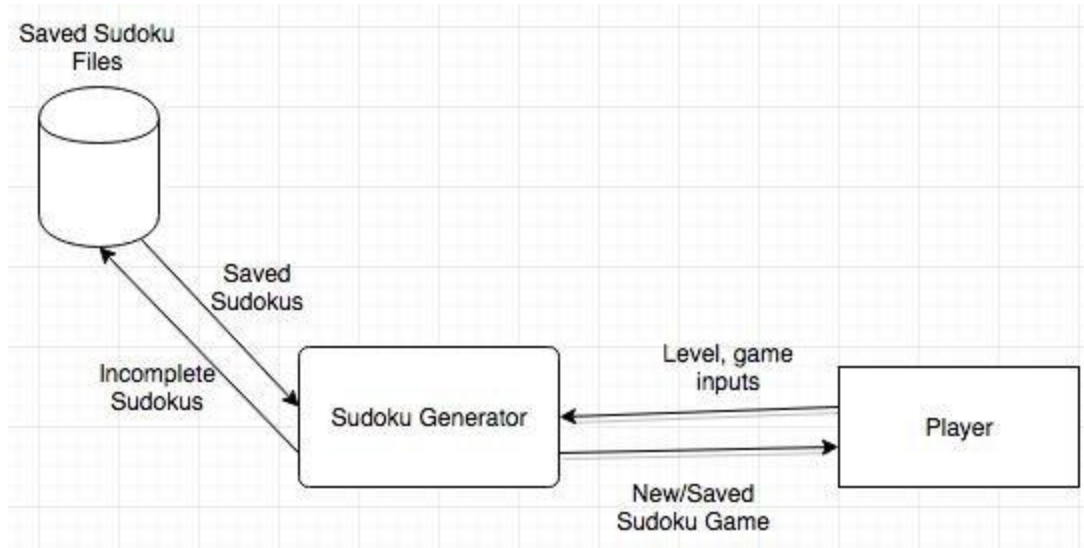
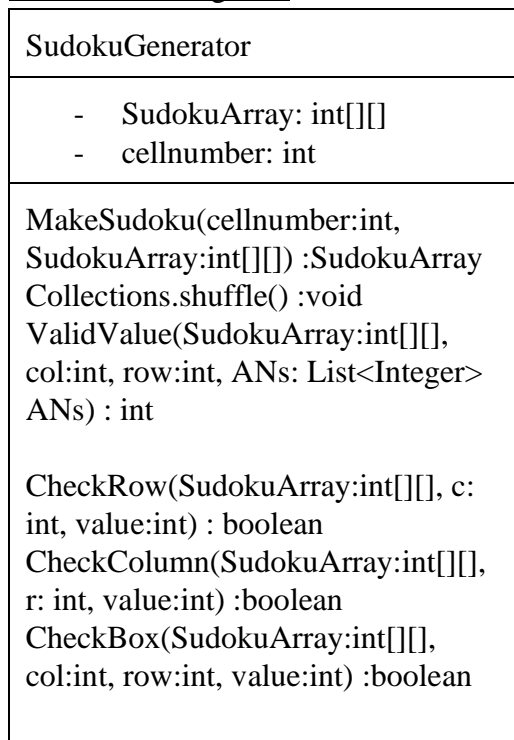


Design

Data Flow Diagram



UML Class Diagrams



MakeGameBoard
- gameArray:int[][] + fullSudoku:int[][]
EasyBoard() :gameArray MediumBoard() :gameArray HardBoard() : gameArray Random(): int



