Matthew Daniel

CS 470

February 3, 2017

Sub Project 1a Write-Up

**Summary:**

I wrote a sliding puzzle program in the programming language “Swift.” I began by making the puzzle playable by a user via the command line. Next, I created a small application that allows a user to play multiple games, clicking on tiles to “slide” them around. I also added a feature that allows games to be played on 4x4 and 5x5 boards.

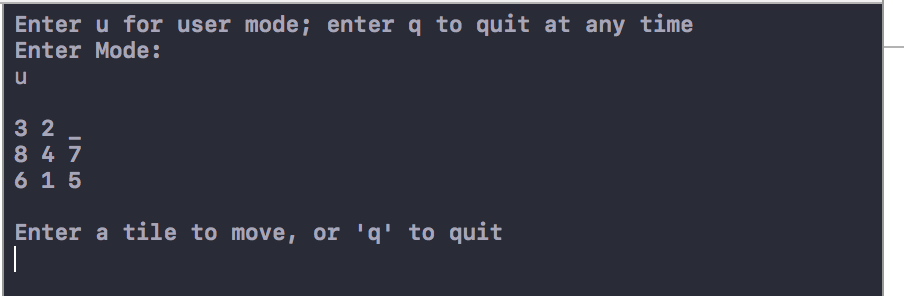


Figure : Beginning of a command line game

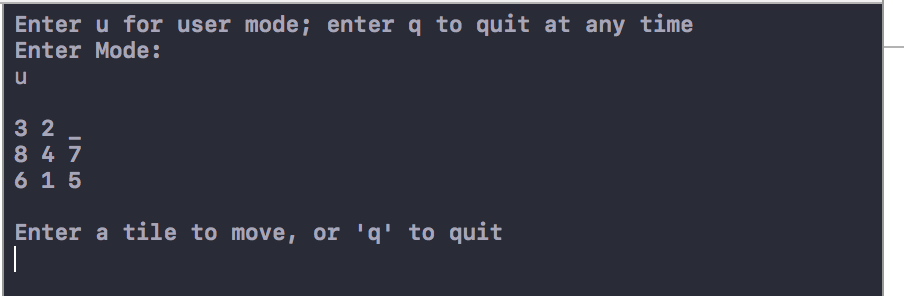


Figure : First move of a command line game

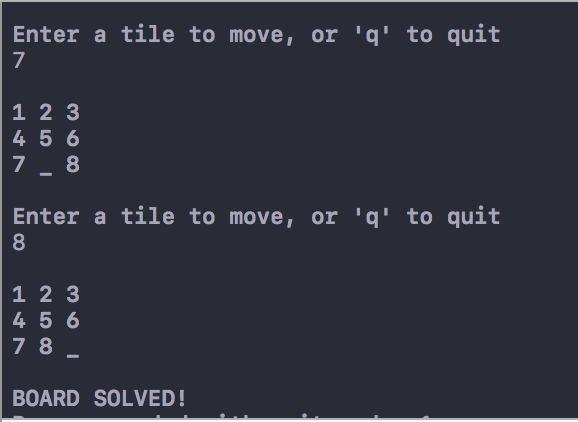
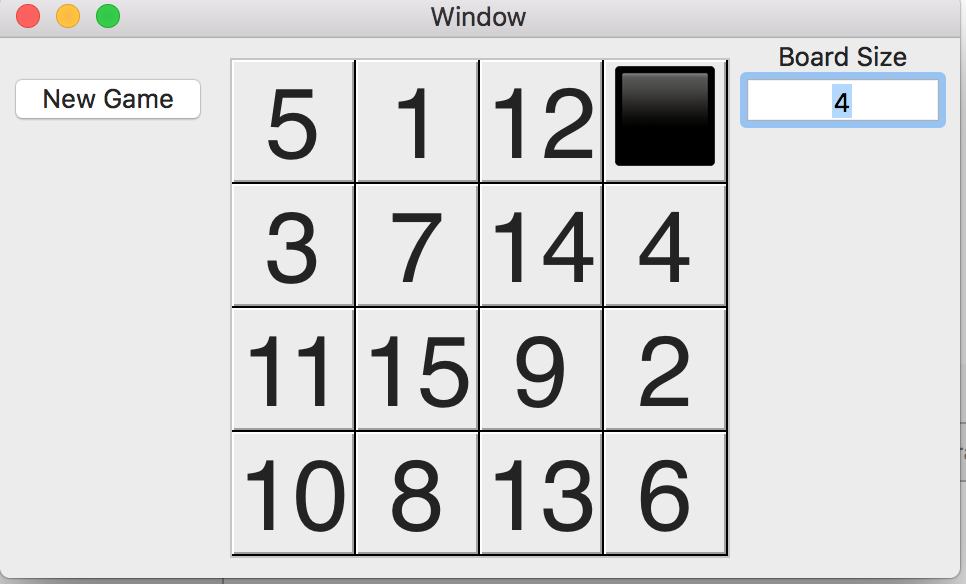
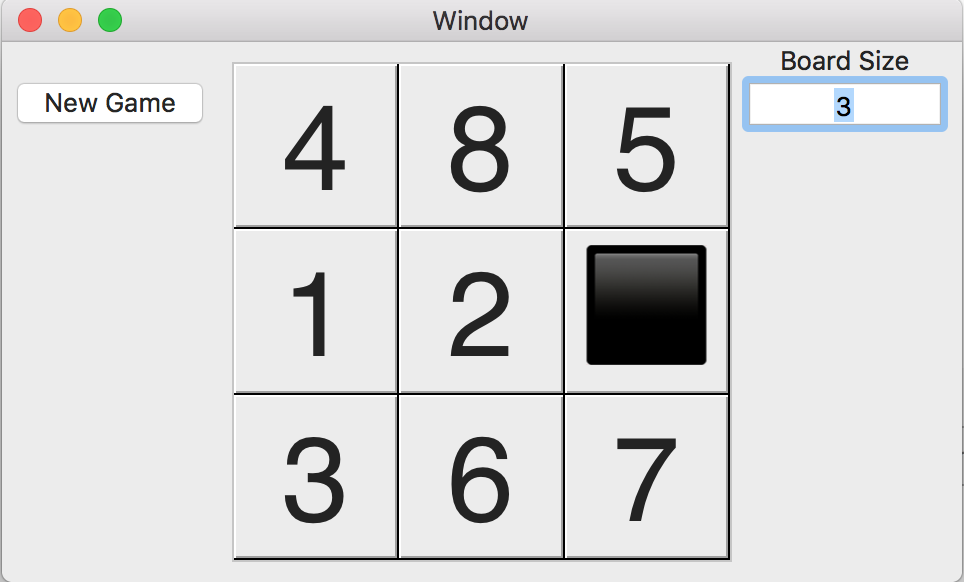


