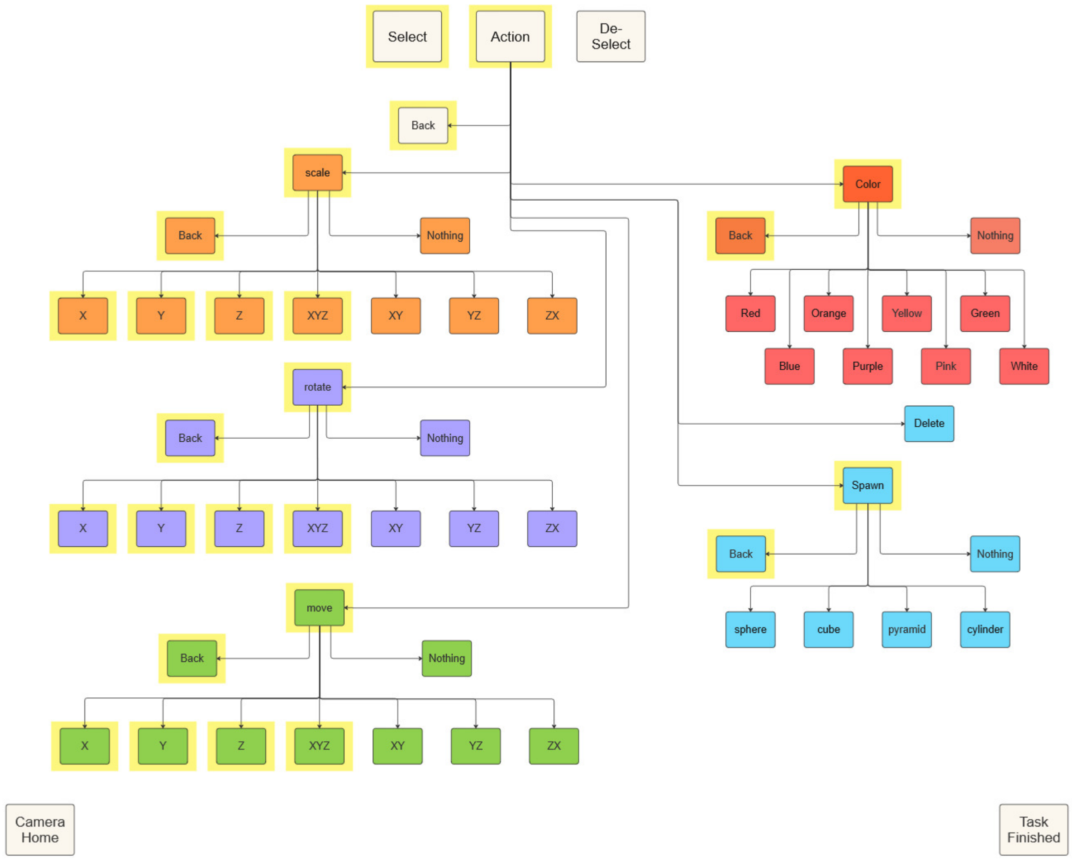

Fig. 1. A diagram displaying the button mapping for the program. Some buttons have a yellow square representing that they would be highlighted similarly in the highlighted button tutorial.

From this base, there are three variants. The first variant includes no tutorial. This version includes all of the features of the base program, but there's nothing that helps guide the user around the program. This variant thus acts as the control in order to measure how useful the UI tutorial is. Figure 1 displays the main button paths available to users. For example, if a user presses the "Scale" button indicated by the Scale icon seen in Figure 2, they will be given a new array of button options to select from. Depending on which axis they select they will see a different manipulation indicator appear around the object that is selected. One of the intentionally most frustrating aspects of the program is the need to go back to the first screen every time the user wants to manipulate a different object, and specifically select the object. Objects appear highlighted in blue when they are successfully selected.

