Noor Aljaber
15-112: Principles of Programming and Computer Science
Term Project Proposal
November 10th, 2019

Sudoku Puzzle

Project Description and Libraries to Be Used:

A Sudoku Puzzle consists of a grid of 81 squares, divided into nine blocks, where each block contains nine squares. To solve the puzzle, each block has to contain numbers from 1-9. This number has to appear only once in each row and column. Each puzzle has only one correct solution. To implement the sudoku puzzle in python, the following libraries will be used: 1) tkinter: to create the interface. 2) Time: to create a timer. 3) Random: to randomly generate numbers on the board. 4) Pygame: for sounds and other purposes. This is not final, other libraries like math might be used if necessary.

2	4	6	8	5	7	9	1	3
1	8	9	6	4	3	2	7	5
5	7	3	2	9	1	4	8	6
4	1	8	3	2	9	5	6	7
6	3	7	4	8	5	1	2	9
9	5	2	1	7	6	3	4	8
7	6	4	5	3	2	8	9	1
3	2	1	9	6	8	7	5	4
8	9	5	7	1	4	6	3	2

Figure 1: Sudoku. Numbers should be arranged in a way that they only appear once in each block, row and column.

User Interface Description and Features:

In the homepage of the game, the user will have three options, either to start, view scores or watch a tutorial to learn the rules of sudoku.

1) If the user chooses to start the game, first he/she will select a level; easy intermediate or hard.

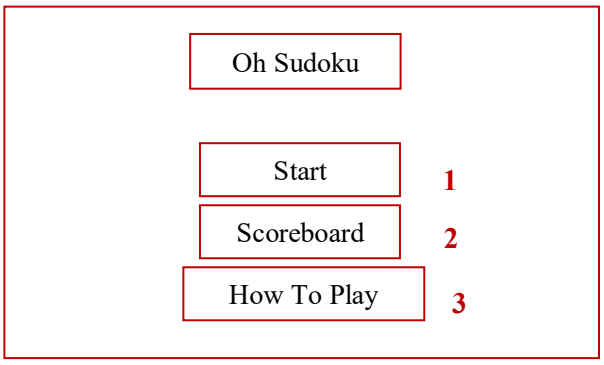
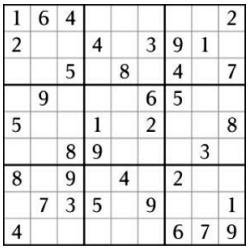
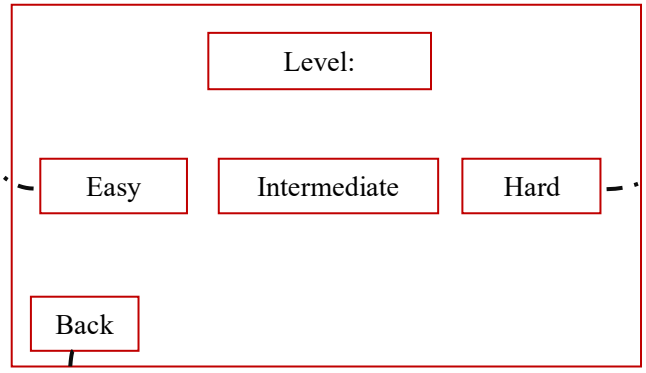


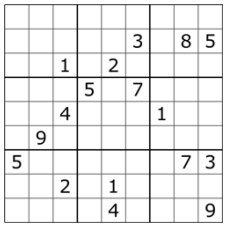
Figure 2: Homepage.



It will be easy for the user to guess the missing numbers as they are few.

