Volume: 4 Issues: 13 [June, 2019] pp.-71-83]
Journal of Information System and Technology Management
eISSN: 0128-1666
Journal website: www.jistm.com

SOLVING DIAGONAL SUDOKU PUZZLE 9X9 GRID BY JAVA PROGRAMMING ALGORITHM

Nur Intan Syafinaz Ahmad[1]
Fakulti Sains Komputer dan Matematik,
Universiti Teknologi MARA (UiTM), Malaysia
(Email: nurin395@uitm.edu.my)
Nurhuda Ismail[2]
Fakulti Sains Komputer dan Matematik,
Universiti Teknologi MARA (UiTM), Malaysia
(Email: nurhudaismail@uitm.edu.my)
Ahmad Khudzairi Khalid[3]
Fakulti Sains Komputer dan Matematik,
Universiti Teknologi MARA (UiTM), Malaysia
(Email: ahmad4829@uitm.edu.my)
Muhammad Syawal Abd Halim[4]
Fakulti Sains Komputer dan Matematik,
Universiti Teknologi MARA (UiTM), Malaysia
(Email: syawal434@uitm.edu.my)

***Accepted date**: 16-03-2019*
***Published date**: 09-07-2019*

To cite this document: Ahmad, N. I. S., Ismail, N., Khalid, A. K., & Halim, M. S. A. (2019). Solving Diagonal Sudoku Puzzle 9x9 Grid by Java Programming Algorithm. *Journal of Information System and Technology Management*, 4(13), 71-83.
DOI: 10.35631/JISTM.413007

Abstract: *Sudoku is one of the well known puzzle game that have been played world wide. There are many types of Sudoku puzzle that exist to be played such as diagonal, diagonal even, diagonal odd, isosudoku, lucky seven, parquet and others. This paper focus on solving the diagonal Sudoku by using Java Programming Algorithm for 9x9 grid. For normal Sudoku, the number 1-9 must be filled up in all grids with no number duplication conflict in the row, column and 3x3 grid inside the Sudoku puzzle itself. While the diagonal Sudoku has one extra condition to be considered that is the number 1-9 must also be filled in the diagonal place which start from the most upper-left or bottom-left to the most upper-right or bottom-right of the puzzle. There is no conflict with other grid in its diagonal region. In order to solve the puzzle, we have to contruct algorithm and translate it using Java Programming Language. The guessing method with backtrack approach has been selected to construct the algorithm. This Java Programming are designed to solve the diagonal Sudoku puzzle for 9x9 grid.*

Keywords: *Diagonal Sudoku, Sudoku Puzzle, Java Programming and Guessing Method*

Introduction

Sudoku puzzle is a puzzle that uses player’s logical thinking to solve it. They need to insert a number according to its rule. Sudoku is a popular game since it has more than 97 million results in Google search.

Playing the Sudoku will improve the capabilities of thinking logically and critically. Sudoku game will enhance mathematical skills and confidence to the players. There are no specific mathematical calculations required to solve Sudoku, but it will use some of mathematical methods and techniques depending on the type of Sudoku that being played.

According to Jr. (2005), Sudoku comes from the Japanese words which are “*su*” and “*doku*”, means “*number*” and “*single*”. In English, it is also being called as “*number place*”. Sudoku presented with square grid. Inside the square grid there is a single square. Basically, Sudoku contains various sizes of grid such as 4x4 grids, 8x8 grids up to 100x100 grids. But 9x9 grids are the most popular among others. There are many types of Sudoku that have been invented such as the normal Sudoku, Geometry Sudoku, Diagonal Sudoku and others.

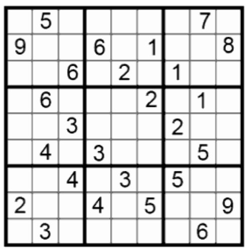


Figure 1: Normal Sudoku

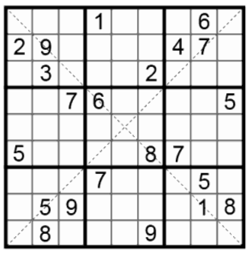


Figure 2: Diagonal Sudoku

Sudoku have been found in United States and United Kingdom many years ago and before it became in Japan. Around 1070’s, the first publisher of Sudoku in New York was “Dell Magazines”. Dell published Sudoku in its Math Puzzles and Logic Problems magazines using the named as “Number Place”. The first designer of the Sudoku is unknown, but Walter Mackey was one of the constructors of Sudoku puzzles in Dell’s Magazine. In 1984, Sudoku was first introduced in Japan by Nikoli. Currently, there are five Sudoku magazines that have been published in Japan every month with a total circulation over 600,000. Semeniuk (2005) claim that the huge popularity of Sudoku mainly because it is very challenging puzzle but have a very simple rule. Sudoku was very popular, so many people examine and studies about Sudoku. They studied and designed many methods to complete this game. Most of the method that will be used is mathematics. From those studied, Sudoku and mathematics are related to each other. There are so many mathematical techniques used to construct or to solve Sudoku. From the different sources that have been found, most of the researchers were discuss about the mathematical techniques that commonly involve in the Sudoku games which are Permutations, Latin Square and Enumeration. This method has been invented to determine the number of possible unique grids that will exist inside the Sudoku. All these invented methods are confirmed by Huckvale (2005) that stated, a good Sudoku puzzle is one that can be solved without guessing method.

