SUDOKU GAME

END TERM REPORT

By

Aditya Vikram, Riya Rai

Section: K19PG

Roll Numbers:
RK19PGA06
RK19PGA32

Department of Intelligent Systems,
School of Computer Science Engineering,

Lovely Professional University, Jalandhar

November, 2020

Student Declaration

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we are shall take full responsibility for it.

Student 1

Signature: Rai

Name: Riya Rai
Roll Number: RK19PGA32

Student 2

Signature:

Name: Aditya Vikram
Roll Number: RK19PGA06

Date: 31/10/2020

TABLE OF CONTENTS

<u>BONAFIDE CERTIFICATE</u>

*Certified that this project report "**SUDOKU GAME**" is the bonafide work of "**ADITYA VIKRAM** and **RIYA RAI**" who carried out the project work under my supervision.*

Signature of the Supervisor
(Due to Covid 19, signature is exempted)

Dr. Dhanpratap Singh

Academic Designation:

Assistant Professor in Intelligent System

ID: 25706

Department: School of Computer Science and Engineering.

BACKGROUND AND OBJECTIVES OF THE PROJECT

Background: Sudoku is also known as the Number Place originally when invented as a game. Sudoku is one kind of a puzzle game in which there is a 9 x 9 grid consisting of nine 3 x 3 smaller grids in it. The 3 x 3 smaller grids are also called “boxes”, “blocks” or “regions”.

Complete games are always an example of a Latin Square, including an additional constraint on the contents of individual regions. For example, the same single integer may not appear twice in the same row, column or any of the nine 3 x 3 sub regions of the 9 x 9 playing board.

Sudoku puzzles was invented in the 19th century, and the puzzle has appeared in puzzle books under the Number Place since 1979. Modern Sudoku only began to gain widespread popularity in 1986 when it was published by the Japanese puzzle company Nikoli under the name Sudoku, meaning “single number”. It first appeared in a U.S. newspaper, and The Times (London), in 2004.

A complete Sudoku grid is a special type of Latin square with the additional property of no repeated values in any of the nine blocks (or boxes of 3 x 3 cells). The relationship between the two theories is known, after it was proven that a first-order formula that does not mention blocks is valid for Sudoku if and only if it is valid for Latin Squares.

The general problem of solving Sudoku puzzles on $n^2 \times n^2$ grids of $n \times n$ blocks is known to be NP-complete. Many computer algorithms such as backtracking and dancing links can solve most 9×9 puzzles efficiently, but combinatorial explosure occurs as n increases, creating limits to the properties of Sudokus that can be constructed, analyzed, and solved as n increases. A Sudoku puzzle can be expressed as a graph coloring problem. The aim is to construct a 9-coloring of a particular graph, given a partial 9-coloring.

The fewest clues possible for a proper Sudoku is 17 (proven January 2012, and confirmed September 2013).Over 49,000 Sudokus with 17 clues have been found, many by Japanese enthusiasts. Sudokus with 18 clues and rotational symmetry have been found, and there is at least one Sudoku that has 18 clues, exhibits two-way diagonal symmetry and is automorphic. The maximum number of clues that can be provided while still not rendering a unique solution is four short of a full grid; if two instances of two numbers each are missing from cells that occupy the corners of an orthogonal rectangle, and exactly two of these cells are within one region, the numbers can be assigned two ways. Since this applies to Latin squares in general, most variants of Sudoku have the same maximum.

OBJECTIVES: There are many reasons behind making this project for the people of modern generation.

1- It is impossible to solve a Sudoku puzzle without concentration. Since this game requires logical thinking, an interruption can break the chain of thought and force the player to restart their analysis. The frustration of having to constantly go back to square one in order to progress will eventually train the brain to block any source of distraction.

The more puzzles you play, the more absorbed in your task you will be each time, improving your concentration skills step by step.

This better-developed skill will not only be felt when playing Sudoku but will also transpire to other activities in your life, be it at work, studying or performing a task that requires your full attention.

2- The two big bad words of today's society can be tamed by a simple numbers puzzle. One of the benefits of Sudoku is that it requires the player to concentrate on the grid and use logical thinking to find the solution for each cell. While doing this, the brain becomes fully focus on the task at hand rather than the source of stress and anxiety.

