CS4386 AI Game Programming

Report of Assignment 1

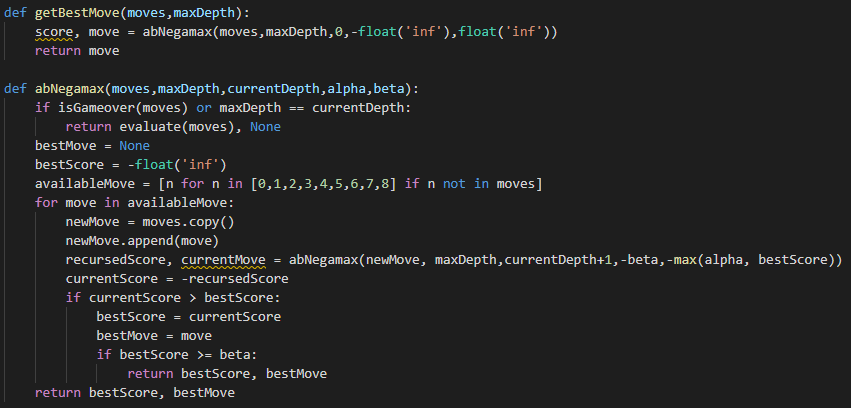
Three Thirteen Game AI

Tsang Ka Kuen

55292702

**Methodology**

The algorithm that I applied to the three thirteen game AI is **Negamax with alpha beta pruning**. This algorithm is suitable for implementing an **unbeatable** three thirteen AI since the number of combinations of game states under winning conditions is small, this algorithm can easily go through most of the combinations, prune all unnecessaries combinations and then get the best move.



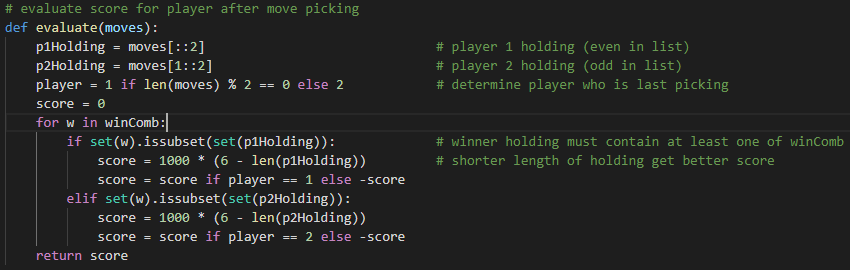
By finding out the combinations of 3 from 0 to 8, we can get 84 combinations, then filter by sum equal to 14, there are 7 combinations left, those are combinations are the subset of the winning state. That means **all winning state must contain at least one set of the following number**.



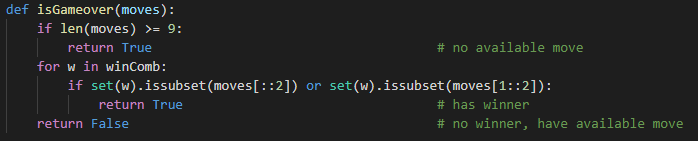
By using the combinations above, we can evaluate score for all game state. The length of number of the player held is also important for the evaluation, **shorter length of number holding will get better score**. **If the game state is beneficial for opponent, we can just negative the score.**

Score for all situation are shown as the following table.

|  |  |
| --- | --- |
| **Situation** | **Score** |
| Win and you holding three number | 3000 |
| Win and you holding four number | 2000 |
| Win and you holding five number | 1000 |
| Draw game | 0 |
| Lose and opponent holding five number | -1000 |
| Lose and opponent holding four number | -2000 |
| Lose and opponent holding three number | -3000 |



The game is over once player win or no number left in pile.



**Test Cases**

The following are some test cases of my python program. We can just run it from command-line.

> python 55292702a1.py <input>

> <output>

|  |  |  |
| --- | --- | --- |
| input | output | description |
| 4 0 1 6 3 | 5 0 1 6 3 8 | Game is over, AI win with 0,6,8 |
| 4 0 1 8 3 | 5 0 1 8 3 6 | Game is over, AI win with 0,8,6 |
| 4 1 0 3 8 | 5 1 0 3 8 6 | Game is not over, AI preventing opponent’s directly win in next pick by choosing 6 |
| 4 1 0 3 6 | 5 1 0 3 6 8 | Game is not over, AI preventing opponent’s directly win in next pick by choosing 8 |

**If the game is over, let say someone win or draw game, the best move would be None, the output of the program would be same as input.** Let say we use <5 0 1 8 3 6> as input, the output would be <5 0 1 8 3 6> since player 1 win, the game is over.

> python 55292702a1.py 5 0 1 8 3 6

> 5 0 1 8 3 6