Simonis (2005) is one of the first who published Sudoku based paper in the computer science literature. He has formulated sudoku as a constraint satisfaction problem. This work has been continued by Lynce (2006). They show that Sudoku puzzle can be solved using propositional satisfiability techniques. This technique demonstrated on how individual Sudoku puzzle may be converted into conjunctive normal form.

One of the advantages of playing this puzzle is high ability in solving logical problems. This is the reason why Sudoku puzzle is commonly used by parents to educate their children during free time. With the popularity of Sudoku, there are various competition that has been held world wide. In order to solve Sudoku puzzle, we must know the technique that basically use in solving the puzzle in a fastest way. Most of the techniques that player used are difficult to understand. Thus, the player will solve the puzzle in longest time. Other than using the Human Solvers' solution to solve a Sudoku puzzle, a player can retrieve the solution by using Computer's Solution. Algorithms to achieve the solutions will be needed in order to construct the Computer's Solution programs. Some of the algorithms are likely to imitate the human solver methods by showing the step by step solving methods, while others are using the guessing strategy that will find and directly show the solution for the given puzzle.

Solving a Sudoku puzzle with the aid of some computer's solver can achieve the solution easily. Due to technology sophisticated, this games not only available in the newspapers, but it is can be found on a computer via internet and digital games via application. Various programming language are used in order to construct the digital games. The programming language commonly used in constructing Sudoku games are *Java*, *Hypertext Markup Language (HTML)* and C or *C++*. All programming language will not give the same result. The result of the games is related to the algorithm of finding the solutions to the puzzle. This paper focus to 9x9 grids Diagonal Sudoku and Java programming language will be used to solve the 9x9 grids of diagonal Sudoku. It can help many people to solve diagonal Sudoku by using Java language. Therefore, the objective of this paper is to construct a diagonal Sudoku solver program using Java programming language.

Literature Review

Lawler et al (1985), stated that Sudoku is a combinatorial optimization problem. In order to solve a normal Sudoku puzzle, the number 1 until 9 must be filled up in all grids with no number duplication conflict in the row, column and box (a 3x3 grid inside the Sudoku puzzle). In other hands, to solve the diagonal Sudoku not only all the above rule have to been considered but number 1 until 9 must also be fill in a diagonal place and we have to make sure the number does not conflict with other grid in its diagonal region. Diagonal place is a set of grid which start from the most upper-left or bottom-left to the most upper-right or bottom-right of the puzzle.

According to Sopitan (2012), there is a question about how many possible Sudoku puzzle can be created. In order to know how many of possible diagonal Sudoku puzzles can be made, some permutations must be used while considering the the symmetrical of a Sudoku grid. By applying a brute-force computation, the result of valid Sudoku grids arrived at 6670903752021072936960 (6.67 x 10^{21}). Furthermore, Sopitan (2012) claim that the minimum number givens or hints must be at least 17-hints number in order to achieved a unique solution for a diagonal Sudoku. Therefore, the puzzle can be considered valid if the puzzle has unique solution. Unfortunately, he said that Sudoku puzzle has the minimum number of clues, which is seventeen does not truly has unique solution because the choice of seventeen as the minimum number of required clues must have for a puzzle is just a general acceptance among the Sudoku enthusasts. According to Semeniuk (2005), the number of given hint does not determine the level of difficulty of the puzzle.

Sopitan (2012) has categorized human solvers' solution into two categories. There are Logical Deduction and Process of Elimination. Logical deduction is a process that can achieve the solution value for the grids, while process of elimination is to reduce the possible numbers (which will be called "candidate" later on) in certain grids. Davis (2005) has discussed about techniques that can be used, the situation effect to the techniques used, the outcome by applying the techniques, and others. He also stated that not all techniques can be used in order to achieve the solution. Meanwhile, to achieve some solutions from certain puzzle, especially the hard and diabolic level puzzle, he suggested to use almost all the techniques that has been discussed. Rhyd (2007) has invented the first application of a metaheuristic technique in order to solve the puzzle. He use simulated annealing to complete logic-solvable puzzle.

