
Final project

Solving EWN

EWN (Einstein würfelt nicht!)

All assignments in this course will be related to EWN.

Rule Description

EWN is a **two player** and **stochastic** (雙人隨機) game.

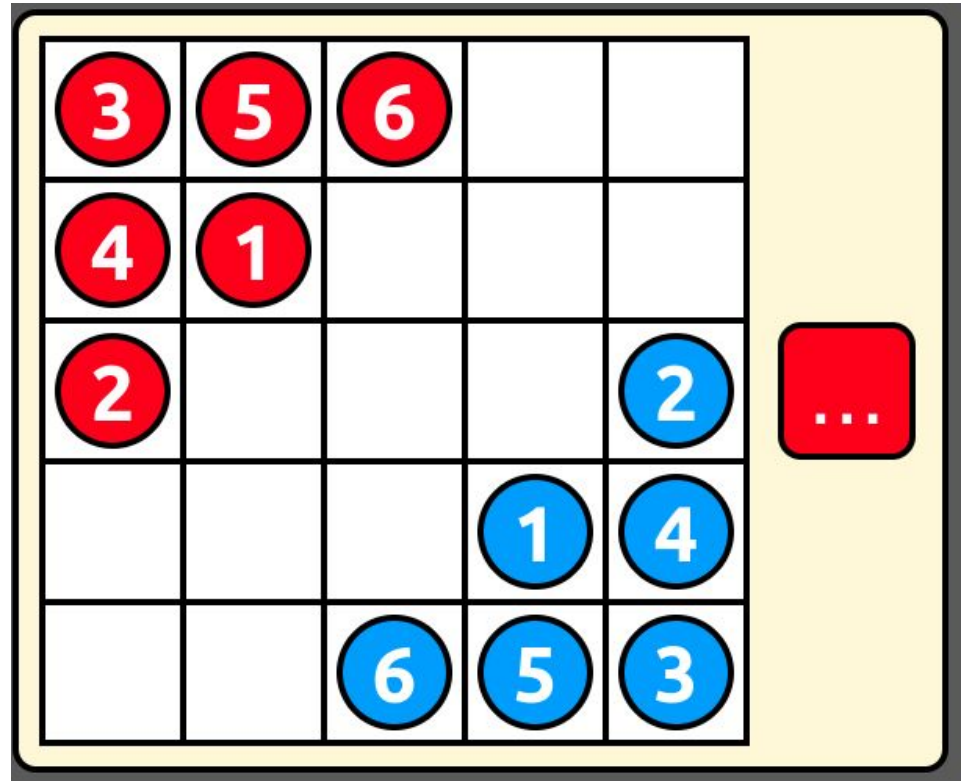
In the final project, we are finally going to solve EWN!

Rules of EWN

1. board:

size: 5x5

pieces: 1,2,3,4,5,6



Rules of EWN

2. moving procedure

a. determine the moving piece :

if the dice number matches an existing piece, you can only move that matched piece.



In this example, you can only move piece 4.

Rules of EWN

2. moving procedure

b. determine the moving piece :

if the dice number doesn't match any existing piece, you can choose to move:

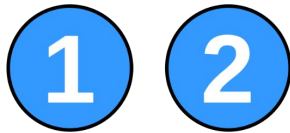
1. The piece that its number is the smallest but bigger than the dice number. (If such piece exists.)
2. The piece that its number is the biggest but smaller than the dice number. (If such piece exists.)

Examples are in the next page.

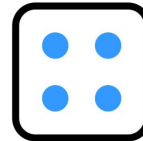
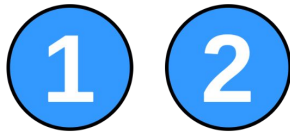
Rules of EWN

2. moving procedure

b. determine the moving piece :



In this example, you can choose to move piece 2 or piece 5.



In this example, you can only move piece 2.

Rules of EWN

2. moving procedure

c. piece movement and capture

After choosing the moving piece, you can move it in 3 directions.

For the red side, it is down, right, down-right.

For the blue side, it is up, left, up-left.

