State space in my program is start from the choices that player can make to end state-no ball in either player. The state space includes multiple states such as board situation, current player, and number of balls in each cell. The current player is a Boolean stored in GameControl class. The number of balls in each cell and board details are stored in Board class as ArrayList and other attributes.

The program has two heuristics function. The first one is calculating how many balls will be in the player side. This heuristic use number of balls in player side after the moves to estimated winning chance that player can have. Because, it is easier to get balls from player own side than opponent side. In addition, at the end of game, all balls in player side will belong to the player. So, more balls in player side, more chance to win the game. However, this may not be the optimal movement, as the balls in the player side are not always got by the players.

The second heuristic function is calculating how many balls will be in the player Mancala. This heuristic use number of balls in player side mancala after the moved to estimated winning chance that player can have. Because, the number of balls in mancala is the most important for the game. What’s more, it is winning condition for the game. Hence, this heuristic is closer to optimal answers than first one.

The number of nodes searched for first heuristic function is 17884 and for second heuristic function is 12067. After Implement the alpha-beta, the number decrease to 200.