Rock, Paper, Scissors in C++ using NCURSES Windows

Ryne Whitaker

Dr. Larry Morell

COMS 2203

7 December 2018

# Program Usage

Compile main.cc and run the program by running the command “g++ \*.cc && a.out” from within the lab6 folder. When the program runs, it will produce a command window at the bottom of the screen that begins with “Command: “ and the active keyboard cursor. A prompt at the top of the screen asks the user to type 0 to enter keyboard mode, or type 1 to enter mouse mode. If 0 is typed, the game will be controlled using the keyboard, and if 1 is typed, the game will be controlled using the mouse. Typing 1 will bring up a blank screen, and after a mouse button is clicked, it will get stuck in an infinite loop playing random play choices against each other and displaying an error message for invalid input. To escape from this, press CTRL + C. Typing 0 and pressing enter twice will bring up the keyboard-controlled version of the game, which works more than the mouse version. A prompt is shown at the top of the screen, explaining the valid inputs. Type “r” to play rock, “p” to play paper, “s” to play scissors, or “q” to quit the game. At any point in the game, if any other input is typed, such as “lizard” or “spock,” this error message with valid inputs will be shown again. The game will loop until “q” is typed or a kill signal is typed. If the player plays the same thing the computer plays, a tie message such as “It’s a tie!” is shown and the tie counter is incremented by one. If the player plays a winning choice against the computer, a win message such as “You win!” is shown and the player win counter is incremented by one. If the player plays a losing choice against the computer, a lose message such as “You lose!” is shown and the computer win counter is incremented by one.

# Scoring

|  |  |  |
| --- | --- | --- |
| Program proposal | 10 | A full program proposal was submitted and approved. |
| Program compiles | 10 | The program compiles and runs. |
| Use of multiple windows | 10 | * A command window is used at the bottom of the page for user input of choice and to choose game mode * A window is used for showing outcome messages * Separate windows are used for showing player and computer selections * Separate windows are used for showing number of player wins, computer wins, and ties |
| Commenting and formatting | 10 | Enough commenting is used to explain how the program works, and formatting is used to make the program readable |
| Additional use of a vector beyond the preceding project | 10 | Separate vectors are used for the win, loss, and tie messages, and the randomly selected messages are chosen by generating a random number and selecting that element of the vector. |
| Additional classes used beyond the previous project (each) | 10\*2 = 20 | Two additional classes, process() and displayMessage(), are implemented.   * process() is used for processing the output of the game (win, loss or tie) based on the choices of the player and the computer * displayMessage() is used for displaying a randomly selected |
| Any use of the mouse | 20 | Although mouse mode gets stuck in an infinite loop, the mouse can be used to get past the blank screen, and the code for reading mouse presses is implemented. This reflects research on mouse usage in NCURSES, an attempt at implementing it, and basic understanding of how mouse events are handled. “Any use of the mouse” does not require that the mouse part of the program works correctly, but that any use of the mouse receives 20 points. Since the mouse can be used to get past the blank screen, the mouse is used. |
| Report on project | 20 | A thorough report is made describing how to use the program. It includes proposed scoring and arguments for why points should be awarded in different areas. |
| Degree to which the program implements the proposal | 15 | The program reflects the proposal because it is a version of the classic game, rock, paper, scissors, implemented in C++ using NCURSES windows. The computer’s selection was designed to use randomly generated integers between 0 and 4, not inclusive, just as proposed. The mouse mode suggested in the proposal does not work correctly, and the scoreboard feature wasn’t used exactly as it was suggested in the proposal. |
| **Total:** | 125 |  |

# Appendix