A
LP-I MINI PROJECT REPORT
ON

# SLIDER PUZZLE


Submitted By

| | |
|---|---|
| Gopal Panigrahi | 103 |
| Nikit Narkhede | 98 |
| Sourabh Pisal | 110 |

Guided By
Prof. G. B. Gadekar




SAVITRIBAI PHULE PUNE UNIVERSITY
In the academic year 2020-21 Department of Computer Engineering
Sanjivani Rural Education Society's
Sanjivani College of Engineering Kopargaon – 423 603

# SLIDER PUZZLE

## INTRODUCTION

The slide puzzle consists of a three by three board with eight numbered titles and a blank space. A tile adjacent to blank space can slide into the space. The objective is to figure out the steps needed to get from one configuration (which is an arbitrary arrangement of the tiles), for example,

| | | |
|---|---|---|
| 4 | 3 | 2 |
| 1 | 8 | 5 |
| 7 | 6 | |

To the goal configuration:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | |

Finding such solution of the general $n^2$-1 puzzle is known to be NP-complete, and furthermore, the best known algorithm for solving the eight puzzle optimally is A*.

## THEORY

**A* Algorithm**

The A stands for ”algorithm”, and the * indicates its optimal property.

The problem with breadth first search is that it eats too much resources and takes too long. One way to make it faster is to prioritize boards according to their distances to the goal board. This will avoid considering boards that are relatively far away from the goal board. Such a class of search algorithms is called heuristic-based searches.

We shall define a heuristic function H(S) that estimates the distance (i.e. the length of the path) from a current board S to the solution board:

$$H(S) = A(S) + E(S)$$

Where A(S) is the actual distance from the initial state to this board, and E(S) is the estimated distance from goal state. We will compute A(S) from the game tree we constructed so far, and we will define E(S) as the "Manhattan Distance" between S and the goal state.

**Manhattan distance**

The Manhattan distance heuristic is used for its simplicity and also because it is actually a pretty good underestimate (aka a lower bound) on the number of moves required to bring a given board to the solution board. We simply compute the sum of the distances of each tile from where it belongs, completely ignoring all the other tiles. For example, the Manhattan distance between "213540678" and "123456780" is 9:

```
2 1 3          1 2 3          1  2  3  4  5  6  7  8
5 4 0  ----->  4 5 6          -------------------------
6 7 8          7 8 0          1+1+0+1+1+3+1+1 = 9
```

This is so, because the tile "1" is 1 move away, the tile "2" is 1 move away, the tile "3" is 0 moves away, the tile "4" is 1 move away, the tile "5" is 1 move away, the "6" tile is 3 moves away, the "7" is 1 move away, and the "8" is 1 move away.

Note that we do not consider the empty space, as this would cause the Manhattan distance heuristic to not be an underestimate.

Another example:

```
6 4 7          1 2 3          1  2  3  4  5  6  7  8
8 5 0  -------> 4 5 6          -------------------------
3 2 1          7 8 0          4+2+4+2+0+3+4+2 = 21
```