Figure : Final two moves



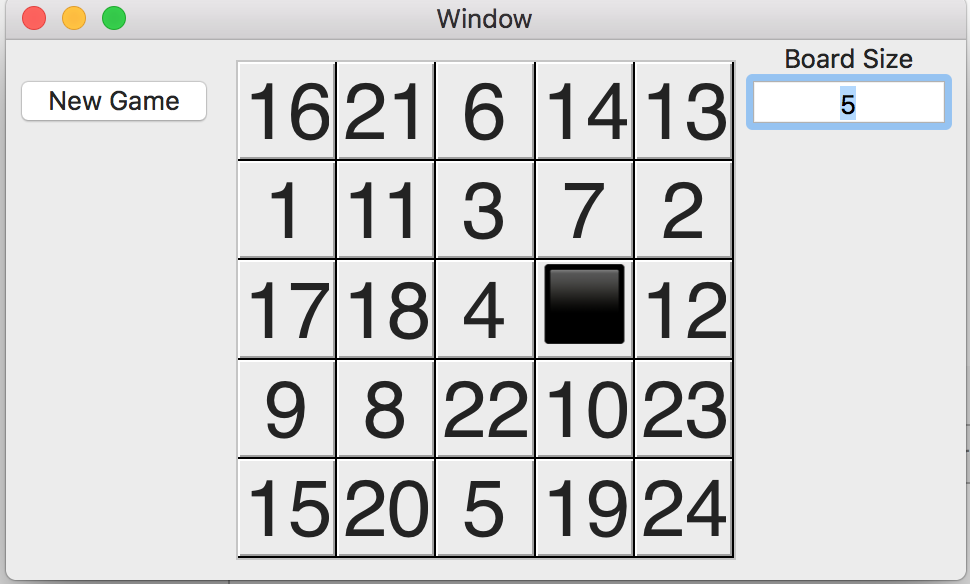


Figure : 3x3, 4x4, 5x5 Initial boards

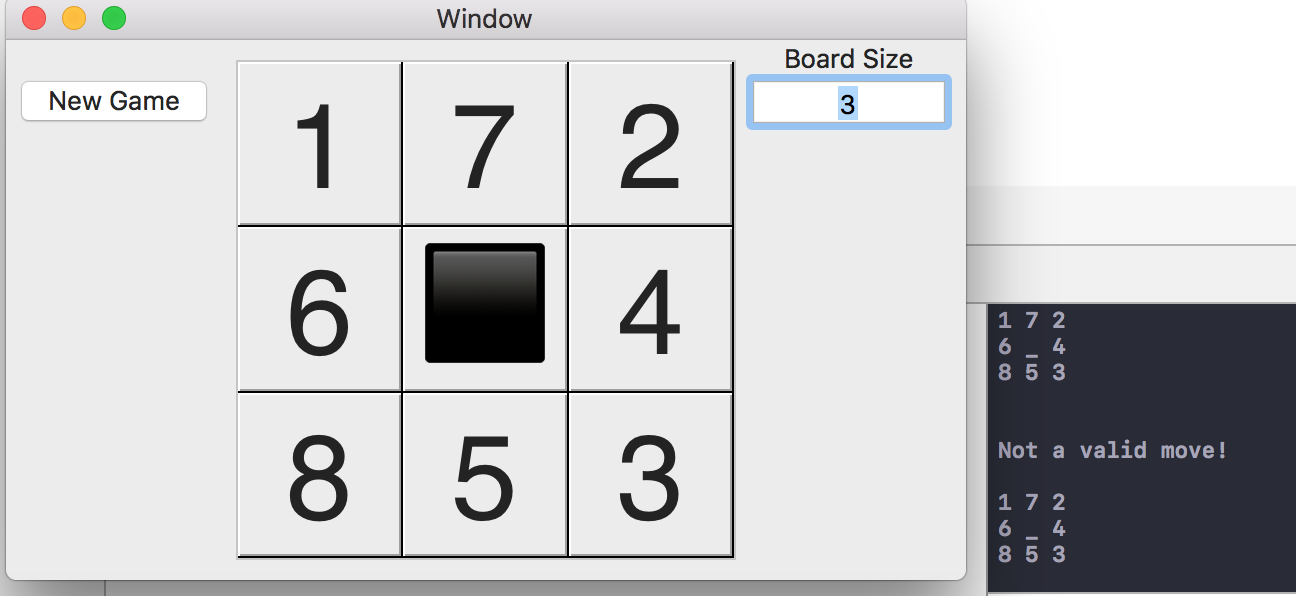


Figure : Gameplay. Note the simultaneous command line output

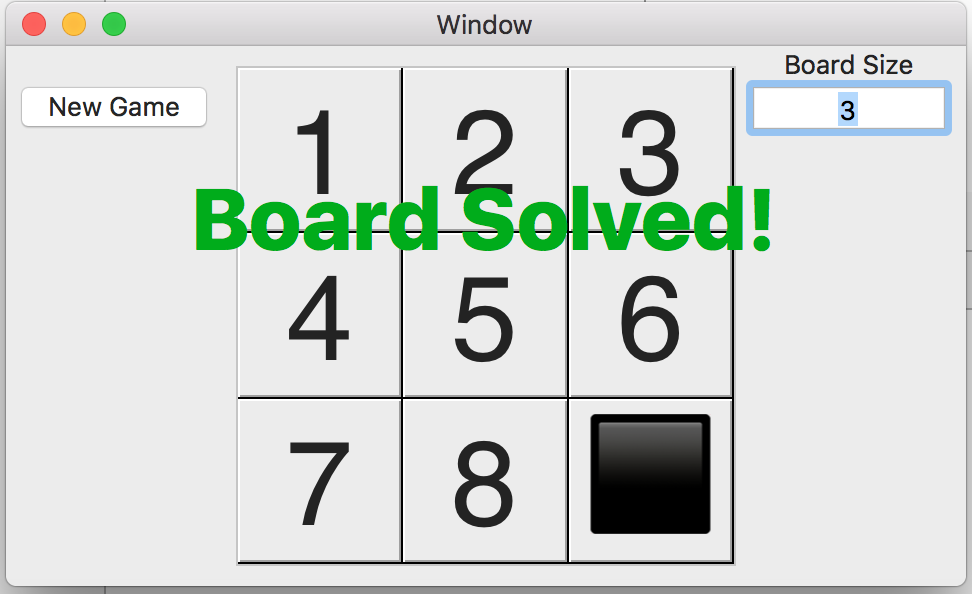


Figure : Solved board

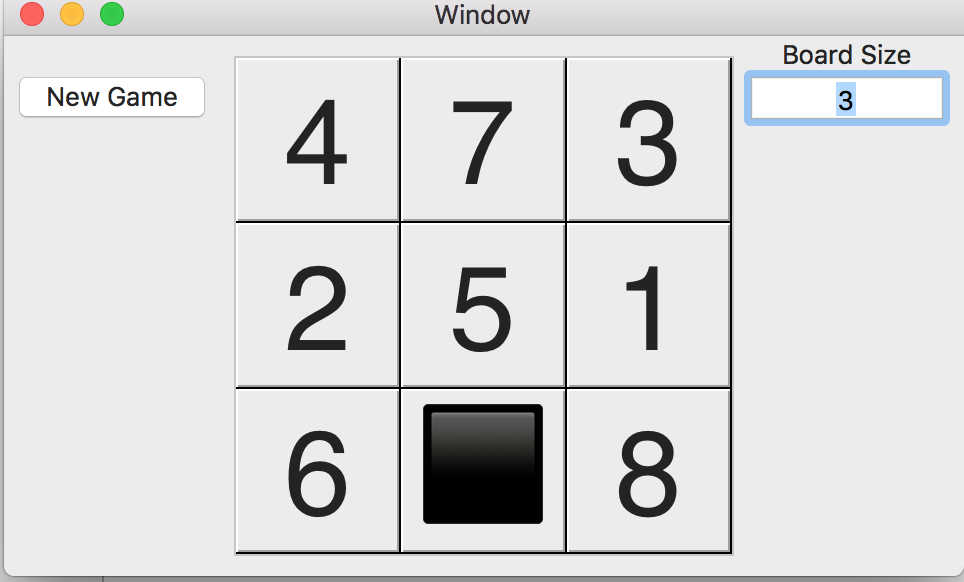


Figure : Start of New Game

Matthew Daniel

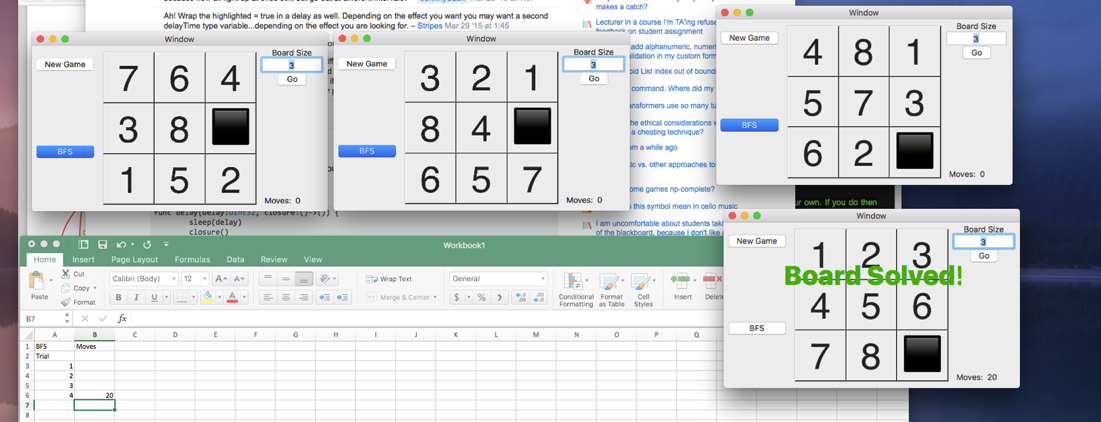