This break can be just enough for the player to regain their sense of balance and become calmer. Once the puzzle is over, they might even find that the task or the problem that generated so much anxiety is not as daunting as initially seemed.

3- When the brain is not stimulated, it tends to dwell on negative thoughts and infuse the person with an overall sense of unhappiness.

Just as exercising can boost your vitality, so does playing challenging thinking games like Sudoku. A fitter and happier brain is the first step to regard the world and your life with a healthier and more positive mind-set.

4- As far as the benefits of Sudoku go, improved thinking skill is likely one of the first players experience.

In the initial stages, solving a puzzle can be a chaotic process and you are likely to jump from the analysis of rows and columns to groups randomly. However, the brain will instinctively begin to find patterns of solutions. As the game progresses, you will come to understand which elements and which patterns are more likely to result in a quicker and easier solution.

Slowly, you will begin to apply this improved skill in your daily life too, and you will be able to identify more efficiently the best way to attain the outcome you desire.

5- Logical skills are the only true requirement to be able to play a Sudoku puzzle. Even in this case, the inference pattern is a simple one: if X is true, then Y is false. The difficulty levels in Sudoku depend solely on the number of allocated clues given at the beginning of each puzzle. The logic behind the game remains the same every time.

However, the more challenging the puzzle, the more complicated the inference pattern will become. As you progress in the game and face more challenging puzzles, the more you will improve your logical thinking.

6- Sudoku is a puzzle with simple, easy to understand rules that any kid can try. The need to engage in logical thinking to fill the grid correctly plus the process of trial and error they must apply will naturally and unconsciously help to develop their problem-solving skills.

Furthermore, the challenge of solving what to a kid will seem like an easy and boring game also helps them to engage more intensely to finish it quickly and improves their concentration skills.

These benefits of Sudoku can also help them in other areas and even improve their school performance.

7- The first time you play an easy Sudoku level, you might find yourself noting down all the candidates for one single cell in order to keep track of your progress. The more challenges you face, the quicker you will drop these notes as your brain will retain the information naturally.

In harder levels, notes will become essential once again, but your memory skills will still be stimulated in different ways. You will become able to remember more complicated strategies and how to apply them without referring to a tutorial. Patterns from previous games will also be memorized and you will find yourself looking for the opportunity to apply them once again.

8- One of the hidden benefits of Sudoku is the way it improves your capacity of analysing the relation and dynamic between tasks and subtasks.

This numbers puzzle cannot be solved by completing each row, column or group individually and once at a time. Their information is intertwined, and you must learn which subtask to tackle each time in order to progress in your goal to fill the whole grid.

If you want to succeed in life, you must also be able to take the information of a project as a whole and how and which subtasks to perform to achieve its successful completion.

DESCRIPTION OF PROJECT

The project deals with constructing a game of Sudoku using python programming language and implementing different techniques and technology of the programming language in it.

Sudoku is played on a grid of 9 x 9 spaces. Within the rows and columns are 9 “squares” (made up of 3 x 3 spaces). Each row, column and square (9 spaces each) needs to be filled out with the numbers 1-9, without repeating any numbers within the row, column or square. Does it sound complicated? As you can see from the image below of an actual Sudoku grid, each Sudoku grid comes with a few spaces already filled in; the more spaces filled in, the easier the game – the more difficult Sudoku puzzles have very few spaces that are already filled in.

By seeing which numbers are missing from each square, row, or column, we can use process of elimination and deductive reasoning to decide which numbers need to go in each blank space. Based on the neighboring rows and squares we cannot clearly deduce which number to add in which space. This means that we should ignore the upper left square for now, and try to fill in spaces in some other areas of the grid instead.

Sudoku is a game of logic and reasoning, so you shouldn’t have to guess. If you don’t know what number to put in a certain space, keep scanning the other areas of the grid until you seen an opportunity to place a number. But don’t try to “force” anything – Sudoku rewards patience, insights, and recognition of patterns, not blind luck or guessing.

Use process of elimination, what do we mean by using “process of elimination” to play Sudoku? One way to figure out which numbers can go in each space is to use “process of elimination” by checking to see which other numbers are already included within each square – since there can be no duplication of numbers 1-9 within each square (or row or column).