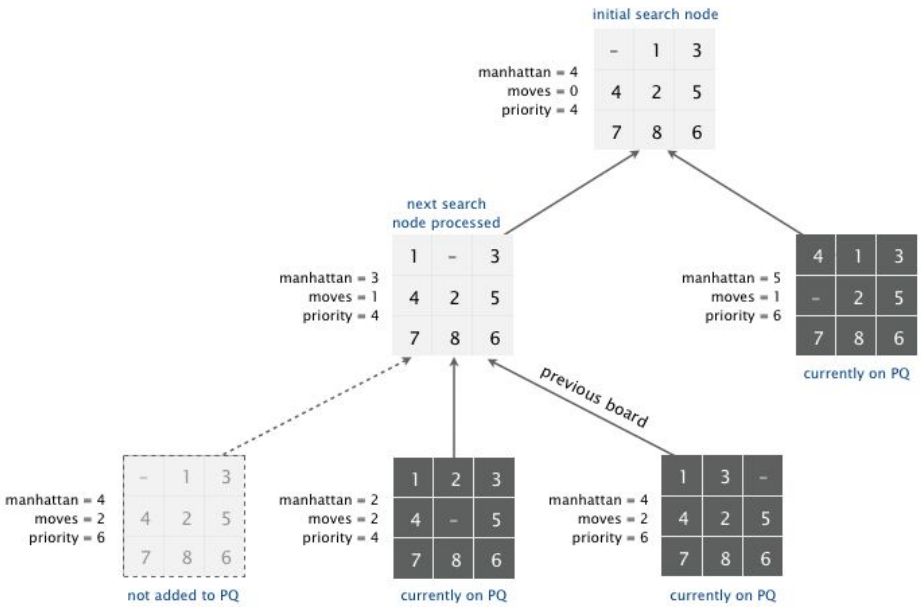


**Fig. Sample Game Tree**

## SCOPE

- Limited to 3*3 board
- Takes time for complex configurations
- Users can play or else the system can solve the puzzle for them.

## OBJECTIVE

- Representing complex problem in workable form
- Construct and analyse game trees
- Understand and implement Heuristic search
- Understanding the A* algorithm
- Using Manhattan Distance as Heuristic

## LITERATURE SURVEY

- The paper “Complete solution of eight puzzle problem using BFS in CUDA environment” by Masuma Sultana, Rathindra Nath Dutta, S. K. Setua of the year 2016, attempts at solving the slider puzzle problem using BFS. Their parallel algorithm is capable of providing a much faster solution using Compute Unified Device Architecture (CUDA). CUDA facility enables the best possible available computation power of GPU (Graphics Processing Unit). They present fast implementation of common graph operation like breadth-first search to find out complete solution of eight puzzle problem on the GPU using the CUDA programming model

- The Paper “Analysis of tree based search techniques for solving 8-puzzle problem” by Arpan Kumar Mishra and P C Siddalingaswamy of the year 2018. They compare the various uninformed and informed search techniques for finding the optimal solution. For, uninformed search they build the complete game tree whereas in informed search by using the heuristic search space is greatly reduced.

## ARCHITECTURE

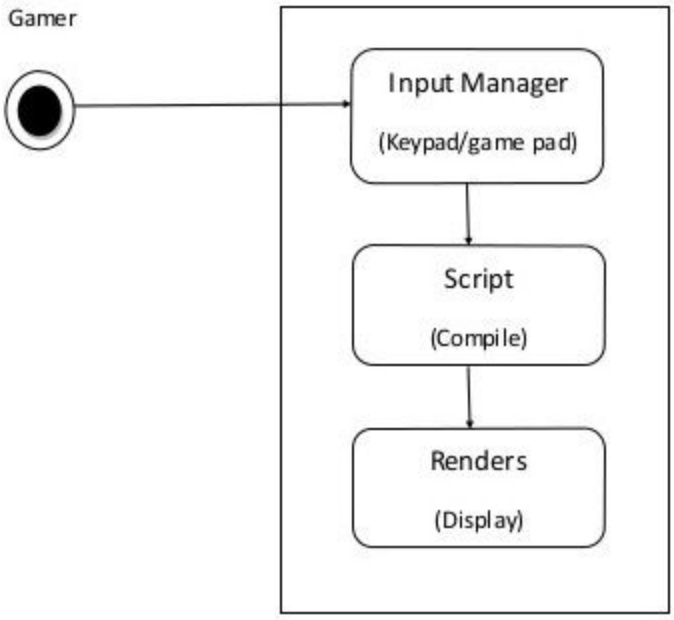


**Fig. Architecture**

## TECHNICAL SPECIFICATION

**Software requirement**

- Operating system (Linux/Windows)
- Browser (Chrome/Firefox)
- JavaScript
- CSS

**Hardware requirements**

- Any processor
- Minimum 1 GB of RAM

## EXPERIMENTAL ANALYSIS

Testing on following board configurations :

| Board | Number of Solution | Moves |
|---|---|---|
| 123405786 | 2 | RD |
| 123745086 | 4 | URRD |
| 123480765 | 5 | DLURD |
| 413726580 | 8 | LLUURDDR |
| 162530478 | 9 | LURDLLRR |
| 512630478 | 11 | LLURRDLLDRR |
| 126350478 | 13 | ULDLDRRULURDD |
| 356148072 | 16 | RRUULLDRDRUULDRD |
| 436871052 | 18 | URRULDDRULDLUURDRD |
| 302651478 | 21 | DRULDLURRDLLDRRULURDD |
| 012345678 | 22 | RDLDRRULLDRUURDDLLURRD |
| 503284671 | 23 | LDDRRULLDRRULLDRUULDDRR |

Here U- Up, D- Down, L- Left, R- Right and 0 indicates a blank space.
These are the optimal solutions for some of the boards given by the system.

