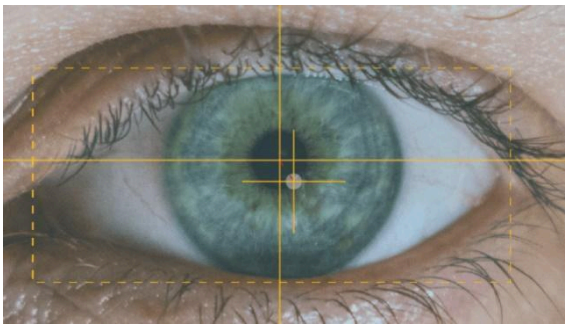
המכללה האקדמית להנדסה בראודה
בכרמיאל

Software Engineering Department

Braude College

Extended Project in Software Engineering



Tal – System

A system that **sees** your needs

24-1-R-23

Capstone Project Phase B

Supervisor:

Dr. Julia Sheidin

Students:

Liraz Akiva – Liraz.Akiva@e.braude.ac.il

Einan Cohen– Einan.cohen@e.braude.ac.il

Table of contents

1. Abstract

Today, technology is an essential part of everyday life, making a significant impact on people's lives. Muscular dystrophy patients, particularly those with ALS, benefit from using eye-tracking technology to interface with computers. This technology lets them control computer actions using only their eyes, replacing traditional mouse and keyboard inputs. Despite these advancements, ALS patients still face challenges with certain tasks. Our project focuses on addressing these challenges.

Our project centers around a man named Tal, who has ALS and utilizes eye-tracking technology to navigate Adobe Premiere's video editing software. Tal has encountered difficulties with specific operations, particularly when moving and correcting files on the timeline. To address these challenges, we have designed a panel with buttons to activate the keyboard shortcuts within the Adobe Premiere software. Our research process and the proposed solution are detailed in the project documentation.

Keywords:

Eye-tracking technology, Adobe Premier Pro, Accessibility technologies, Hands-free technology, Virtual keyboards, Remote communication devices.

2. General Description

Performing minimalist tasks such as moving a mouse or executing simple computer operations like pressing a key on a keyboard may appear effortless to many. Yet for a significant portion of the population, particularly individuals with physical disabilities such as Amyotrophic Lateral Sclerosis (ALS) patients, these actions pose significant challenges. ALS patients, who experience problems with muscle movements and rely entirely on others, find operating a mouse and keyboard to be exceptionally difficult. Using eye-tracking technology can have significant psychological and social benefits for ALS patients. It gives them more independence and a sense of belonging in society, which can directly improve their quality of life [1,2]. However, there are also some challenges with eye tracking. Not everyone is suitable for working with this technology. For example, people who wear contact lenses or glasses may have difficulty with accuracy. Additionally, only 10-20% of the population is considered suitable for eye tracking [3].

2.1. Challenges in Patient-Computer Interaction Using Eye Tracking Systems

There are also some difficulties in the interaction between the patient, the eye tracking system, and the computer, such as:

- Spatial inaccuracy: The eye sees a wide image, making it hard to know precisely which pixel on the screen it's focusing on.
- Midas touch: It's difficult to distinguish between a "casual look" and a look meant to give a command.
- Mobility: The constant eye movement makes it hard to move the mouse cursor.
- Calibration: The system needs to be calibrated for each specific user.
- Multiple actions: It's challenging to look at an object and perform an action when both interactions are done through eye movement.
- Adjustment: The size of objects and the interface need to be adjusted to work with gaze focus.

To overcome these difficulties, there are currently solutions like making objects and buttons larger to improve accuracy and provide clear feedback. Hence, the user knows what action is being performed and uses dwell time to differentiate between a "casual glance" and focusing on a desired action [4]. Therefore, the main goal of this research project was to propose, develop, and evaluate an alternative solution that aims to improve the existing interaction between the user and the computer interface, specifically in the Adobe Premiere Pro software. This was done using keyboard shortcuts already present in the software that helped to perform automatic actions. The goal was to optimize the actions that the user performs in the software, both in terms of efficiency (accuracy and time) and independence, which the user can acquire for himself following the use of the panel.

2.2. The gap between Tal's desired outcomes and the current functionality

Our project takes place within the framework of Brauda's flagship project, an ambitious initiative that aims to foster the integration of people with disabilities in mainstream society and the field of employment and is intended for a specific client named Tal, who suffers from ALS. Tal is confined to a wheelchair and interacts with the computer through eye tracking, where most of his computer activity is focused on working with the Adobe Premiere Pro software for video editing. Hence, we aimed to explore alternative solutions for enhancing Tal users' experiences during film editing using the Adobe Premiere Pro software.

Our client, Tal, utilizes the PCEye solution developed by Tobii. The solution contains TD Control software that replaces the mouse and keyboard actions using eye movements. Tal uses the TD Control software to select the file he wants to insert into the timeline. He then clicks on the drag action icon (figure 1) and tries to mark the specific point on the timeline where he plans to place the file (figure 2). As he does this, Tal concentrates on the exact location where he aims to drag the file. This process is cumbersome and interferes with precise operation.

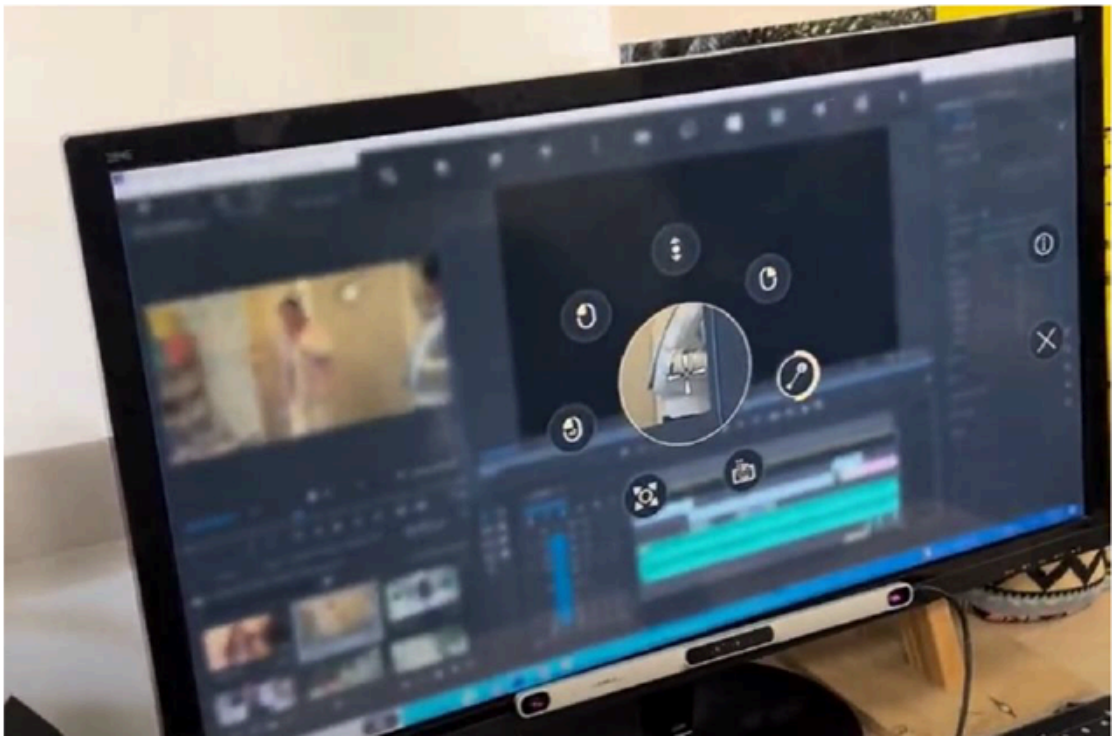


Figure 1: Selecting the icon

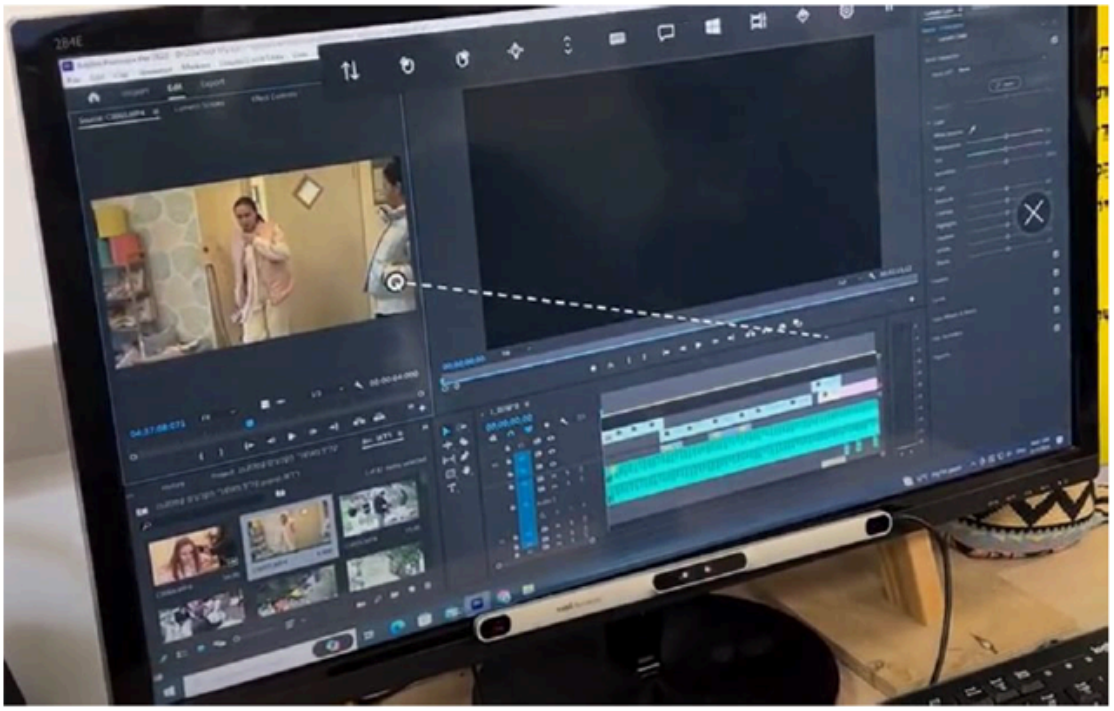


Figure 2: Choosing the location

In the current situation, Tal uses its software to perform a drag operation by marking a start point, then marking an end point, and the drag is carried out between the points. In addition, the accuracy of the marking of the points is difficult to measure using the software and sometimes, he even needs help from his assistant in order to make the operation precise because she uses the mouse, and it is easier for her to be precise.

3. Solution Description

Our goal was to develop a side panel for Tal that would address his challenges in using Adobe Premiere. The panel includes buttons that activate keyboard shortcuts to perform specific functions within the software, which we will detail later.