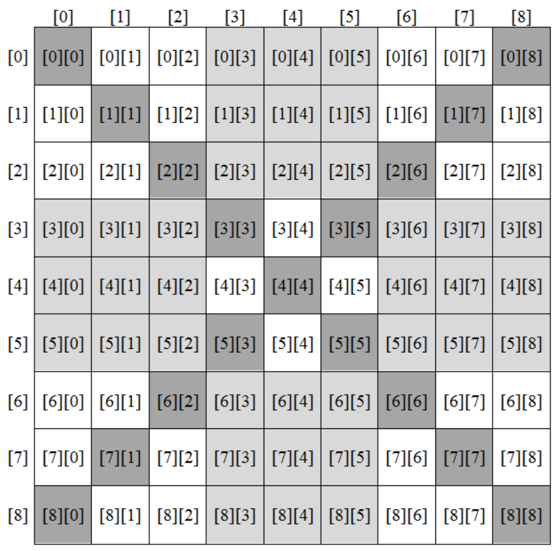
The other methods that can be used are Genetic Algorithm (GA). Many researcher have been using GA to solve the puzzle. Mantere (2006), has run a test to find if a genetic algorithm optimization is an efficient method to solve sudoku and generate the Sudoku puzzle. They also investigate whether Sudoku puzzle are difficult for human solver and difficult to solve using genetic algortihm. In 2006, Nicolau (2006) has presented a different approach to solve this puzzle. They optimizes the sequence of logical operations and applied it to find the solution by developing a system named GAuGE (Genetic Algorithm using Grammatical Evolution). Perez (2008), they have been using four different methods. There are cultural genetic algorithm, repulsive particle swarm optimization, quantum simulated annealing and genetic algortihm or simulated annealing hybrid (HGASA). By applying all methods, the result indicated that HSAGA was the most efficient method to solve the puzzle. Mantere (2009) has also proposed another method to solve the puzzle using evolutionary algorithm (EA) example genetic algorithm (GA), cultural algorithm (CA), ant colony optimization(ACO) and genetic algorithm or ant colony optimization hybrid (GA/ACO). Based on these methods, the researcher found that GA/ACO method is the most efficient of all. Xiu (2011) has improved the previous method by using hybrid genetic algorithm. However, Gold (2005) stated that genetic algorithm seem inefficient since it need 35700 generations to come up with a new puzzle.

Another approach that can be used is Artificial Bee Colony algorithm. This algorithm was first developed in 2005. Jaysonne (2009), has explored the possibility of using improved variant of the Artificial Bee Colony algorithm. He concluded that this algorithm can be used to solve Sudoku problem effectively and efficiently. According to Li (2009), they have introduced a graph search strategy on the basic of knowledge representation and deduction base in artificial intelligence.

Methodology

This study has introduced algorithm to solve diagonal Sudoku 9x9 grid. This algorithm created by using guessing method with backtracking approach for constructing Java programming language. According to Sopitan (2012), some of the computer's solvers algorithms that solve Sudoku puzzles imitate the human solver while others use the guessing strategy. Guessing involves inserting values and seeing if they lead to a solution. If guess breaks diagonal sudoku rules, another guess must be made. This strategy is really difficult for human solver because a guess could look correct until very late into solving of puzzle until at which point, it become almost impossible to keep in track. However, this strategy is much easier for computer to handle due to its processing power. Carpenter (2013) has used Java programming language to develop his Sudoku solver program by using guessing algorithm with bactracking approach. Algorithm for solving diagonal Sudoku puzzle is similarly to the standard Sudoku puzzle. Due to some of differences of rule to solve, some modification from the algorithm to solve the standard one has been made due to the rules to get the diagonal Sudoku solver's algorithm.

The diagonal Sudoku puzzle that being used in this paper is 9x9 grids, it is square and has two dimensions (2D). Thus, to construct the coding for this paper, 2D-arrays has been used to store all the value for the Sudoku puzzle in the correct order due to its ability to store several values with same data type in a single variable in 2D. To implement this coding, the coordinate of the Sudoku puzzle grids is needed in order to put them in 2D-arrays in the coding. The coordinate of the grids is stated in the figure 3.



	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
[0]	[0][0]	[0][1]	[0][2]	[0][3]	[0][4]	[0][5]	[0][6]	[0][7]	[0][8]
[1]	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]	[1][5]	[1][6]	[1][7]	[1][8]
[2]	[2][0]	[2][1]	[2][2]	[2][3]	[2][4]	[2][5]	[2][6]	[2][7]	[2][8]
[3]	[3][0]	[3][1]	[3][2]	[3][3]	[3][4]	[3][5]	[3][6]	[3][7]	[3][8]
[4]	[4][0]	[4][1]	[4][2]	[4][3]	[4][4]	[4][5]	[4][6]	[4][7]	[4][8]
[5]	[5][0]	[5][1]	[5][2]	[5][3]	[5][4]	[5][5]	[5][6]	[5][7]	[5][8]
[6]	[6][0]	[6][1]	[6][2]	[6][3]	[6][4]	[6][5]	[6][6]	[6][7]	[6][8]
[7]	[7][0]	[7][1]	[7][2]	[7][3]	[7][4]	[7][5]	[7][6]	[7][7]	[7][8]
[8]	[8][0]	[8][1]	[8][2]	[8][3]	[8][4]	[8][5]	[8][6]	[8][7]	[8][8]

Figure 3: Grids Coordinate

The guessing method with backtrack algorithm that has been selected will search the first suitable value for each grid in the puzzle. The searching for the value will be done from column to column, which means starting from column [0] where the initial grid [0][0] be and go down to grid [8][0]. Then, it is continuing to next column, column [1] and so on until column [8].

