Minesweeper Final AI Report

Team name: Slatty
Member #1 (name/id): Tristan Bradfield
tbradfie
64453924
Member #2 (name/id): N/A

I. Minimal AI

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

I didn't have a minimal AI functioning by the specified due date, but my initial AI used only simple logic to complete worlds. It would first check for any guaranteed safe moves, like zeroes or spaces that could be proven to not be a mine by surrounding spaces. These spaces would be added to a list of "safe spaces," continuing to uncover spaces in this list until it is empty. Any uncovered spaces are added to a list of "available moves", which are checked to see if simple logic can be used to determine any surrounding safe spaces/mines. If any spaces are determined to be a mine, they are added to a list of "mine spaces," which is checked immediately after checking for any safe spaces. Each call to getAction would repeat this process. An internal 2-d array was used to keep track of the board, initialized as a grid of '.' characters, which are updated with '?' characters for flags and numbers for uncovered spaces.

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

Board Size	Sample Size	Score	Worlds Complete
5x5	N/A	N/A	N/A
8x8	100	23	23
16x16	N/A	N/A	N/A
16x30	N/A	N/A	N/A
Total Summary	100	23	23

II. Final AI

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

After constructing my initial AI that only used simple logic, I attempted to add more complex model checking to go beyond what simple logic is capable of, though I ran into difficulties figuring out the best way to properly implement this. The logic I added would select a move from the remaining "available spaces" after all other checks (mines and safe spaces) had been performed and all available spaces had been checked using the simple logic of the Minimal AI. If no further mine/safe spaces could be determined, a move likely to be a safe space would be randomly chosen from the "available spaces" list. Expert worlds continued to give me trouble, with the AI only ever reaching 2% success rates with these levels but there was improved performance for beginner and intermediate boards, consistently passing close to half of all intermediate worlds and slightly over half of beginner.

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

Board Size	Sample Size	Score	Worlds Complete
5x5	1000	1000	1000
8x8	1000	556	556
16x16	1000	860	430
16x30	1000	36	12
Total Summary	4000	2416	1998

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

The section I could most improve in is in the model checking of tiles after simple logic can no longer be effective. My method for determining which tiles are likely to be safe is extremely rudimentary and using the U and C method described in the Minesweeper Ideas slides would greatly increase the efficacy of my AI, likely boosting the score of the Intermediate and Expert worlds by a significant amount. Specifically, keeping track of all bomb probabilities in a separate local array as I saw some other students implementing during class discussions would significantly help with this process.