The project uses Python, with PyAutoGUI to control keyboard inputs and Tkinter to design the panel. Each button on the panel triggers a unique function that activates the relevant keyboard shortcuts through PyAutoGUI's `hotkey` function. For example, the `insertClick` function inserts a file into the Timeline:

```
def insertClick():
    time.sleep(0.1)
    pyautogui.hotkey('alt', 'tab')
    time.sleep(0.1)
    pyautogui.press(',')
```

```
pyautogui.hotkey('shift', '3')
```

We've added a short delay in this function (and all others) to allow the `alt + tab` key combination to execute appropriately, switching back to the Premiere window from our panel. Once the switch is made, the relevant keyboard shortcut—here, pressing the comma (`,`) key to place the file into the Timeline—can be executed.

The process works as follows:

- The user presses a button on the panel.
- The button triggers its associated function.
- The function activates the corresponding keyboard shortcut in Premiere.
- The operation is completed automatically and efficiently.

As mentioned, we used the Tkinter library to design the panel. The library's built-in functions allowed us to define:

- Panel window size.
- Button sizes.
- Font sizes.
- Button placement.

Our software employs the `mainloop` function from Tkinter, ensuring that the buttons remain active and responsive at all times. This "listening" feature allows the user to press any button whenever needed, keeping the panel buttons ready to respond.

Using the panel enables automatic operations, enhancing both the efficiency and ease of use in Premiere, meeting our primary goal of creating an efficient and valuable tool.

Most importantly, we aimed to provide Tal with a solution for the challenge of dragging files into the Timeline. The panel allows Tal to select the placement of his file using arrow keys and a setting to adjust the number of segments (details provided later). When the button is pressed, the file is placed in the Timeline, eliminating the need for dragging. This feature gives Tal greater independence, and according to him, "this software is a breakthrough for me." We believe that our solution provides Tal with a versatile, efficient, and fast tool that offers both ease of use and a sense of security.

3.1. Success criteria for our project

- The proposed solution will improve our client's accuracy while working with Adobe Premiere.
- The proposed solution will shorten the time spent while working with Adobe Premiere.
- The proposed solution will make Tal more independent, meaning less dependent on his main caregiver.

4. Research Process

To ensure that the suggested concept meets the user's requirements, we followed the User-Centered Design approach phases [Shneiderman and Plaisant, 2005]**:**

4.1. Define: Initial Planning and Decision Making

The first stage of our work process focused on meetings with Tal to understand how the panel could be tailored to make his workflow as easy and efficient as possible. We began by researching Adobe Premiere and its capabilities. Following this, we conducted several additional meetings with Tal to observe how he uses the software. Tal demonstrated his video editing process, and we noted the steps he took to achieve the final result.

In the second stage, we aimed to improve the dragging operation, which posed a significant challenge for Tal. We discovered that the software has keyboard shortcuts that allow for automated actions with a single button press, and we decided this was the best way to enhance Tal's user experience. We explored various programming languages and ultimately chose Python, a popular language in automation, combined with the Tkinter library to display the software's GUI.

We knew Tal used TD Control to replace mouse and keyboard functions, so we wrote a single function that inserts the video file into Premiere's Timeline to test whether Tal could use our panel. This test confirmed that the panel could indeed support a variety of actions for Tal. After receiving confirmation, we discussed with Tal any additional actions he wanted to include on the panel. Tal suggested that it would be beneficial if he could not only drag files to the desired location but also delete video sections. Moreover, he wanted the deletion to automatically merge the remaining parts of the video, eliminating the need to drag the remaining segments after deletion.

4.2. Development: Code Implementation

At this stage, we began developing the necessary functions:

- **INSERT** – A button that allows Tal to insert the file he wants to edit into the Timeline.
- **LEFT & RIGHT ARROWS** – Using these arrows, Tal can move the cursor to select the starting position of the file he wants to edit in the Timeline.
- **CUT** – This function lets Tal set the beginning and end points of the segment he wants to delete.
- **DELETE** – This function enables Tal to remove the selected segment from the video file.
- **SAVE PROJECT**– Pressing this button opens a window where Tal can choose the location to save the project file.
- **COPY** – This button lets you select a video segment or a portion of it and copy it for placement at the desired position in the timeline.

- **PASTE** – This button allows you to paste the copied video segment or part of it into the desired location in the timeline.
- **TIMELINE** – This button lets you select the Timeline window when you want to focus your work on it.
- **RIPPLE & DELETE** – This button allows you to delete a selected segment and automatically merge the remaining edges, eliminating any gaps.
- **OPEN PROJECT** – This button lets you import a project from your computer for editing.

4.3. Prototype

During another meeting with Tal to let him test the panel, he requested an additional feature that could further improve his workflow in Premiere. Tal asked if we could introduce an option for him to choose a number between 1 and 10, allowing him to move the cursor on the Timeline by the selected number of segments using the arrow keys.

We explored how to implement this feature and decided to add three more buttons to the panel:

- **Display the Number** – This button displays the number selected by the user.
- **"+" Button** – This button allows the user to increase the variable's value which determines the number of segments the cursor will move.
- **"-" Button** – This button allows the user to decrease the variable's value which determines the number of segments the cursor will move.

Integrating these buttons into the panel gives Tal the flexibility to choose how many segments the cursor should jump rather than moving it step by step as we originally planned.

Additionally, we added a feature to enhance efficiency: when the arrow keys are pressed, the panel automatically returns to the forefront of the window. This allows Tal to avoid reopening the window each time he uses the arrows, keeping the panel conveniently accessible.

4.4. Evaluate and Refine: Final Code Refinement

In our last meeting with Tal, we installed the final version of the panel, allowing him to use it actively. We asked him to assess whether the panel truly improved his ability to work with Premiere, whether his working time decreased, whether his level of accuracy improved, and whether he was able to work independently without the assistance he previously needed.

To prove our user experience Tal in the panel we created for him, we tested every button on the panel together with Tal, we saw that the size of the buttons and their location allows Tal to work effectively with the help of focus. We tried to find bugs that

could be in the panel while trying together with Tal for a scenario of several actions to see that he is indeed able to edit videos with the help of the panel. The actions were a test of several different paths such as: inserting a video segment into the TIMELINE, deleting an unwanted segment and deleting the spaces of the video segment that were created as a result of the deletion. And we saw that we were able to eliminate any option of unnecessary disorder that could affect Tel's performance. This effort is intended to prove that our panel is efficient and useful for Tal and meets all the requirements that Tel demanded at the beginning.

Tal and his assistants presented us with very good feedback regarding the use of the panel and how Tal can work with it and improve his independence.

Tal shared his thoughts with us, saying, "This is a breakthrough for me. Using the panel has improved my work wonders."

As part of the evaluation conclusions, we saw that Tal asked for a slightly smaller panel screen to appear, but we explained to him that in order for the panel to work by focusing the view, the window size must be large enough.

We explained to Tal that in order to improve the use of the panel, the panel is minimized in every action he uses and this is so that it does not interfere with the Adobe Premiere window, except for the use of arrows where it is important that the panel not be minimized because this action can be continuous when using it.

In addition, we examined our system's usability – e.g., learnability, efficiency and ease of use, using the system usability scale – SUS – developed by Brooke (1995)[5]. The final result is 97.5, so it is understandable that the panel we prepared succeeds in meeting the goals and objectives that were set before it and improves the user experience of Tal. The filled SUS Queries can be found in the ‘Appendixes’ chapter.

5. Challenges

Premier Software Versions:

After building a prototype on our personal computers and conducting an initial test on Tal's computer, we noticed that the functionality wasn't working as expected. Further testing on other computers in Tal's workroom revealed that the software settings on Tal's computer were different. To address this, we updated the Premier software on Tal's computer to the latest version and familiarized ourselves with its settings. This allowed us to make the necessary adjustments to ensure our software functions optimally on Tal's system.

Security Issues:

When we attempted to run our software on Tal's computer, we encountered a security block. After investigating the root cause, we discovered that the method we used to create the software—designed to make it as light and fast as possible—triggered a

security warning on other computers. To resolve this issue, we compared the advantages and disadvantages of various development approaches, prioritizing one key parameter: ensuring that our software does not raise any information security concerns.

Panel Design:

Initially, the panel design featured images and text on each button (see figure 3). However, during our meeting with Tal and his assistant, we realized that this combination could overwhelm Tal and cause delays in his routine tasks. To simplify the interface, we decided to remove the images from the buttons and use minimal colors and text(see figure 4).

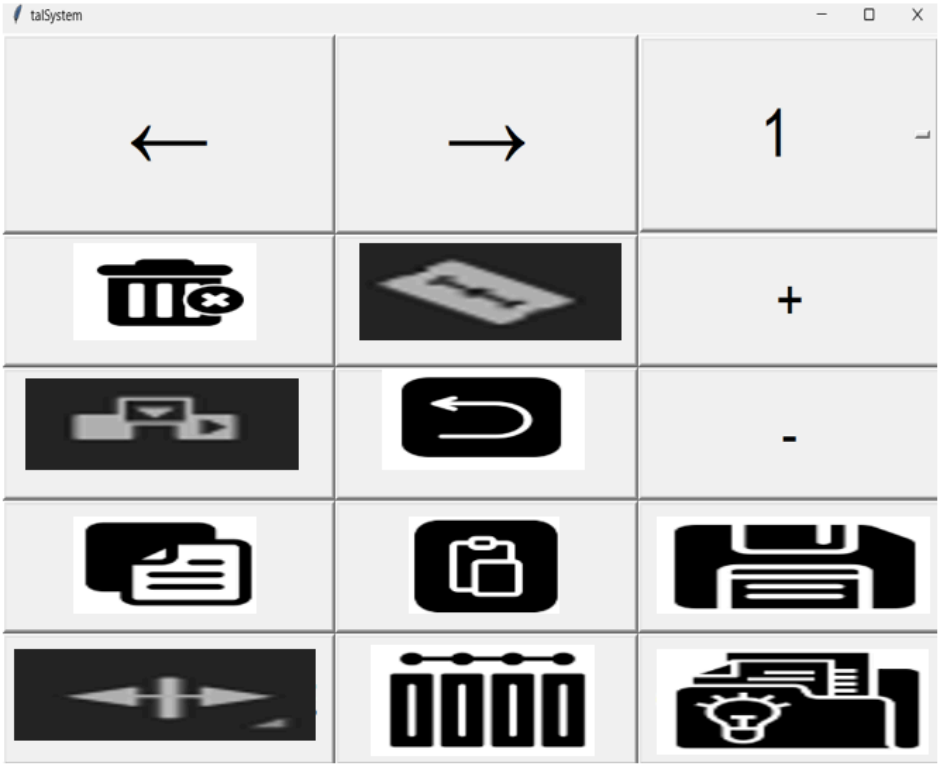


Figure 3: The initial system panel offered to tal with icons.

