Hello Madhukar Temba here, thanks for downloading my Minesweeper AI game! Here is the explanation of how the AI works:

Method 1 (Simple Method):

1. If the tile number is equal to the number of hidden tiles adjacent to it then all the adjacent tiles are mines.
2. If a tile has the same number of flags around it as the number on the tile then all the remaining hidden tiles around it are mines.
3. If the above case does not exist then we calculate the arbitrary probability and select a square. Each cell which is connected to a numbered cell is given an arbitrary probability based on the 8 adjacent numbered cells.

Time complexity: O (8 x lengthofboard x widthofboard)

Method 2 (Brute Force):

1. The method of Brute Force is used by the AI agent to predict between safe squares and unsafe squares, by using probability.
2. The method of Brute Force gives an optimal solution by checking every cell combination in the perimeter and storing the valid combinations.
3. After that we can see in how many valid combinations is a cell a mine, by that we can calculate the probability of each perimeter cell being a mine.

Method 2.1: Improved method 2:

1. There are a lot of cases which method 2 checks which are not possible.
2. For example while checking from 00000 to 11111 we are sure that some of the in between cases are always not valid (example perimeter size = 5).
3. To improve on that, instead of calculating the whole sequence we calculate the combinations of mines starting from 0 to the number of tiles in the min (perimeter, number of mines left). This saves a lot of time.
4. The number of combinations can be significantly reduced.

5. But this method will still not give results if perimeter size > 32 as the combinations become too high and the integer limit is also exceeded. This method is good for solving small boards.

Time Complexity for method 2.1:

$$O\left(\sum_{r=0}^{\min(\text{perimeter size},\ \text{remaining mines})} {}^{n}C_{r} \right) \quad \begin{array}{l} n = \text{perimeter size} \\ r = \text{number of mines} \end{array}$$

Method 3: Stitching solutions together:

1. This method was invented by: https://www.youtube.com/watch?v=G2kd745uYuo

2. Instead of calculating the whole perimeter at once we instead calculate local segments of the perimeter and stitch them together to form a global result. It is a slightly complicated algorithm.
3. The time complexity of this is still exponential but is dependant of the number of squares being calculated at a time.
4. If the number of squares calculated at a time are less then this algorithm will run very quickly.
5. Its time complexity is O (2^(avg cells calculated at once) x size of perimeter/avg number of cells calculated at once)
6. This is the best method for solving large boards.

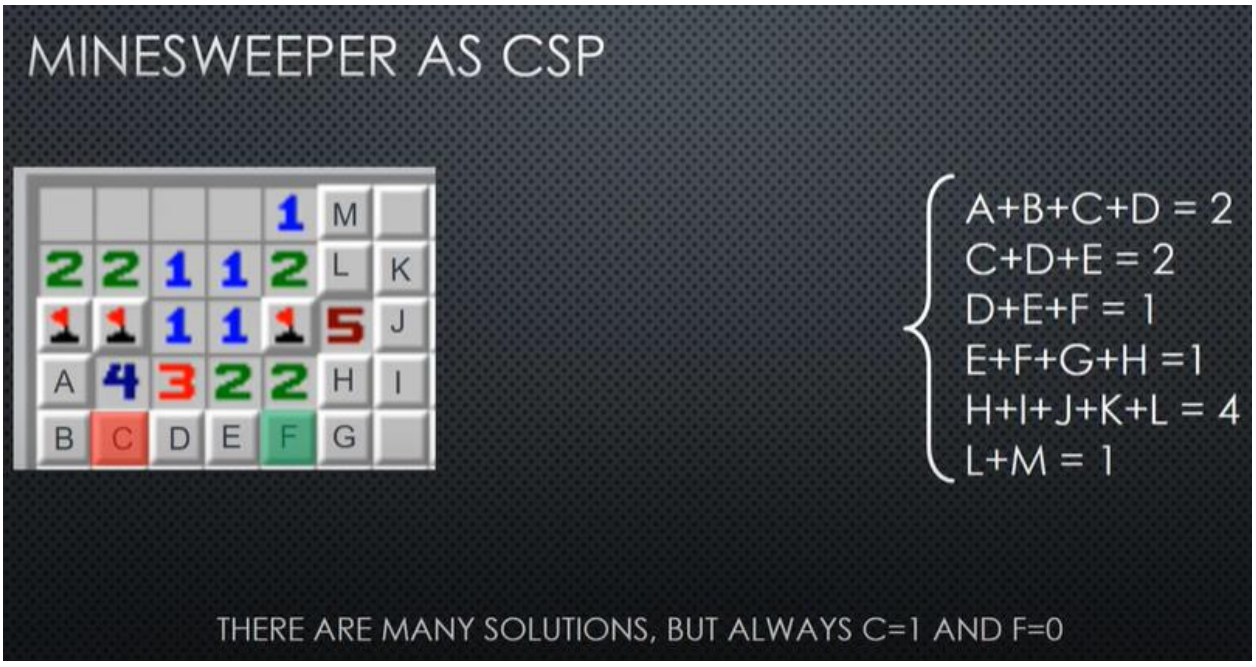
How it Works:

