



## **PROJECT REPORT ON 8 PUZZLE GAME**

**SUBJECT : ARTIFICIAL INTELLIGENCE**

**Submitted to :- Mr. Ishan Kumar**

**PROJECT SUBMITTED BY :**

**Aman Sehrawat : 11801364**

**Amrit Raj : 11801363**

**Mahima Singh: 11801291**

# **Contents**

**What is AI?**

- **What does it mean by AI based Game?**
- **8 puzzle Game**
- **Algorithms Used**
- **Snapshot of the Game**

# WHAT IS AI?



Artificial intelligence (AI) is wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence. AI is an interdisciplinary science with multiple approaches, but advancements in machine learning and deep learning are creating a paradigm shift in virtually every sector of the tech industry. AI systems will typically demonstrate at least some of the following behaviours associated with human intelligence: planning, learning, reasoning, problem solving, knowledge representation, perception, motion, and manipulation and, to a lesser extent, social intelligence and creativity

## WHAT DOES IT MEAN BY AI BASED GAME?

In video games, artificial intelligence (AI) is used to generate responsive, adaptive or intelligent behaviours primarily in non-player characters (NPCs) like human-like intelligence. Artificial intelligence has been an integral part of video games since their inception in the 1950s

## **The 8 puzzle GAME**

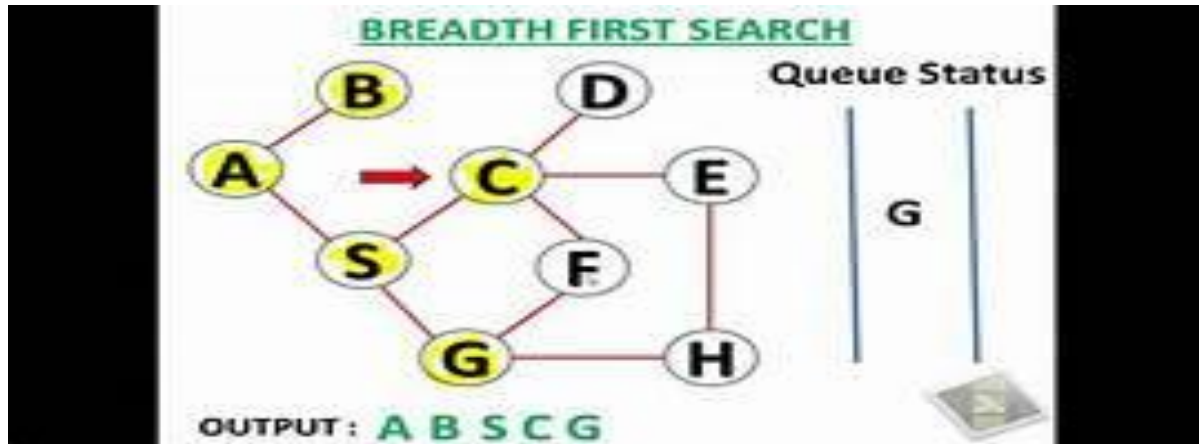
This project deals with the Production System for solving 8 Puzzle Problem. For solving the 8 puzzle problem we have implemented the Breadth First Search Algorithm to generate the initial state from the final state (reverse order). About the program, first we have to provide the initial matrix and then the final matrix. The Algorithm will search for the parent or initial state and when it does it will return the steps that were taken to reach it.

### **Introduction**

The 8-puzzle problem is a puzzle invented and popularized by Noyes Palmer Chapman in the 1870s. It is played on a 3-by-3 grid with 8 square blocks labelled 1 through 8 and a blank square. Our goal is to rearrange the blocks so that they are in order. We are permitted to slide blocks horizontally or vertically into the blank square.

# ALGORITHM USED

## Breadth First Search



Breadth-first search is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'), and explores all of the neighbour nodes at the present depth prior to moving on to the nodes at the next depth level.

## HISTORICAL Review

The puzzle was "invented" by Noyes Palmer Chapman, a postmaster in Canastota, New York. It was invented as 15 puzzle which consisted of 15 tiles from 1 to 15.

Two approaches were proposed to solve the 8 puzzle problem.

Hammingpriorityfunction - The number of blocks in the wrong position, plus the number of moves made so far to get to the state. Intuitively, a state with a small number of blocks in the wrong position is close to the goal state, and we prefer a state that have been reached using a small number of moves.