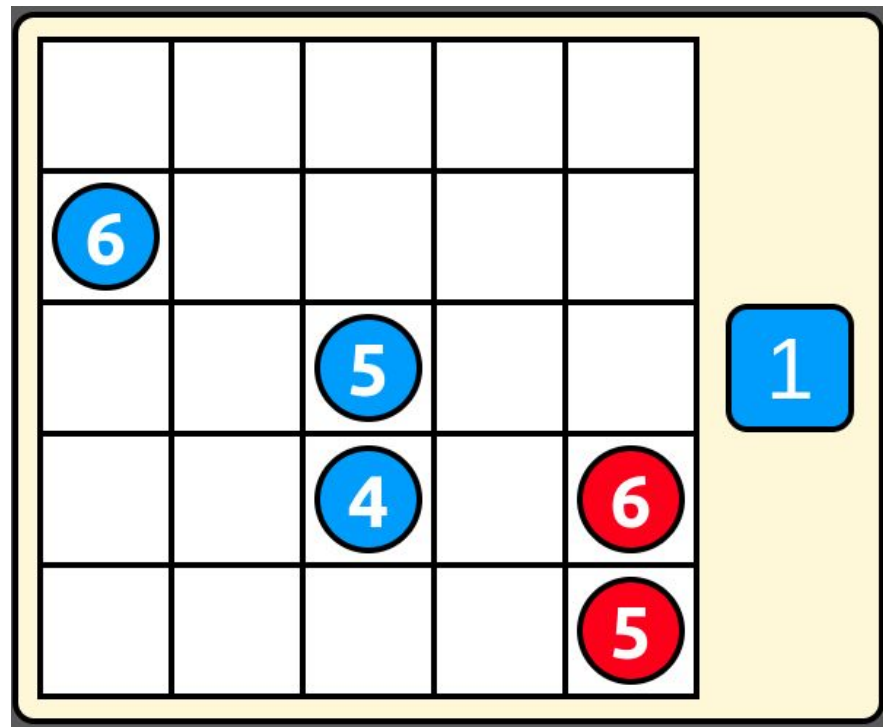
If there is a piece at the destination, the piece at the destination will be removed.

Rules of EWN

3. win condition:

Each side has a goal position. For the red side, the goal is the bottom-right corner. For the blue side, the goal is the top-left corner.

For example, the red 5 reaches the bottom-right corner. Red wins the game.

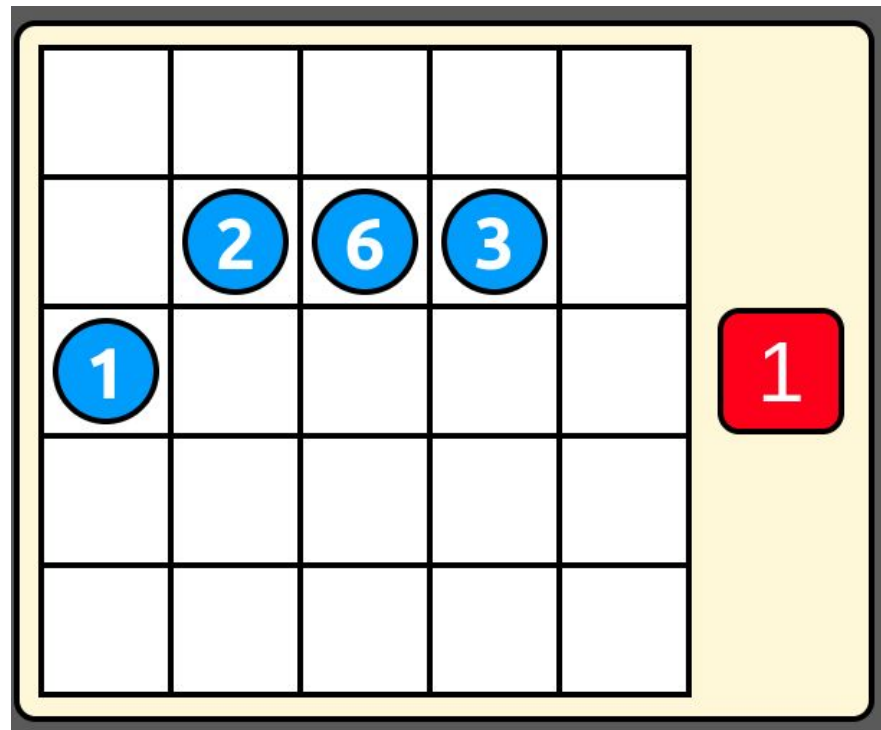


Rules of EWN

4. win condition - capture all:

If every piece of the player is captured. The opponent wins.