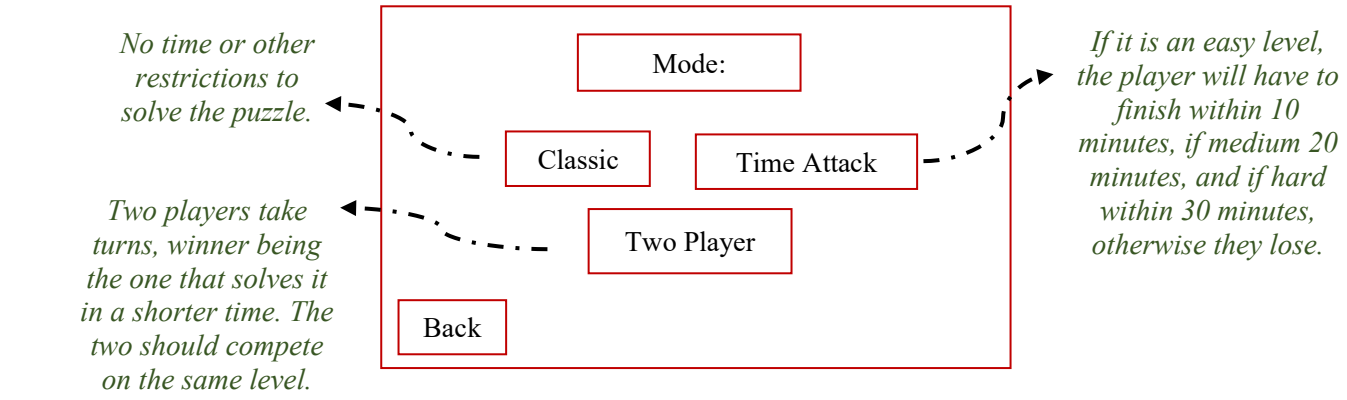


This button will appear on all windows after homepage, for the user to go back to the previous window.

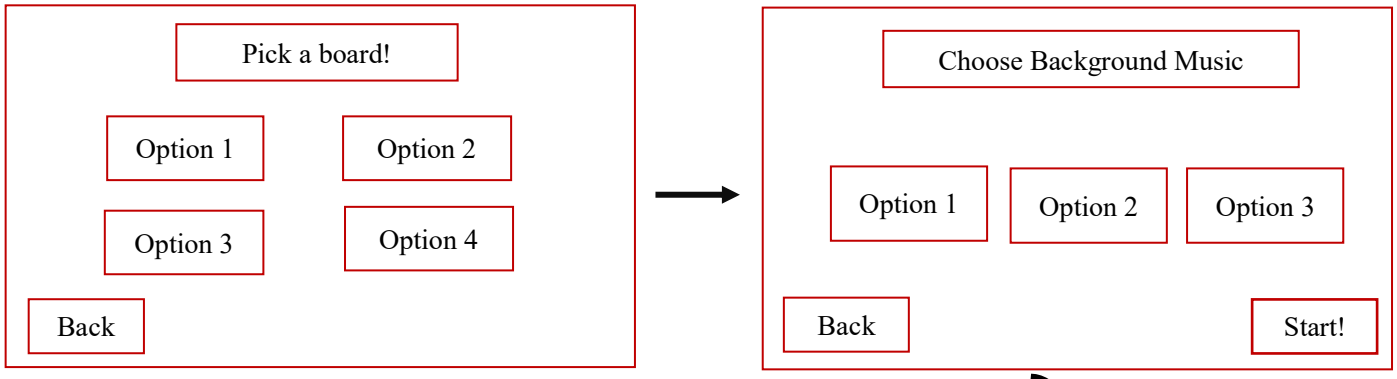


Less numbers are given so the player will have to guess most of them. The numbers provided are distributed in a way that it is hard to predict unknown ones.

