

# COURSE PROJECT DOCUMENTATION

CS101

## SUDOKU AUTOSOLVER

Team Id:229

Kunal Telgote	14D070030
Nitesh Kaushal	14D070050
JeetkumarNamha	140070021
Anil Pawar	14D070051

## **Table Of Contents**

1. Introduction.....	3
2. Project description.....	4
3. Requirements.....	5
4. Functionality.....	6
5. Testing Strategy and Relevent Diagrams.....	6
6. Interfaces.....	10
7. Challenges And Their Solution.....	11
8. Future Work.....	12
9. References.....	12

# 1. INTRODUCTION:

Sudoku(normal) is a puzzle game in which a 9x9 grid with a few numbers given by the user and our program is expected to solve it by filling in all numbers from 1 to 9 in every row, column and certain 3x3 squares without repeatation . Because of the dimensions of the grid, no number is repeated along a row, Column or a 3x3 square.

## Different types of sudoku:

1. Normal Sudoku : Usual general sudoku.

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

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

## 2. Diagonal Sudoku :

Diagonal Sudoku is also played over a 9x9 grid divided to 3x3 sub grids or boxes. The objective is to fill a grid with digits so that each column, each row, and each of the nine boxes that compose the grid contains all of the digits from 1 to 9 and each digit appears once. Also the main diagonals of the grid contains all of the digits from 1 to 9 and each digit appears once.

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

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

### 3. Windows Sudoku :

Window Sudoku is also played over a 9x9 grid divided to 3x3 sub grids or boxes. The objective is to fill a grid with digits so that each column, each row, and each of the nine boxes that compose the grid contains all of the digits from 1 to 9 and each digit appears once. Also there are four additional boxes in the grid that contain all of the digits from 1 to 9 and each digit appears once

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

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

.

## 2.PROJECT DESCRIPTION:

### ➤ PROBLEM STATEMENT :

Goal 1:

User will give inputs in a 9x9 grid on canvas wherever he want and our program will check whether the given inputs are valid according to rules of sudoku puzzle.

Goal 2:

To solve a 9x9 sudoku puzzle whose inputs are given by the user and to show the solved output on the canvas.

### ➤ APPROACH :

In the program, the user is given a blank 9x9 grid. He can input certain values at any positions. The validity of the input values will be checked according to the rules of Sudoku and the grid will

be solved by the program.

### ➤ **STEP BY STEP DESCRIPTION :**

1. The user is asked to input the Sudoku.
2. The user inputs the Sudoku.
3. The program checks the given input, if the given input is invalid appropriate error message is displayed and program exits.
4. The solution is given by our program if it exists.

## **3.REQUIREMENTS :**

GNU-GCC compiler and Simplecpp interpreter must be installed on the user's computer.

## **4.FUNCTIONALITY :**

- 1.Grid:** The user is provided with a choice to solve a 9x9 Sudoku.

