Report of Theory of Computer Games project 3, 2022

109550027 紀竺均

1. Implementation

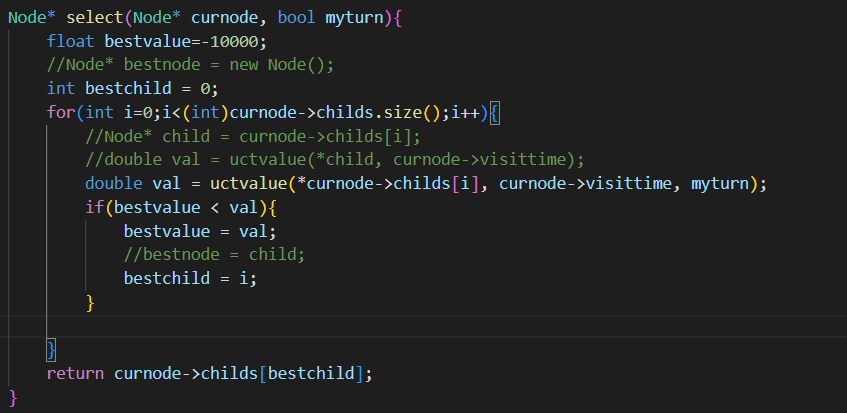
In this assignment, I create two classes – “MCTS” and “mcts\_agent” which inherited from class “agent”.

In MCTS class, there are four mainly function for constructing MCTS tree – select, expand, simulate, and update. And a function “best\_action” that decide the action based on the tree.

In class “MCTS”:

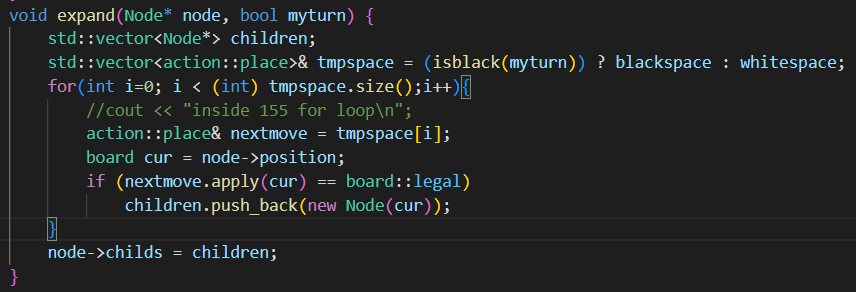
1. select():

In function “select”, I select the current node’s child with the highest uct value. I’ll discuss the way I calculate uct value in section b.



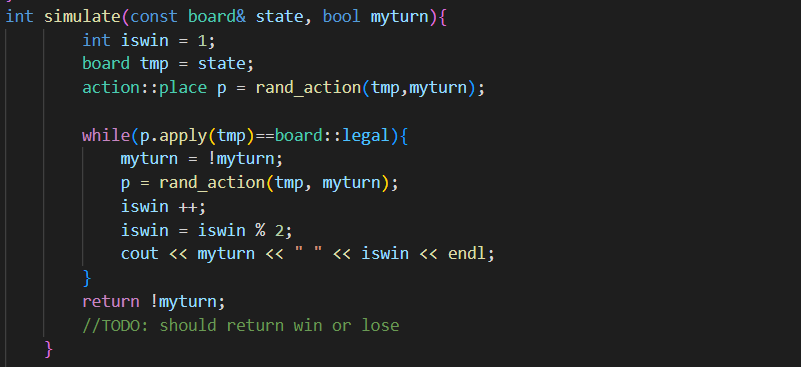
1. expand():

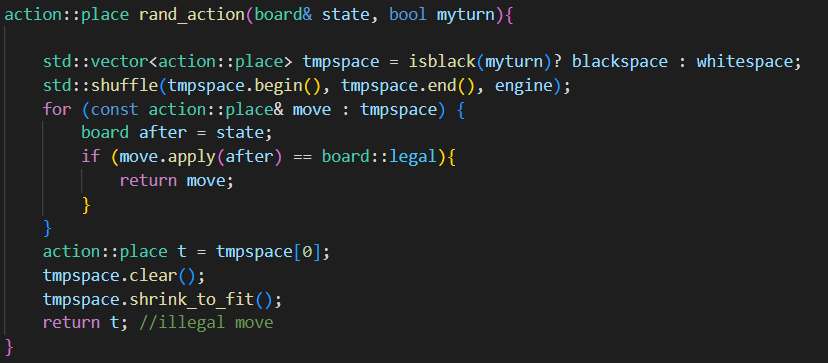
In function “expand”, I append every legal move in current space to the node’s child list.