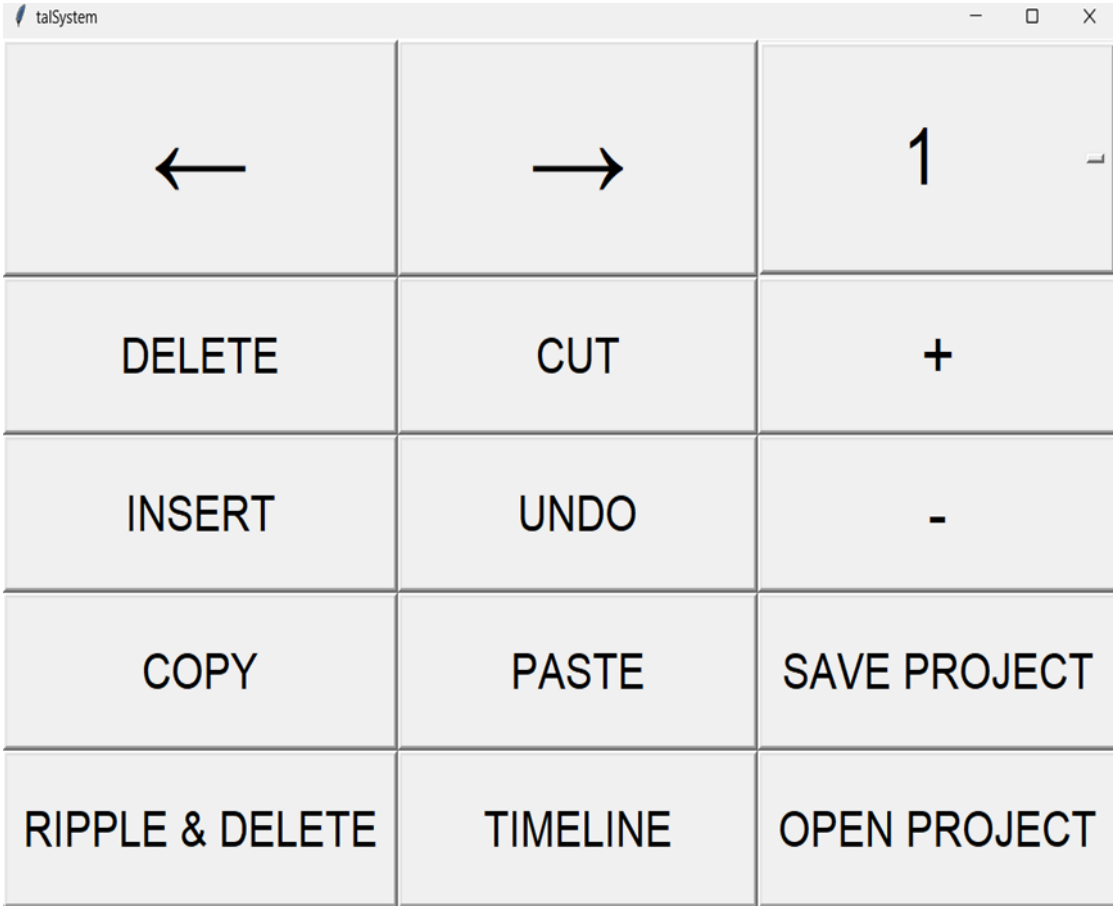


Figure 4: The final panel with text Tal chose as his preferred option

Button Size:

During our meetings with Tal, we realized that the size of the buttons is a crucial aspect of the panel's design. We needed to strike a balance between making the buttons large enough for Tal to focus on them with the help of the eye-tracker sensor while also ensuring that the panel wouldn't take up too much screen space. To resolve this dilemma, we recorded a video of our client using the interface during a routine activity. This allowed us to estimate the optimal button size, minimizing the chances of missed presses.

Panel location:

As previously explained, Tal uses the TD Control software to replace keyboard and mouse functions. One of our challenges was determining the optimal placement of the panel so that it wouldn't disrupt Tal's workflow. We created several panel sizes for Tal and experimented with positioning the options in different locations within the window using pixel-based adjustments. We found that the panel works best when positioned on the right side of the screen, considering the primary windows Tal uses in Premiere. As illustrated later in the book, the panel is quite large and occupies significant space on the screen. However, given Tal's use of an EYE-TRACKER and his limited eyesight, we concluded, together with him, that a larger panel would be the most effective solution.

Figure 5, Shows the option that we initially thought of producing for Tal, but it is the option that was not accepted in the end, because of the width of the panel that hides the rest of the Premiere window.

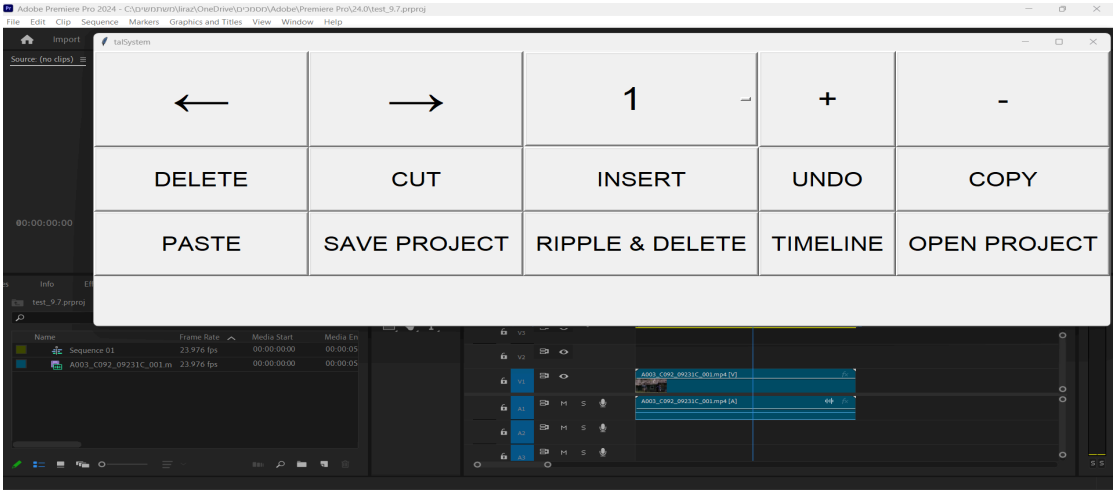


Figure 5: The first option for tal.

Figure 6, Shows the option chosen by Tal as his preferred option when he can see the rest of the screen of the Premiere software and the size of the panel helps to integrate Tal's work with the software alongside the panel we prepared for him.

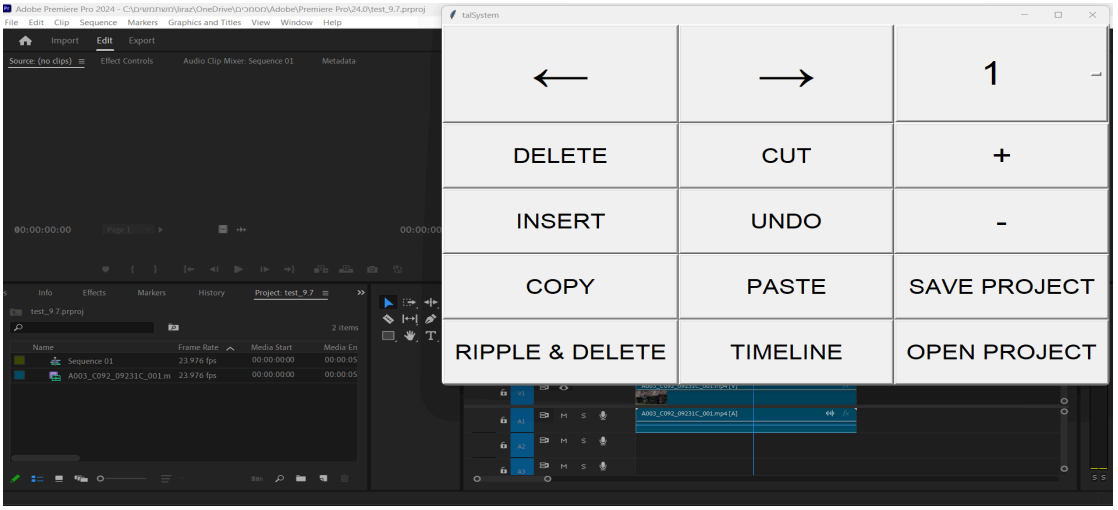


Figure 6: The final panel for tal.

6. Results and Conclusions

At the beginning of the process, Tal used the TD CONTROL software to edit his video files. He emphasized the dragging operation as a significant challenge he wanted us to address.

Initially:

- Tal would select the drag function in TD CONTROL.
- He would click on the desired file and drag it.
- Tal had to choose the exact point in the TIMELINE where he wanted to place the file.

In practice, this method often resulted in imprecise placement. The act of dragging with his eyes was challenging for Tal, and it sometimes caused him to lose focus with the TOBII device.

After using the panel:

- Tal would click on the TIMELINE window, an action that he finds easy.
- Tal will select the number of segments he wants to jump using the "+" button
- He would use the arrow buttons on the panel to select the exact position where he wanted to place the video.
- Tal would then click on the video he wanted to edit and press the INSERT button.
- The video would be placed precisely at the starting point Tal had selected with the cursor in the TIMELINE window.

*You can see later under the **User Guide** chapter the use of the panel.

6.1 Future Work

The panel we've created is designed to be compatible with keyboard shortcuts in Premier software. However, if needed, we can also customize the panel for other programs that utilize keyboard shortcuts. While our software is currently tailored to specific individuals and software, it is designed in a way that allows the code to be integrated into other software. This could provide more opportunities for individuals with disabilities who use software other than Premier.

7. Lessons Learned

Methodology

Our research process was conducted in collaboration with Tal and his assistant. We decided to film Tal as he performed routine video editing tasks, allowing us to ask questions and gain a deeper understanding of the challenges that needed to be addressed. This hands-on approach taught us that relying solely on documented requirements is insufficient; it's essential to engage directly with the client and assess real-world difficulties in the field. During this process, we uncovered additional issues beyond the initially anticipated requirements, and we promptly adapted our approach to address them.

Project Management

At the beginning of the project, we planned to divide the tasks and functions among ourselves to work in sync with Git, with each team member responsible for specific parts. Initially, this worked well, but as we progressed, we constantly consulted each other on various aspects, such as panel design and button placement, among other topics. We decided it would be more efficient to work together, so we created a schedule that accommodated everyone's personal constraints and allowed for regular meetings to work on the project. This turned out to be the best decision we made, as it enabled us to support and assist each other, leading to the successful completion of the project.

One thing we would have liked to change was to have more frequent meetings with Tal. However, scheduling conflicts with work hours made this difficult. We managed to compensate for this by holding Zoom meetings with Tal's supervisor, teaching her how to use the software so she could assist Tal with any difficulties he encountered.

Technical Skills

During the research phase, while selecting the technologies we wanted to work with, we encountered numerous mentors online, each offering in-depth insights into their areas of expertise. This exploration expanded our knowledge of various technologies, including programming languages, code libraries, and different work environments. Although we ultimately chose the technologies best suited to our project, we gained a broad set of skills and tools that we believe will be valuable as we continue our journey in the industry.

