

Expert Minesweeper Player

Chad Bloxham
chadbloxham10@gmail.com

1 MyAI.py Purpose

The program MyAI.py was written as part of the final project for the graduate course CS271P at UC Irvine. It is a Minesweeper player capable of completing Expert level boards (16 x 30 board size containing 99 mines). It maintains an internal representation of the board and identifies safe tiles (those not containing mines) and mine tiles using the four methods described in the Program Overview section.

2 Minesweeper Rules

The player is given a board of covered tiles. Some tiles are empty and others contain mines. The player wins the game by uncovering all tiles which do not contain a mine. The player is allowed to place flags on tiles known to be mines. An uncovered empty tile reveals a hint number indicating the number of mines in the surrounding tiles.

Our player cannot lose on the first move because the program which creates the board provides a starting position which is known to be a safe tile.

3 Run Instructions

REQUIRES: Python 3 (to be run from a terminal)
Modules: Numpy, Itertools

1. Download the Minesweeper_Student repository from https://gitlab.ics.uci.edu/ai-projects/Minesweeper_Student.
2. Download this repository and replace the MyAI.py file found in the Minesweeper_Student repository with the MyAI.py file found in this one.
3. In a terminal, enter the directory Minesweeper_Student/WorldGenerator and execute the bash script generateTournament.sh to generate text files which represent the Minesweeper boards MyAI.py will play on. It will generate the directory Minesweeper_Student/WorldGenerator/Problems which will contain 1000 Beginner boards (8 x 8, 10 mines), 1000 Intermediate boards (16 x 16, 40 mines), and 1000 Expert boards (16 x 30, 99 mines):

```
$ ./generateTournament.sh
```

4. Enter the directory Minesweeper_Student/Minesweeper_Python/src. Run Main.py on a text file using -f followed by the file name:

```
cbloxham@odin 21:30:06 ~/Minesweeper_Student/Minesweeper_Python/src
$ python3 Main.py -f ~/Minesweeper_Student/WorldGenerator/Problems/Expert/Expert_world_2.txt
WORLD COMPLETE
```

Figure 1: MyAI.py completing an expert world.

Including the -d flag displays the board with each move made by MyAI.py:

```

Number of mines: 99
Number of flags left: 54
16| . . . . . . . . . . . . . . ? 2 1 2 3 ? ? ? 1 0 0 1 . . . .
15| . . . . . . . . . . . . . . ? 2 1 ? ? 3 3 2 1 0 0 1 . . . .
14| . . . . . . . . . . . . . . 3 1 2 3 3 1 0 0 0 0 0 1 1 . . . .
13| . . . . . . . . . . . . . . 2 0 1 ? 1 0 0 1 1 1 0 1 . . . .
12| . . . . . . . . . . . . . . 2 0 1 1 1 0 0 1 ? 2 1 2 . . . .
11| 1 1 . . . . . . . . . . . . . 3 1 0 1 1 2 1 2 2 4 ? 4 3 . . .
10| 0 1 . . . . . . . . . . . . . ? 2 0 1 ? 3 ? 2 2 ? ? ? . . 3 .
9 | 0 1 . . . . . . . . . . . . . ? 2 0 1 2 ? 2 2 ? 5 ? ? 3 1 2 ?
8 | 0 2 . . . . . . . . . . . . . 3 1 0 0 1 2 3 3 3 ? 3 2 1 0 1 1
7 | 0 1 . . . . . . . . . . . . . ? 2 0 0 0 0 1 ? ? 2 1 1 1 1 0 0
6 | 0 2 3 . . . . . . . . . . . . . 3 1 2 1 1 1 2 2 1 0 0 1 ? 1 0 0
5 | 0 1 ? . . . . . . . . . . . . . 2 ? 3 ? 2 0 0 0 0 0 0 0 1 1 1 0 0
4 | 0 2 2 . . . . . . . . . . . . . 2 3 3 5 ? 3 2 1 1 0 0 0 0 1 1 2 1
3 | 0 1 ? . . . . . . . . . . . . . 3 ? ? 3 ? 2 ? 1 0 0 0 0 1 ? 2 ?
2 | 1 2 3 . . . . . . . . . . . . . . 3 3 4 4 4 2 1 0 0 1 1 2 2 4 3
1 | 1 ? . . . . . . . . . . . . . . 1 1 ? ? ? 1 0 0 0 1 ? 1 1 ? ?
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Tiles covered: 267 | Flags left: 54 | Last action: FLAG on (3, 3)
Press ENTER to continue...