The second variant is the UI tutorial. For this version, when a button is clicked, the program highlights "commonly used" related actions. For example, when the user clicks the "Rotate" button, the four most helpful axis manipulation buttons are highlighted yellow to indicate that the user might want to click one of them. These commonly selected axes are X, Y, Z, and XYZ. The idea is to help guide the eye and build up muscle memory for multi-step actions, while not

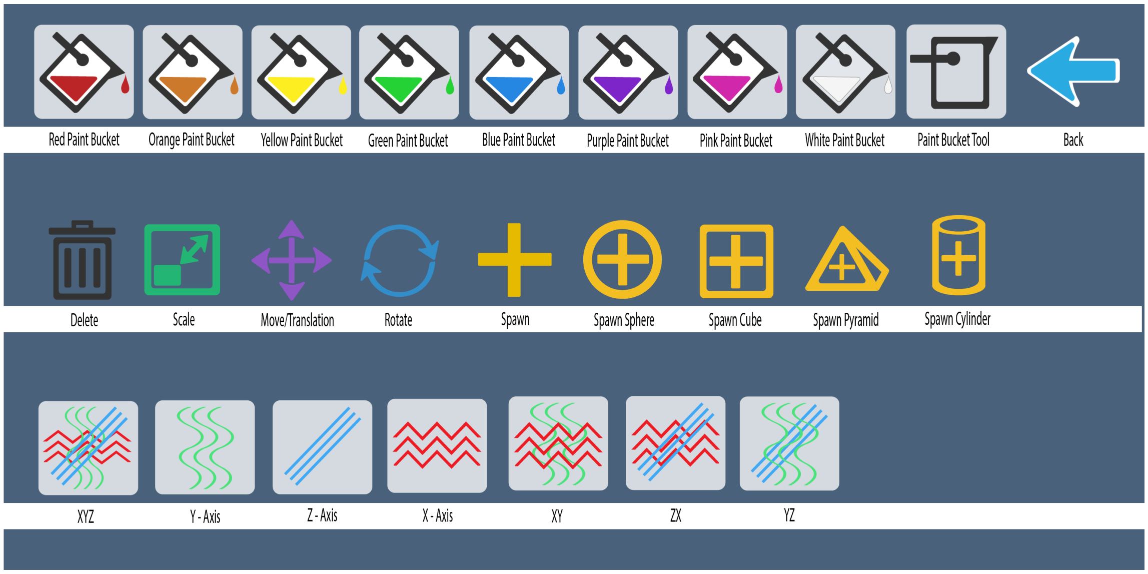

Fig. 2. All of the icons used in the final product with labels for each underneath. The icons are displayed on a blue background with the same Hex code as the background UI in the program to demonstrate how they look on this base color.

making the functionality so specific that it would only work in situations where they are being told the specific task we want them to complete.

The final variant includes a "standard" tutorial to compare the UI tutorial's usefulness compared to how most tutorials function now. For this variant, users were shown a short video of how to complete a related action, but not the specific task. They are only able to watch this video once, and the watch time is added to their total task completion time, as following application tutorials add to actual work time in applications like the experiment. These three variants ideally give the best visualization of the usefulness of the UI tutorial in improving user learning by testing it not only against the presence of no tool, but also against current tutorials.

The icons designed for the axes were abstracted the most of all the button icons used, each axis being represented by a different type and color of line, seen in Figure 2. These abstract designs were then layered over each other to create the combination axes buttons. The non-standard depiction of these otherwise common operations was intended to confuse users briefly, but once they played with each button and discovered how each one reacted, they would theoretically understand the concept and learn to look for specific buttons thereafter. The application was built with Unity and ran locally in the Unity file, with each user's data also being saved locally to a file for further analysis.

5 RESULTS

After collecting the data for each participant's trial times and taking the averages for each trial in Table 1, we ran an ANOVA on the resulting data to detect any meaningful patterns, shown in table 2. The mean time across all trials for the highlighted tutorial (htut) group was 199.63767s, for the standard tutorial (stut) was 189.49033s, and for no tutorial (notut) was 171.33302s. However, the effects of the participants' groups (standard tutorial [stut], highlighted tutorial [htut], or no tutorial [notut]) was not statistically significant ($F_{2,15} = 0.623, ns$). The interaction effect of trials x group was also not statistically significant ($F_{12,90} = 1.526, p = 0.129445$).

The effect of trials on task completion time was statistically significant ($F_{6,90} = 54.970, p = 0.000000$). However, because the trials all varied greatly in both difficulty and number of actions required to complete, this data is not considered significant to this study.

While there was not a statistically significant difference between the three tutorial groups, table 1 shows that surprisingly, on average, the participants with no tutorial completed most tasks slightly faster than the other two groups.