Critical Thinking

One of the challenges we encountered during the project was Tal's limited availability for meetings, in which we needed to conduct essential tests and assessments. To overcome this, we had to adopt the perspective of Tal using our software, considering his likely reactions to the hardware, response times, and physical limitations. By thinking in this way, we identified constraints during the development process, allowing us to make the most of our few meetings with Tal and successfully complete the software development within the given timeframe.

Future Research Directions

The primary gap we identified, which forms the basis of our project, is the challenge faced by individuals with physical disabilities in operating computer hardware when using certain software. While modern computers offer extensive accessibility features and some programs provide a wide range of keyboard shortcuts to activate various functions, individuals with disabilities still require specialized auxiliary software tailored to their specific needs. This software is crucial to enabling them to work as seamlessly as possible within their professional environments.

Broader Impact

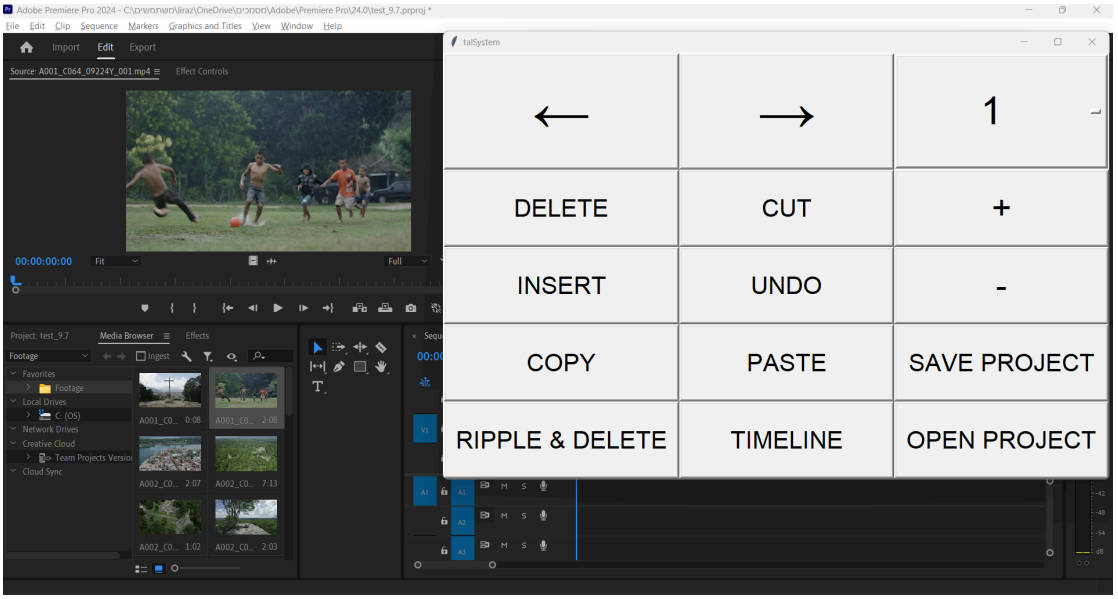
We believe this project will inspire future students to pursue research and development in creating assistive software for individuals with disabilities. We hope that more companies will embrace this approach to developing assistive technologies. Collaboration between inquisitive students and development companies with advanced tools can significantly enhance the quality of life for those with disabilities, provide students with valuable experience, and boost the reputation of companies committed to inclusive innovation.

8. User guide

TalSystem is designed to simplify and automate complex computer actions with just a click. Our grid panel, equipped with customizable buttons, executes keyboard shortcuts to help our customer perform tasks that might otherwise be challenging or time-consuming for him.

In this user guide, we will provide detailed explanations of the functionality of each button and how to use it.

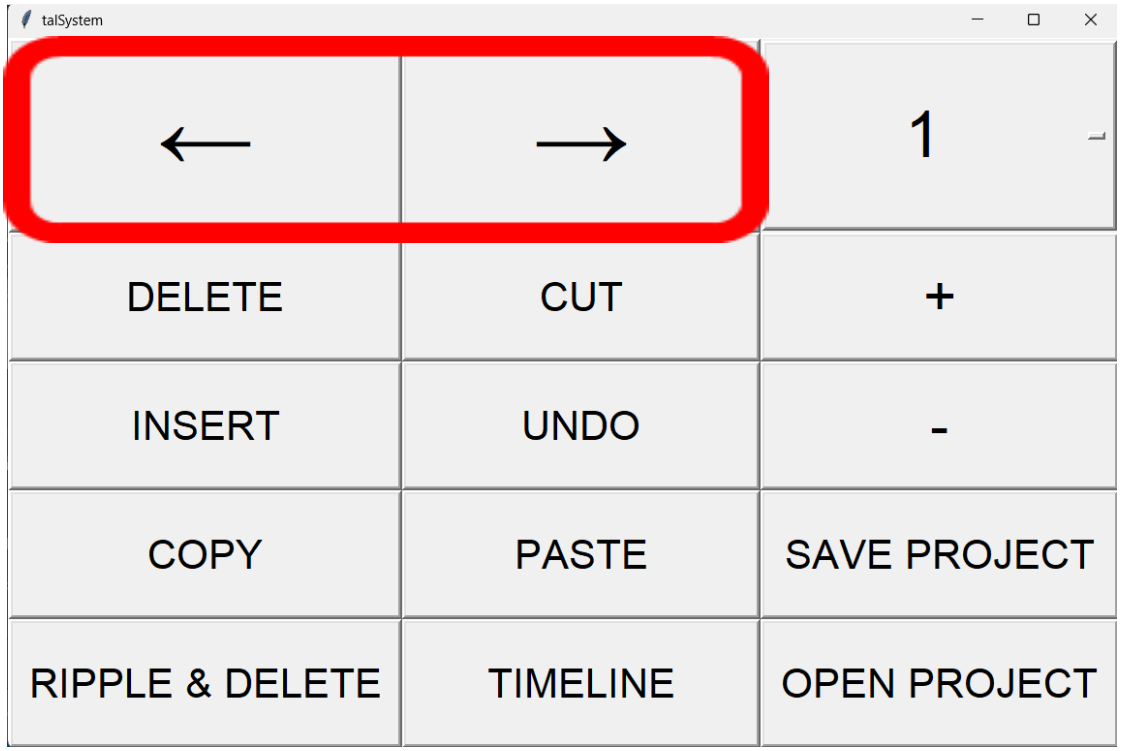
TalSystem Panel

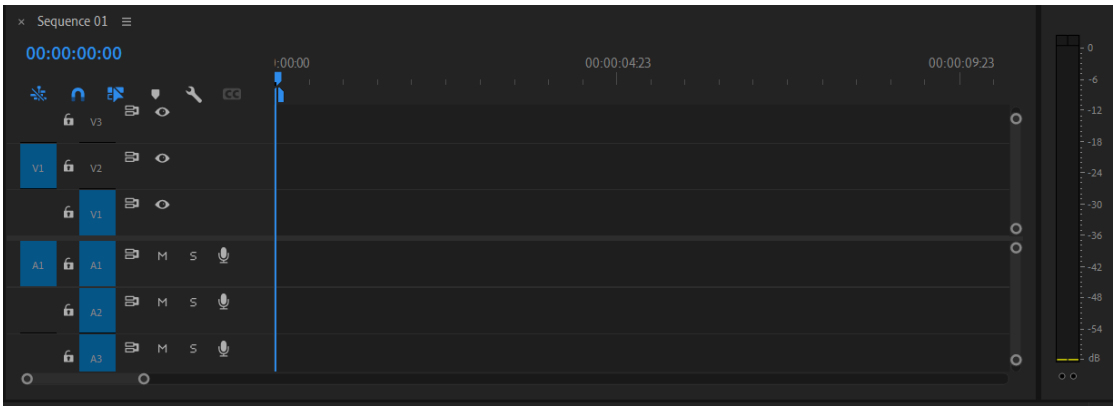


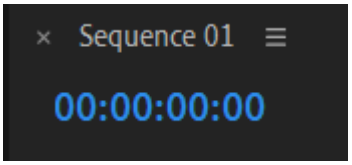
Arrows buttons

Allows the user to move the cursor on the timeline forward and backward.

Note: The timeline should be marked when working with the arrows.