In order to solve a diagonal Sudoku puzzle using this Java programming algorithm, the clues need to be key in first. The clues will be stored in order to be used in searching the solution for the puzzle. The clues that have been keying will be check whether it follows the rule of the diagonal Sudoku or not. If it follows, then the searching process for the unoccupied grid will be started. If it follow the rule, the searching process will begin. Otherwise, it will halt and stated that the puzzle is invalid. Then, to start searching the value for the grid, it will check whether the grid already has a clue given or not. If a clue has existed in the grid, move on to next grid. If it is empty or 0, the searching process will be start by trying from 1 until 9. The value that is being trying will be compared in row, column, box and diagonal region one by one. If the first trying value has been found and following the rules, the value will be stored in the grid and the searching will be move to the next grid. If it is not found until number 9, the grid will be set to 0 and will be move backward to the previous searched grid. Then, the previous searched grid will be trying to search again for another new value until it has found a new one or until number 9. This process will continue until it has found the solutions for the puzzle by successfully found all the values for the empty grids, or if it has not found any suitable values for the empty grids, which means the puzzle has no solution.

The result will shows whether it has successfully found the solution by displaying the solution, or an error message stated that there is no solution for the puzzle. The figure 4 shows Java programming algorithm.

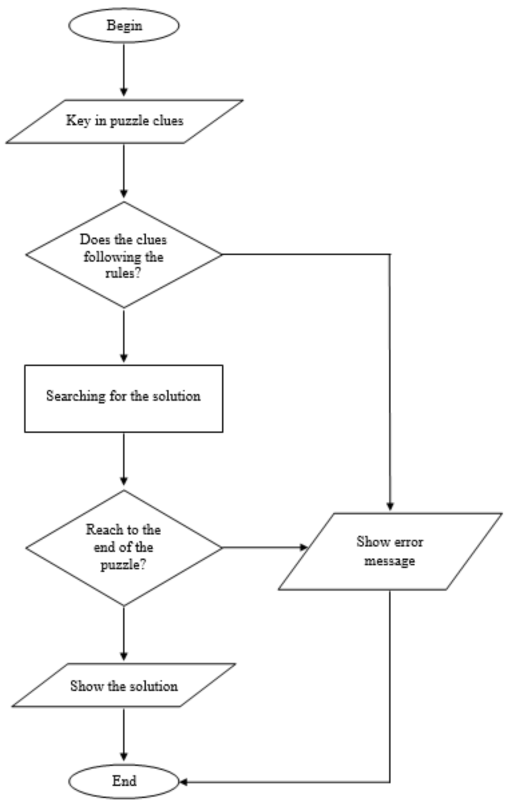

Begin
Key in puzzle clues
Does the clues following the rules?
Searching for the solution
Reach to the end of the puzzle?
Show error message
Show the solution
End

Figure 4: Java Programming Algorithm

Implementation

Java programming language is used to create the diagonal Sudoku 9x9 grid only. Graphic User Interface (GUI) has been implemented to make it look nicer rather than just using the black-screened command prompt window. With some menu options that can be chose, pop-up message boxes, and also option message box. Pop-up message boxes is created to interact with the user while option message box to give options to the user to choose the clue’s value in the grids. It can be convenience to the user to use it.

In the next section is the full description manual of Java programming output.

Main Graphic User Interface (GUI)

This is the main GUI for the 9x9 diagonal sudoku puzzle programming where it has menu bar that contains options and help. Other than that, this main GUI also has diagonal Sudoku grids panel for user to insert the puzzle and it will shows the solution for the puzzle.

Menu bar

Diagonal

Sudoku

grids

panel

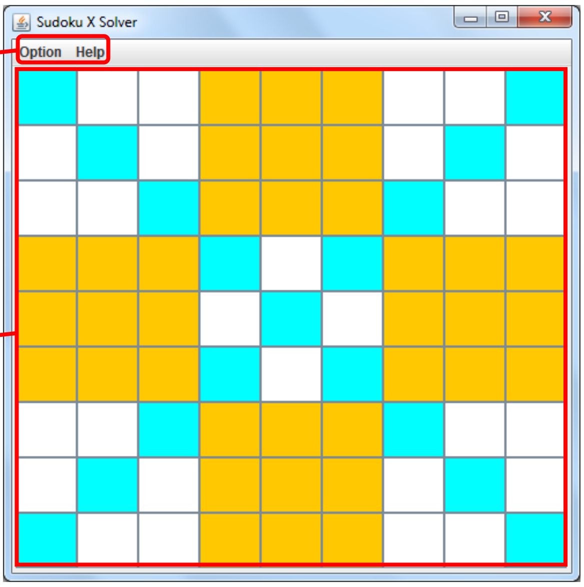


Figure 5: Main GUI

Option Menu

In the option menu, there will be “Solve”, “Clear” and “Exit” options. “Solve” option will solve the diagonal Sudoku puzzle that has been inserted in the diagonal Sudoku grids panel. “Clear” option will clear the diagonal Sudoku grids panel. “Exit” will close the diagonal Sudoku programming.

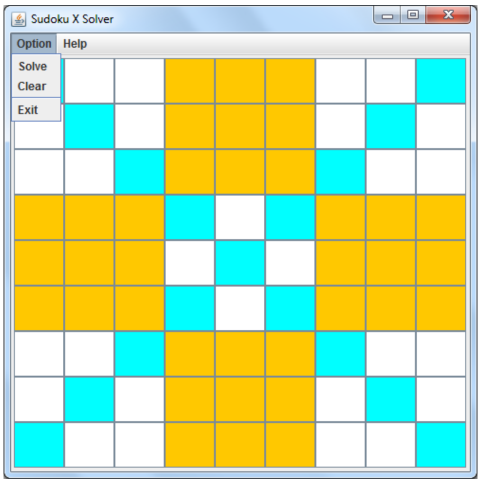


Figure 6: Option Menu Options

Help Menu

In the help menu, there will be “How to use?” and “About” options. “How to use?” option will tell the user how to use the program. However, “About” option will tell the user briefly about the diagonal Sudoku programming.