The goal of Sudoku is simple: fill in the numbers 1-9 exactly once in every row, column, and 3x3 region.

The classic Sudoku game involves a grid of 81 squares. The grid is divided into nine blocks, each containing nine squares.

The rules of the game are simple: each of the nine blocks has to contain all the numbers 1-9 within its squares. Each number can only appear once in a row, column or box.

The difficulty lies in that each vertical nine-square column, or horizontal nine-square line across, within the larger square, must also contain the numbers 1-9, without repetition or omission.

Note: Every puzzle has just one correct solution.

DESCRIPTION OF WORK DIVISION IN TERMS OF ROLE OF STUDENTS

The main base of the project that is building the interface and writing the code for its working forming the hard code was the major part handled by **Aditya Vikram.**

Usage of certain libraries in python first one being the **random** library. This module implements pseudo-random number generators for various distributions.

For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement.

Almost all module functions depend on the basic function random (), which generates a random float uniformly in the semi-open range [0.0, 1.0). Python uses the Mersenne Twister as the core generator. It produces 53-bit precision floats and has a period of 2**19937-1. The underlying implementation in C is both fast and threadsafe. The Mersenne Twister is one of the most extensively tested random number generators in existence. However, being completely deterministic, it is not suitable for all purposes, and is completely unsuitable for cryptographic purposes.

Usage of **time** library. This module provides various time-related functions. For related functionality, see also the datetime and calendar modules.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

Usage of **os** library

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module.

Usage of **pickle** library

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

OOP (Object Oriented Terminology) in python has been also used

Class − A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.

Class variable − A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variable are.

Data member − A class variable or instance variable that holds data associated with a class and its objects.

Object − A unique instance of a data structure that's defined by its class. An object comprises both data members (class variables and instance variables) and methods.

Operator overloading −The assignment of more than one function to a particular operator.

Usage of **loops**

In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on. There may be a situation when you need to execute a block of code several number of times.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times. The following diagram illustrates a loop statement

Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.

The GUI portion that is Graphical User Interface was handled by **Riya Rai**

Handling the RGB color part and bug detection was under this section.

Python provides various options for developing graphical user interfaces (GUIs).

The one used in our project is

Tkinter − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

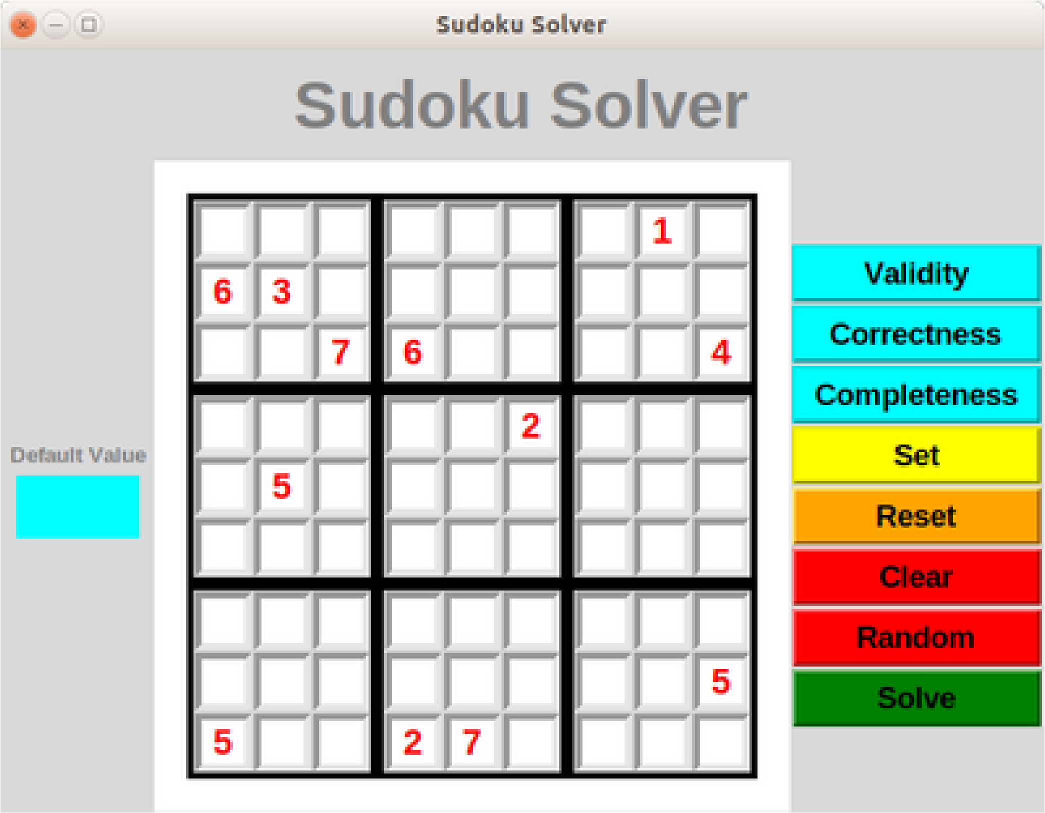
- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Using **Canvas** Operator