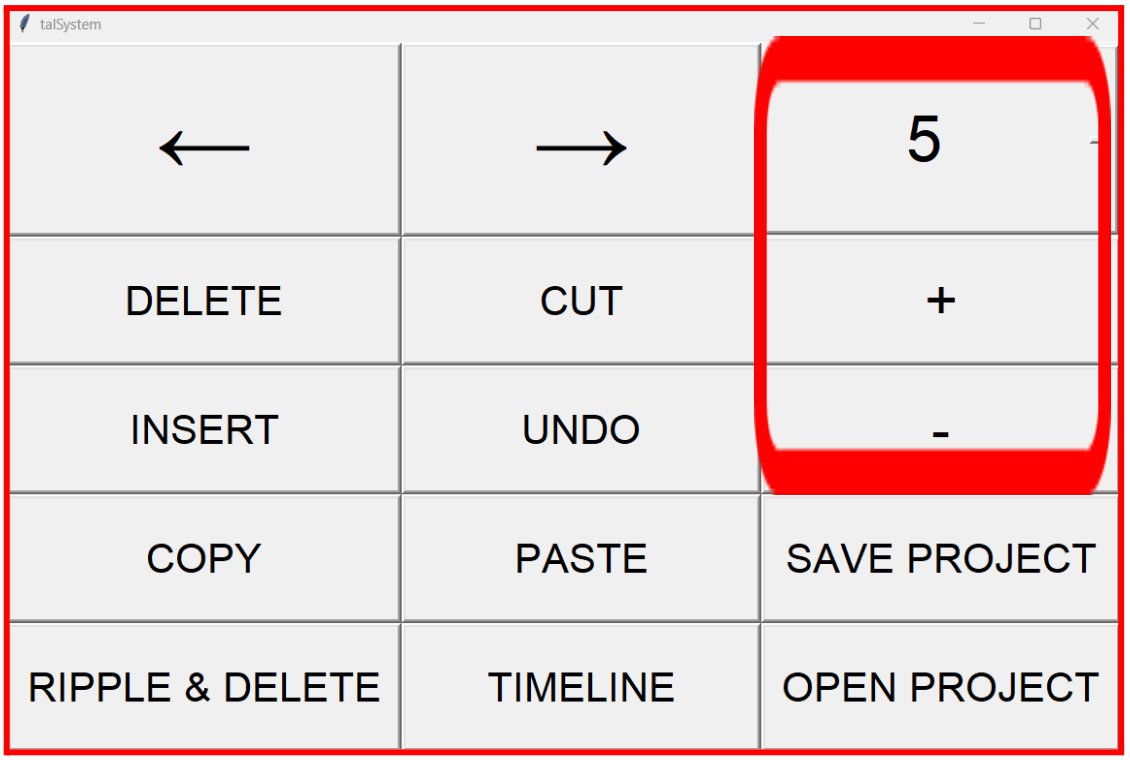

talSystem
←
→
1
DELETE
CUT
+
INSERT
UNDO
-
COPY
PASTE
SAVE PROJECT
RIPPLE & DELETE
TIMELINE
OPEN PROJECT

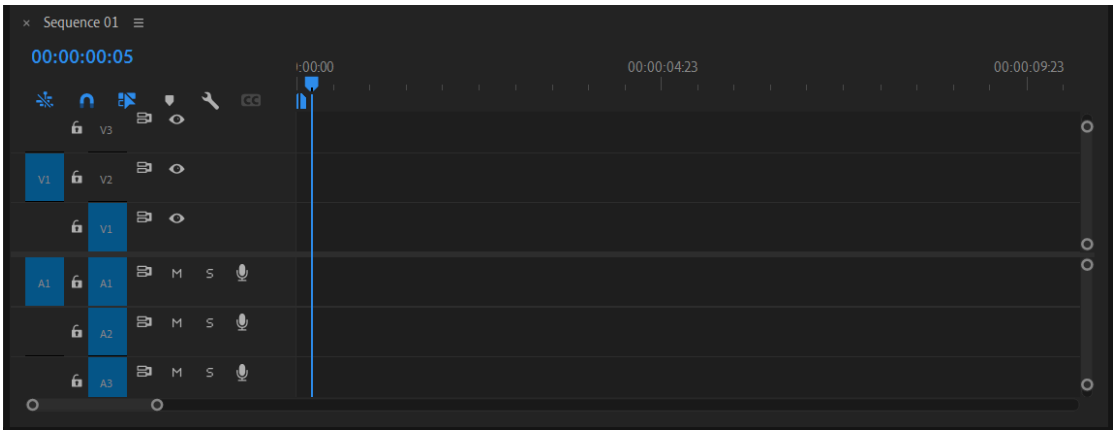

Sequence 01
00:00:00:00
:00:00
00:00:04:23
00:00:09:23
V3
V2
V1
A1
A2
A3
M
S

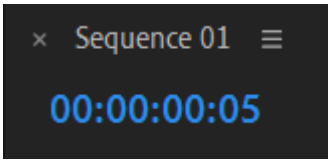

Sequence 01
00:00:00:00

Time intervals

Allows the user to increase and decrease the time interval for moving the cursor in the timeline. This action replaces dragging the cursor with the mouse. The number 5 is an example of the number of segments the user chooses to move the cursor within the TIMELINE. The number 5 is an example of the number of segments the user chooses to move the cursor within the TIMELINE. In the picture of the TIMELINE window, allow the user to press the + button five times to set the cursor jumps and then press the "right arrow" button once.



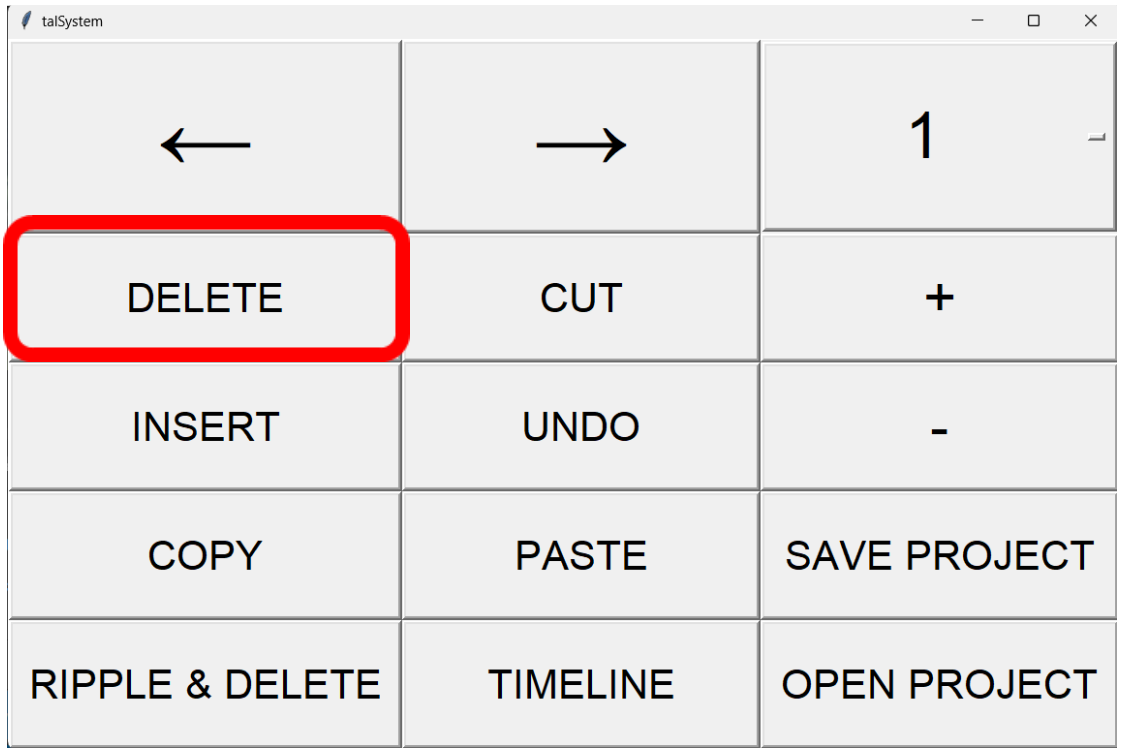




Delete

Deleting a timeline video segment.

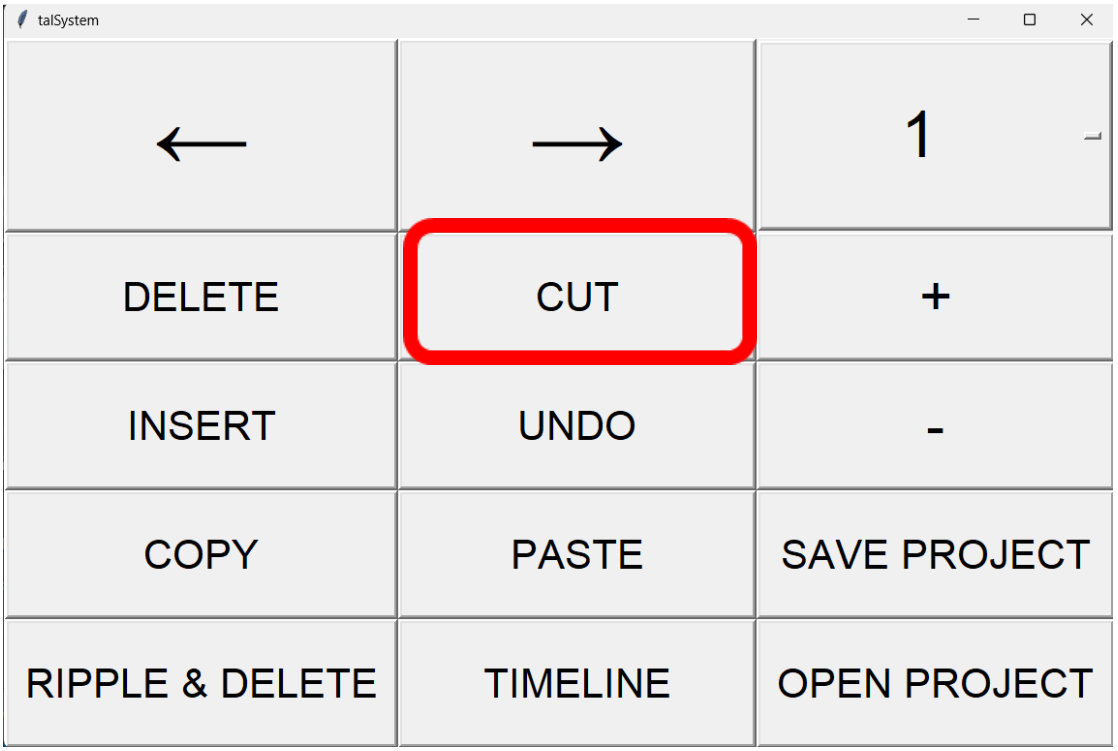
Note: The cursor should be on the video segment to be deleted.



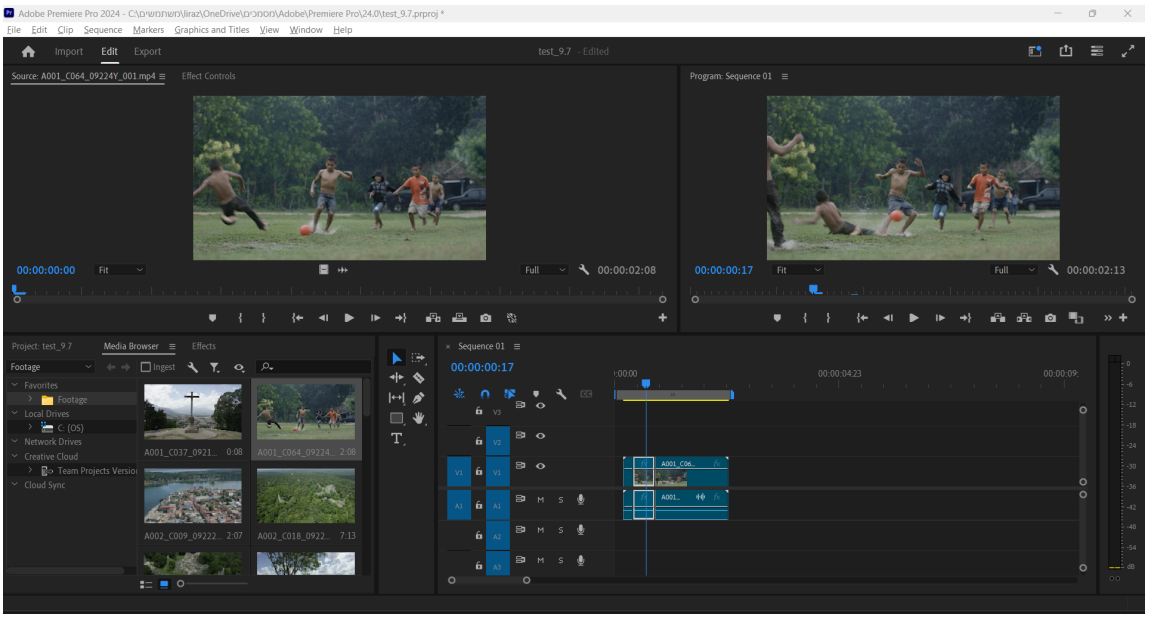
Cut

Trimming a video clip.