**2.Checker:** The user gets to check if the entry done by him is correct or not.

**3.Solver:** The user can input his own Sudoku problem and get it solved.

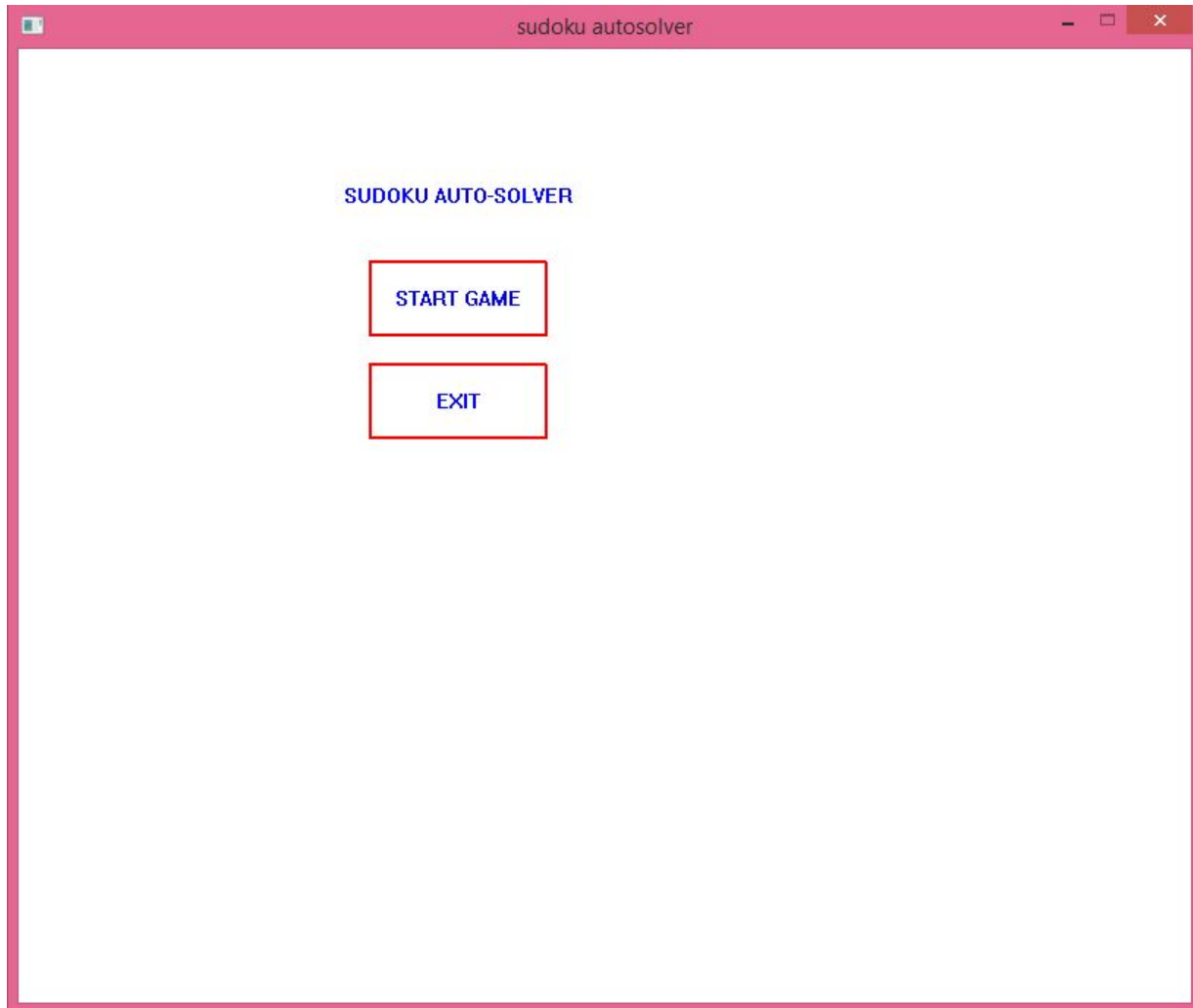
## **5. Testing Strategy and Relevent Diagrams :**

We simply test our program by giving any inputs of a given solved sudoku and check the output with the remaining grids of given sudoku.