1. We scan and find out the cells in the perimeter of the uncovered cells.
2. Then we select a cell from the group and select all of its neighbouring cells which connect to it by a numbered square.
3. We calculate and store all the combinations of the mines in those cells and store each combination separately.
4. We include another neighbouring perimeter cell and remove the cells which are not connected to it by a numbered square.
5. The removed squares will be marked as calculated and then we only store the number of combinations where it was or was not a

mine and we club those scenarios where the ‘not calculated squares’ have the same combination.

6. We compute the combinations with the ‘not calculated squares’ and select the valid ones for the next round. Repeat till all squares in the perimeter are complete.

We can also solve the minesweeper game as constraint satisfaction problem (not implemented this method):

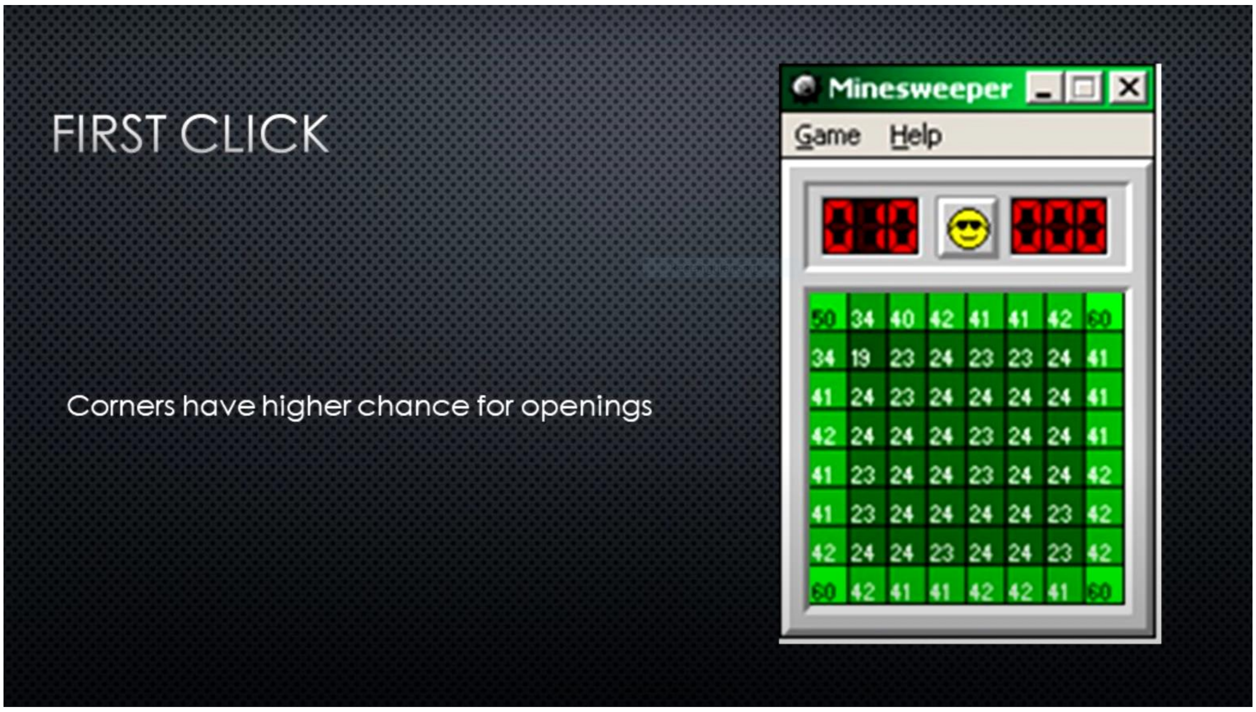


MINESWEEPER AS CSP

$$\begin{cases} A+B+C+D = 2 \\ C+D+E = 2 \\ D+E+F = 1 \\ E+F+G+H = 1 \\ H+I+J+K+L = 4 \\ L+M = 1 \end{cases}$$