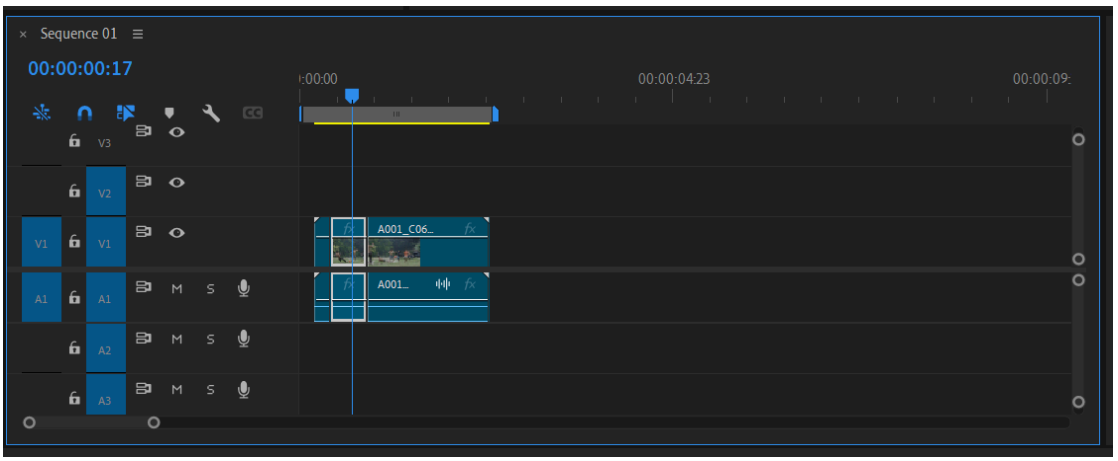
Note: The cursor should be on the video segment to be trimmed.



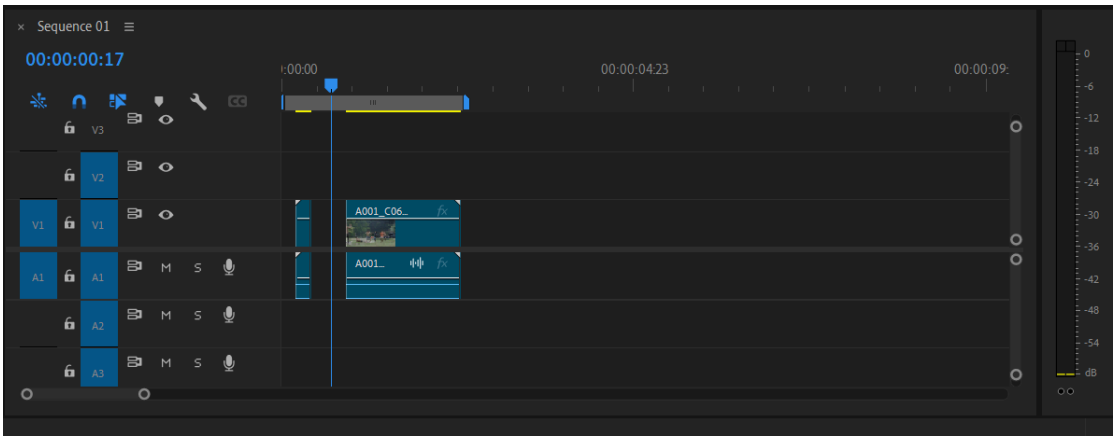
You can see in the pictures an example of using the CUT and DELETE buttons when the CUT button is used to set the start and end points of the segment that we want to delete using the DELETE button.



The following image shows the section cut using the CUT button before deletion.



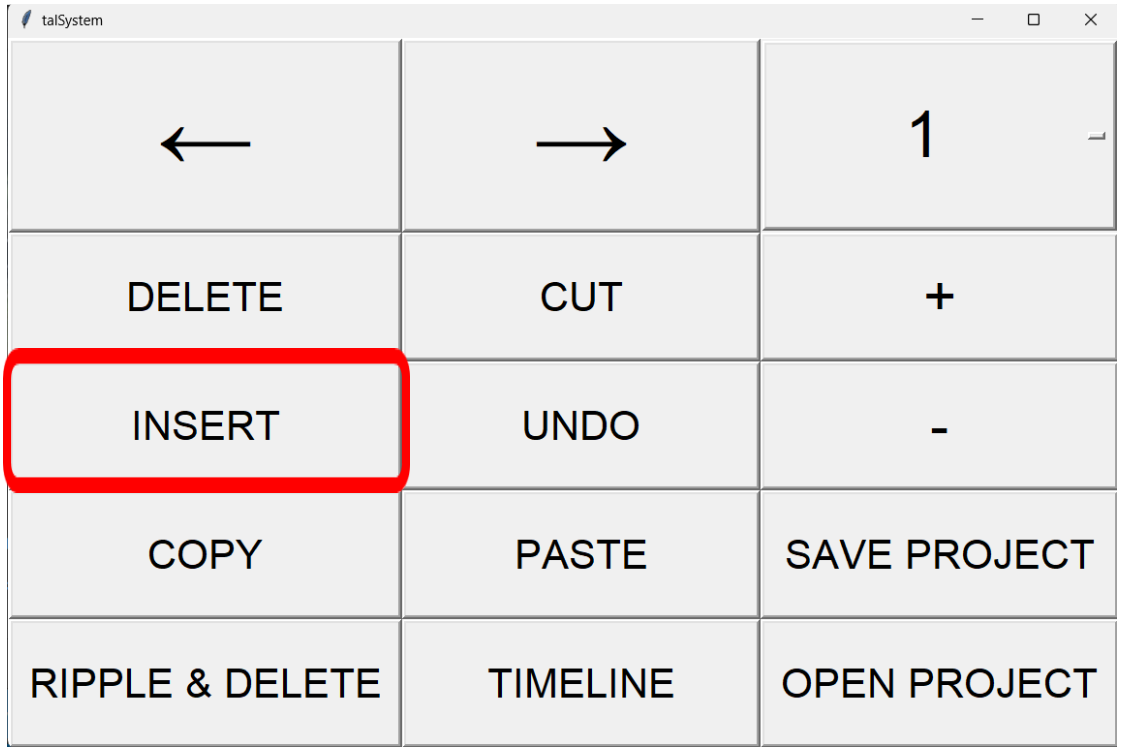
After using the DELETE button.



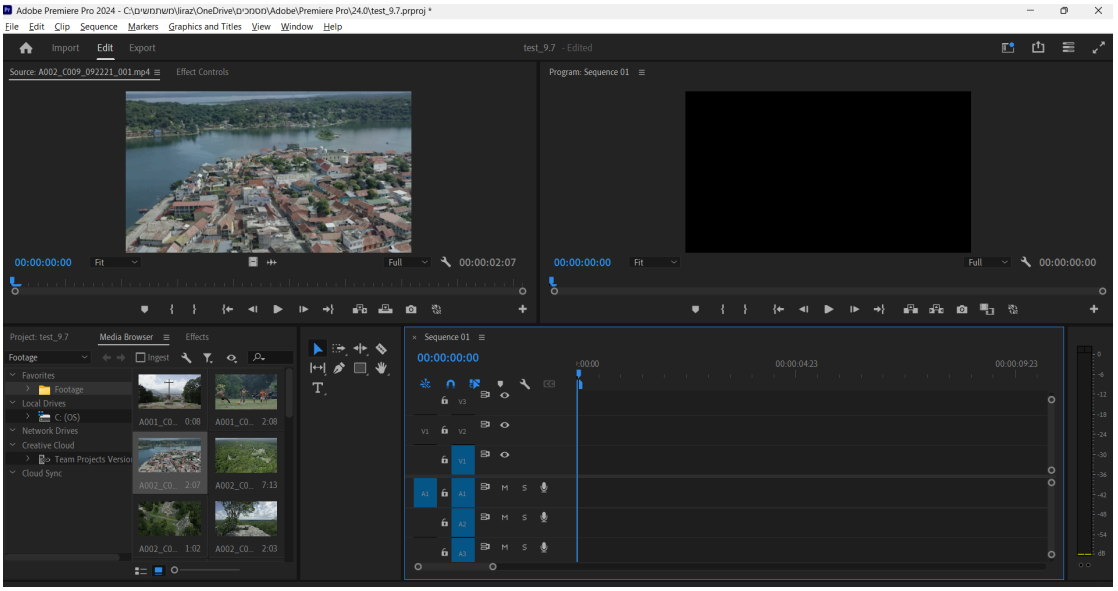
Insert

Sending the video segment from the Source monitor to the Timeline at a specific point where the cursor is. This action is actually a replacement of the drag & drop action to the timeline.

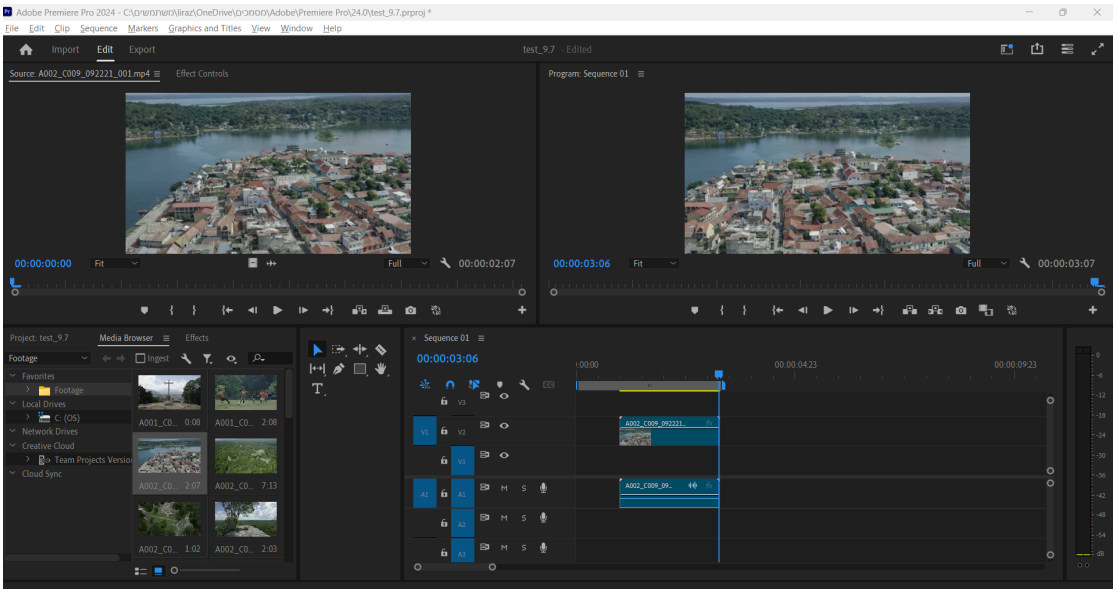
Note: The Source monitor should be marked when working with the Insert button.



