Build Adversarial Search Agent Project Report.

I started the project with copying Minimax agent from sample agents file. This gave custom agent around 70-75% win rate against GREEDY agent. The maximum depth possible was 3.
I changed opening movies from random to center of the board or any of 9 central squares if the center was taken.
Then I added iterative deepening with best moves memoisation that simply added mappings from states to best moves to dictionary by using "setdefault()" to either call them from memory or saving moves to dict. After successfully implementing that the max depth increased to 5, and win rate increased to 75-83%.
After that I have added new evaluation function that checked for moves that would divide the board with more liberties for my player. That would allow my agent to find the potentially winning movies way above the last layer of search tree. After some experimenting, I set the value of that move as difference of the liberties available and compared that value to whatever value recursion would return. It would then return the max value of both. That increased my win rate to almost 100%.
At that point I decided to stop and submit the project.

Table below compares performance of different configurations of the custom agent.

Technique	Depth	Win Rate
Minimax Minimax with random first move selection and own_liberties - opp_liberties heuristic. Fair Matches ON	3	65%
Minimax with own_liberties - opp_liberties. Choosing middle tiles in the beginning. Fair Matches ON	3	70-75%
Above with Iterative deepening.Fair Matches ON	5	75-83%
Above with my own Divided board heuristic. Fair Matches ON	5	93-100%

Conclusion:
Using iterative deepening with memoisation allowed for more depth of the search which gave custom player additional 5-8% of success rate.
Searching for the split states and finding a good weight for the move added additional 10-17% of won games!!!