THERE ARE MANY SOLUTIONS, BUT ALWAYS C=1 AND F=0

We should also use the fact that the corner cells and edge cells have a higher probability of being safe, this will improve our chances of winning:



FIRST CLICK

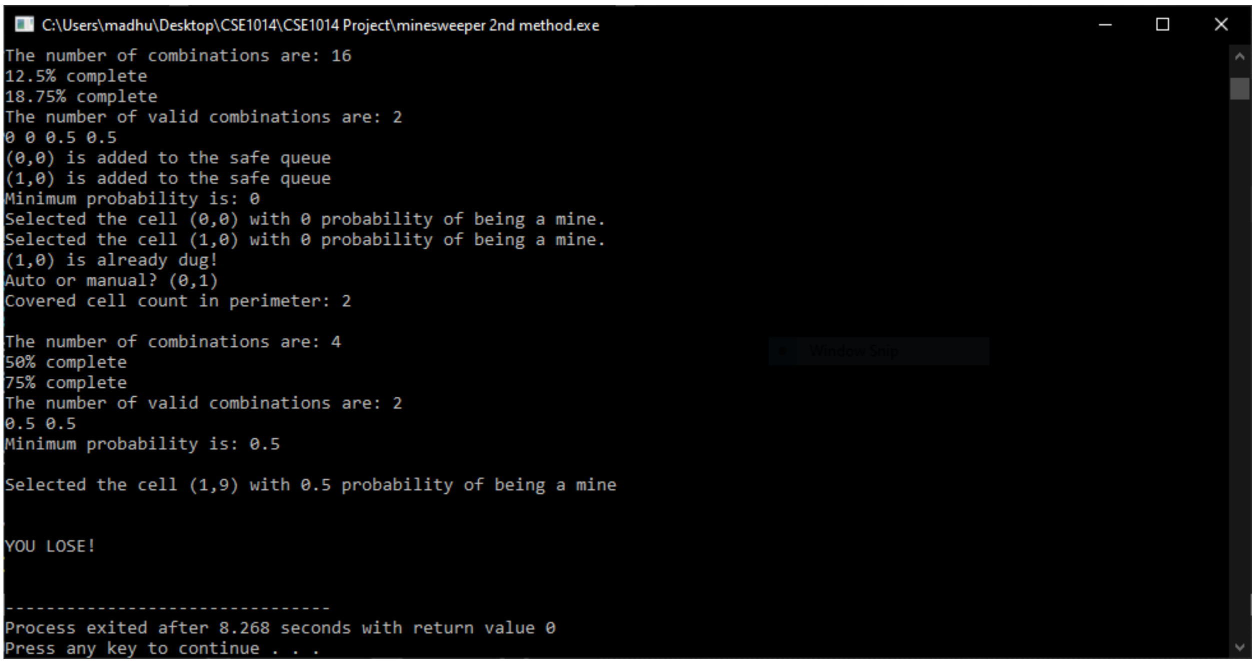
Corners have higher chance for openings

60	34	40	42	41	41	42	60
34	19	23	24	23	23	24	41
41	24	23	24	24	24	24	41
42	24	24	24	23	24	24	41
41	23	24	24	23	24	24	42
41	23	24	24	24	24	23	42
42	24	24	23	24	24	23	42
60	42	41	41	42	42	41	60

Also minesweeper is a NP-complete problem meaning it cannot be solved in polynomial time: In 2000, Richard Kaye published a proof that it is NP-complete to determine whether a given grid of uncovered, correctly flagged, and unknown squares, the labels of the foremost also given, has an arrangement of mines for which it is possible within the rules of the game. The argument is constructive, a method to quickly convert any Boolean circuit into such a grid that is possible if and only if the circuit is satisfiable; membership in NP is established by using the arrangement of mines as a certificate. If, however, a minesweeper board is already guaranteed to be consistent, solving it is not known to be NP-complete, but it has been proven to be co-NP-complete. In the latter case, however, minesweeper exhibits a phase transition analogous to k-SAT: when more than 25% squares are mined, solving a board requires guessing an exponentially-unlikely set of mines.

Kaye also proved that the infinite Minesweeper is Turing-complete.