Before

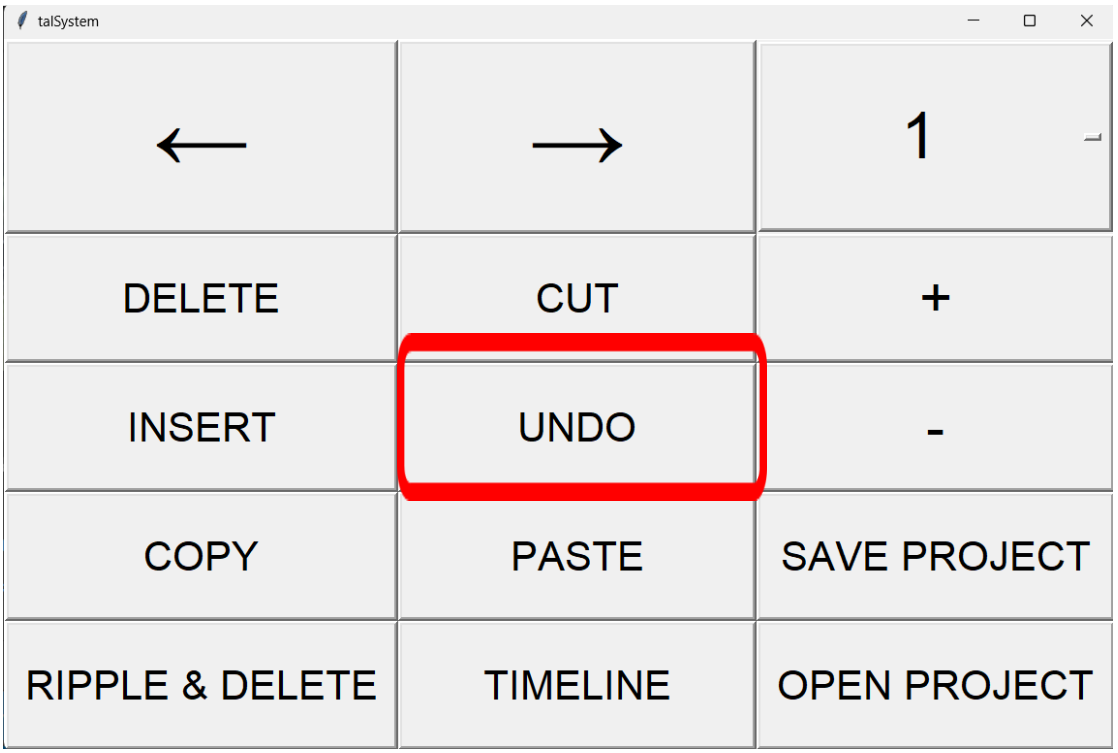


After


Import
Edit
Export
test_9.7 - Edited
Source: A002_C009_092221_001.mp4
Effect Controls
Program: Sequence 01
00:00:00:00
00:00:02:07
00:00:03:06
00:00:03:07
Project: test_9.7
Media Browser
Effects
Footage
Favorites
Local Drives
Network Drives
Creative Cloud
Cloud Sync
Sequence 01
00:00:03:06

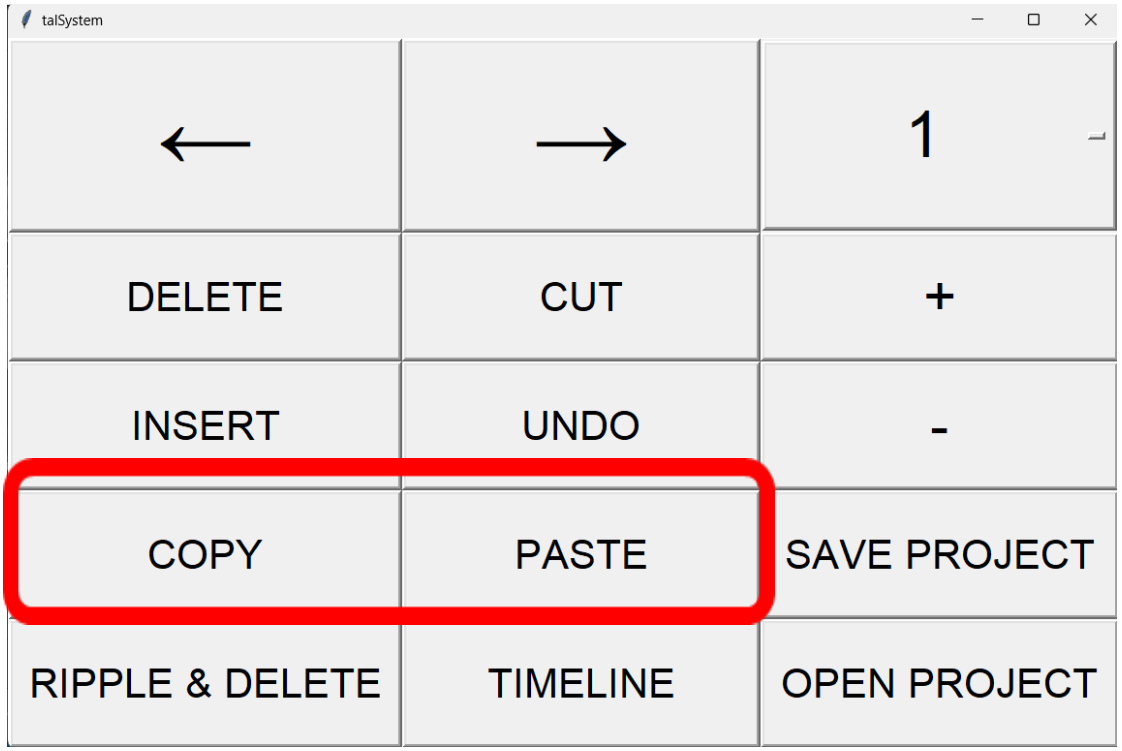
Undo

Allows undoing the last action performed.

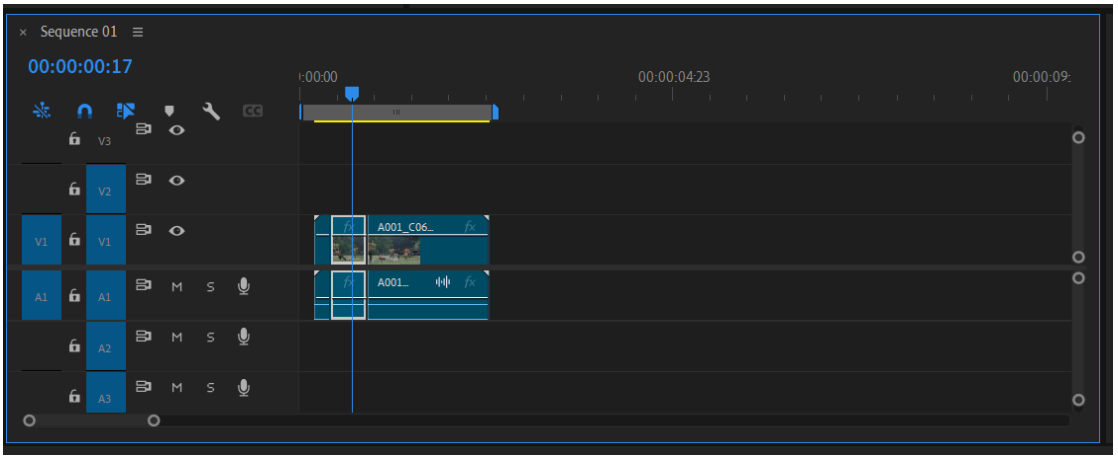


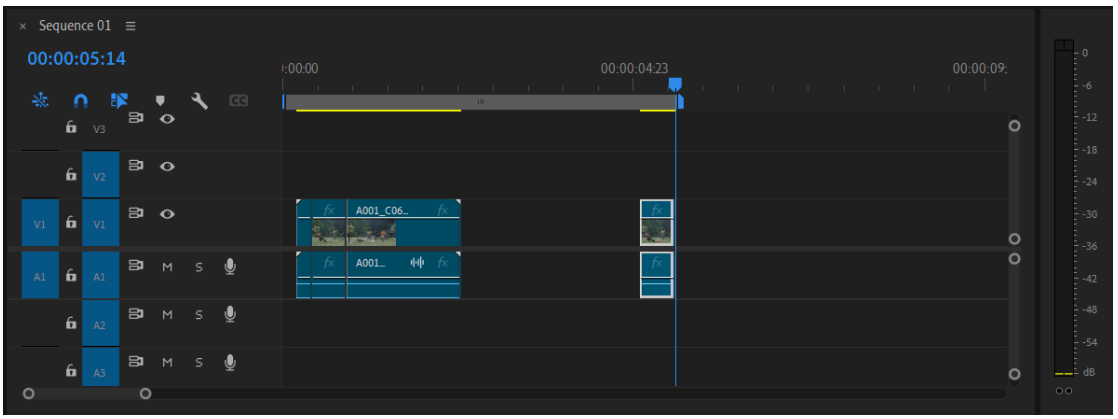
Copy and paste

Copy and paste actions like the familiar ctrl+c / ctrl+v actions.



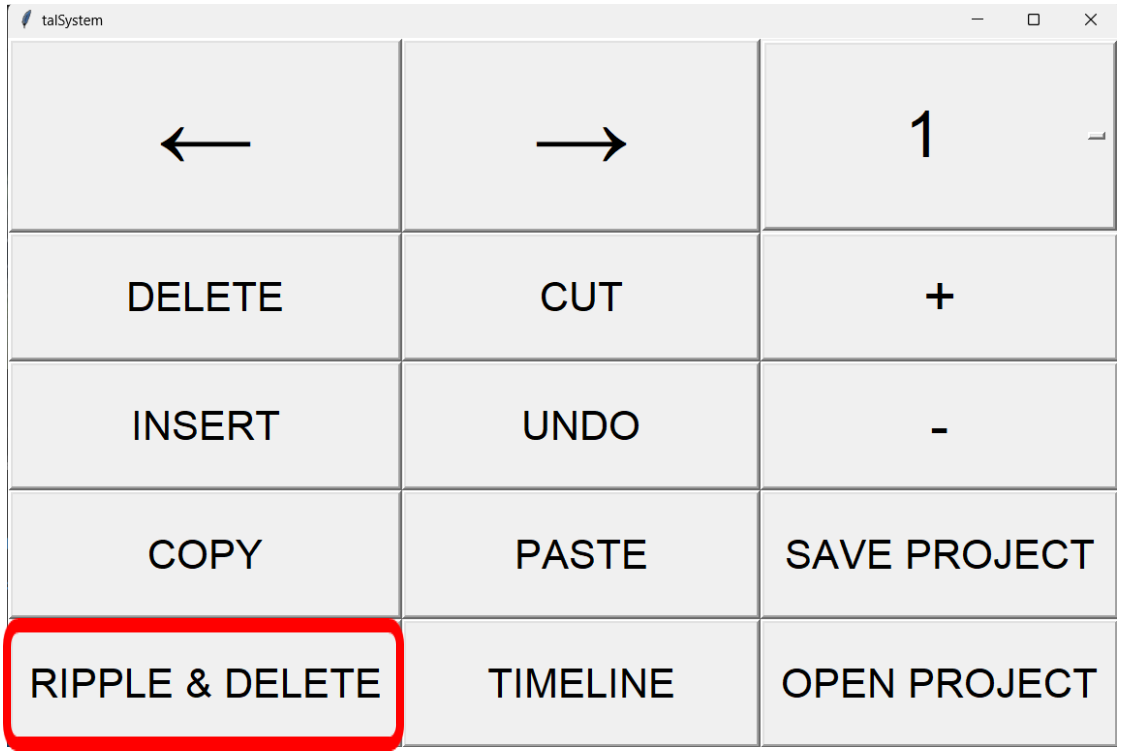
In the pictures here, you can see an example of using the COPY + PASTE buttons when the user can choose to copy the section in its entirety or cut using CUT the part he wants to copy (as in the example below) and move this part using the PASTE button to the place he wants in the TIMELINE.