```

Figure 2: The state of an Expert board as MyAI.py plays. Question marks correspond to flags.

```

Number of mines: 99
Number of flags left: 0
16| 0 0 0 1 B 1 0 0 2 B 3 1 0 2 B 2 1 2 3 B B B 1 0 0 1 1 2 B 1
15| 2 2 1 1 1 1 1 2 4 B B 3 2 4 B 2 1 B B 3 3 2 1 0 0 1 B 2 1 1
14| B B 2 1 1 1 2 B B 5 B 4 B B 3 1 2 3 3 1 0 0 0 0 0 1 1 1 0 0
13| 4 B 2 1 B 2 4 B 4 3 B 3 3 B 2 0 1 B 1 0 0 1 1 1 0 1 1 1 0 0
12| B 2 1 1 1 2 B B 3 3 3 3 3 2 2 0 1 1 1 0 0 1 B 2 1 2 B 2 1 1
11| 1 1 0 0 0 1 2 3 B 2 B B 3 B 3 1 0 1 1 2 1 2 2 4 B 4 3 3 B 1
10| 0 1 1 1 0 0 1 3 3 3 2 2 4 B B 2 0 1 B 3 B 2 2 B B B 3 B 3 2
9 | 0 1 B 1 0 0 1 B B 2 1 0 2 B B 2 0 1 2 B 2 2 B 5 B B 3 1 2 B
8 | 0 2 2 2 0 0 2 3 4 B 2 1 2 3 3 1 0 0 1 2 3 3 3 B 3 2 1 0 1 1
7 | 0 1 B 3 2 1 1 B 3 3 B 2 3 B 2 0 0 0 1 B B 2 1 1 1 1 1 0 0
6 | 0 2 3 B B 2 2 2 2 B 3 4 B B 3 1 2 1 1 1 2 2 1 0 0 1 B 1 0 0
5 | 0 1 B 4 3 3 B 1 2 4 B 4 B 3 2 B 3 B 2 0 0 0 0 0 0 1 1 1 0 0
4 | 0 2 2 3 B 2 1 1 2 B B 3 2 2 3 3 5 B 3 2 1 1 0 0 0 0 1 1 2 1
3 | 0 1 B 3 2 1 0 0 2 B 3 1 1 B 3 B B 3 B 2 B 1 0 0 0 0 1 B 2 B
2 | 1 2 3 B 1 0 0 0 1 1 2 1 3 3 B 3 3 4 4 2 1 0 0 1 1 2 2 4 3
1 | 1 B 2 1 1 0 0 0 0 0 1 B 2 B 2 1 1 B B B 1 0 0 0 1 B 1 1 B B
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Tiles covered: 0 | Flags left: 0 | Last action: LEAVE on (2, 2)
WORLD COMPLETE

```

Figure 3: The completed Expert board (with mines revealed).

Entering a directory after -f will run Main.py on all the text files in that directory and report the performance of MyAI.py:

```

cbloxham@odin 21:30:19 ~/Minesweeper_Student/Minesweeper_Python/src
$ python3 Main.py -f ~/Minesweeper_Student/WorldGenerator/Problems/Beginner/
-----Your agent's results:-----
Beginner: 799   Intermediate: 0           Expert: 0
Cumulative Score: 799

```

Figure 4: MyAI.py completing 799 out of 1000 Beginner boards.

NOTE Refer to the file Minesweeper_Student/Minesweeper_Student_Manual.pdf for more detailed instructions on the various optional arguments Main.py can handle. It also gives instructions on how to create your own text files representing Minesweeper boards.

4 Program Overview

NOTE For a more detailed description of the methods used to identify safe and mine tiles, refer to the comments in MyAI.py.

Given the current state of the board, MyAI.py finds safe tiles and/or mine tiles using one of four methods. A method is only called if needed i.e. if none of the previous methods yielded a move.

1. **Hint number comparisons.** If the hint number for an uncovered tile is 0 and there are covered tiles surrounding it, these tiles can be uncovered. If the hint number for an uncovered tile is equal to the number of flagged tiles surrounding it, then all surrounding tiles which are covered and non-flagged can be uncovered. If the hint number for an uncovered tile is equal to the number of covered tiles surrounding it then all surrounding tiles which are covered and not already flagged can be flagged.

2. **Solve mine equations.** Create mine equations for pairs of uncovered tiles touching each other in the same row or same column. Create a reduced equation by subtracting one equation from the other. Calculate the maximum and minimum possible values which could be created from the LHS of the reduced equation considering each mine variable can only take a value of 0 or 1. If the max. or min. equals the RHS of the equation, mines and/or safe tiles have been discovered.

3. **Calculate mine probabilities.** Generate all possible configurations for the mines which have yet to be identified. For each of these configuration, check if it satisfies all relevant mine equations. For each position, keep a count of how many times a mine appears in that position in a satisfiable configuration. Divide each count by the total number of satisfying configurations to get “mine probabilities” for each position. Tiles with a mine probability of 1.0 are guaranteed to be mines while tiles with a mine probability of 0.0 are guaranteed to be safe.

If the number of covered tiles is equal to the number of mines, we have won and will LEAVE. Otherwise,

4. **Guess.** Iterate through the board and uncover the first covered tile identified.

If multiple safe tiles and/or multiple mine tiles are identified from one of the methods described above, all of these moves will be performed before any methods are called again. The current state of the board is updated with each move.

5 Performance

MyAI.py was run on 5000 Beginner, Intermediate, and Expert boards to give the following results:

Trial	Beginner	Intermediate	Expert
1	807	777	191
2	830	769	188
3	823	757	187
4	832	761	184
5	799	770	204
Avg	818.2	766.8	190.8
Win %	81.8%	76.7%	19.1%

Figure 5: MyAI.py wins 81.8% of the time on Beginner boards, 76.7% of the time on Intermediate boards, and 19.1% of the time on Expert boards.