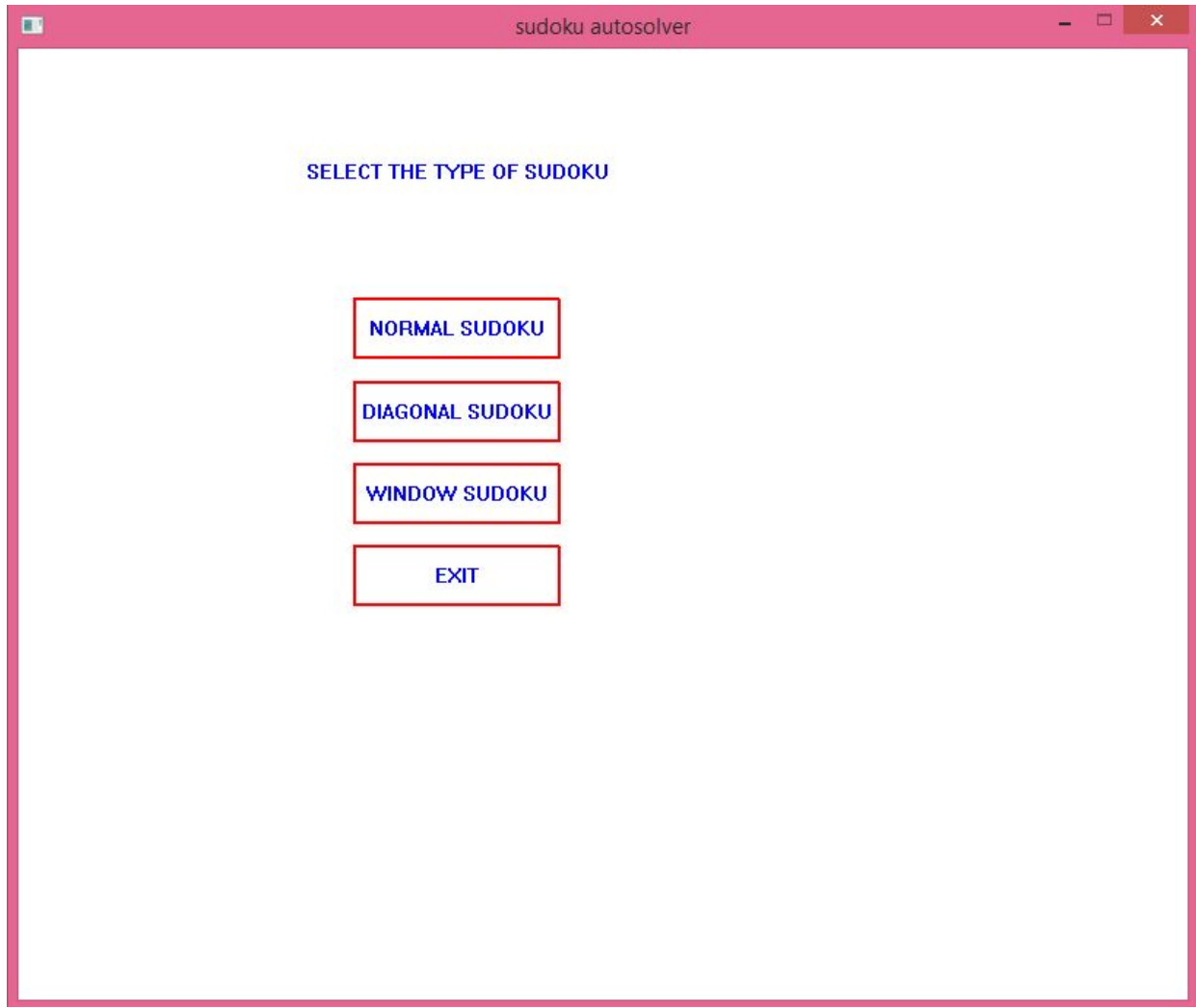


## Project Screenshots:

### Start Window



# Varient Of Sudoku



# Input Window For Diagonal Sudoku

sudoku autosolver

GIVE INPUTS

1								7
				2				
6								
				5				8
		3						
3								

DIAGONAL SUDOKU

SOLVE

1

2

3

4

5

6

7

8

9

INSTRUCTIONS

First click in one of the box in 9x9 grid where you want to give inputs  
Then click on one of 1 to 9 buttons shown below the 9X9 grid  
After giving inputs click on SOLVE button on right side of canvas  
Then solution is displayed  
Then Click once again anywhere on the Canvas to Exit

# Output Window For Diagonal Sudoku

sudoku autosolver

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

DIAGONAL SUDOKU

SOLVE

INSTRUCTIONS

First click in one of the box in 9x9 grid where you want to give inputs  
Then click on one of 1 to 9 buttons shown below the 9X9 grid  
After giving inputs click on SOLVE button on right side of canvas  
Then solution is displayed  
Then Click once again anywhere on the Canvas to Exit

