Theory of Computer Games(Full 2022)

Homework #2

National Taiwan University

Due Data: 14:20(UTC+8), December 8,2022

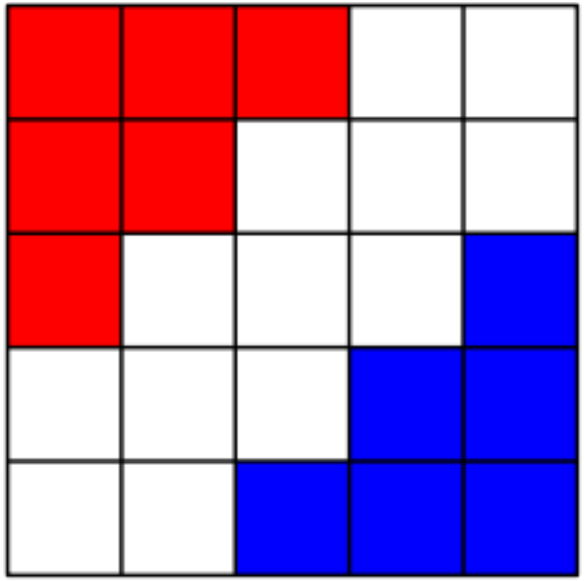
Homework Description

In this homework, you are required to

- Implement an agent of **modified-Einstein Würfelt Nicht** using Monte-Carlo Tree Search
- Beat the 3 baseline AI

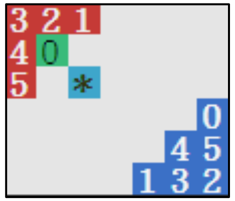
Original game - Einstein Würfelt Nicht

- get one of their cubes to the far corner square in the grid
- or remove all of their opponent's cubes from the board
- more detail can see wiki



Our game - modified-Einstein Würfelt Nicht

- The game is played on a 6x7 board. Initially there are 6 red cubes and 6 blue cubes on the board.



- Each cube has a number between 0 and 5
- Initial positions of both sides are randomized. 1st player's pieces start from the North-West.
- In each turn the 1st player chooses a red cube to move, and subsequently (if the game is not over) the 2nd player chooses a blue cube to move.

Move

- a player can move any piece of its color
- The top-left player can only move a cube to **east**, **south**, or **southeast** adjacent square
- The bottom-right player can only move a cube to **west**, **north**, or **northwest** adjacent square
- If there is another cube in the adjacent square, that cube is **captured**. **A player is not allowed to capture a cube of its own**.
- A player is not allowed pass if there is at least one legal move.

Terminal Condition

- A red cube has number a reaches the Southeast corner, and a blue cube has number b reaches the northwest corner
    - a < b then red win
    - a = b then draw
    - a > b then blue win
- If the last red cube is captured, blue player wins
- If the last blue cube is captured, red player wins

Execution Files

- 2 folders, game and baseline
- Under game, make for the executable gaming environment – game
- The game supports AI-AI mode, AI-human mode and human-human mode
- Under baseline, make for 3 given agents, **random**, **greedy**, and **heuristic**
- To begin with, use

```
$ ./game –p0 ./greedy
```

to start playing Human vs AI with the agent greedy

Protocol

- An agent receives the last move of the opponent from game and sends its move accordingly back
- We've handled most **parts of the** communication. Receive messages by reading from stdin and send messages by writing to stdout
- Read everything character-by-character; if you expect a message of length k to be received, read one character k times instead of directly reading a string
- Remember to flush every time after writing a message to stdout.

Frame of an Agent

```
• While true do
    • Receive $R_1, R_2$
    • $B$ <- the initial board given $R_1$
    • Your Turn <- $R_2$ = “f”? true: false
    • While true do
        • if “terminal” then
            • Break
        • if your turn = false then
            • Receive $R_3$
        • else
            • Choose a move $M$
            • Do the move $M$ on $B$
            • Send $M$
        • Your turn <- ! Your turn
```

Formats of Received / Sent Messages

- R_1[0:1][0:5] : a permutation of “012345”
  - number of cubes
  - (0,0),(0,1),(0,2),(1,0),(1,1),(2,0)
  - (3,6),(4,5),(4,6),(5,4),(5,5),(5,6)
- R_2 : a single character
  - ‘f’: you are the 1st player in this round
  - ‘s’: you are the 2nd player in this round
- R_3 : can be “??” (pass), or ND ,where
  - N = number of cube to be moved
  - D = direction , 0(vertical) 1(horizontal) 2(diagonal)
- M: a 2-sized string, can be “??” or ND only

Homework requirement

- You’re required to implement Monte-Carlo Tree Search
  - ucb score
  - uct tree search
  - amaf/rave
  - Anything taught or found in literatures about MCTS improvements
- Your agent should send a valid move within 10 seconds
- If game receives an invalid move or donesn‘t reveive a move within the time limit, the opponent wins immediately
- Beat the 3 baseline AI
- Write a Report

Submission and Grading Policy

- Directory Hierarchy:
    - Student_id
        - Makefile
        - src // a folder contains all your code
        - report.pdf
- Compress your folder into a zip file and submit to

  https://www.csie.ntu.edu.tw/~tcg/2022/hw2.php
- We will compile and run your code on csie workstations, therefore make sure your code can run on csie workstations

Report

- Your report should include but not limit to the following:
    - How to compile your code into an agent
    - What algorithms and heuristics you’ve implemented
    - Experiment results and findings of your implementation
    - Some detail about your implementation
    - Discuss benefits of various enhancements
- Add your name and student id in the report

Grading Policy

- Beat the Random agent (20%), Greedy agent(30%), heuristic agent(30%)
- Random will play 10 rounds, Greedy and heuristic will play 15 rounds:
  - Win: +2
  - Draw: +1
  - Loss: +0
- Your agent will be tested by

  ```
  • $ ./game  –p0  [your agent]  –p1  ./random  –r  10
  • $ ./game  –p0  [your agent]  –p1  ./greedy  –r  15
  • $ ./game  –p0  [your agent]  –p1  ./heuristic  –r  15
  ```

- Report (20%)
- Bonus: We will make tournaments among students and those scores are among the best will be given bonus points