The number of button clicks that each participant made as they completed each of the tasks was also recorded and from it a Table 3 was created showing the average number of clicks per trial per group as well as the standard deviations. The group that received the standard tutorial, clicked an average of 378.5 buttons, the highlight group, clicked an average of 562.67 buttons and the no tutorial group, clicked an average of 427.83 buttons. With a standard deviation of 95.33 buttons clicked over all trials this puts the standard tutorial group outside one standard deviation from the highlight group that had the most clicks on average.

An ANOVA was also run on the data from button clicks and the results shown in Table 4. The effect of the group on the number of clicks was statistically significant ($F_{2,15} = 4.406, p = 0.0312$). The effect of trial on the number of clicks was also statistically significant ($F_{6,190} = 22.554, p < 0.0001$), however as mentioned previously since each trial varies it is not considered significant to this study.

Tutorial Type	AVG Trial Time (s)							AVG Total Time (s)
	T1.1	T1.2	T1.3	T2.1	T2.2	T2.3	T3.1	
Standard	76.43591	133.7893167	249.3819833	63.74405	184.7947167	236.13505	382.1512667	1326.432293
Highlighted	113.2698483	238.7494667	179.9751667	46.74275	168.961	226.0118833	423.7536	1397.463715
None	61.56285667	142.47185	231.4055333	44.164	167.5265	205.12975	347.0708167	1199.331307

Table 1. Average Trial Times Per Group

Effect	df	ss	MS	F	p
Group	2	17273.338	8636.669	0.623	0.5498
Participant(Group)	15	208017.336	13867.822		
Trials	6	1272852.065	212142.011	54.970	0.0000
Trials_x_Group	12	70666.817	5888.901	1.526	0.1294
Trials_x_P(group)	90	347330.162	3859.224		

Table 2. ANOVA Table for Participant Trial Times

Tutorial Type		Average Button Click and Standard Deviation by Trial							Average Total Button Click per Group
		T1.1	T1.2	T1.3	T2.1	T2.2	T2.3	T3.1	
Standard	Average	12.16	31	58.33	34.66	68.16	60.83	123.33	378.5
	Standard Deviation	1.83	11.54	23.37	3.5	7.47	22.96	25.59	
Highlighted	Average	66.33	112.16	66	26.5	75.5	69	147.16	562.67
	Standard Deviation	90.48	80.85	35.18	5.50	16.31	20.65	36.48	
None	Average	21.17	48.33	70.67	27	72.17	66.17	122.33	427.83
	Standard Deviation	11.70	19.71	13.56	1.67	11.2	22.55	24.26	
Standard Deviation of Average Total Button Clicks									95.33

Table 3. Average Number of button clicks, and Standard Deviation per trial for Group

Effect	df	ss	MS	F	p
Group	2	15580.333	7790.167	4.406	0.0312
Participant(Group)	15	26520.810	1768.054		
Trials	6	124642.984	20773.831	22.554	0.0000
Trials_x_Group	12	19679.111	1639.926	1.78	0.633
Trials_x_P(group)	90	82898.190	921.091		

Table 4. ANOVA Table for Participant Button Clicks

6 DISCUSSION

As the ANOVA data results in table 2 clearly show, neither the highlighted tutorial nor the standard tutorial had a significant impact on the speed with which it took participants to complete the trials. For the standard tutorial, this is likely due to the time taken to finish watching the provided videos. Participants seemed to do most of their learning during trials 1.1-1.3, meaning that the time it took the other two groups to learn how to navigate the menus took approximately as long as it took the standard tutorial participants to watch the videos. As for the highlight tutorial, it is likely that the highlighting simply was not helpful enough or dynamic enough to have any real impact on the efficiency of this group. In fact, while we were originally trying to create a tutorial that worked almost subconsciously, this seemed to work a little too well, as some participants with the highlight tutorial reported forgetting they had highlighted buttons to aid them. However, with only 6 participants per group, determining whether these patterns are consistent or not would likely require more participants.

Where participants' times seemed to have been impacted little or none by the highlighting and standard tutorial, the data collected by an ANOVA on button clicks that can be seen in Table 4 supports that there was an effect of tutorial type on number of buttons clicked. As seen in Table 3 the group that received the standard tutorial had the lowest number of clicks overall, with the highlight tutorial group having the highest. This would support that the standard tutorial did aid in learning but since there was the time it took to view the tutorial it was not generally faster than trial by fire for task 1.