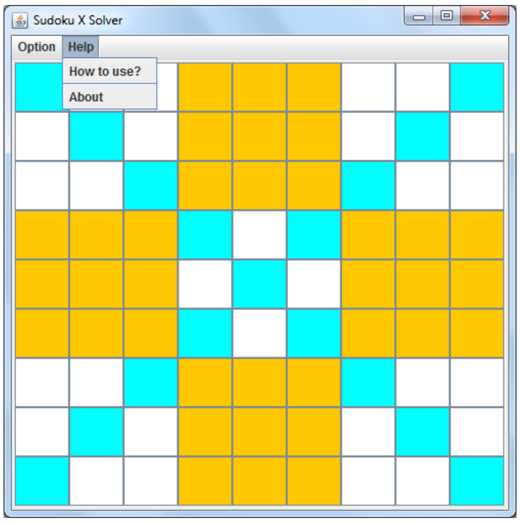


Figure 7: Help Menu Options

Diagonal Sudoku Grids Panel

Solving a puzzle by using this program will need a some clues to be inserted. Thus, to insert a clue in a grid in the panel, just simply click any of the grids in the panel, and a pop-up window will be shown to select the clue's value in the clicked grid. The value that can be chose to be inserted in the grid panel is from 1 until 9. In order to erase back the input value that has been chose, the user can click again the wronged inserted grid. Then, click "Erase" button to erase the value in the current grid.

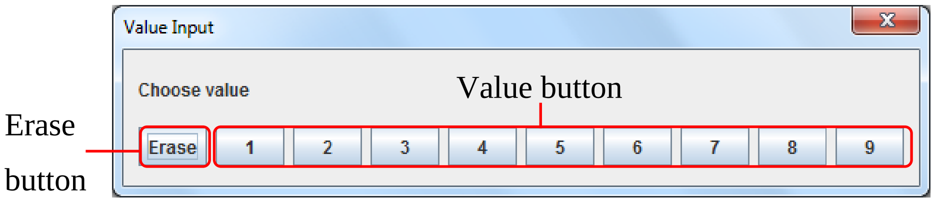
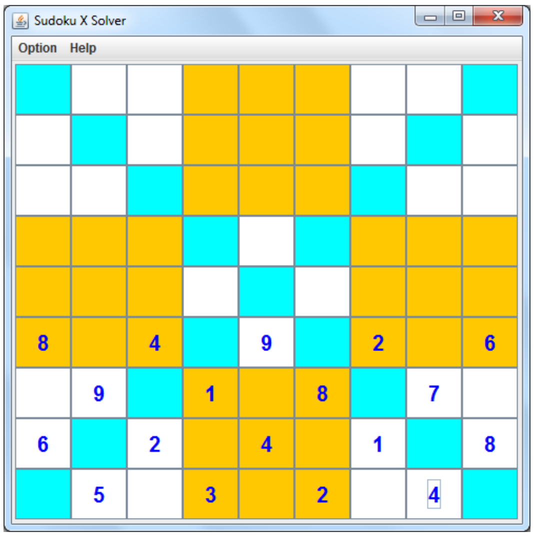


