AI LAB 6 REPORT

Group 2

Algorithms

We have used the Min-Max Algorithm with Alpha Beta pruning to reduce the search space by reducing the number of branches we visit using the MinMax recursion tree.

The Algorithm works as follows

1. Starts with all the possible moves at a currentNode.
2. The finds all the possible moves of the next player if a move from the previous step is made.
3. It repeats until no possible move is left or the max search depth is reached.

Heuristics

We have considered two heuristic functions.

1. **Static Heuristic**

   We searched the internet and found a paper that had a heuristic that the other university researchers found out.
   This gives values to all possible discs and adds or subtracts the values based on which player is playing.

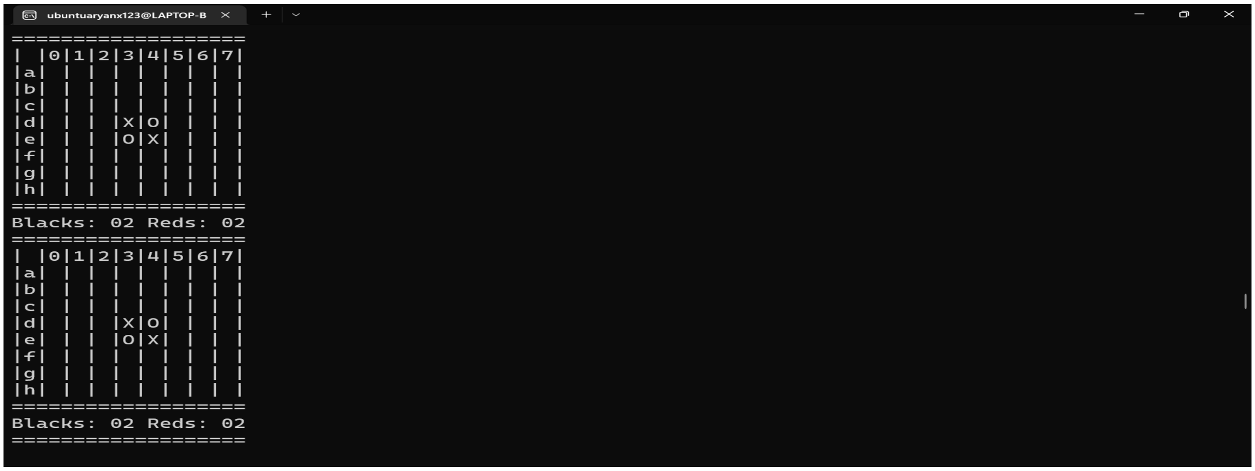
2. **Living Heuristic**

   We searched for another article and found out that we can use different heuristic values depending on these parameters

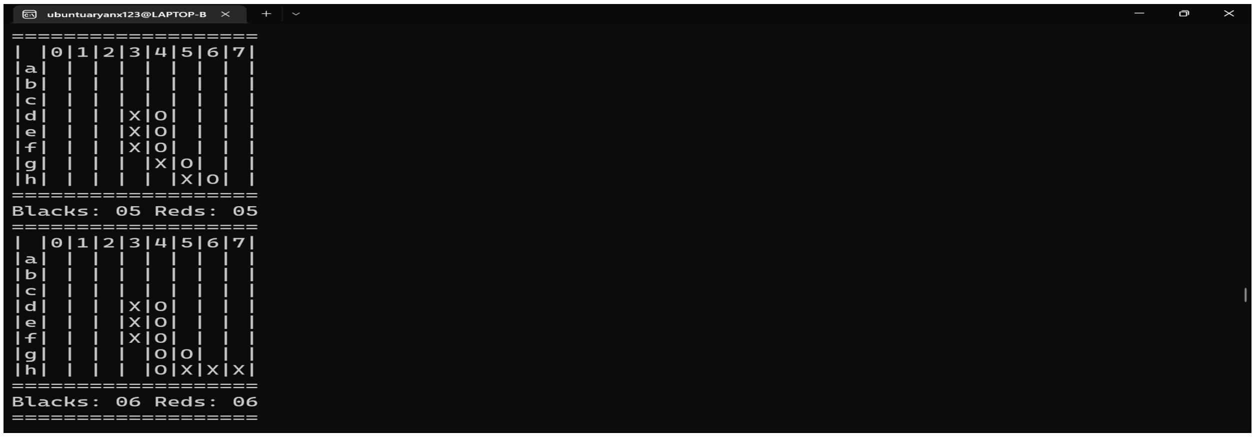
   1. **Disc Balance**: The number of discs each side has.
   2. **Possible Moves**: The number of moves that each player has available.
   3. **Corner Discs**: If we make a move, we find that the disc can be placed on the corner, then it is given the highest priority.