Supporting the findings of [15], mentioned previously, our results show that a predictive or suggestion-based user support decrease the users speed. This could be due to an increase in the cognitive load placed on the user when buttons

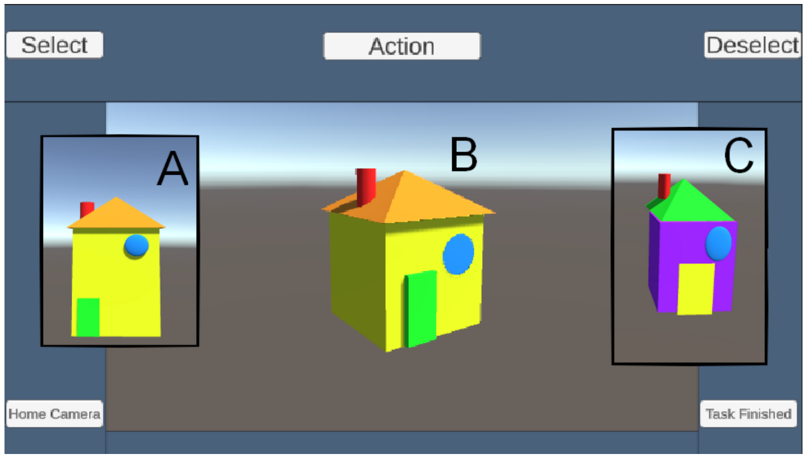

Fig. 3. Three houses all built using our 3D modeling program. A and C are participant made, B is the example product seen in the task instructions.

were highlighted that were not the buttons that the user wanted to use placing an extra layer of decisions on the user that would not have been there with no suggestions at all. Another consequence of more decisions for the user to make was an increase in problem difficulty. Instead of learning the application the user also needed to understand how the button highlighting worked. Since it was an imperfect system that recommended more buttons than was helpful it had an negative impact on the participant. Sweller states that problem solving can incur a high cognitive load and to learn a specific tool or domain, the required information to do so may differ highly [22].

Relating to one of the sources about predictive text influencing the elicited responses from users, we did find that there were some participants who created drastically different "answers" to the tasks than seen in the example image they were provided [2]. For instance, Figure 3 demonstrates three houses. 3A and 3C are screenshots of the "houses" that two participants built for their final task, whereas 3B is the example house used in the task instructions. Figure 3A is the same colors as the example house, whereas Figure 3C uses more creative liberty, though neither was a perfect recreation of scale. However, like the other study had concluded, most participants created models that were as similar to the example picture as possible.

There are outliers in each of the groups that may be causing some higher levels of variance. This can be seen in Table 3 in T1.1 the highlight tutorial group has an average of 66.33 with a standard deviation of 90.48 with means that there were some participants that had wide ranges in the number of clicks to finish the task. Task 1.2 also has a high variance with a standard deviation of 80.85 for the highlight group. This is in contrast to the standard tutorial group for T1.1 which had a standard deviation of 1.83. This also supports that the standard tutorial promoted learning, as each of the participants completed the task within two button clicks of one another.

Another reason for this large variance could be the different interpretations of the finished task. Each participant received the same task instructions and picture of the completed task, and each participant had a different perspective on how close to the picture that they needed to be in order to complete a task. Some participants would spend more time than others getting certain aspects of the task perfect where other participants would get close and decide that was good enough for them and mark the task as completed. This wide definition of done caused variation in the data and could be done again, we would have formed a more precise definition of complete.

The post-experiment user survey revealed a couple of trends as well. It was mentioned by some participants with the highlighted button tutorials that the highlights weren't effective either because users forgot that the yellow indicated "suggested" buttons or that they didn't recognize the buttons as being highlighted in the first place. One way that this experiment could be improved upon in future iterations is to make the highlighted buttons more obvious by animating the yellow glow to pulsate or flicker rather than it just being a static box behind the regular buttons. This might have solved one of the most common issues that the users with the "predictive" highlighted button tutorial experienced, as multiple users provided this feedback. Multiple studies also support this theory, connecting the idea of flashing lights and movement as potentially even more effective to drawing the eye than color or static brightness due to experiments with traffic lights and website pop-ups [11, 13]. This would also connect to another of our survey questions regarding previous experience with 3D modeling software, as one study claims that people with extensive video gaming experience may have increased perception of small visual stimulus [26]. Thus, a future version of this experiment might also consider the effect that gaming experience has on the effectiveness of the flashing highlighted buttons in determining the participants' decisions.

