Minesweeper Final AI Report

Team name___TopPlayer____
Member #1 (name/id)_______Yaoming Zhan/18205717_________ ______________
Member #2 (name/id)________Jiada Ye/32964504_________________________

I. Minimal AI

I.A. Briefly describe your Minimal AI algorithm. What did you do that was fun, clever, or creative?

The algorithm for Minimal AI is very simple. It starts with the uncovered tile, gradually expands to its neighbor tiles to find the potential mine. Based on the number returned from the previous action, we can tell if a tile is safe of not. If a tile is safe, add itself and its neighbors into a list of safe tiles, else add them into potential mine tiles.

Then the AI will poll from our list of safe tiles, uncover them, then poll from unsafe tiles and flag them. When we run out of actions, leave the game (since there's only one mine).

I.B Describe your Minimal AI algorithm's performance:

Board Size	Sample Size	Score	Worlds Complete
5x5	1000	380	380
8x8	N/A	N/A	N/A
16x16	N/A	N/A	N/A
16x30	N/A	N/A	N/A
Total Summary	1000	380	380

II. Final AI

II.A. Briefly describe your Final AI algorithm, focusing mainly on the changes since Minimal AI:

First, the algorithm still finds all tiles that are guaranteed to be safe/unsafe and add them into a queue with UNCOVER/FLAG actions. Then, when there's no enough information for us to determine tiles' status easily, run formula(), and formula2() and they will find possible other actions available.

Formula() was applied to deal to two tiles with the same given bomb number and for the surrounded unknown tiles, one was the subset of the other. That inferred that the disjoint tiles were guaranteed safe.

Formula2() was to deal with 2 tiles, tile a with 1 unknown bomb and tile b with 2 unknown bombs. If the disjoint set of surrounded tiles for tile b had only one tile, that tile must be a mine.

However, there were still many complex cases are dead cases in many games. A good guessing algorithm was applied. In simple words, the tile with the smallest mine number and the most unknown neighbor would be best choice for the center tile. The tile to be uncovered could be drawn out from its neighbors.

The finalturn() was used to deal with the end of the game in which the total mine number was equal to the discovered mine number or the remained unknown tile number was equal to the left mine number.

II.B Describe your Final AI algorithm's performance:

Board Size	Sample Size	Score	Worlds Complete
5x5	N/A	N/A	N/A
8x8	1000	786	786
16x16	1000	1522	761
16x30	1000	765	255
Total Summary	3000	3073	1802

III. In about 1/4 page of text or less, provide suggestions for improving this project (*this section does NOT count as past of your two-page total limit.*)

There were many fundamental functions supporting the algorithms. Valid() was used to determine whether the tile was in the board. Many functions were used to find many kinds of neighbors like unknown neighbors and mine neighbors. updateBoard() was applied through every movement. Some functions were used to search the border tiles, flagged tiles or safe tiles in the whole board.

The process of flagging/uncovering is divided into 3 part: first, process tiles surrounded by the center tile according to the bomb number of the center tile; second, process the tiles surrounded by multiple tiles based on the bomb number of these tiles; third, take a logical guess of uncovering an unknown tile.

At the phase of Minimal AI 5x5 board, we only used the specific bomb number 0 of the center tile to uncovering nearby tiles and do so iteratively. That was straightforward and simple enough to find out only 1 mine.

At the second phase, the first part mentioned above was brought to deal with the ambient tiles of the center tile of which mine number was identical to the nearby flag number or unknown tile number. All the actions were added into an action queue. For the remaining unknown tiles, the random uncovering mechanism was applied and the result was great enough. for 8x8 board game, about half of 1000 cases was successful and 20% for 16x16 board.

For the expert level of Minesweeper, the difficulty rises sharply, and many complex algorithms were necessary to overcome at least 20% of 1000 games. For us, formula(), formula2() and randomTile() were applied. More tile relationships were needed to ensure the information of surrounding tiles, like the unknown surrounding mine number larger than 2. Some relationships between tiles for 3 or more are expected to explore..