

PROJECT REPORT

THE SUDOKU

GROUP MEMBERS OF THE PROJECT

N.SANDEEP	140100098
CH.VENKATA SAI KIRAN	140100083
REVATI NANAWARE	140260013

Contents

1) Rules of Sudoku.....	1
2) System Requirements.....	2
3) References.....	2
4) Algorithm for preparing the code.....	3
5) Brief Description of C++ coding.....	4
6) Working Details.....	8
7) Individual Contribution.....	9
8) Limitations.....	10

Rules for the game

A standard Sudoku puzzle consists of a grid of 9 blocks. Each block contains 9 boxes arranged in 3 rows and 3 columns.

The Basic Rules of Sudoku:

- There is only one valid solution to each Sudoku puzzle. The only way the puzzle can be considered solved correctly is when all 81 boxes contain numbers and the other Sudoku rules have been followed.**
- When you start a game of Sudoku, some blocks will be pre-filled for you. You cannot change these numbers in the course of the game.**
- Each column must contain all of the numbers 1 through 9 and no two numbers in the same column of a Sudoku puzzle can be the same.**
- Each row must contain all of the numbers 1 through 9 and no two numbers in the same row of a Sudoku puzzle can be the same.**
- Each block must contain all of the numbers 1 through 9 and no two numbers in the same block of a Sudoku puzzle can be the same.**

A valid Sudoku may look as follows

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

System Requirements-

C++, Code blocks integrated with SIMPLECPP

References

- 1) <http://en.wikipedia.org/wiki/Sudoku>
- 2) <http://en.wikipedia.org/wiki/Backtracking>

Algorithm for preparing the Code

We want to write a C++ code which will give the user a solvable Sudoku puzzle having a unique solution, checking his gameplay against the rules of conventional sudoku and finally checking the correctness of the solution.

For this to happen,

We first made use of SIMPLECPP to open a window consists of options like NEW GAME and QUIT and then wrote a code which will show the user options EASY,MEDIUM,HARD to select the level of difficulty of the Sudoku which he wants to play. Then we wrote the code such that user on choosing the desired level then he in the window he will see the numbers from 1 to 4, from which he needs to select one game. We did the coding using SIMPLECPP for these things.

When he selects the game now we make use of FILES to copy the question and solution of Sudoku which are stored in TEXT DOCUMENT FILES to the corresponding arrays and again using simplecpp we displayed it on the window. After he selects the

game number which he wants to play.

Then we arise the situation where the user should see 9x9 grid to play the game and numbers from 0 to 9 so that he can input the number he wants to the board by directly clicking on it and other options like VERIFY, HINT, ANSWER so that he can play the game. So we wrote code for all the necessary things which we stated above.

While the user is playing the game he may choose to take a hint by clicking on the HINT option available and he can also verify. So for all these things we wrote the code.

Brief Description of C++ Coding

We decided to make the Sudoku as the project and started doing research on it as it is a common game we found it little interesting as we can make the game little bit advance by checking the input entered by the user is correct or not.

By discussing between ourselves and with our TA Gaurav we first decided the outline of our body. And we discussed among ourselves and made a list of C++ programs required for our project.

In Our Project We make use of the following Libraries

- 1) **SIMPLECPP**: Used for Graphics like opening the Game Window, other graphics things.
- 2) **IOSTREAM**: Contains the basic input output functions.
- 3) **CSTDIO**: **C** **S**tandard **I**nput and **O**utput Library, this library uses what are called *streams*, Streams are handled in the `cstdio` library as pointers to [FILE](#) objects. A pointer to a [FILE](#) object uniquely identifies a stream, and is used as a parameter in the operations involving that stream.
- 4) **CSTRING**: The `cstring` header provides functions for dealing with C-style strings — null-terminated arrays of characters.
 - **Next We Used defined arrays and pointers like**
 - 1) `int board[9][9]`: Stores the input while user playing the game

2) `int solvedboard[9][9]`: Stores the Solution of the Sudoku which user is playing.

3) `int helpboard[9][9]`: This is a support array which contains only 0 and 1 and is used so that user cannot change the number in corresponding row and column of board array (because the number in the question cannot be changed) if it is 1 else it can change.

4) `FILE*fpin`: Used while opening the file to copy the question of Sudoku.

- Then comes the declaration of structure which contain the rectangle and pointer to the text and it useful for displaying the 9x9 grid and displaying the numbers in the array board.
- Then we used the functions like
 - `Levelcheck(int v)`: It takes the position of the mouse click as parameter and is used for checking which game did the user select.

- `Void Copyboard(int board[9][9],FILE*fpin,int helpboard[9][9]):`
After user selected the game we opened the file which is assigned with user selection and passes the pointer along with arrays board and help board and copy the question to board and stores 0 or 1 in helpboard as per situation and then returns.
- `Void Copysolvedboard(int solvedboard[9][9],FILE*fpin):`
Immediately after coping the question it copy's the solution to solvedboard.
- `bool CheckingValidity(int board[9][9],int i,int j):`
If the user asked to verify the number at a certain position then main function passes the row and column of that grid along with array board then it validates and returns true if it is correct else false.
- `int GetValue(int m,int n):`
If user wants to input the number he will click on the canvas then main functions passes x and y coordinate of the click and the function determines the number and returns that value.
- `void Copysolution(int board[9][9],int solvedboard[9][9]):`
If the user wants to see the answer he will click on the user button present on the canvas then main function calls this function by passing arraus board and solved board then it will copy solvedboard to board and then returns.

In the main function we made use of SIMPLECPP functions like g.hide(), g.reset(), etc.. We also made use of files during the coding to copy the question and solution to the corresponding arrays.

Work Distribution:

Coding with Graphics: N.SANDEEP

General Coding: N.SANDEEP and CH.SAIKIRAN
REVATI(Minor)

Documentation for Stage 2:

SRS and DRAFT USER MANUAL: N.SANDEEP

PROJECT REPORT : CH.SAI KIRAN

INDIVIDUAL Contribution:

N.Sandeep:

Main Program in Cpp file and some of the functions mentioned above(i.e, Copyboard, Copysolvedboard, CheckValidity,Getvalue) and done documentation for some files.

Ch.Sai Kiran:

Coded Functions such as Levelcheck, Copysolution. Done documentation for files, and helped Sandeep for Main Program lightly.

Revati :

Done documentation for files in Stage 1.

LIMITATIONS OF OUR PROJECT:

1. We are not solving the Sudoku(Just copying the answer using files).
2. We are not calculating the time elapsed.