Project Overview: We are designing a basic graphical user interface for the traditional game "Rock Paper Scissors" with Python and the Tkinter toolkit. The player chooses from three possibilities (Rock, Paper, Scissors) and competes against the computer, which makes a random selection. The application tracks victories for the user, wins for the computer, and draws; the UI shows the current score. To reset the scores and begin a new game, the game also has a reset button.

Members of the group:

Carlos Cruz, Cole Gatewood, David Taylor

User Guide:
Use the program by doing the following actions:

Launch the given Python script.
The main game window will pop up with a scoreboard, buttons to pick Rock, Paper, or Scissors, a reset button, and a label indicating the outcome of each round.
Press the button that corresponds to your selection (Scissors, Paper, Rock).
The outcome will be shown in the result label once the computer makes a random selection.
As soon as the game is over, the scoreboard will refresh to show the result.
To start a new game and wipe the scores, click the reset button.

Design Documentation:
Our program is organized around a primary window that was made with Tkinter. This window has buttons allowing the user to pick an option, a reset button, and labels that show the scoreboard and game outcome. Several functions manage the game's primary functionalities:

When the user makes their pick, the function user_choice(user_selection) is invoked. It produces the computer's selection, chooses the winner, refreshes the scoreboard, and shows the outcome.
determine_winner(user, computer): To identify the winner, this function contrasts the user's and the computer's selections.
update_score_board(): This method applies the current scores to the scoreboard labels.
reset_game(): This method removes the result label and resets the scores.

Challenges and Solutions:
Implementing the algorithm to choose the winner based on user and computer selections was one of the biggest challenges we faced. To solve this problem, we developed a distinct function called determine_winner that can handle choice comparison and correctly identify the winner. Using Tkinter to develop the graphical interface presented additional difficulties, particularly in correctly aligning and organizing the widgets on the window. We overcame this difficulty by trying out several layouts and experimenting until we got the ideal UI design. In order to guarantee that the global variables for the scores were updated accurately following each game, further thought had to be given to their management. By adding universal variables and

updating them inside the relevant routines, we were able to resolve the problem. Overall, we were able to effectively develop the Rock Paper Scissors game with a working scoring system through teamwork and problem-solving.