Theory of Computer Games (Fall 2023)

Homework 2

NTU CSIE

Due: 2023/12/7 14:20

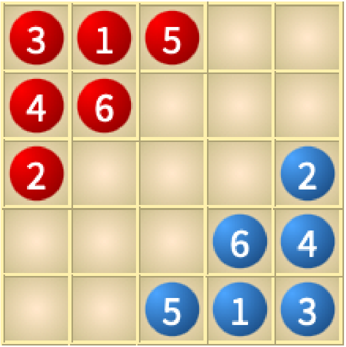
Outline

1. Game Description

2. Homework Requirements

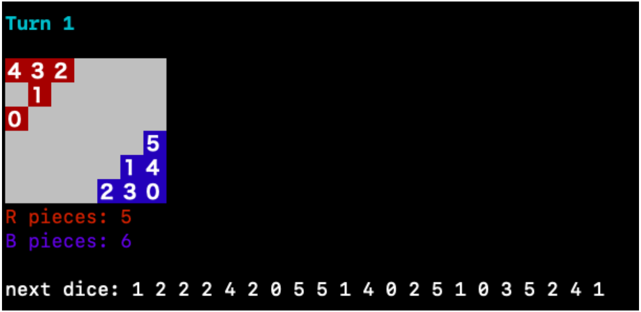
3. Submission and Grading Policy

Original game - EWN



- EWN-wiki
- 愛因斯坦棋-中文版維基

Modified EWN



```
Turn 1

4 3 2
  1
0
            5
          1 4
        2 3 0
R pieces: 5
B pieces: 6

next dice: 1 2 2 2 4 2 0 5 5 1 4 0 2 5 1 0 3 5 2 4 1
```

- The value of the dice is fixed
  - the dice sequence is cyclic with period 21
- Range of the number is 0 $\sim$ 5 not 1 $\sim$ 6

How to select a cube to move

- Assume the dice shows the number x.
- If the cube with number x still exists, then you can only choose x.
- If the cube with number x does not exist, then you can choose
  - a: the cube with the biggest number smaller than x
  - b: the cube with the smallest number bigger than x
- The top-left player can only move $\rightarrow$, $\downarrow$, $\searrow$, and $\nearrow$.
- The bottom-right player can only move $\leftarrow$, $\uparrow$, $\nwarrow$, and $\swarrow$.

Terminal Condition

- If the last red cube is captured, then blue player wins
- If the last blue cube is captured, then red player wins
- If a red cube reached the southeast corner, then red player wins
- If a blue cube reached the northwest corner, then blue player wins

Homework Requirements

- Implement a MCTS based program with UCB.
- Beat the baseline program
  - easy: random move
  - normal: alpha beta with depth 2
  - hard: alpha beta with depth 8
- Write a report
- Limitation
  - Time limit: 3s per ply.
  - Memory: no more than 1G.
  - Thread limit only one.
  - We will run your code on csie workstations (ws1).

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, easy, normal, and hard
- To begin with, run

  ```
  $ ./game -p1 ./normal
  ```

  to start playing Human vs AI with the normal agent.

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 $R_1, R_2, R_3$
    $B \leftarrow$ the initial board given $R_1$
    Your Turn $\leftarrow R_3 =$ 'f' ? true : false
    while true do
        if terminal then
            break
        if your turn then
            Receive $R_4$
        else
            Choose a move $M$
            Do the move $M$ on $B$
            Send $M$
        change to next player
```

Formats of Received/Sent Messages

- R_1: two permutations of "012345"
  - initial positions
  - (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 dice sequence of period 21
- R_3: a single character
  - 'f': you are the first player in this round
  - 's': you are the second player in this round
- R_4: ND, where
  - N: number of cube to me moved
  - D: direction, 0(horizontal), 1(vertical), 2(diagonal), 3(↗, ↙)
- M: ND

Submission

- Directory Hierarchy:
    - student_id
        - Makefile
        - src // a folder contains all your codes
        - report.pdf
- Compress "student_id" into a zip file named student_id.zip.
- The first letter of your student id should be lowercase.
- Send your zip file to ntu.theory.of.computer.games@gmail.com.
- Due to server limitation, the file size is restricted to 2 MB.
- You will get some penalty (-10 points) if you don't follow these rules.

Report

- Your report should be named report.pdf.
- Your report should include but not limit to the following:
    - What algorithms and heuristics you've implemented.
    - Experiment results and findings of your implementation.

Grading Policy

- Generate the agent named agent after running “make” (5%)
- Beat the easy agent (20%), normal agent(20%), hard agent(20%)
  - Win: $+1$
  - Lose: $+0$
  - Due to the given dice sequence, this game has an element of luck. If you win ≥ 14 at a part, your score is $\min\{\text{win} + 3, 20\}$
- Your agent will be tested by
  - $ ./game –p0 [your agent] –p1 ./easy –r 20
  - $ ./game –p0 [your agent] –p1 ./normal –r 20
  - $ ./game –p0 [your agent] –p1 ./hard –r 20
- Correct implementation of the required parts:
  - UCB (8%)
  - MCTS (12%)
  - More techniques taught in class (Bonus, at most 5%)
- Report (15%)