1. simulate

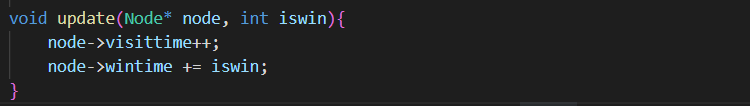
In function “simulate”, I apply the legal moves based on rand\_move function, which return a random legal movement drawn from current space, and return the win/lose result.





1. update():

update current node’s visit time and the count of win time.

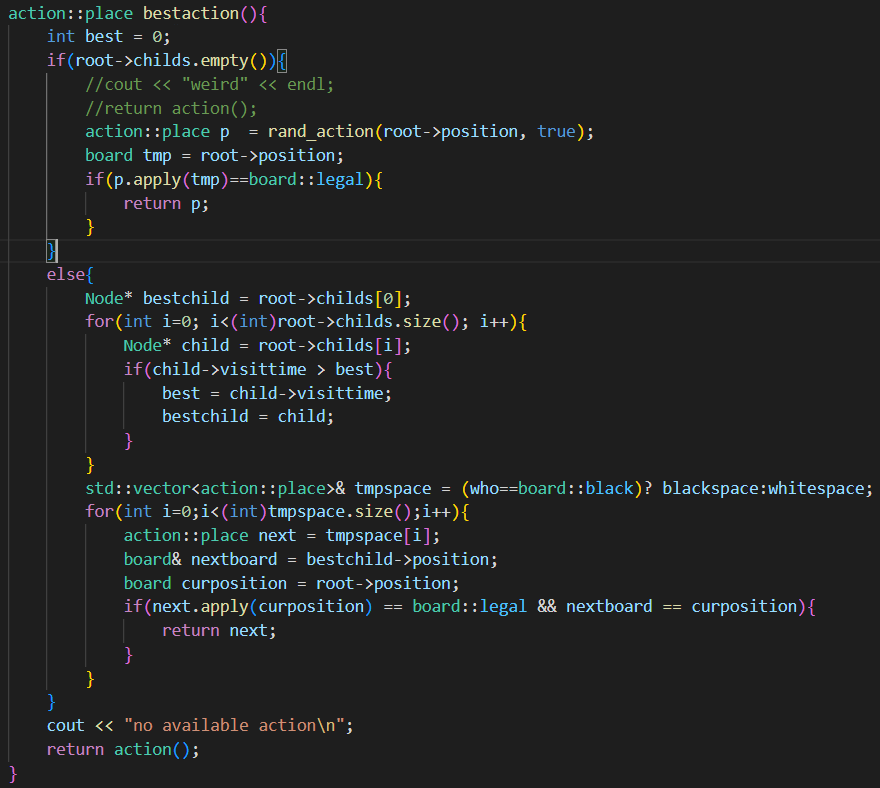


1. best\_action():

If the childs list is empty, it means that there isn’t any legal move for current root, so we simply return a random action.

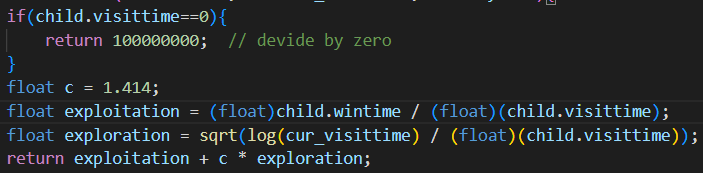
If there are childs for root, we could choose the child with greatest visit time to be our next position.

Once we have the next position, we can traverse thru current space to find what is the legal move that can lead to that position, and finally return that legal move.