7 CONCLUSION

In this study, we looked at how tutorials affect a user's ability to navigate a new piece of software and how user acceptance rate changes based on what type of tutorial that participant did or did not receive. We found that participants task completion time was not improved by tutorials of any kind, but the number of buttons clicked per task was improved for the group that received the standard video tutorial. It was also determined that the highlighting tutorial that we designed was not sufficient in teaching the application and would need to be modified and the study ran again to get data that would be more representative of what an actual predictive tutorial might be like.

Since we were limited both in time and participants some simplifications were made to the tutorials and the application that may have hurt the overall results of the study. The highlighted tutorial was supposed to be a predictive highlighting tool that selected the button that the participant was most likely to press based on the previous actions. Since this would require a level of machine learning that we were not able to implement in such a short time, a simple highlighting procedure was implemented based on the previous button press. This method created more confusion in participants compared to no tutorial since buttons were highlighted even if they were not relevant to the task at hand. Even with these limitations, however, there is still data that supports that learning occurs when a tutorial is provided and the tutorial teaches the required skills. If we were able to run the study again, more investment would be made into the highlighted tutorial and making it a tool that promotes learning and speed in a way that we were unable to for this study.

REFERENCES

[1] Al Rekhawi, Hazem Awni and Abu Naser, Samy S. (2018). "An Intelligent Tutoring System for Learning Android Applications UI Development". *International Journal of Engineering and Information Systems (IJEAIS), 2(1)*, 1-14.

[2] Arnold, Kenneth C., Krysta Chauncey, and Krzysztof Z. Gajos. (2020). "Predictive text encourages predictable writing". *Proceedings of the 25th International Conference on Intelligent User Interfaces (IUI '20).* Association for Computing Machinery, New York, NY, USA, 128–138. https://doi.org/10.1145/3377325.3377523

[3] Bobek, E., & Tversky, B. (2016). Creating visual explanations improves learning. Cognitive research: principles and implications, 1(1), 27. https://doi.org/10.1186/s41235-016-0031-6.

[4] Brutzman, Don. (2002). *Teaching 3D modeling and simulation: virtual kelp forest case study.* Association for Computing Machinery, New York, NY, USA, 93–101.

[5] Fernquist, J., Grossman, T., & Fitzmaurice, G. (2011). "Sketch-Sketch Revolution". *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, 373–382.

[6] Foster, Jerod. (2013). *Color: A Photographer's Guide to Directing the Eye, Creating Visual Depth, and Conveying Emotion*. Pearson Education.

[7] Gong, Y., Hua, L., & Wang, S. (2016). "Leveraging user's performance in reporting patient safety events by utilizing text prediction in narrative data entry". *Computer Methods and Programs in Biomedicine*, 131, 181–189.

[8] Gray, Shawn R., Ye, Liang, Ye, Jing Yong, Paukert, Martin. (2021). "Noradrenergic terminal short-term potentiation enables modality-selective integration of sensory input and vigilance state." *Science Advances, Vol 7 Issue 51*. https://www.science.org/doi/10.1126/sciadv.abk1378.

[9] Hamel, S., Houzet, D., Pellerin, D., & Guyader, N. (2015). "Does color influence eye movements while exploring videos?". *Journal of Eye Movement Research*, 8(1).

[10] Hillyard, S.A., Münte, T.F. "Selective attention to color and location: An analysis with event-related brain potentials". *Perception & Psychophysics 36*, 185–198 (1984).

[11] Hong, Weiyin, James Y. L. Thong, Kar Yan Tam. (2004). "Does Animation Attract Online Users' Attention? The Effects of Flash on Information Search Performance and Perceptions". Information Systems Research 15(1):60-86. https://doi.org/10.1287/isre.1040.0017

[12] Hua, L., Wang, S., & Gong, Y. (2014). "Text prediction on structured data entry in Healthcare". *Applied Clinical Informatics, 05(01)*, 249–263.

[13] Jashami, H., Hurwitz, D. S., Monsere, C., & Kothuri, S. (2019). "Evaluation of Driver Comprehension and Visual Attention of the Flashing Yellow Arrow Display for Permissive Right Turns". Transportation Research Record, 2673(8), 397-407. https://doi.org/10.1177/0361198119843093.

