

# **Software Requirement Specifications**

## **Sudoku Quest**

### **INTRODUCTION**

#### ***1. Purpose:***

The Purpose of this Document is to present a brief Description of the Project and an Analysis of the Algorithms on which this Project is based on.

#### ***2. Scope of this Project:***

- Generate a Sudoku of User-Specified Toughness.
- Solve a Sudoku input by the User.

### 3. Overview:

#### a) Generating a Sudoku Grid:

An Incomplete Sudoku Grid is Generated based on the level of toughness chosen by the user such that it has a unique solution.

#### b) Solving a Sudoku:

The Sudoku grid input by the user is solved by using Backtracking Algorithm along with Brute force technique. If the Sudoku Grid is unsolvable, an error message is displayed. If the Grid has Multiple Solutions, the first solution encountered by the program is displayed and asks if another solution is required.

### 4. Resources:

- Comments are included to make the program more interesting.

## 5. *Web references*

WIKIPEDIA: To Understand the Backtracking Algorithm.

<https://developer.gnome.org/gtk3/stable/gtk-getting-started.html>

<http://www.cplusplus.com/forum/unices/40030/>

<https://developer.gnome.org/gtk3/stable/GtkWindow.html>

<http://www.yolinux.com/TUTORIALS/GTK+ProgrammingTips.html>

<https://www.youtube.com/watch?v=vcU8AE8mdNs>

<https://developer.gnome.org/gtk3/stable/GtkEntry.html>

<https://developer.gnome.org/gtk3/stable/GtkLabel.html>

<https://developer.gnome.org/gtk3/stable/GtkWidget.html>

## 6. *Software Used:*

Code Blocks

GTK+ 3.0 : To Create a GUI for the program.

# GTK +3.0 INSTALLATION

GTK +3.0 can be downloaded from

<http://www.gtk.org/download/>

Installation of GTK can be proceeded by following the instructions given in

[http://www.gtk.org/download/win\\_tutorial.php](http://www.gtk.org/download/win_tutorial.php)

GTK+3 is known to work on Windows XP, Vista, 7 and 8 at this date.

## ALGORITHM

### To Solve the Sudoku :

The Sudoku is solved using Backtracking Algorithm along with Bruteforce Technique.

- The Program sequentially visits empty cells and places the digit “1” in the cell.
- It Checks if the digit is allowed in that cell.

- If there are no Violations (checking row, column, and box constraints) then the algorithm advances to the next cell, and places a "1" in that cell.
- If the constraints are violated, the digit "2" is placed in the cell and so on.
- In a cell, if none of the digits 1-9 are allowed, the program leaves the cell blank and moves back to the previous cell and increments the digit by 1.
- The Algorithm is repeated until a valid solution is found.
- Once a solution is obtained, it goes back to previous cell and increments the digit by 1 and repeats the algorithm to search for another solution.

## To Generate a Sudoku Grid:

- The first row of the Grid is filled randomly with the digits 1-9 and it is solved using the solver algorithm.
- The first solution encountered by the program is taken
- A List of all the 81 cell positions of the obtained Grid is Shuffled randomly.

- As long as the list is not empty, take the next position from the list and remove the number from the related cell.
- Test the Uniqueness with the Solver Algorithm.
- If the Grid has unique solution, go to step 2 and repeat.
- If the current Grid has more than one solution, undo the last removal (step 2), and continue step 2 with the next position from the list.
- The Difficulty level is based on the number of removals from the Grid.