For my final project, I decided to implement a Sudoku game that can also solve any board the user enters. The user can select between 3 different sudoku board difficulties (which are hard coded into the program), or the user can enter their own custom board that the program will solve. If the easy, medium, or hard boards are selected, then the user attempts to solve the board; after three wrong guesses, the game ends. If the user successfully completes the board, then a winning message is displayed.

This project is related to the class topics of logic and sets. The program uses a method called iterative backtracking to solve each board. Starting at the first empty tile, it iterates from 1 to 9 until it finds a valid number that can be placed in the current empty tile. It determines if a number is valid by looking at the numbers in the current tile’s row, column, and subgrid (each of which could be considered a set). If the number it is currently trying to place in the empty tile already exists in one of these sets, then it iterates to the next number and tries again. If the program runs out of options for the current tile, then it backtracks to the previously solved tile and finds a new solution for that tile (and then continues to solve the rest of the empty tiles). Thus, the program logically solves each sudoku board by comparing possible values to the sets of numbers that are already taken (and backtracking when necessary).

The program also uses logic in several other aspects of the game. When using one of the pre-set sudoku boards (either easy, medium, or hard difficulty), the program logically determines if a user’s guess is correct by comparing the user’s guess to the solution board. The program also gets the user’s mouse position every time it clicks somewhere within the game. If the user clicks outside of the grid, the program realizes this and does not let the user enter input. Additionally, whenever buttons appear (on the main menu and after a game has ended or a board has been solved), the program continually gets the user’s mouse position. The program logically determines when the mouse is hovering over a button and changes the color of the button to indicate this (the change in color also helps the user realize that it is a button).