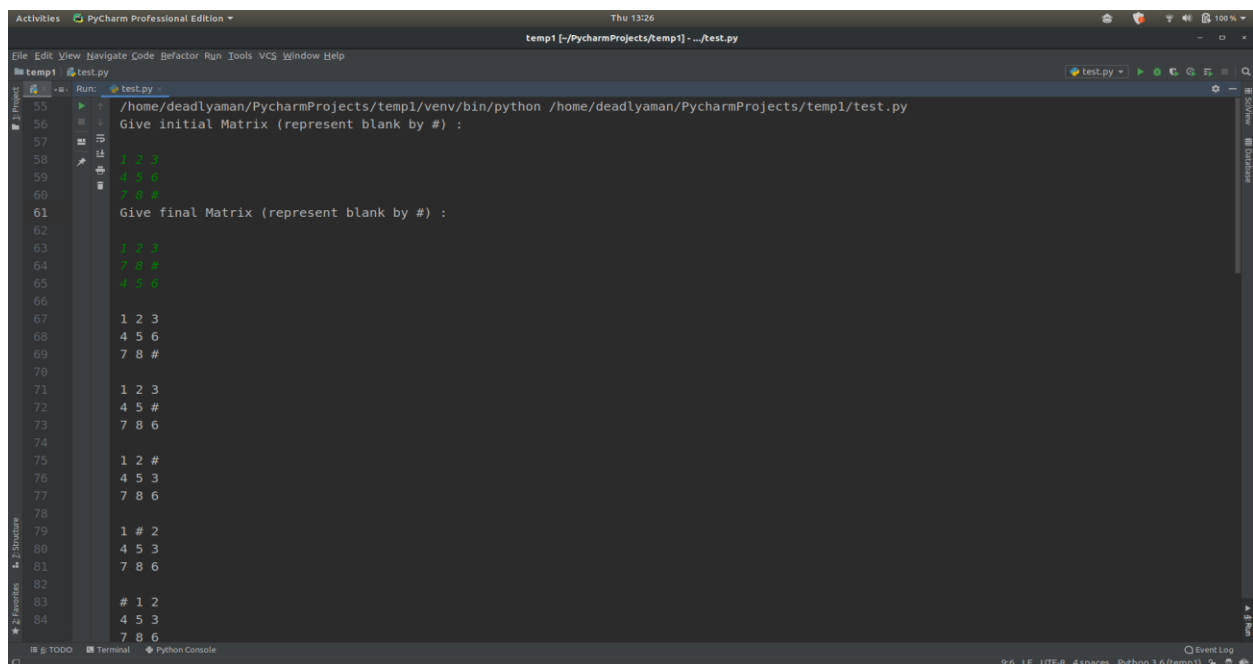
Manhattan priority function - The sum of the distances (sum of the vertical and horizontal distance) from the blocks to their goal positions, plus the number of moves made so far to get to the state.

## Methodology

For solving the 8 Puzzle we have proposed Breadth First Search as the key. Using BFS we will analyse the final or goal state of the puzzle and generate a number of states and out of these states we will have our goal state. For searching the initial state we are implementing BFS algorithm.

Although Breadth First Search is not the optimised algorithm for solving the 8 Puzzle problem. Best First Search is the optimized Algorithm through which we can solve the puzzle in a quicker time. Breadth First Search increases the complexity and takes more time than Best First Search.

## OUTPUT



```
Activities PyCharm Professional Edition Thu 13:26
temp1 [-/PycharmProjects/temp1] - .../test.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
temp1 test.py
Run: test.py
/home/deadlyaman/PycharmProjects/temp1/venv/bin/python /home/deadlyaman/PycharmProjects/temp1/test.py
Give initial Matrix (represent blank by #) :
1 2 3
4 5 6
7 8 #
Give final Matrix (represent blank by #) :
1 2 3
7 8 6
4 5 3
1 2 3
4 5 #
7 8 6
1 2 #
4 5 3
7 8 6
1 # 2
4 5 3
7 8 6
# 1 2
4 5 3
7 8 6
9.6 LF UTF-8 4spaces Python 3.6 (temp1) Event Log
```

```

1  # Project
2  # temp1
3  # test.py
4  # Run: test.py
5  #
6  # 55
7  # 56
8  # 57
9  # 58
10 # 59
11 # 60
12 # 61
13 # 62
14 # 63
15 # 64
16 # 65
17 # 66
18 # 67
19 # 68
20 # 69
21 # 70
22 # 71
23 # 72
24 # 73
25 # 74
26 # 75
27 # 76
28 # 77
29 # 78
30 # 79
31 # 80
32 # 81
33 # 82
34 # 83
35 # 84
36 #
37 #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #

```

```

1  # Project
2  # temp1
3  # test.py
4  # Run: test.py
5  #
6  # 55
7  # 56
8  # 57
9  # 58
10 # 59
11 # 60
12 # 61
13 # 62
14 # 63
15 # 64
16 # 65
17 # 66
18 # 67
19 # 68
20 # 69
21 # 70
22 # 71
23 # 72
24 # 73
25 # 74
26 # 75
27 # 76
28 # 77
29 # 78
30 # 79
31 # 80
32 # 81
33 # 82
34 # 83
35 # 84
36 #
37 #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #

```

From the above program we are able to conclude that we can solve the 8 puzzle problem by using Breadth First Search. The complexity is  $O(V + E)$ , where  $V$  is the number of vertices, and  $E$  is the number of Edges.

## **BIBLIOGRAPHY**

**1. [www Wikipedia.com](http://www.Wikipedia.com)**

**2. [www geeksforfreeks.com](http://www.geeksforfreeks.com)**