Team Name: One Team One Dream

COSC326 Sequencium - Etude 5: Project Diary

Monday 3rd August:
Dream wrote an implementation that focused on filling the centre of the board first, using larger values to represent squares closer to the centre of the board forcing the algorithm to choose these centre cells more often.

This current implementation chooses where to place the next number based on how many cells are surrounding the possible cell choices.

Wednesday 5th August: 2pm - 6pm
Group meeting based around trying to find the best way to better Dream's previous implementation, as it is currently not enough to determine a players choice solely on how many cells are surrounding the possible cell choices.

Ideas thought to include to implementation:

- Try to find a way to block the opposition off, to reduce the final number it can get.
    - Attempt to alternate time between blocking the opposition off and increasing the own players final number.
- Attempt to implement the blocking off of the opposition in a way that extends the cells possible to use by the player.

We were quite stuck for a while because we couldn't think of how to create an implementation to block the opposition's moves, whilst still ensuring that we reach the highest possible number.

Monday 10th August: 4pm - 8pm
Implementation of a six step mechanism, where each step reduces the amount of next move cell choices by a specific factor:

- Step 1: Based on the possible number of opponent moves
- Step 2: My highest possible number of moves
- Step 3: Based on the possible number of future moves
- Step 4: My current highest number (currently in the game/generated via moves from the game)
- Step 5: My opponents highest number in the game
- Step 6: 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 5 steps are implemented.

This sixth step look ahead mechanism attempts to look ahead to the next three possible moves from the states remaining in step six (after all five steps have been implemented). The state associated with the best outcome, after three steps, will be chosen by the player as its next move.

Team Name: One Team One Dream

Example case that requires the lookahead approach:

```
possible move = [[1, 0], [2, 0], [3, 0]]
[1, 0]: 0 | 2       & -1 | 5
[2, 0]: 0 | 2       & -1 | 6
[3, 0]: 0 | 2       & -1 | 6
**** p2:         [[2, 0], [3, 0]]
**** p2:         [[2, 0], [3, 0]]
  1  4  5 -5
  0  2  5 -4
  0  4  3 -3
  0  5 -4 -1
                  DEPTH 0
```

```
6TH PICK : [[1, 3], [3, 2]]
# [1, 3]
5 == 5 ?
5 == 5 ?
!!!!! ----------------CASE 2------
 -1 -2  4 -4
 -2  3 -3  0
  0  4  2 -4
  0  0  0  1
```

Friday 14th August:
We adapted the six step mechanism into a 5 step one, with our previous 3rd step being taken out as it seemed to have no real effect on choosing our player's next move. The fifth step acts as the look ahead mechanism.

- Step 1: Based on the possible number of opponent moves
- Step 2: My highest possible number of moves
- Step 3: My current highest number (currently in the game/generated via moves from the game)
- Step 4: My opponents highest number in the game
- Step 5: Look ahead mechanism occurs if the steps do not result in a final movement choice.

This mechanism is shown below:

Team Name: One Team One Dream

```
ArrayList<int[]> best_pick = new ArrayList<int[]>();

if(state == 1){ //one fixed behaviour, focus on getting higher score
    best_pick = pick_by_my_highestNumber(board,next_move);
    best_pick= pick_by_my_number_of_move(board,best_pick);
    best_pick = pick_by_opponent_highestNumber(board, best_pick);
    best_pick = pick_by_opp_number_of_move(board,best_pick);
}

if(state == 2){ // behaviour of the player adapt to current state of the board

    // if my player's highest number is higher than opponent's highest number, focus on blocking opponent from getting higher score
    if(score_diff(board) > 1) {
        best_pick = pick_by_opponent_highestNumber(board,next_move);
        best_pick = pick_by_opp_number_of_move(board,best_pick);
        best_pick = pick_by_my_highestNumber(board, best_pick);
        best_pick = pick_by_my_number_of_move(board,best_pick);
    }
    // if opponent's highest number is higher than my player's highest number, focus on getting higher score
    else {
        best_pick = pick_by_my_highestNumber(board,next_move);
        best_pick= pick_by_opp_number_of_move(board,best_pick);
        best_pick = pick_by_my_number_of_move(board, best_pick);
        best_pick = pick_by_opponent_highestNumber(board,best_pick);
    }
}

/* from testing, the first option tends to win with higher score but the second option has more reliable winning chance */
```

We also implemented a technique that changed our player behaviour according to the current state of the game. If our current highest number is higher than our opponents, our player will focus on blocking. Else our player will focus on getting our number higher.