Team Name: One Team One Dream

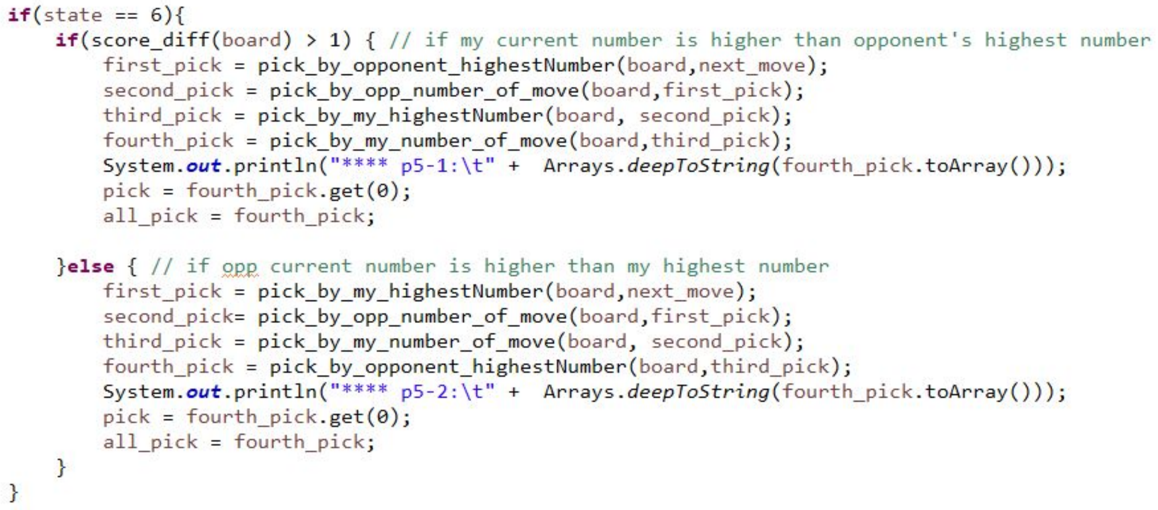
COSC326 Sequencium - Etude 5: Formal Report

Main Implementation:

The current implementation used by our player is a five step mechanism, with each step taking into account a different factor of the game and reduces the amount of next move cell choices by this factor. The fifth step only occurs if the first steps result in more than one possible movement choice. The five step mechanism functions as follows:

- Step 1: Chooses the best possible next step based on the possible number of opponent moves
  - In this step it is best for there to be as little number of opponent moves as possible.
- Step 2: Chooses the best possible next step based on my player's highest possible number of moves
  - In this step it is best for there to be as many moves as possible.
- Step 3: Chooses the best possible next step based on my player's current highest number (currently in the game/generated via moves from the game)
  - In this step we want this value to be as high as possible
- Step 4: Chooses the best possible next step based on my opponent's highest number in the game
  - In this step we want this value to be as low as possible
- Step 5: Look ahead mechanism occurs if the steps do not result in a final movement choice. E.g. If there are more than one possible movement choices left after the first four steps are implemented.

Figure representing the step mechanism:



```
if(state == 6){
    if(score_diff(board) > 1) { // if my current number is higher than opponent's highest number
        first_pick = pick_by_opponent_highestNumber(board,next_move);
        second_pick = pick_by_opp_number_of_move(board,first_pick);
        third_pick = pick_by_my_highestNumber(board, second_pick);
        fourth_pick = pick_by_my_number_of_move(board,third_pick);
        System.out.println("**** p5-1:\t" +  Arrays.deepToString(fourth_pick.toArray()));
        pick = fourth_pick.get(0);
        all_pick = fourth_pick;

    }else { // if opp current number is higher than my highest number
        first_pick = pick_by_my_highestNumber(board,next_move);
        second_pick= pick_by_opp_number_of_move(board,first_pick);
        third_pick = pick_by_my_number_of_move(board, second_pick);
        fourth_pick = pick_by_opponent_highestNumber(board,third_pick);
        System.out.println("**** p5-2:\t" +  Arrays.deepToString(fourth_pick.toArray()));
        pick = fourth_pick.get(0);
        all_pick = fourth_pick;
    }
}
```

This fifth step look ahead mechanism attempts to look ahead to the next three possible moves from the states remaining in step five (after all four steps have been implemented and the best possible next moves have been narrowed down). The state associated with the best outcome, after three steps, will be chosen by the player as its next move. This mechanism currently only functions when our opponent player has no more possible moves, though could be further implemented in future projects.

Further Implementation:

Within our program we have also implemented a technique that changes our player’s behaviour according to the current state of the game. If our current highest number is higher than our opponent’s, our player will focus on blocking our opponent from increasing their current highest number by prioritising picking its next move closer to where the opponent’s highest number is.

If our opponent’s highest number is higher than our player’s, our player will focus on getting it’s number higher. This mechanism gives our player the opportunity to increase the lead it has (when it has a lead), or gives our player the chance to catch up when it doesn’t have the current highest number.

This increases the chances our player has to win, when losing, as it forces our player to focus on increasing its highest number instead of blocking our opponent, when our player is losing the game.