   The precedence is as follows
   Disc Balance < Possible Moves < Corner Discs

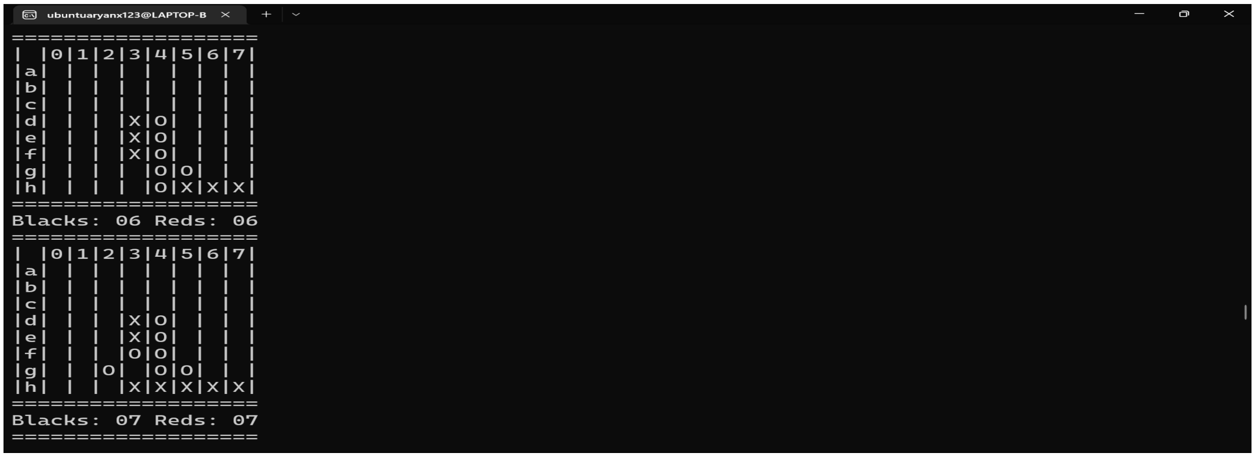
Trees



```
===================
| |0|1|2|3|4|5|6|7|
|a| | | | | | | | |
|b| | | | | | | | |
|c| | | | | | | | |
|d| | | |X|O| | | |
|e| | | |O|X| | | |
|f| | | | | | | | |
|g| | | | | | | | |
|h| | | | | | | | |
===================
Blacks: 02 Reds: 02
===================
| |0|1|2|3|4|5|6|7|
|a| | | | | | | | |
|b| | | | | | | | |
|c| | | | | | | | |
|d| | | |X|O| | | |
|e| | | |O|X| | | |
|f| | | | | | | | |
|g| | | | | | | | |
|h| | | | | | | | |
===================
Blacks: 02 Reds: 02
===================
```



```
===================
| |0|1|2|3|4|5|6|7|
|a| | | | | | | | |
|b| | | | | | | | |
|c| | | | | | | | |
|d| | | |X|O| | | |
|e| | | |X|O| | | |
|f| | | |X|O| | | |
|g| | | | |X|O| | |
|h| | | | | |X|O| |
===================
Blacks: 05 Reds: 05
===================
| |0|1|2|3|4|5|6|7|
|a| | | | | | | | |
|b| | | | | | | | |
|c| | | | | | | | |
|d| | | |X|O| | | |
|e| | | |X|O| | | |
|f| | | |X|O| | | |
|g| | | | |O|O| | |
|h| | | | |O|X|X|X|
===================
Blacks: 06 Reds: 06
===================
```



```
===================
| |0|1|2|3|4|5|6|7|
|a| | | | | | | | |
|b| | | | | | | | |
|c| | | | | | | | |
|d| | | |X|O| | | |
|e| | | |X|O| | | |
|f| | | |X|O| | | |
|g| | | | |O|O| | |
|h| | | | |O|X|X|X|
===================
Blacks: 06 Reds: 06
===================
| |0|1|2|3|4|5|6|7|
|a| | | | | | | | |
|b| | | | | | | | |
|c| | | | | | | | |
|d| | | |X|O| | | |
|e| | | |X|O| | | |
|f| | | |O|O| | | |
|g| | |O| |O|O| | |
|h| | | |X|X|X|X|X|
===================
Blacks: 07 Reds: 07
===================
```

Comparing Algos

Regarding space and time complexity, alpha-beta pruning has the upper hand because it quickly assigns values to the discs based on their position on the board.

But regarding winning criteria, the min-max takes the lead though a slower algorithm but has a better search space and compares many crucial aspects before making a decision.