For example, blue captures every red piece, so blue wins.



Overview of final project

1. random baseline 25%
2. code 55%
3. report 20%
4. contest 15%

Note that:

1. **total grade won't exceed 100%**
2. **You will not get any points for the entire final project if you didn't attend the competition !!!**

Programming part

— grading criteria —

Programming Part - Baselines

For our HW2 and final, we will use this platform written by TA.

<https://weilin97462.github.io/EWNPlatform/>

You can play the game online, or use the client to play the game by your AI.

The sample code already handles all communication with the client. You can focus on implementing the algorithm.

Programming Part - Baselines

random baseline: **25%**

We will open a contest to let you play against random agent with 106 rounds.

Your score = $(\min(\text{wins}, 100) / 100) * 25\%$

Means that if you win 100 times in this 106 times game, you get full score.

Programming Part - Submission

[any name you like].zip
└── code
 └── makefile
 └── [other files]

Your code must successfully compile an executable named **agent** after running **make** command.

The submit system will auto rename your zip file, no need to change the name.

submission site:

www.csie.ntu.edu.tw/~tcg/2024/final

Coding part

— grading criteria —

Coding Part

To make sure you did implement all these following techniques, we will read your code.

In each part, you can get full points if you correctly implement the functions.

However, if your program has bug, you will lose some points.

Coding Part

1. Search on chance nodes 20%
 - a. Star1 (negamax) 20%
 - b. Star1 (minimax) 10%
2. Search on non-chance nodes 10%
 - a. Negascout 10%
 - b. Negascout (minimax) 5%
3. Transposition table 10%
 - a. Zobrist hash 10%
 - b. Other hash 0%
4. Time control
with Iterative deepening search 5%

Coding Part - Transposition table

When using zobrist hashing, there is no point of using `unordered_map` or `map` if you know how they work at bottom layer.

If you insist on using them, you get no points.

Coding Part - Time control

You **should** use iterative deepening + dynamic time control for your final project.

You can check chap 16. for implementation details.

Coding Part

5. readability (10%)

To make sure your code is humanly readable, we encourage you to make your code as readable as possible.

- don't name your variable, function name with meaningless names
- Helpful comments in your code are highly encouraged
- use `#define` to define your return value of your function

Handwriting part

— report —

Report

Your report should include:

1. Explanation of your code. (10%)
2. The location of your functions (5%)
(e.g. `star1` at `main.cpp:99`, `negascout` at `main.cpp:199`, ...)
3. Experiment results and findings of your implementation. (5%)
(recommend showing the win rate between new and old agent)

The number of pages of your report should not exceed 6. (-5% if #pages > 6)

Because **report has different deadline**, it has its own submission site:

www.csie.ntu.edu.tw/~tcg/2024/final_report

Only pdf format is accepted.

Contest

— fight for the glory!!! —

Contest - Date

Practicing session: 2024/12/26 14:20 (UTC+8)

Recommend attending to test your system and device.

Contest - Date

2025/1/2 14:20 (UTC+8)

Players are expected to prepare for the tournament between 1pm and 2pm.

Late comers will be treated as no show.

You will not get any points for the entire final project if you didn't attend the competition !!!

Contest - Game settings

Tournament scheduler: **Swiss-system**

Game number: **200 games per round**

Time limit: **5s for each game**

You lose if your program crash in a game.

In case of any violation, the referee has the right to decide the result of a game or if it may be restarted.

Contest - Grading

$((\text{your_score} - \text{min_score}) / (\text{max_score} - \text{min_score})) * 15\%$

score means how many wins you have, win more, earn more!!!