# Input Screen For Window Sudoku

sudoku autosolver

GIVE INPUTS

2								
				9		3		
	6							
		4				7		
	2						5	

WINDOW SUDOKU

SOLVE

1

2

3

4

5

6

7

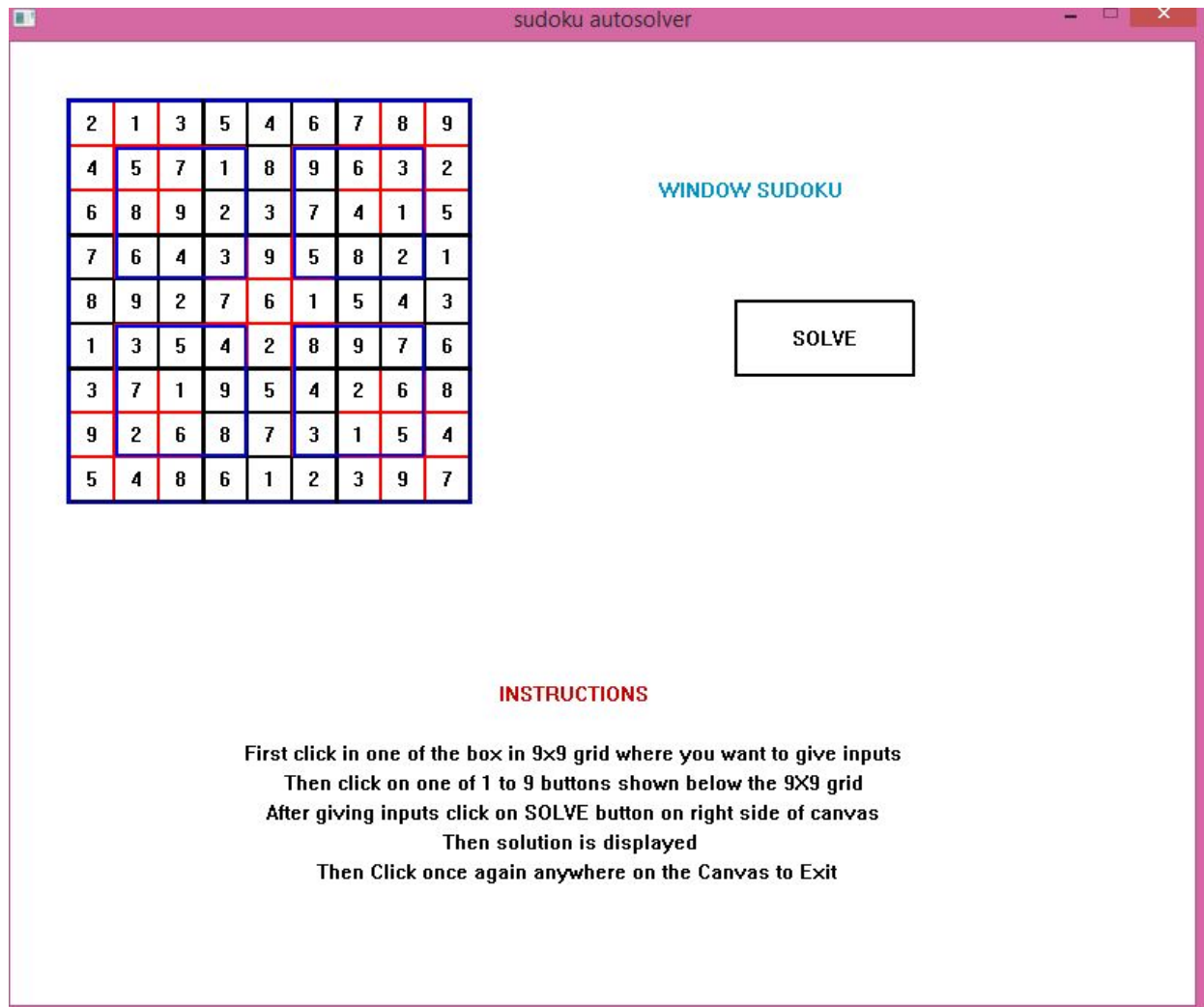
8

9

INSTRUCTIONS

First click in one of the box in 9x9 grid where you want to give inputs  
Then click on one of 1 to 9 buttons shown below the 9X9 grid  
After giving inputs click on SOLVE button on right side of canvas  
Then solution is displayed  
Then Click once again anywhere on the Canvas to Exit

# Output Screen For Window Sudoku



## 6. INTERFACES:

### ➤ USER INTERFACES:

The software currently uses a mouse and keyboard interface with graphics included

### ➤ SYSTEM INTERFACES:

The software opens a new window which serves as mode of interaction between user and system.

### ➤ HARDWARE INTERFACES:

The software currently uses a mouse and keyboard interface with graphics included.

### ➤ SOFTWARE INTERFACES:

We are using codeBlocks .Program will run on any version of UNIX (including Linux) and also on windows.

### ➤ COMMUNICATION INTERFACES:

This game is confined to only one device. Hence, there is no association with communication functions, including e-mail, web browser, network

server communications protocols, electronic forms, and so on.

## 7.Challenges And Their Solutions:

- Backtracking
- To connect the interface part with the algorithm part.

## Solutions:

❖ BACKTRACKING :

The logic behind the solving algorithm is :- Firstly in the (0,0) box the function checks whether the box has value zero or not by calling a function which checks valid inputs according to the rules of Sudoku . Then the function puts 0 in that box. Then it moves to the next adjacent box and checks whether the input is zero. If zero it then puts the smallest possible number in that box and moves to the next empty box. Now if the



initially inserted values are not matching with the final output then there will come one box in which no value can be inserted from 1 to 9. In that case , the function goes to the last box where changes have been made. Now it puts zero in it and then again puts next possible smallest number in it. Now it goes to the next empty box if still it doesn't work then it goes to the second last box where changes were made and there the inserted value is reset to next possible smallest number. In this way the process continues until the last (8,8) box comes. If not then no solution is possible. In this way a recursive function works which solves the Sudoku puzzle.

#### ❖ TO CONNECT THE INTERFACE PART WITH THE ALGORITHM PART :

We made a switch command in which the output array i.e. The solved Sudoku puzzle's values are copied and displayed on the canvas and also on the console window.

## 8.FUTURE WORKS :

- Implement it to solve more different type of sudoku like jigsaw sudoku, samurai sudoku, neighbor sudoku, etc.
- If one does not want to physically solve a sudoku puzzle, he/she can scan the puzzle by making use of image processing from which our program would take inputs and give output.

## 9.REFERENCES :

- <http://www.cse.iitb.ac.in/~cs101/project.html>
- [http://www.cse.iitb.ac.in/~cs101/Project/Manual\\_Code::Blocks\\_Simplecpp.pdf](http://www.cse.iitb.ac.in/~cs101/Project/Manual_Code::Blocks_Simplecpp.pdf)
- Past year projects
- WIKIPEDIA  
<http://en.wikipedia.org/wiki/Sudoku>

