

Sravya Kuchipudi  
Professor Kaisler, Hasanov  
CSCI 6917  
30 July 2023

### Project Report: Building a GUI for Unix Commands Execution on MacOS

The focus of this week's work was to add guidelines, improve any small issues, and doing some testing. From the introduction of a GUI Instructions window to the thoughtful reorganization of buttons and the inclusion of a cautionary note, each improvement aimed to help users with comprehensive knowledge and streamline their interactions. Furthermore, the creation of a README file complements the GUI, offering a comprehensive guide for running and using the application. Together, these enhancements improve the Unix Commands GUI and creating a better user experience.

The initial improvement introduced to the Unix Commands GUI is the addition of a GUI Instructions window. The purpose of this window is to guide users on how to effectively interact with the GUI. By clicking the "GUI Instructions" button, users gain access to comprehensive usage guidelines. These instructions serve as a valuable resource, allowing users, especially those new to Unix commands, to navigate the GUI with confidence. This section was coded similar to the Command Information window other than some minor formatting changes.

The instructions emphasize the following key steps: Selecting a Command: Users are guided on how to choose a command from the dropdown list. Providing Arguments: For commands that require additional input, users are prompted to enter arguments in the designated field. Executing Commands: The "Run" button triggers the execution of the selected command with the provided arguments. Viewing Output: The output is displayed in a separate window, ensuring clear presentation of results.

There was also the inclusion of a cautionary note in the GUI instructions which was an essential add on. This note emphasizes the potential impact of certain commands, such as 'rm', 'mv', and 'mkdir', on the user's file system. By urging users to exercise caution and ensure they have backups or are well-informed about the commands' effects, the GUI prioritizes user safety and minimizes the risk of unintentional actions.

The next improvement involves updating the Command Info window to better assist users in understanding the available commands. The window title was renamed as "How to Use the Commands" and it now offers a bit more detailed information about each command. The command information is presented in a user-friendly format, making it easy to comprehend. Each command's description and template are clearly displayed, enhancing readability and accessibility.

Also, with the introduction of a new button, to further improve the GUI's navigation, the placement of buttons has been thoughtfully reconsidered. The "GUI Instructions" button is placed above the "Command Info" button to enable users to access the instructions first, allowing them to familiarize themselves with the GUI's usage before exploring the available commands. The main window also has a quick sentence to indicate the buttons and to encourage using them. A few other small aesthetic changes were made at this point to better balance the window visually.

Another improvement is the addition of a tooltip for the Arguments entry field, which displays the description and template of the selected command when the user hovers the mouse over it. This provides users with quick and contextual information about the command they are using. This was done by adding two functions: `show_tooltip` and `hide_tooltip`. The `show_tooltip` function is triggered when the user hovers the mouse over the Arguments entry field. It sets the tooltip text (description and template of the selected command) and displays a separate toplevel window (`tooltip_toplevel`) slightly offset from the

mouse cursor using `geometry()` and `config()` functions to prevent overlap or obstruction. The `hide_tooltip` function is triggered when the user moves the mouse away from the 'Arguments' entry field. It simply hides the tooltip window using `withdraw()`.

One rather important addition to the overall project was the creation of the README file accompanying the Unix Commands GUI as an essential guide, offering clear and concise instructions for installation, setup, and usage. It provides clear guidelines for launching the GUI as well as highlighting the significance of different aspects such as the "Command Info" and "GUI Instructions" buttons, ensuring users can make the most of the available features. By complementing the GUI's visual enhancements with textual guidance, the README file enhances the overall user experience, making the Unix Commands GUI accessible and user-friendly for all.

The improvements made to the Unix Commands GUI have significantly elevated its functionality and user experience. By providing a dedicated GUI Instructions window, users are now empowered with comprehensive guidelines on navigating the GUI effectively. The updated Command Info window offers helpful information on available commands, ensuring users are well-informed before executing any action. Furthermore, the reorganization of certain features like the buttons streamlines user access, making the GUI even more user-friendly. Finally, the inclusion of a cautionary note demonstrates the GUI's commitment to user safety, encouraging responsible usage of potentially impactful commands.