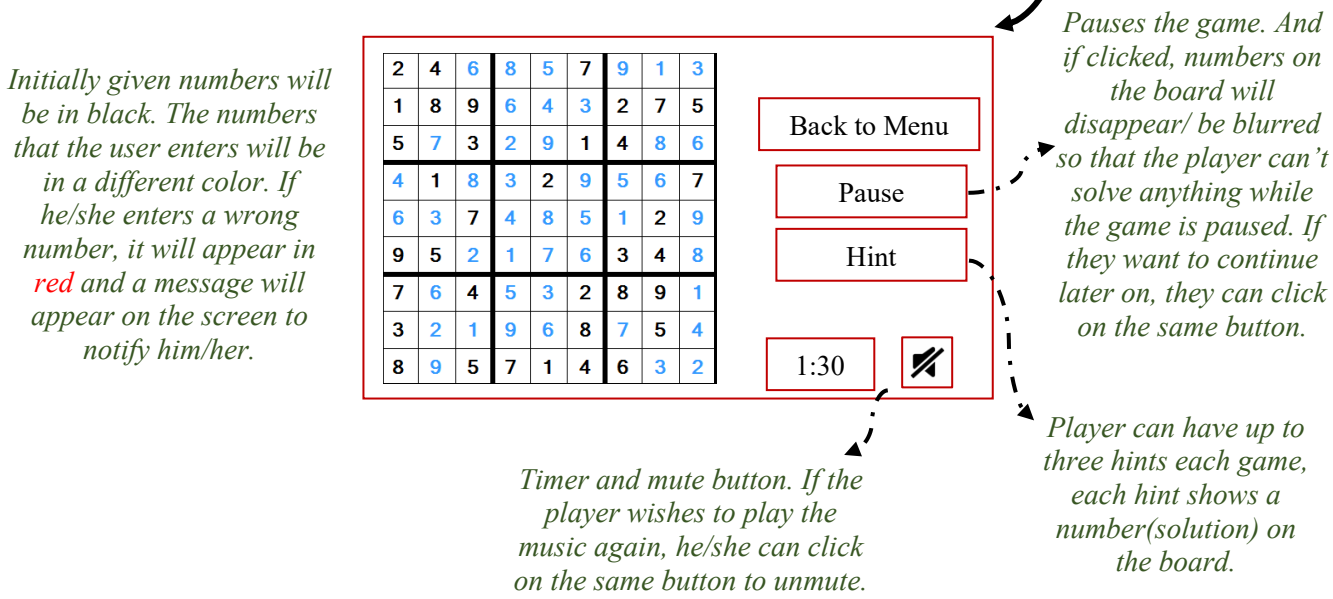
The user will then choose one of the following modes:



After, the user will enter his/her name (two names if there is two players) and will have the choose the board color/design, and background music.



Game starts after choosing the music.



* If the user wins a classic game, a window will show the score (time taken to solve) and lead him/her to homepage.
* If it is a time attack mode and the user loses, a window will indicate that when time is up. Otherwise if they win the window will show their score.
* If it is a two player mode, they take turns and after the second player finishes, a window will indicate the winner.

2) If the user chooses to view the Scoreboard, a list of their, and other users if any, will show scores in a descending order, according to levels and modes.

3) If the user chooses How to Play, he/she will watch an introductory video.

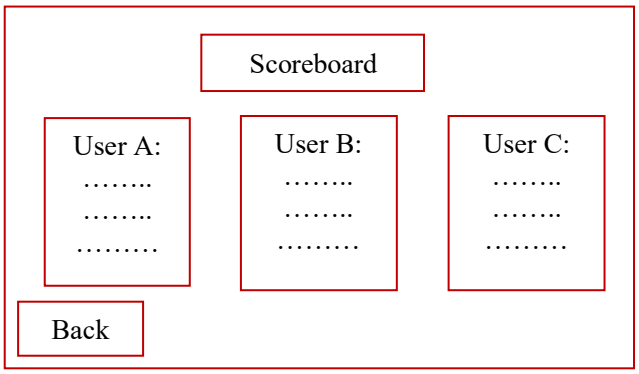


Figure 3: Scoreboard.

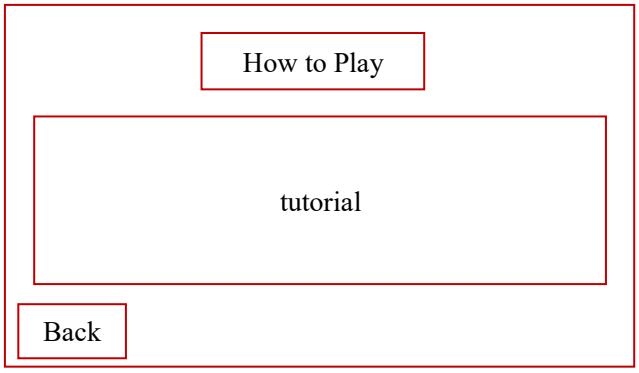


Figure 4: How to Play.

For the first checkpoint, everything described above will be implemented except that the game will only have a single option for background and music. Scoreboard will also be inactive, that is it will not keep scores for users. These are to be added for the final submission, and the project will be complete!