Figure 8: Value Input Panel Box

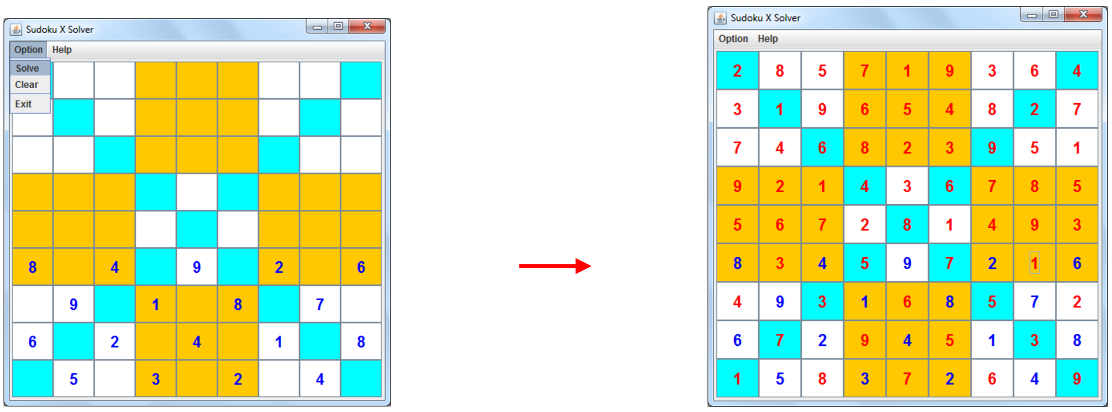


8		4		9		2		6
	9		1		8		7	
6		2		4		1		8
	5		3		2		4	

Figure 9: A Grid Panel That Has Been Inserted With Clues

Solving a Given Puzzle

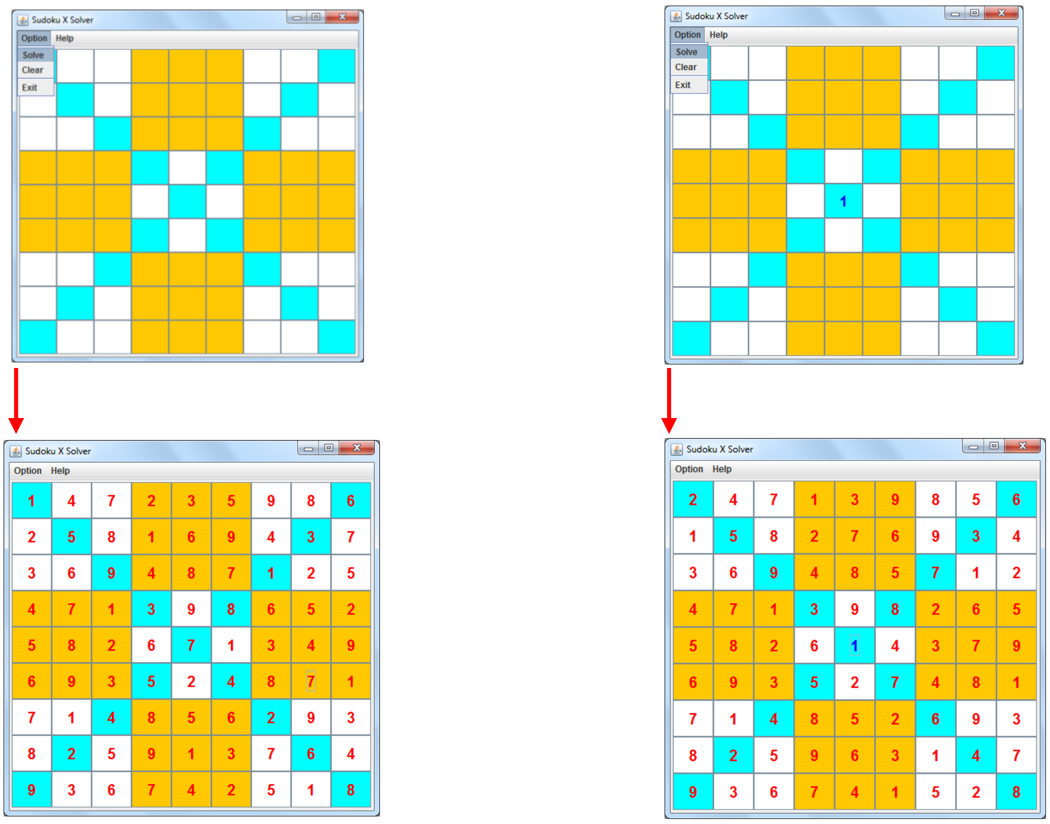
After inserting the clues, to show the solution for the given puzzle by click the Option menu, then click Solve option. However, if there is no clue being given or less than seventeen clues is given in the panel to be solved, the program will definitely show the solution from the given clue in the grid. The solution will be shown in just a few seconds. From figure 10 and figure 11, the blue value in diagonal Sudoku grids panel is the clues, and the red value is the solution.



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

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

Figure 10: Solving a Puzzle



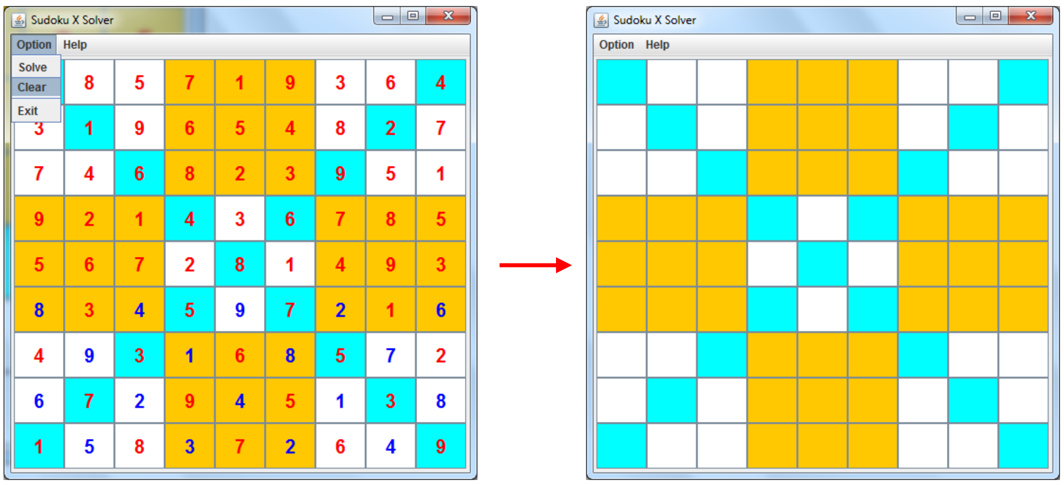
1	4	7	2	3	5	9	8	6
2	5	8	1	6	9	4	3	7
3	6	9	4	8	7	1	2	5
4	7	1	3	9	8	6	5	2
5	8	2	6	7	1	3	4	9
6	9	3	5	2	4	8	7	1
7	1	4	8	5	6	2	9	3
8	2	5	9	1	3	7	6	4
9	3	6	7	4	2	5	1	8

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

Figure 11: Examples of Solution For Empty Grids Panel and A Clue Given Puzzle

Clearing Grids Panel

Figure 12 shows how to clearing the grids panel. For clearing the grids panel, click the Option menu, then choose Clear option.