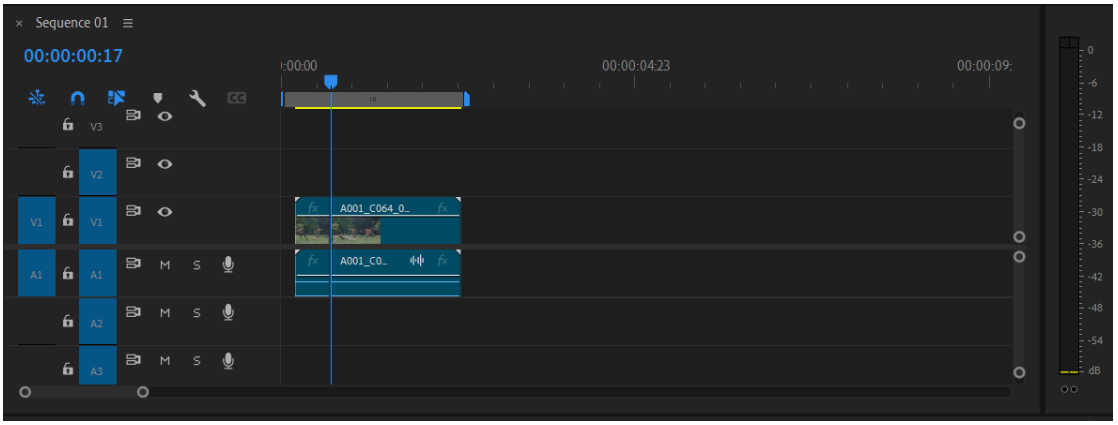


Ripple & Delete

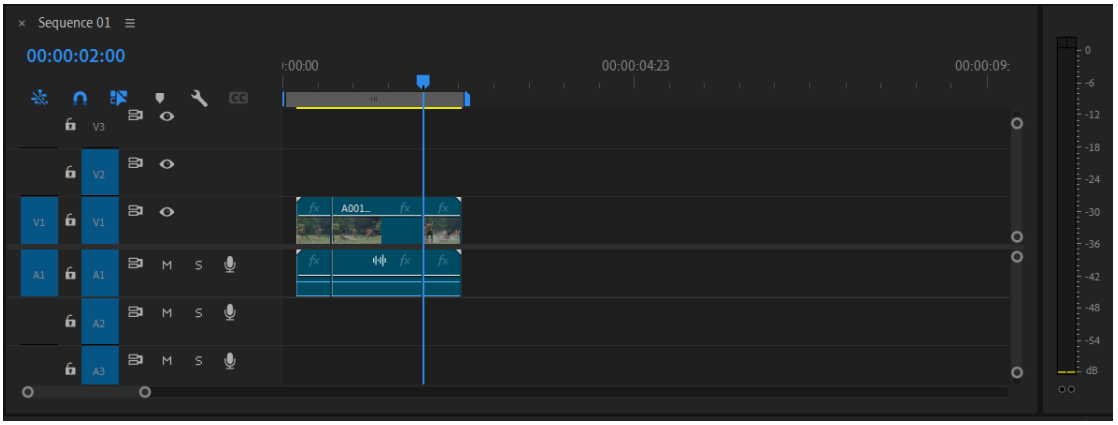
Using this button allows the user to delete the segment he wants from the video file. Unlike the DELETE button, the RIPPLE & DELETE button deletes the segment and unites the two ends caused by deleting the segment.

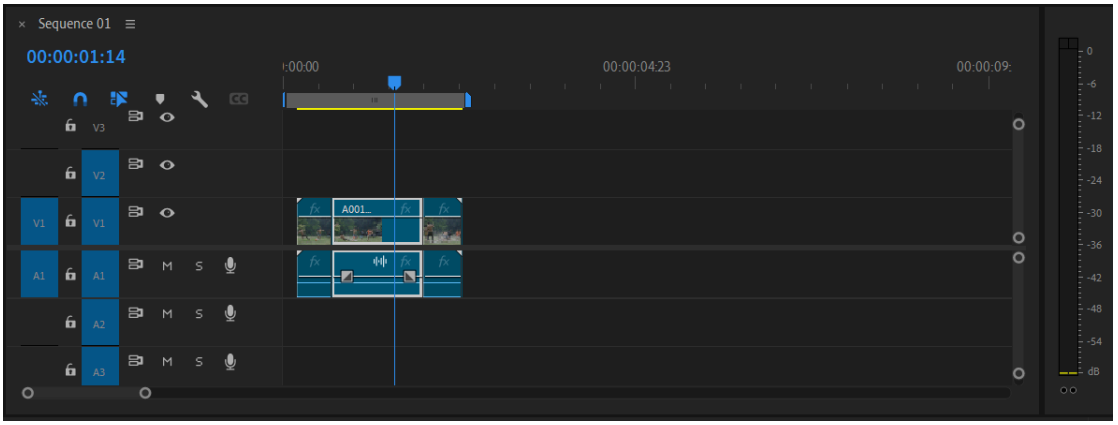


Start point

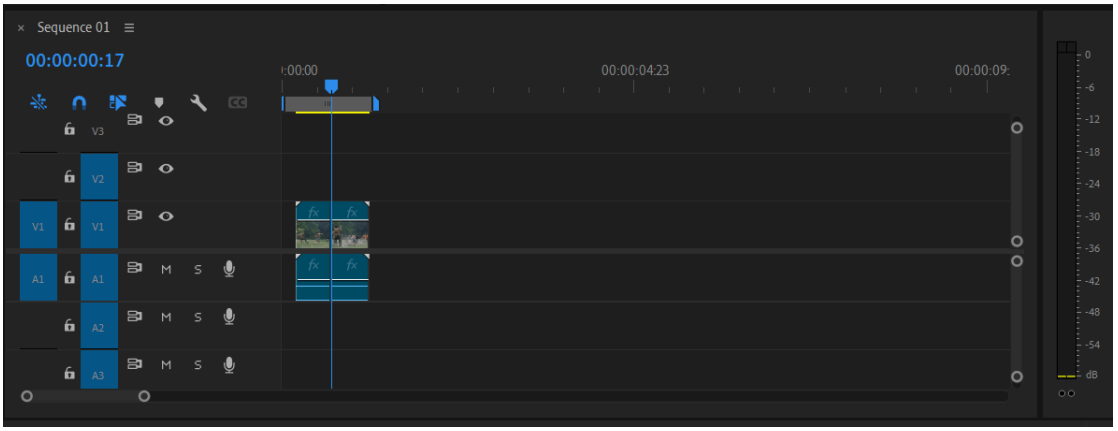


.End point





Use Ripple & Delete button



Save and open the project

Keyboard shortcuts for saving and opening a project to enable continuous editing work.


talSystem
←
→
1
DELETE
CUT
+
INSERT
UNDO
-
COPY
PASTE
SAVE PROJECT
RIPPLE & DELETE
TIMELINE
OPEN PROJECT

9. Maintenance Manual

Instructions for Using the Software:

- **Install Adobe Premiere -** Adobe Premiere is not a free software. First, create an account through the company's website, then download and install the software on your computer.(https://www.adobe.com/products/premiere.html)
- **Place and Run the Panel Executable**:- You can place the exe file on the desktop and run it from there. The software is currently on Tal's computer. You can transfer the exe file to other computers using a flash drive.

Enable Num Lock:- Before starting work, the user must press the "Num Lock" button on the left side of the keyboard to ensure the panel's arrow keys align with the keyboard shortcuts.

10. Appendixes

שאלון שביעות רצון מהמערכת שפותחה לטל:
אנא סמן/ני את רמת הסכמתך עם ההיגדים הבאים, עבור כל שאלה סמן/ני X מתחת לרמה שהכי מתאימה לך.

מס"ד	שאלה	מסכים בהחלט				לא מסכים בכלל
		5	4	3	2	1
1	נראה לי שהייתי מעוניין/נת להשתמש במערכת מסוג זה לעיתים קרובות.	x				
2	לדעתי המערכת מורכבת מידי שלא לצורך.					x
3	מצאתי את המערכת נוחה לשימוש.		x			
4	נראה לי שאזדקק לעזרתו של איש טכני, כדי שאוכל להשתמש במערכת.					x
5	נראה לי שהפונקציות השונות של המערכת שולבו היטב ביחד.	x				
6	נתקלתי במקרים רבים של חוסר עקביות לאורך השימוש במערכת.					x
7	נראה לי שרוב האנשים יכולים ללמוד להשתמש במערכת במהירות.	x				
8	לדעתי המערכת הינה מסורבל לשימוש.					x
9	הרגשתי מאד בטוח באופן השימוש שלי במערכת.	x				
10	הייתי צריך ללמוד הרבה לפני שהצלחתי להשתמש במערכת.					x

<u>נקודות לשימור:</u> המערכת נוחה לשימוש ומאפשרת לי לערוך יותר באופן עצמאי, אני מצליח לבצע את פעולות העריכה באופן מדויק ויעיל באמצעות כפתורי הפאנל מה שלא הצלחתי לעשות לפני.

<u>נקודות לשיפור:</u> הפאנל גדול מידי, אך לפי מה שאני מבין יש צורך שהפאנל יהיה גדול על מנת שיהיה אפשר לבצע את הפעולות עם מיקוד מבט והכ]תורים יהיו מספיק גדולים.

Calculate

((5-1) + (5-1) + (4-1) + (5-1) + (5-1) + (5-1) +(5-1) + (5-1) + (5-1) + (5-1)) * 2.5 = **97.5**

11. References

1. Kübler, A., Holz, E. M., Sellers, E. W., & Vaughan, T. M. (2015). Toward independent home use of brain-computer interfaces: a decision algorithm for selection of potential end-users. *Archives of physical medicine and rehabilitation*, *96*(3), S27-S32.
2. Caligari, M., Godi, M., Guglielmetti, S., Franchignoni, F., & Nardone, A. (2013). Eye tracking communication devices in amyotrophic lateral sclerosis: impact on disability and quality of life. *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, *14*(7-8), 546-552.
3. Edughele, H. O., Zhang, Y., Muhammad-Sukki, F., Vien, Q. T., Morris-Cafiero, H., & Agyeman, M. O. (2022). Eye-tracking assistive technologies for individuals with amyotrophic lateral sclerosis. *IEEE Access*, *10*, 41952-41972.
4. Majaranta, P., & Bulling, A. (2014). Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing* (pp. 39-65). London: Springer London.
5. Brooke, J. (1995). SUS - A quick and dirty usability scale.
6. Shneiderman, B. and Plaisant, C: Designing the User Interface Strategies for Effective Human-Computer Interaction fourth edition. Addison Wesley Longman, Inc., (2005).
7.