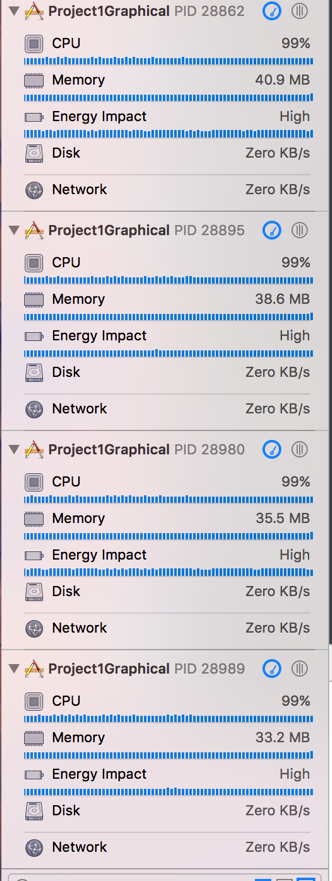
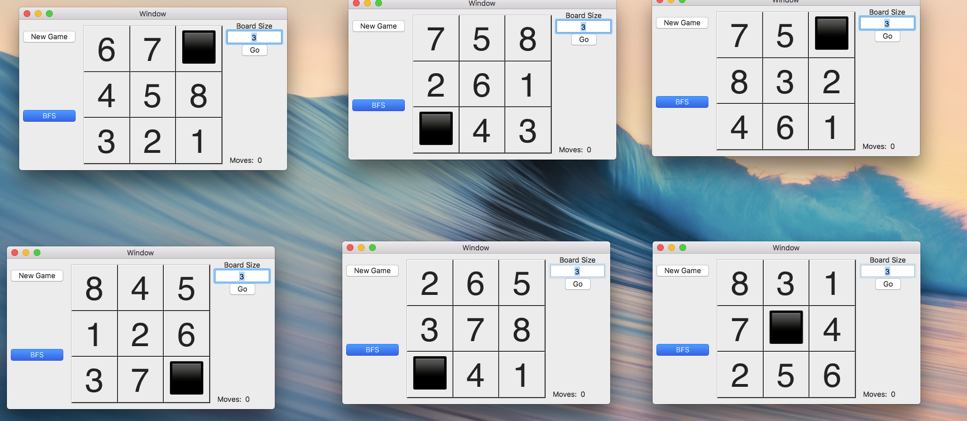
CS 470

February 10, 2017

Sub Project 1b Write-Up

**Summary:**

The breadth-first search solution to the sliding 8-puzzle found a solution in 20.77 moves on average, searching through an average of 72364 boards. Full results can be found on the attached spread sheet. I used a closed list, and did not check the moves as I put them on the open list. The sequence of moves used to solve the board by the BFS prints to the console, as well as is shown to the player via an animation (though the animation sometimes skips displaying moves). The function I used to represent boards in the closed list only works for solving 3x3 boards at the moment, though I may extend it to work for the 4x4 and 5x5 boards next week.



Screenshots of the application running and boards being solved.