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

Figure 12: Clearing The Grids Panel

Giving Error Puzzle To Be Solved

Figure 13 shows the examples of violated rule puzzle. If the given puzzle is violated the diagonal sudoku puzzle rule, there was an error pop-up message box appeared to notify the user like in figure 14.

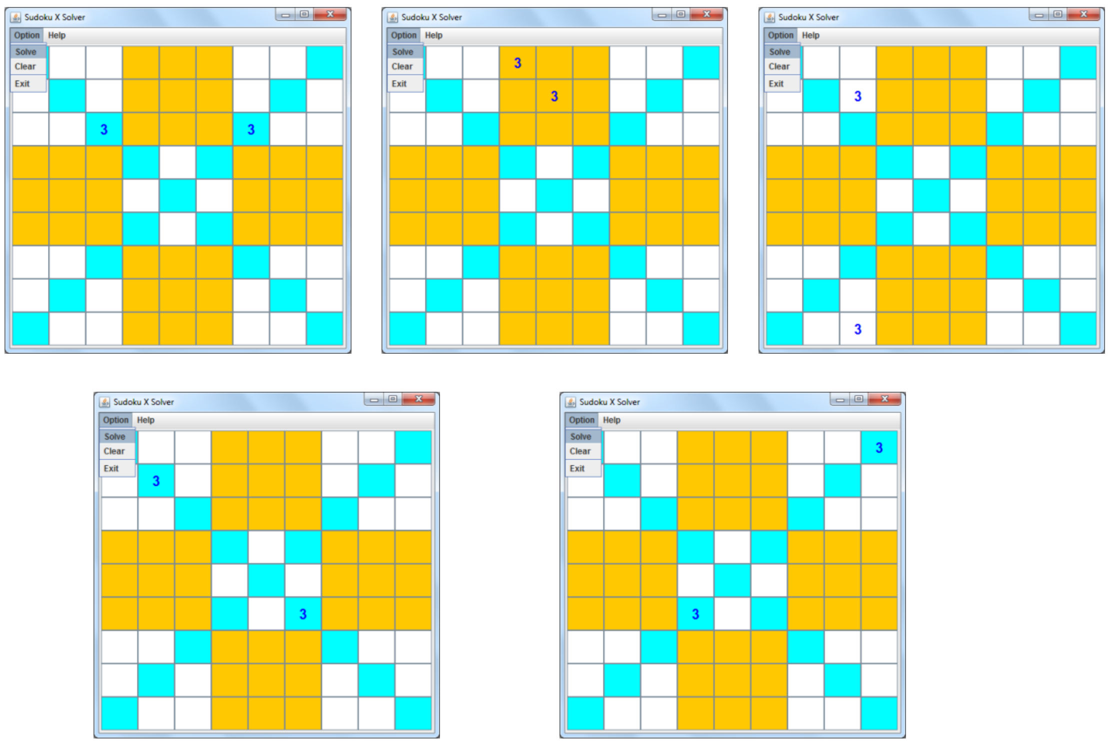


Figure 13: Examples of Violated Rule Puzzle

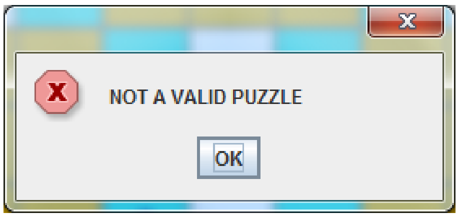
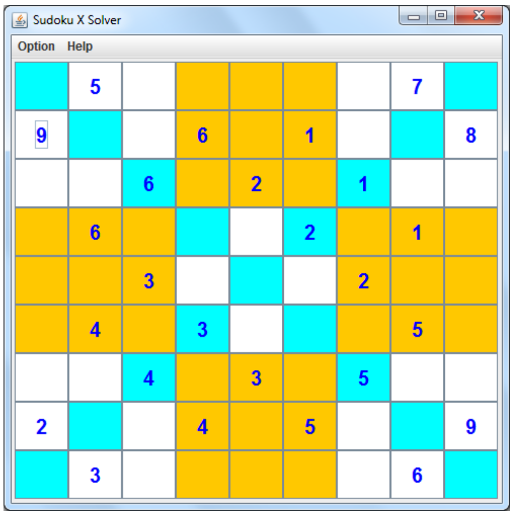

NOT A VALID PUZZLE

Figure 14: Error Message Box For Solving Violated Rule Puzzle

Given Puzzle Does Not Has Solution

If the given puzzle does not has any solution, a pop-up error message box will come to state that the given puzzle does not has solution. Figure 15 shows an example of a puzzle that does not have solution and its error message box.