Even the best player can be stuck with a 50% chance of winning:


C:\Users\madhu\Desktop\CSE1014\CSE1014 Project\minesweeper 2nd method.exe

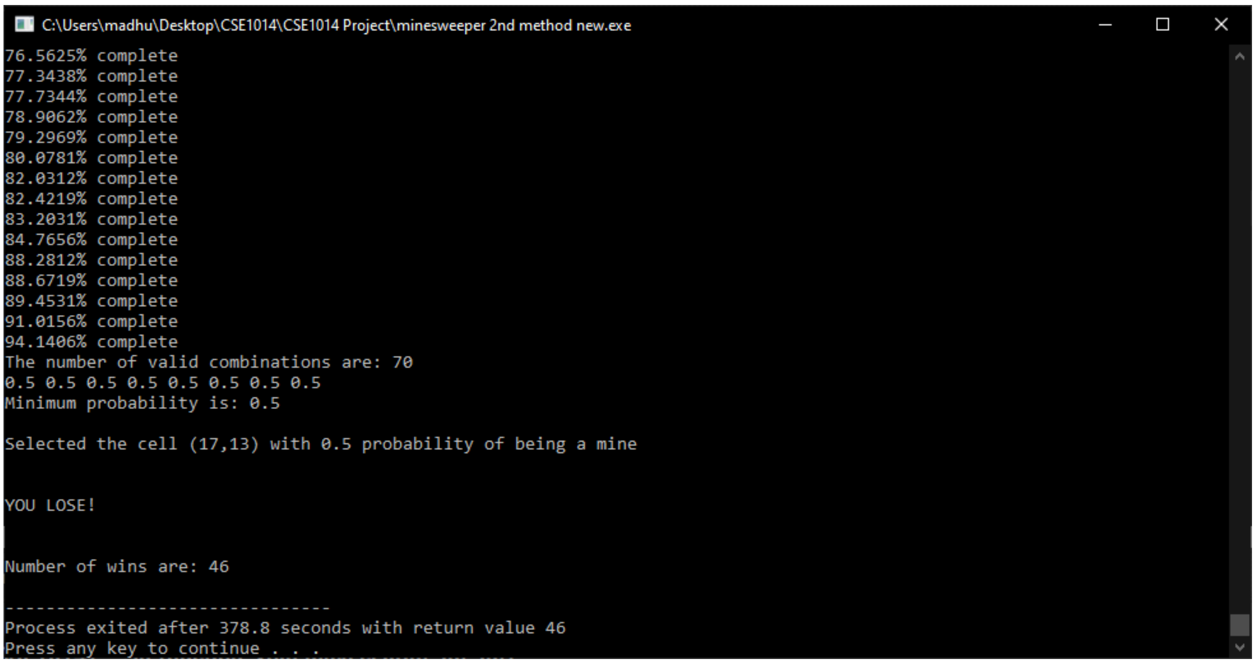
```
The number of combinations are: 16
12.5% complete
18.75% complete
The number of valid combinations are: 2
0 0 0.5 0.5
(0,0) is added to the safe queue
(1,0) is added to the safe queue
Minimum probability is: 0
Selected the cell (0,0) with 0 probability of being a mine.
Selected the cell (1,0) with 0 probability of being a mine.
(1,0) is already dug!
Auto or manual? (0,1)
Covered cell count in perimeter: 2

The number of combinations are: 4
50% complete
75% complete
The number of valid combinations are: 2
0.5 0.5
Minimum probability is: 0.5

Selected the cell (1,9) with 0.5 probability of being a mine


YOU LOSE!


--------------------------------
Process exited after 8.268 seconds with return value 0
Press any key to continue . . .
```


C:\Users\madhu\Desktop\CSE1014\CSE1014 Project\minesweeper 2nd method new.exe

```
76.5625% complete
77.3438% complete
77.7344% complete
78.9062% complete
79.2969% complete
80.0781% complete
82.0312% complete
82.4219% complete
83.2031% complete
84.7656% complete
88.2812% complete
88.6719% complete
89.4531% complete
91.0156% complete
94.1406% complete
The number of valid combinations are: 70
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
Minimum probability is: 0.5

Selected the cell (17,13) with 0.5 probability of being a mine


YOU LOSE!


Number of wins are: 46

--------------------------------
Process exited after 378.8 seconds with return value 46
Press any key to continue . . .
```

Hope you enjoy this game! Thanks for downloading!

References:

1. https://www.youtube.com/watch?v=cGUHehFGqBc
2. https://www.youtube.com/watch?v=G2kd745uYuo
3. https://www.youtube.com/watch?v=uvRQUWxOHqo
4. https://www.youtube.com/watch?v=btFwzZVFv0M
5. https://en.wikipedia.org/wiki/Minesweeper_(video_game)