The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.

Using **Button** Operator

The Button widget is used to display buttons in your application.



Sudoku Solver

Sudoku Solver

							1	
6	3							
		7	6					4
					2			
	5							
								5
5			2	7				

There are various buttons used in the Sudoku platform the description of all of them are given below

Description of each of the button:

Validity: It checks whether the inputs entered are valid or not. No number should be repeated in any row or column.

Correctness: Checks whether the Sudoku is solvable or not.

Completeness: Whether all the rows and columns of each 3 x 3 cells of 9 x 9 sudoku is completely filled or not.

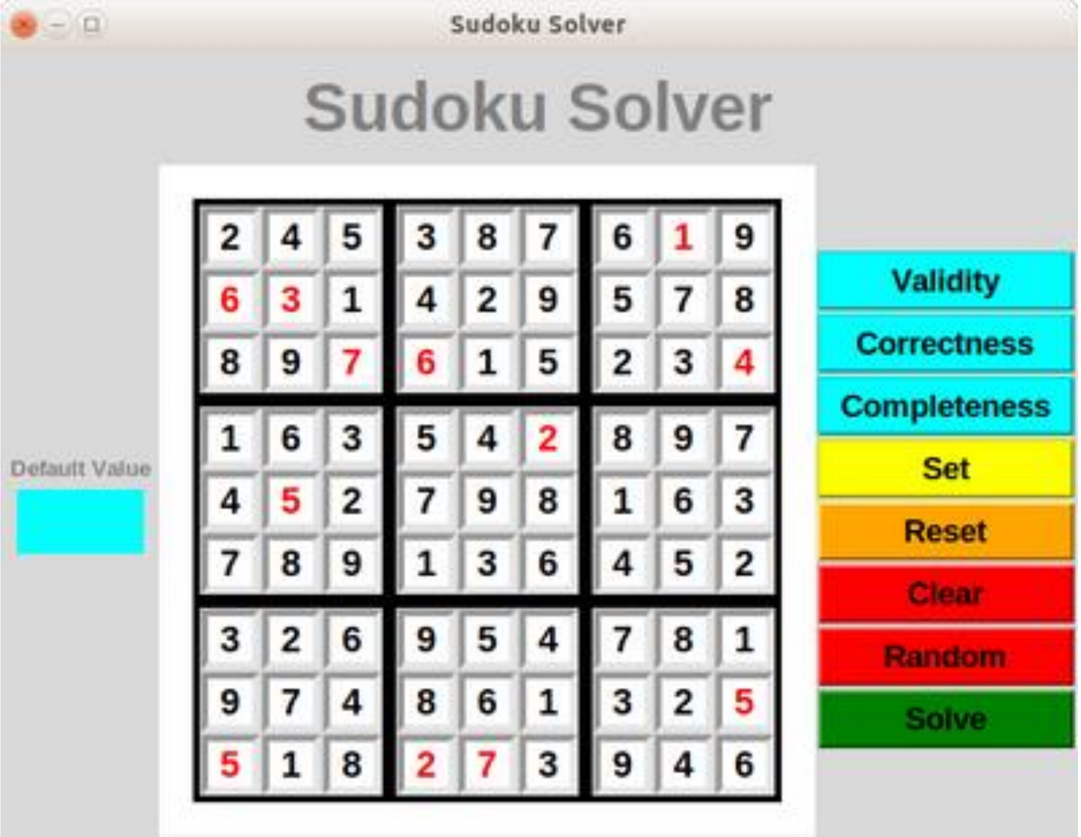
Solve: Used to solve the Sudoku.