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

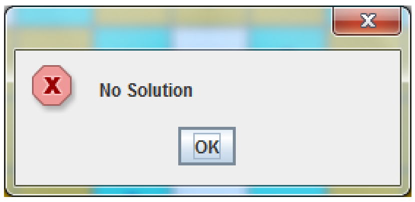

No Solution

Figure 15: A Puzzle That Does Not Have Solution (Left) and Its Error Message Box (Right)

Results and Discussion

Java programming is built to help human for solving Sudoku problem. The puzzles that have been inserted can be solved using Java Programming. Either the problem is easy, intermediate, difficult or even diabolic level; all can be solved within a few seconds. Unfortunately, this program cannot differentiate the given puzzles whether it has unique or many solutions. Furthermore, even the puzzle has many solutions, the program will only show the first solution that it has found while searching the solution.

Conclusion and Recommendations

In conclusion, Java programming is needed for solving diagonal Sudoku 9x9 grid because it is usefully to Sudoku player to find the solution. Just by inserting the clue of the puzzle, add with some clicks, and the solution will be shown in the panel. Even any level of puzzle has been inserted; the program still can show the solution of the given puzzle. It is very helpful for human, especially for the diagonal Sudoku player to use it as an aid in their way to find their diagonal Sudoku puzzle solution.

For the future research, who is interested in this field can use another alternative like making a modification or improvement in creating the diagonal Sudoku problem solver like adding it can be solve step by step by showing the human solvers' techniques. Other than that, researcher can make further research about the human solvers' techniques because the human solvers' techniques always being discussed in order to improve the techniques and make it simpler to be used. Besides that, the researcher can apply another programming language to construct the diagonal Sudoku computer problem solver, such as C/C++, Python, Hypertext Markup Language (HTML) and others.

References

Carpenter (2013). *Soduku.java.* [Online]. [Accessed 2nd January 2019]. Available from World Wide Web: https://github.com/bob-carpenter/java-sudoku/blob/master/Sudoku.java

Davis, T. (2005). Sudoku Code for Solving Puzzles. Retrieved March 22, 2013, from http://www.geometer.org/puzzlex/index.html

Gold, M.. (2005). Using Genetic Algorithm to Come Up with Sudoku Puzzles. http://www.csharpcorner.com/UploadFile/ mgold/Sudoku09232005003323AM/Sudoku.aspx ?ArticleID=fba36449-ccf3-444f-a435a812535c45e5 (cited 11.9.2006)

Huckvale M.. (2005). Sudoku Puzzle. Online Resource accessed July 2005. http://www.phon.ucl.ac.uk/home/mark/sudoku/

Jaysonne, A.P, Glaiza, M.M.S & John, P.T.Y. (2009). Solving Sudoku Puzzle uisng Improved Artificial Bee Colony Algorithm. *4th International Conference on Innovative Computing, Information and Control.* 885-888.

Jr., E. P. (2005, September 15). Sudoku Variations. http://www.maa.org/editorial/mathgames/mathgames_09_05_05.html

Lawler, E.L, Lentra, J.K, Rinnooy, A.H.G & Shmoys, D.B. (2005). The Travelling Salesman Problem- A guide Tour of Combinatorial Optimization. *John Wiley & Sons, New York.*

Li, Y.D. & Deng, X.Q. (2009). Solving Sudoku Puzzles Base on Improved Genetic Algorithm. *J. Tonghua Nomal Technol*. 43 -45

Lynce, I. & Quaknine. (2006). Sudoku as a SAT problem. *In proceeding of the 9th Symposium on Artificial Intelligence and Mathematics*.

Mantere, T. & Janne K. (2006). Solving and Rating Sudoku Puzzles with Genetic Algorithm. *12th Finnish Artificial Intelligence Conference, Finland*. 86-92.

Mantere, T. & Koljonen, J. (2009). Ant Colony Optimization and a Hybrid Genetic Algorithm For Sudoku Solving. *15th International Conference on Soft Computing, Brno, Czech Republic.* 41-48.

Nicolau, M. & Ryan, C. (2006). Genetic Operators and Sequencing in The GauGE system. *IEEE Congress on Evolutionary Computation*. 1561-1560.

Pendlebury, P. (2005). Can You Sudoku. Article 'Can you Sudoku'. Article appearing in The Mail on Sunday, London, 8th May 2005. http://www.mailonsunday.co.uk/pages/live/articles/news/news.html?in_article_id=348348&in_page_id =1770&in_a_source=

Perez, M. & Marwala, T. (2008). Stochastic Optimization Approaches for Solving Sudoku In Computer Science. *Neural Evolutionary Computing*.

Semeniuk, I. Stuck On You. (2005). *New Scientist*. 45-47.

Simonis, H. (2005). Sudoku as a Contraints Problem. *CP Workshop Modelling and Reformulating Constraint Satisfaction Problem, Spain*. 13-27.

Sopitan, G. A. (2012). Sudoku and Mathematics. *Courant Institute.* www.cs.nyu.edu/courses/summer08/ G22.2340-001/projects/GS_Sudoku.pdf.

Rhyd, L. (2007). Metahueristics Can Solve Sudoku Puzzles. *Journal of Heuristics*. 387-401.

Xiu, Q.D. & Yong D.L. (2011). A Novel Hybrid Genetic Algorithm for Solving Sudoku Puzzles. *Springer* 2011. 241-25