**FINAL SUBMISSION OF CS 101 PROJECT**

MAIN REPORT

**THE PROJECT AT A GLANCE**

This project aims at the generation and solving of Sudoku puzzles using programs coded using C++ as the programming language. The project has been designed as a two way interface for the user working on the program. On the one hand the program accepts a Sudoku puzzle from the user and generates a solution to it, whereas, on the other hand, the user can also use the program as a game. In this mode the program generates a Sudoku and presents it before the user which the user can solve himself and then checks back the solution given by the user. Thus the program designed under this project serves the purpose of simple Sudoku solver as well as that of a Puzzle Game.

**BRIEF DESCRIPTION OF VARIOUS PROGRAMS USED**

* SUDOKU SOLVER

This program is written to solve any given Sudoku. For this initially, we defined four functions , three of which are similar which are used to check whether a given number (1-9) exists in a particular row, column or 3\*3 box. The fourth one will assume the presence of a particular digit different from the ones already existing in the 3\*3 box then it will check whether that digit can exist anywhere else in that box, if it doesn’t then it will assign that digit to that place, else it will do so for other numbers. Then the function which actually solves the Sudoku (Sudoku solver). It first finds how many numbers can exist at a particular position and also finds the number of positions where a given number can exist. One part of this function works only for easier Sudoku i.e, it only works if in two columns or two rows of a particular 3\*3 box a number exists.

* SUDOKU GENERATOR

For the generation of Sudoku the basic algorithm we used is to store a certain number of solved Sudoku puzzles in different files and use them for the further process of Sudoku generation. For this we created a function (name) which will randomly select one of the created files, take the solved puzzle from there and then will remove elements from a fixed number of positions; these positions again, are selected randomly. For this we used the ‘srand’ and ‘rand’ functions from the C++ library and included time as the SEED.

* GRAPHIC INTERFACE

The graphic interface was designed using EzWindows. In the beginning of the program we generated a window which displays two options, one for Sudoku solver and another for Sudoku game. The user choses one of them using Mouse click. For the former option he is presented with a new window where the user can provide his puzzle using a file, the program then solves the puzzle and returns the solution to the user on the terminal. For the later option the user is directed to a different window which presents a puzzle and allows the user to solve it by filling digits from 1 to 9, again using mouse click.

**STATUS OF COMPLETION**

In the finalized algorithms:

Solver working properly.

Generator working properly.

Program for Graphic Interface working properly.

The program for graphic interface is working properly but has not been included in this project submission as we were facing some problems while integrating it with the codes of Sudoku solver and generator and we were running short of time for working over this problem.

*Initially we worked on a different algorithm for Sudoku generator which proved to be wrong when we finally tested it (because of the phenomena called backtracking which require extensive use of recursion which we were not able to code)after the coding and therefore, we had to change the algorithm. So the initial Sudoku generator which is not used in the finalized program did not work and is attached in the file containing all the codes used in the project (Working as well as Non-working).*

*Our Sudokus Solver is not solving difficult codes since we could not come out with a way to use recursion properly.*

**BRIEF DESCRIPTION OF INDIVIDUAL CONTRIBUTION OF ALL THE TEAM MEMBERS**

* ANKIT AGRAWAL (Roll no. 110040017)

Being the team leader and co-ordinator, he has done both programming and managing tasks. On the programming part his main work was in the designing of the Sudoku generator. He searched for and thought of algorithms for generation of a Sudoku using random number generation and then designed a code for the generation of random Sudoku and then a code which could eliminate random elements from a given solved Sudoku. On the managing part he managed to call meetings of the working group and ensured that each member is present and if not conveyed to him the major decisions of that meeting. Ensured timely allotment of tasks to group members.

* ANAND AGARWALLA (Roll no. 11D170024)

The major task assigned to him was to design the Sudoku solver. He worked on this task with Anushrut Sharma . He designed the algorithm for the solver and then independently wrote a code for this. During the process he studied for functions in C++ library which were best suited to code his algorithm and has successfully used them in his code for Sudoku solver which is a major contribution in the project. Initially, he also wrote a function to check for numbers repeating in various rows and columns. He learned file handling which was used to call files for all three Sudoku generator, solver and graphics.

* ANAND SONI (Roll no. 110050037)

The major task assigned to him was in concern with the Sudoku generator. He proposed an algorithm for random Sudoku generation and working with Ankit Agrawal wrote a code for the generator on the initial algorithm which was dropped as stated above. But, his work is acknowledged. Also he wrote a function to accept the Sudoku as a problem from the user. He has given his major contribution in the documentation required for the project submission i.e. in the drafting of SRS(Software Requirement Specification), the Main Project Report and compilation of the final Consolidated diary.

* ANURAG SRIVASTAVA (Roll no. 110040087)

He was assigned one of the major tasks of the project which was designing of the Graphic interface of the program. For this he was to work with Ankit Kumar Singh. He individually studied EzWindows which is the main graphic interface used in C++ and designed a code which could take inputs from user as well as from other programs using files and display them in a Sudoku grid . Initially, he also worked on the basic solver algorithm in which he designed a code which could search all 3\*3 blocks for repeated elements. His major contribution was the graphic interface which had to be dropped because of the above stated problem but, his hard work is acknowledged.

* ANUSHRUT SHARMA

He was assigned the work of Sudoku solver with Anand Agarwalla. Working on this he wrote a code on Anand Agarwalla’s algorithm for which he also studied various new functions of the C++ library and successfully completed his code. (However we have taken pieces from codes written by both of them) Initially, working with Ankit Kumar Singh, he worked for a Sudoku solving algorithm.Thus his major contribution in the project was in the designing of the solver.

* ANKIT KUMAR SINGH

He was mainly assigned the task of Graphic interface with Anurag Srivastava. He gave some suggestions for this task and also helped in debugging some erroneous parts of the code. He sent a generator code at the last moment which no one understood and therefore, not used.

**IDEAS FOR FUTURE WORK**

The current Sudoku program or the Game has been made as a part of an introductory programming course and was meant to be completed within a given course of time and henceforth, does not cover all that we had thought of and which can be possible in such a project. However, at the professional level of software development too, there is always a scope of further improvement on the work performed.

Therefore, this project too, has a wide scope of further additions as well as improvements. Some of these which the constraints of our knowledge and time did not allow us to add to this project are included as ideas for further work below:

* The kind of algorithm we used for the Sudoku solver can be made more efficient in terms of the extent of difficulty of the Sudoku being solved. Further work is possible on this algorithm.
* The graphic interface can be improved both in terms of working time as well as presentation. Bitmapping can be introduced.
* Random number generation can be used for the generation of Sudoku.
* Levels can be introduced.