## RESULT SNAPSHOTS

Here, some random board appears on screen.

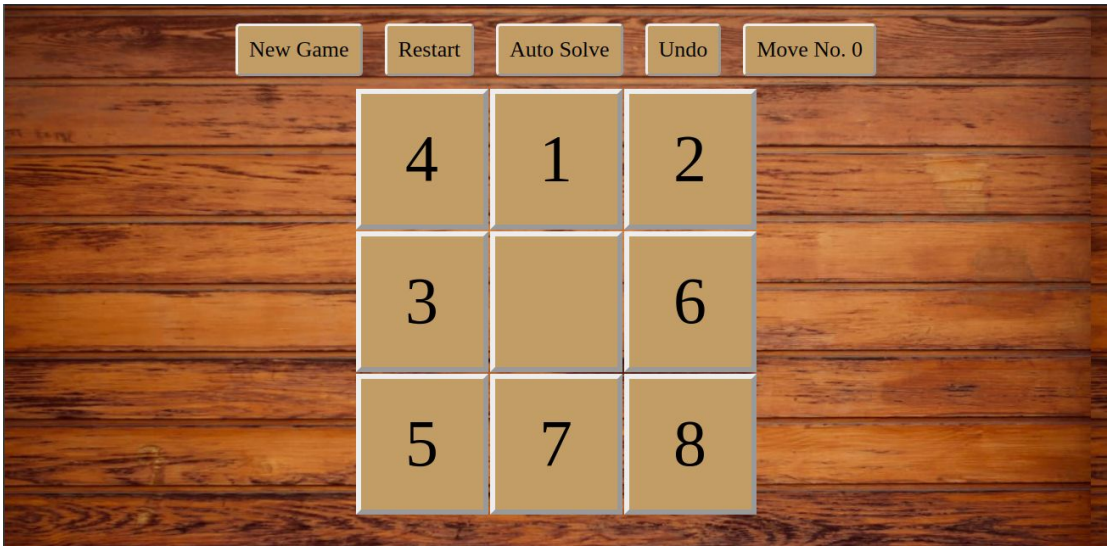


**Fig. Game Board**

To get the solution by system click on ‘Auto Solve’ button. After that the following result will be generated by system.