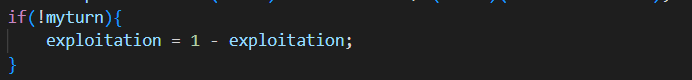


1. Improvements
2. UCT value calculation

Originally, I calculate uct value based on simple function:



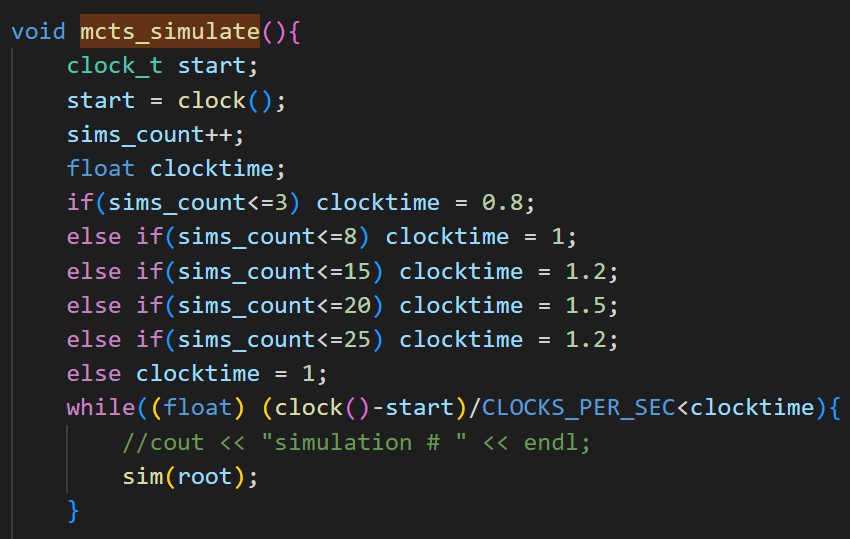
Then, I tried to improve the function by adding who’s turn is it while calculating uct value because if it’s the opponent’s turn, their win rate should be our loss rate ( 1-winrate).



1. Time distribution

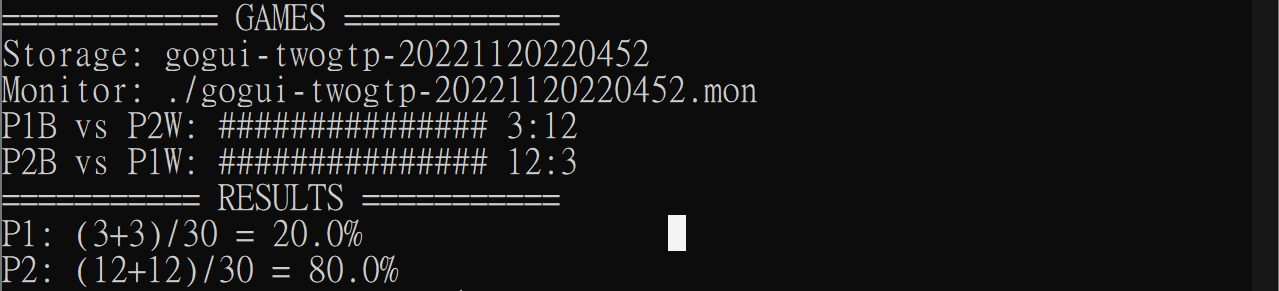
Because there are total 73 spaces in the board, there are at most 73/2 = 32 times that we will do a mcts simulation. Originally, I let every simulation to run at most 0.99 seconds (so it won’t exceed the time limit of 40 seconds per game).

My improvement is that, spend more simulations in the middle of the game.

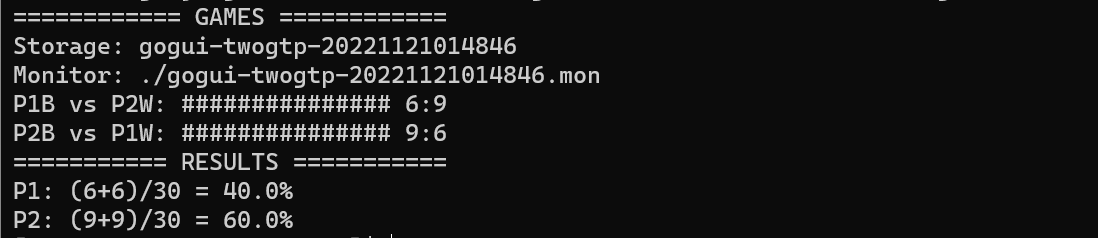


1. Results

Origin mcts player against strong agent



Improved mcts player against strong agent



1. Problems encountered and solutions

bad\_alloc error:

During the running time, I found out all the memory in workstation is depleted and thus get this error.

Solution: recursively “delete” every node after I take my action, clear and shrink\_to\_fit the vectors after I allocate them.