Justin Grubbs Final Project Design Document

Overview:

My project will be a game of sudoku. I will likely be using sudoku.com as a general layout guide for my project. I chose this project as it is something I made in Python last year; however, it was purely text based.

I intend on having four difficulties, with each having a folder containing different puzzles and their respective keys. Depending on the difficulty chosen, it could randomly select a puzzle in the respective difficulty folder.

Example puzzle:

000659000400300000307800000210000007000001300000090008005000000006008090030000560

Example key:

182659734469317285357842619218463957594781326673295148945126873726538491831974562

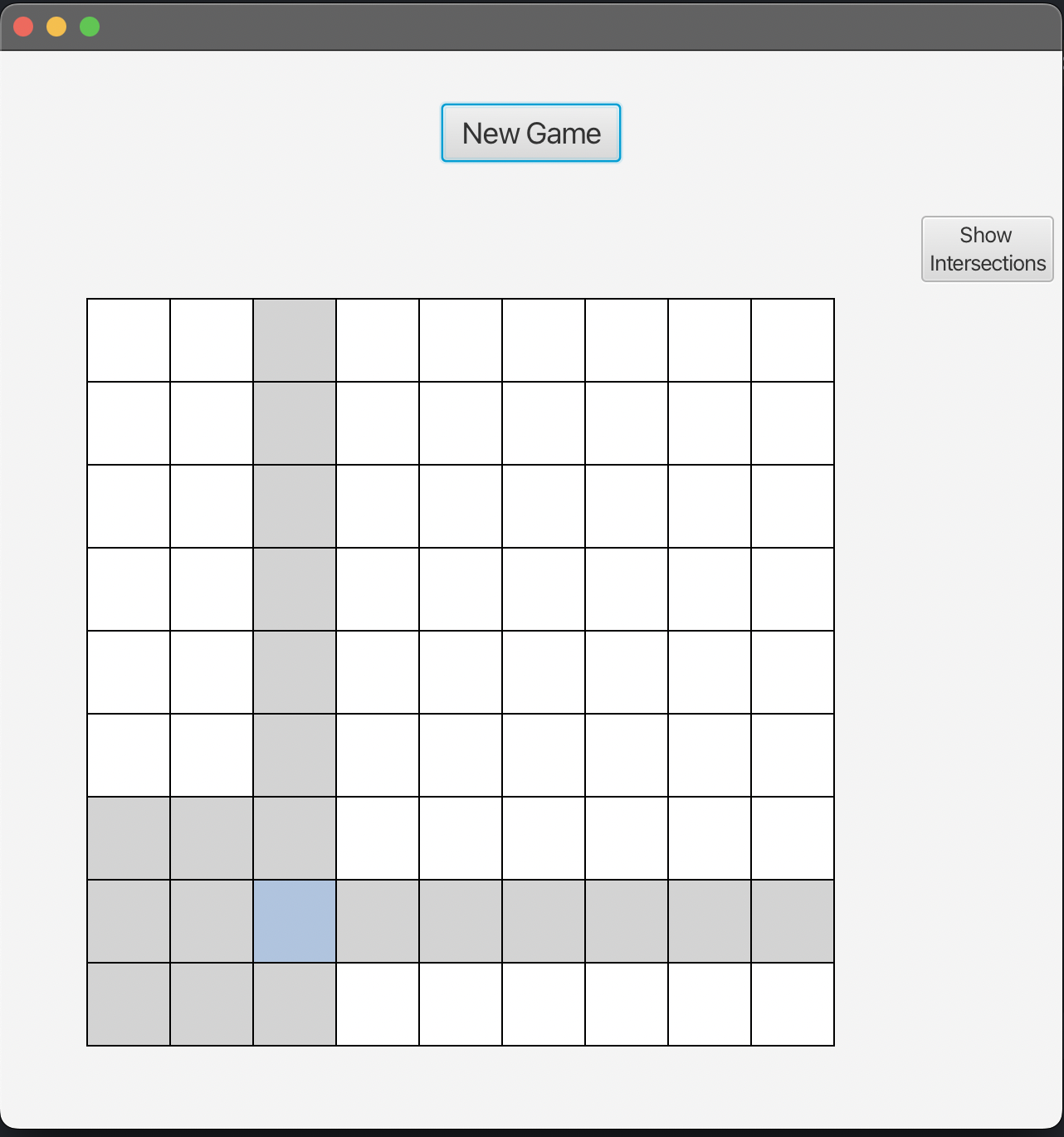
I could take these keys and turn them into maps. The former would represent whatever the current grid is. To check if a player has inputted a correct answer, I could just check it against the answer key. I could either manually get a few of these puzzles so I can show the functionality of the game, or I could make a puzzle generator.

Description of Classes:

* Sudoku class
  + Three or four HashMaps
    - AnswerKey
    - CurrentKey
    - IntersectionKey
    - CurrentCommonNumbers (when clicking on a number in the grid, this will be used to highlight all cells with that same number)
  + getIntersectionMap()
  + textToHashMaps()
  + getCommonNumberMap()
* Difficulty enum?
  + Contains the four difficulties
* Controller
  + reset()
  + toggleHighlightingIntersections()
    - Toggles the grey highlights in the screenshot below
  + toggleHighlightingSimilarNums()
  + toggleShowIncorrectInputs()
    - Can be disabled to make the game more like a paper game, doesn’t tell you whether the number you inputted was correct or incorrect until the end
  + drawGrid()
  + initialize()
  + gameSolved()
* Puzzle generator?
* Possibly use a view to add the numbers to the grid

General Layout:

Will have more buttons and the layout will be cleaner once the game is functional



Timeline:

I have already made the skeleton for the game, but I have not begun to add the game itself.

November 22nd - 26th:

* Be able to see the current grid with all respective numbers on the screen

November 26th - 28th:

* Be able to interact with the grid, adding and erasing numbers

November 28th - 29th:

* Add the option to highlight similar numbers
* Add the option to select different difficulties

November 29th – December 1st:

* Clean up the GUI

December 2nd - 7th:

* If needed, use this time to clean up and optimize the game
* If the project is fully done by this point, may consider creating a grid generator to create **solvable** puzzles
  + Create a valid, completely full grid
  + One by one, remove a number
    - If that number can be found from the rest of the grid, it is a valid removal
  + Depending on how many numbers removed, and the complexity of moves required to solve, assign a difficulty to the puzzle, and put it and the answer key in its respective folder
* I plan on doing this last so that if I encounter unexpected troubles with it, I can drop it while still having a functional, presentable project

December 7th:

* Create the presentation for the project