**Minesweeper Final AI Report**

**Team name\_\_\_Mineithon\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Member #1 (name/id)\_\_Clyton Dantis / 53649044\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Member #2 (name/id)\_\_Sunnyhith Kanakanti / 74403964\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**I. Minimal AI**

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

Keep uncovering all cells surrounding a zero. When you encounter a one, do not open its surrounding cell, walk past them. Repeat step 1 until only one cell remains. The remaining cell will be a mine.

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

|  |  |  |  |
| --- | --- | --- | --- |
| Board Size | Sample Size | Score | Worlds Complete |
| 5x5 | 1000 | 1000 | 1000 |
| 8x8 | NA | NA | NA |
| 16x16 | NA | NA | NA |
| 16x30 | NA | NA | NA |
| Total Summary | 1000 | 1000 | 1000 |

**II. Final AI**

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

The algorithm consists of multiple layers. In the first layer, we keep uncovering zeros and storing neighboring tiles in a ‘Safe to visit’ queue. If the queue is not empty we pop a tile from it and return uncover action on it.

If the first layer fails, we proceed to the second layer where we analyze opened tiles. If a tile has exactly as many neighbors as the tile number, then mark all those neighbors as mines and reduce the count of the neighboring tiles. If new zero tiles get discovered then mark their neighbors as ‘safe’

If the second layer fails, we proceed to the third layer where we utilize patterns. We have coded 1-2-1, 1-2-2-1, 1-2, 2-1, 1-1 patterns in our algorithm to identify mines quickly. This layer again may discover new zeros which will help accelerate our search in the first layer.

In the third layer, we model the minesweeper board state as a set of linear algebra equations and reduce the system to row-reduced echelon form. Then we perform analysis on the board using the following steps mentioned [here](https://massaioli.wordpress.com/2013/01/12/solving-minesweeper-with-matricies/comment-page-1/)

Set the maximum bound and minimum bound to zero For each column in the row (not including the augmented column of course)

if the number is positive add it to the maximum bound and if it is negative then add it to the minimum bound.

If the augmented column value is equal to the minimum bound then    All of the negative numbers in that row are mines and all of the positive values in that row are not mines else

if the augmented column value is equal to the maximum bound then    All of the negative numbers in that row are not mines and all of the positive values in that row are mines.

This layer is present in the la branch.

In the last layer, all covered/unopened tiles are candidate mines. The opened neighbors of these tiles will cast votes. We choose the mine having the minimum votes as SAFE and uncover it. This layer is less reliable in detecting mines but is used as a last resort.

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

|  |  |  |  |
| --- | --- | --- | --- |
| Board Size | Sample Size | Score | Worlds Complete |
| 5x5 | 1000 | 1000 | 1000 |
| 8x8 | 1000 | 1400 | 700 |
| 16x16 | 1000 | 1324 | 630 |
| 16x30 | 1000 | 374 | 134 |
| Total Summary | 4000 | 4794 | 2464 |

**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.*)**

The coding clinics should be focused on AI techniques that can be applied to the project.

The course should be organized in a way that the topics that can help us solve the project are taught earlier. For eg: 3-arc consistency technique could be applied to minesweeper but the topic was covered in the last week of the course. Each week’s coding clinic could also focus on small problems where we apply the algorithms taught in class that week. This will not only help in our project but also reinforce the concepts taught in class