# Brain Storm

Wednesday, February 3, 2021 8:51 PM

Constraints:

- Design a program that follows the rules of Sudoku and generate the solution
- Have each grid randomly created
- Create a visual interface that allows user interaction
- Allows either the user to solve the puzzle or allow the program to complete the puzzle in a efficient manner
- Create the program 100% in python to learn the fundamentals of the language
- Create a difficulty menu that determines how many empty spaces will be in the grid
- Set Deadline is February 22 for the grid creator and algorithm implementation. More time will be allow for creating visual design and player input.

Overview:

Sudoku is a puzzle game played on a 9x9 "grid" broken down by 9 "squares" made by 3x3 spaces. The objective is to fill the all "spaces" in the grid with number from 1-9 without repeating any numbers in the same row, column or square.

Algorithm:

Will be using the backtracking algorithm to find to the solution of each randomly generated grid. This algorithm explores one branch of a possible solution by placing a "1" in the first possible space and then checking if it does not interfere with any of the given rules. If there are no violations then the algorithm goes to the next available space and fills in a "1" and the process continues. If a violation occurs then the number is incremented by one and checked again. In the event that all 9 digits are not allowed, then the algorithm works backwards to the previous space. The previous place will be incremented by one. The algorithm will repeat until call spaces are filled and placed correctly.

Ideas:

- Create a 2d array to represent the grid
- Divide array in to 9 groups representing the squares
- Make random spaces filled given the difficulty
- Create the rules of Sudoku as a condition method
- Implement the backtracking algorithm
- Test if the solution works by text format
- Create away for a user to input numbers and check it with algorithm
- Create a visual format of the puzzle