[14] Lamontagne, Charles, Sénécal, Sylvain, Fredette, Marc, Labonté-LeMoyne, Élise, Léger, Pierre-Majorique. (2021). "The effect of the segmentation of video tutorials on User's training experience and performance". *Computers in Human Behavior Reports, Volume 3 100071*. https://doi.org/10.1016/j.chbr.2021.100071.

[15] Lehmann, F., Kornecki, I., Buschek, D., & Feit, A. M. (2023). "Typing behavior is about more than speed: Users' strategies for choosing word suggestions despite slower typing rates". *Proceedings of the ACM on Human-Computer Interaction, 7(MHCI)*, 1–26. https://doi.org/10.1145/3604276.

[16] Malthed, A., & Hjelmqvist, M. (2020). "UI-based, NPC-guided or No Tutorial at All? A Qualitative Study Comparing the Effects of Different Tutorials in the Same Game".

[17] Marsh, K. R., Giffin, B. F., & Lowrie, D. J. (2008). "Medical student retention of embryonic development: Impact of the dimensions added by multimedia tutorials". *Anatomical Sciences Education, 1(6)*, 252–257.

[18] Mayrath, M. C., Nihalani, P. K., & Robinson, D. H. (2011). "varying tutorial modality and interface restriction to maximize transfer in a complex simulation environment: Correction to mayrath et al. (2011)'. *Journal of Educational Psychology, 103(2)*, 268–268.

[19] Ramesh, Vidya, Charlie Hsu, Maneesh Agrawala, and Björn Hartmann (2011). "ShowMeHow: translating user interface instructions between applications". *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. Association for Computing Machinery, New York, NY, USA, 127–134. https://doi.org/10.1145/2047196.2047212.

[20] Shabiralyani, Ghulam, Hasan, Khuram Shahzad, Hamad, Naqvi, Iqbal, Nadeem. (2015). "Impact of Visual Aids in Enhancing the Learning Process Case Research: District Dera Ghazi Khan". *Journal of Education and Practice, Vol.6, No.19, 2015*.

[21] Surynková, Petra. (2011). "3D Modeling in Teaching and Learning Geometry". Charles University in Prague, Faculty of Mathematics and Physics, Sokolovská 83, 186 75. Praha 8, Czech Republic

[22] Sweller, J. (1988) "Cognitive load during problem solving: Effects on learning." *Cognitive Science, vol. 12, no. 2, June 1988*, pp. 257–285, https://doi.org/10.1016/0364-0213(88)90023-7.

[23] van der Meij, H., van der Meij, J., Voerman, T., & Duipmans, E. (2017). "Supporting motivation, Task Performance and retention in video tutorials for software training". *Educational Technology Research and Development, 66(3)*, 597–614.

[24] Vanichvasin, Patchara. (2020). "Effects of Visual Communication on Memory Enhancement of Thai Undergraduate Students, Kasetsart University". *Higher Education Studies, Vol. 11, No. 1; 2021*.

[25] Whitman, L. (2013). "The effectiveness of interactivity in multimedia software tutorials (dissertation). The Effectiveness of Interactivity in Multimedia Software Tutorials". *ProQuest LLC*.

[26] Wilms, Inge L., Anders Petersen, Signe Vangkilde. (2013). "Intensive video gaming improves encoding speed to visual short-term memory in young male adults", *Acta Psychologica*, Volume 142, Issue 1, 2013, Pages 108-118, ISSN 0001-6918, https://doi.org/10.1016/j.actpsy.2012.11.003.

[27] Xie, Charles, Corey Schimpf, Jie Chao, Saeid Nourian, Joyce Massicotte. (2018). "Learning and teaching engineering design through modeling and simulation on a CAD platform." Computer Applications in Engineering Education. 26. 824-840. 10.1002/cae.21920.

[28] Yi, M. Y., & Hwang, Y. (2003). "Predicting the use of web-based information systems: Self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model". *International Journal of Human-Computer Studies*, 59(4), 431–449.

[29] Zhang, Ying, Gang Huang, Nuyun Zhang and Hong Mei. (2009). "SmartTutor: Creating IDE-based interactive tutorials via editable replay." *IEEE 31st International Conference on Software Engineering*, Vancouver, BC, 2009, pp. 559-562, doi: 10.1109/ICSE.2009.5070555.

[30] Zhiyong, X., Jiani, W., & Lijun, J. (2021). "How Game Tutorials in Mobile Online Games Affect User Adoption". *2021 10th International Conference on Educational and Information Technology (ICEIT)*, 264–268.