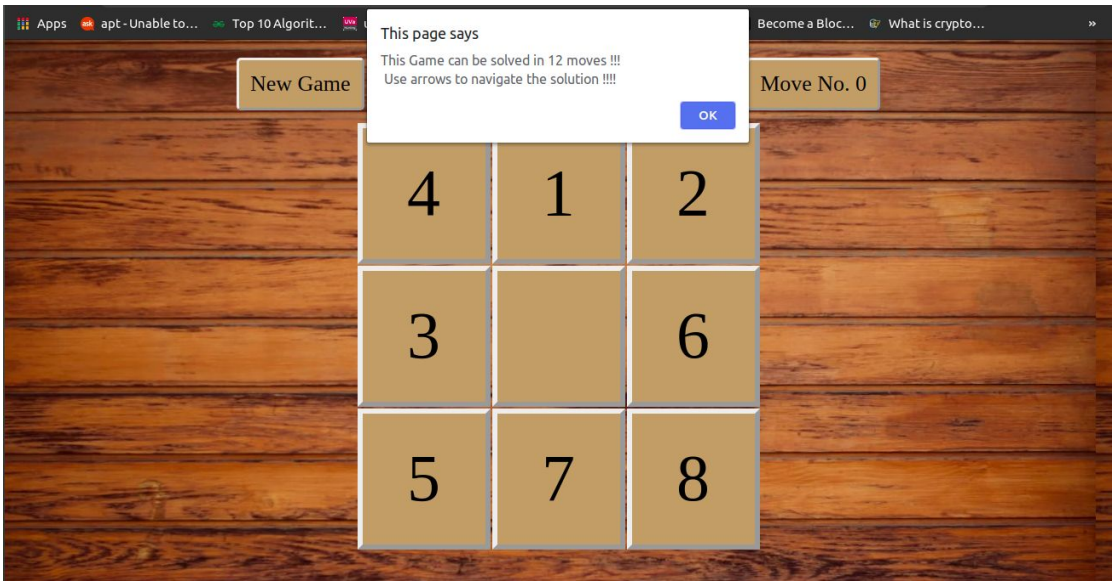


**Fig. Shows the minimum number of moves required to get solution**

The above figure shows the number of moves required to get the optimal solution which is generated by the system.

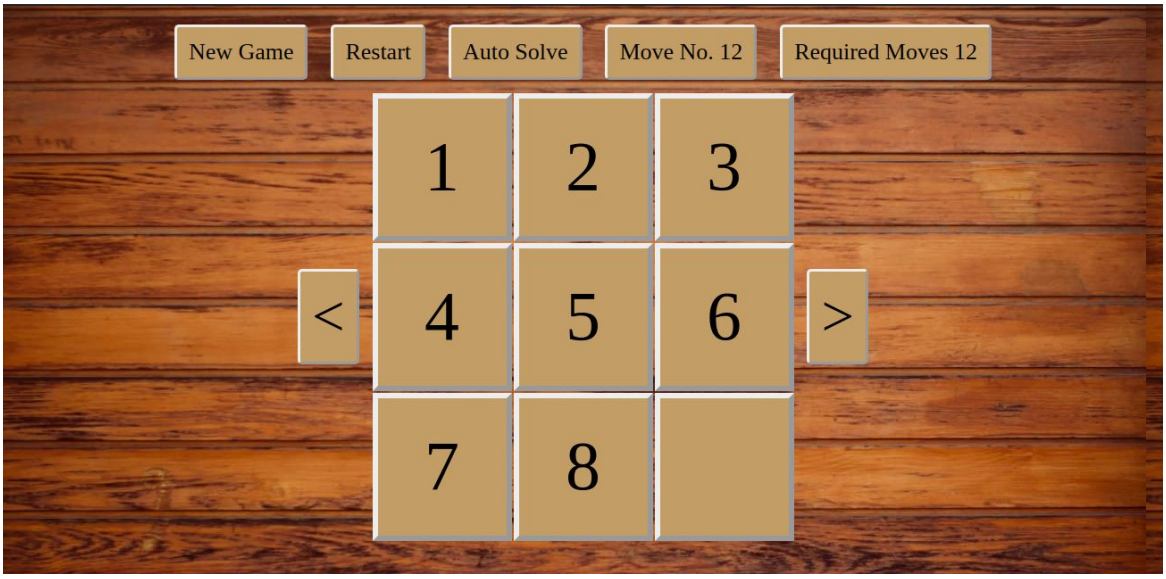


**Fig. Final result**

The above figure shows the final result i.e. after the 12 moves we get the solution for the puzzle.

## CONCLUSION

This project attempts at solving NP Complete problem by using A* informed search technique. Many games are very complex to solve, even by the human brain, thus being able to solve complex games might give insight into solving other real world problems.

## FUTURE SCOPE

- This can be extended to larger board size for eg NxN boards.
- This doesn't differentiate between solvable and unsolvable board configurations

## REFERENCE


- Coursera – Algorithms Part 1 by Princeton University
- M. Sultana, R. N. Dutta and S. K. Setua, "Complete solution of eight puzzle problem using BFS in CUDA environment," *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, Dhaka, 2015, pp. 333-337, doi: 10.1109/WIECON-ECE.2015.7443931
- A. K. Mishra and P. C. Siddalingaswamy, "Analysis of tree based search techniques for solving 8-puzzle problem," *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vellore, 2017, pp. 1-5, doi: 10.1109/IPACT.2017.8245012