# Theory of Computer Games (Fall 2019) Homework #2

National Taiwan University

#### Due Date: 14:20 (UTC+8), December 19, 2019

Theory of Computer Games (Fall 2019)Homework #2

・ 一 マ ト ・ 日 ト ・

#### Homework Description

In this homework, you are required to

- Implement an agent of modified-Einstein Würfelt Nicht! (Kari) using Monte-Carlo Tree Search.
- Beat the conservative AI and the greedy AI.

**F 4 3 F 4** 

# Basics

• The game is played on a  $6 \times 6$  board. Initially there are 6 red cubes and 6 blue cubes on the board.



- Each cube has a number between 0 and 5, and no two cubes of the same color shares the same number.
- Initial positions of both sides are randomized. 1st player's pieces start from the North-West, while 2nd player's pieces start from the South-East of the board.
- 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.

## Moves

- In a turn, a player can move any piece of its color.
- The top-left player (red) can only move a cube to the east, south, or southeast adjacent square.
- The bottom-right (blue) player can only move a cube to the west, north, or northwest adjacent square.
- If there is another cube in the adjacent square, that cube is captured. A player is allowed to capture a cube of its own.
- If there is no movable cube, a player should pass in that turn.
  A player is NOT allowed pass if there is at least one legal move.

・ 同 ト ・ ヨ ト ・ ヨ ト

## **Terminal Condition**

#### The game is over when

- A red cube reaches the SouthEast corner, and a blue cube reaches the NorthWest corner...
  - If the SouthEast red cube has numbers smaller than the NorthWest corner, the red player wins.
  - If the SouthEast red cube has numbers bigger than the NorthWest corner, the blue player wins.
  - If the SouthEast red cube has numbers equal to the NorthWest corner, then it is a draw.
- 2 If the last red cube is captured, blue player wins.
- IF the last blue cube is captured, red player wins.

**F 4 3 F 4** 

# **Execution** Files

- Unzip, there will be 2 folders, game and baseline.
- Under game, make for the executable gaming environment game
- The game game supports Al-Al mode, Al-human (1P) mode, and human-human (2P) mode.
- Under baseline, make for 3 given agents, random, greedy, and conservative.
- To begin with, use
  - \$ ./game -p0 ./greedy

to start playing Human vs. Al with the agent greedy.

- 4 同 ト 4 ヨ ト 4 ヨ ト

# 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.

- 4 同 ト 4 ヨ ト

# Frame of an Agent

1: \	while true do
2:	receive $R_1, R_2$
3:	$B \leftarrow$ the initial board given $R_1$
4:	$YourTurn \leftarrow R_2 = "f"?True : False$
5:	while true do
6:	if "game has reached terminal condition" then
7:	break
8:	end if
9:	if YourTurn = False then
10:	receive $R_3$
11:	do the opponent's move $R_3$ on $B$
12:	else
13:	choose a move <i>M</i>
14:	do the move <i>M</i> on <i>B</i>
15:	send M
16:	end if
17:	YourTurn ←!YourTurn
18:	end while
19: end while	

æ

#### Formats of Received / Sent Messages

**2**  $R_2$ : a single character.

- 'f': you are the 1st player in this round
- 's': you are the 2nd player in this round
- So  $R_3$ : can be "??" (pass), or *nd* (otherwise), where
  - *n* = number of cube to be moved
  - d = direction: 0 (vertical), 1 (horizontal), 2 (diagonal)
- M: a 2-sized string, can be "??" (pass) or *nd* (otherwise) only.

- You can assume that every move your agent receives is valid.
- Your agent should send a valid move within 10 seconds. If game receives an invalid move, or doesn't receive a move within the time limit, your agent will be killed, and your opponent wins immediately.

・ 同 ト ・ ヨ ト ・ ヨ ト

# Implementation Requirement

- You're required to implement the following algorithms:
  - Perform UCT tree search with tree expansion based on UCB score
  - Add at least one one of Progressive Pruning (PP) or RAVE
- Your execution file should be named with your student ID, with all alphabets in lower case, e.g., b08902000, not B08902000.
  - If your programming language is python3, add
    #!/usr/bin/env python3 in the first line and remove .py from the filename.
- Single core, and no more than 4GB RAM
- You can add plug-in learning + training data, but the training needs to be done by TA in 30 minutes using hardware described above.
- Your agent will be tested by
  - \$ ./game -p0 [your\_id] -p1 [our\_agent] -r 5

# Report

- Your report should include but not limit to the following:
  - How to compile your code into an agent (if your code must be compiled). Don't upload the compiled executable file!
  - What algorithms and heuristics you've implemented.
  - Experiment results and findings of your implementation.
  - Observe your refinement on UCT tree search and (PP/RAVE), try to measure the improvements
- Your report should be named report.pdf.

伺 ト イヨト イヨト

# **Directory Hierarchy**

- [your\_id] // e.g. b08902000
  - src // the directory contains your code
  - makefile
  - report.pdf
- Compress your folder into a zip file.

< ロ > < 同 > < 三 > < 三 >

э

# Grading Policy

- Beating the Agent: 15%
  - Beat the Simple Conservative Agent (SCA) : 5%
  - Beat the Simple Greedy Agent (SGA) : 5%
  - report.pdf 5%
- Bonus:
  - Dominate the Simple Greedy Agent (SGA)
  - Peer Competition
  - Beat the Very Strong Benchmark (VSB)

▲ 同 ▶ ▲ 国 ▶ ▲ 国

# Beating the Agent

- Each round means 2 games with alternating first for each player
- Points you get each round can be S,  $S \in \{-1, -2, 0, 1, 2\}$ 
  - Win: +1 points
  - Draw: 0 points
  - Lose: -1 points
- 5 rounds will be performed between your agent and the baseline agents.
- We consider the total net score no less than zero as beating the agent.

- 4 同 ト 4 ヨ ト 4 ヨ ト

#### Bonus: Dominate SGA

• Received 1 extra credit point you have a net score of **strictly more than** 2 when playing against the Simple Greedy Agent (*SGA*) in 5 rounds

## Bonus: Peer Competition

- Assume number of homework participants ... N
- We will host a 5-round game between each participant.
- Net score is calculated after all games are performed.
- Participants with net score stricly more than 5 ∗ (N − 1) gets 1 bonus point.
- Top *K* players will be awarded more points, where *K* will be decided base on the results.

#### Bonus: Hidden Boss

- You will have 3 rounds to fight the Very Strong Benchmark
- Let your bet score after 3 rounds be W.
- Additional bonus points of  $\min(\max(W, 0), 3)$