Set: After you know that the values you have filled are correct accordingly then you are required to click on the set button to fix them.
Reset: If some of the random values entered in the Sudoku are not correct and you want to change them then you need to click on the Reset button.

Clear: To erase all the pre filled data in the Sudoku. In order to restart.

Random: To fill in the random numbers in the Sudoku not completely but partially.

Completely Solved Sudoku:



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

Description of Technologies and Platforms used in the project

Python Programming is used to execute the overall project and get the output.

Python is object-oriented, interpreted, and a high-level programming language. It has incredible built-in data structures that are combined with the dynamic typing and binding to render hassle-free app development. It poses as a scripting or glue language to combine several components together.

Python is renowned for its simple and easy-to-learn syntax which supports readability and reduces the expenses incurred in the program maintenance. It also favours modules and packages, which in return promote modularity and code reusability. To add another jewel to its crown, it totally favours cross-platform, making Python ideal for mobile app development.

GUI (Graphical User Interface) usage is also included

Python and tkinter are excellent for developing GUIs. While this is true of most scripting languages, I think these two make a particularly good combination.

For most GUIs, functionality is more important than form (i.e.: eye candy is not top priority). Even then, it's possible to make very nice looking GUIs with Tkinter. Tk has been my go-to toolkit for close to twenty years. It is most easily used with Tcl since that is its native language, but there's no denying the advantage python has over Tcl as far as built-in functionality goes.

That being said, all of the toolkits available to a python programmer are good. Tkinter's main advantages are that it is extremely easy to use and you probably already have it installed, so the barrier for entry is very low.

Used Anaconda for compilations of code

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole[18]
- Spyder
- Glue
- Orange

- RStudio
- Visual Studio Code

Also Used IDLE to run the code just in case to check whether the code is compatible on all the platforms or not

IDLE (short for **Integrated Development and Learning Environment**) is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit (wrapper functions for Tcl/Tk).

IDLE is intended to be a simple IDE and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

SWOT Analysis (STRENGTH, WEAKNESS, OPPORTUNITIES and THREATS) of the Project

- **Strengths:** Sudoku can help us to improve on our algorithm making skills. Memory and logic work side-by-side when you are playing Sudoku. We use our memory to memorize the numbers, when we use our logic to figure out the next blank. It keeps you practicing your logical thinking process when you are solving a puzzle, and eventually improve your number skills by keeping your brain active. Not only playing Sudoku is interesting but it helps to increase your sense of time. You will learn how to make a decision and take an action with less hesitation. Sudoku requires players to think strategically and solve problems creatively. Once you stop playing in the middle of the game, you have to start the whole thinking process, which helps you to develop your concentration power and re-focus skills. Sudoku gives you a sense of accomplishment when you can solve a puzzle, especially the puzzle is a difficult one.
- **Weaknesses:**

1. **Addictive and Frustrating**
2. **Time-consuming**
3. **Distraction**
4. **Sense of incompletion (You will never finish all the puzzles)**
5. **Sudoku is a threat to crossword**
6. **Can lead to unbalanced lifestyle**
7. **Responsibility shirking**
8. **Can lead to anxiety**
9. **May lead to people abandoning their mental health strategies**
10. **Becomes less challenging**

- **Opportunities:** Solving Sudoku puzzles requires a sense of attentiveness to opportunity. Good Sudoku players know how to recognize opportunities and spot future chances as they emerge. Ideally, if you want to be a great Sudoku player, you can develop a mindset that helps you see good opportunities and think “two moves ahead” so you can anticipate the next places on the grid where you might be able to place new numbers. Sudoku is a game of momentum – each number placed on the grid opens up additional opportunities; each number placed brings you one step closer to solving the entire puzzle. However, sometimes the momentum will grind to a halt. The best Sudoku players know how to “switch gears” mentally and start turning their attention to a different part of the grid, or start looking for opportunities in a different corner of the grid. Sudoku rewards focus and single-mindedness, but it also rewards adaptability.

- **Threats:** Since bug protection is used there is no threat possessed by the code until and unless there is some other interference by the external malwares and viruses.