Check
checker() : boolean SudokuFrame():void Integer.parseInt() : int

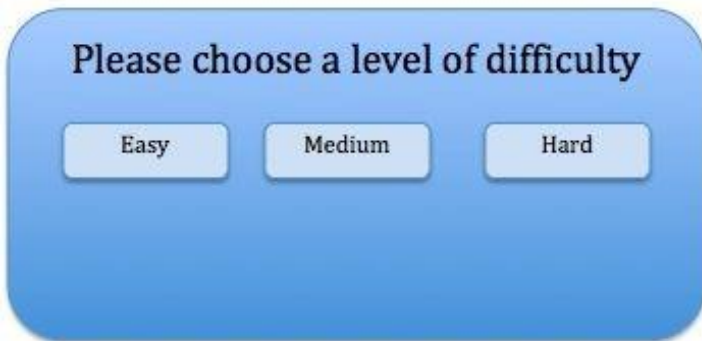
Prototype

Start page



Itamar Yacoby 2/22/2017 1:28 PM
Comment [1]: This window will pop up once the application is run. The player will be given the opportunity to start a new Sudoku puzzle or load a previously saved one.

Choosing a level window:



Itamar Yacoby 2/22/2017 1:27 PM
Comment [2]: This window is reached once the player has clicked "play!" in the previous window. Once a player clicks on a difficulty they will be immediately transferred to the game.



Pseudocode for SudokuGenerator:

1. Initialize [9,9] 2D array of type int named SudokuArray, initialize cell count named cellnumber of type int
2. Generate arraylist ANs of type integer and populate it with numbers 1-9 using a for loop
3. Shuffle the arraylist randomly
4. Get the column and row value of the cell using the cell count.
5. Begin while (ANs.size()>0) loop
 - a. Gets a value from the validvalue() algorithm
 - b. If it is valid store it in the current cell and if not return null
 - c. Make a new array and make it equal to the recursive call of the method with an increase of one cell count.
 - d. If this method doesn't return null, return the value in the array, else set the current cell equal to 0

ValidValue Algorithm

1. (while ANs.size()>0)
 - a. Int value = the value of the zeroth position in the arraylist
 - b. Remove this position from the arraylist
 - c. If CheckRow and CheckColumn and CheckBox equal true return this value
2. Return -1

Check Row Algorithm

1. Use for loop to iterate through the current row.
 - a. If any existing value equals the randomly chosen value return false
2. Else return true

Check Column Algorithm

1. Use for loop to iterate through the current column.
 - a. If any existing value equals the randomly chosen value return false
2. Else return true

Check Box Algorithm

1. Find the lower boundary row and column values for the box by dividing the current row and column values by 3 (no remainder) and multiplying the result by 3.
2. Iterate through the box using a nested for loop counting up thrice.
 - a. If any existing value equals the randomly chosen value return false
2. Else return true

Test Plan:

Test Number	Action Test	Expected Result
1	Run Program	Welcome window pops up with the correct title, the option

		to enter a name, and click “play” or “how to play”
2	Click on “How to Play” button	A separate window should pop up containing the rules of the game.
3	Enter name	The name appears in the text field
4	Click play	A separate window pops up with the levels of difficulty
5	Click each level of difficulty (easy, medium and hard) on separate occasions.	Check that the new game window that pops up has a Sudoku with the corresponding amount of boxes already filled out as per the level chosen
6	Click on empty box and add number	A number should be added in that box and should stay there.
7	After filling up half of a Sudoku, click on “Check Progress” button	Boxes that have been filled up by the player should have their background turn green if the box is filled correctly or red if it is filled incorrectly
8	Finish a Sudoku and click “Done!” button	<p>A window should pop up with the option to quit or start a new sudoku.</p> <p>If it has been filled out wrong, a window should pop up saying “you have some mistakes”.</p> <p>The Sudoku board should display those boxes that are filled out wrong in red</p>
9	While solving a Sudoku, click on the “save for later” button	<p>A window should pop up saying “saved! You may only save one sudoku at a time.” and the option to quit</p> <p>The sudoku should be saved in a text file in a prescribed workspace</p>
10	Quit the program and run it again, click on the “Open Saved Sudoku” button	Game window should pop up with the progress from the saved sudoku.