The Game of Yinsh (Phase II)

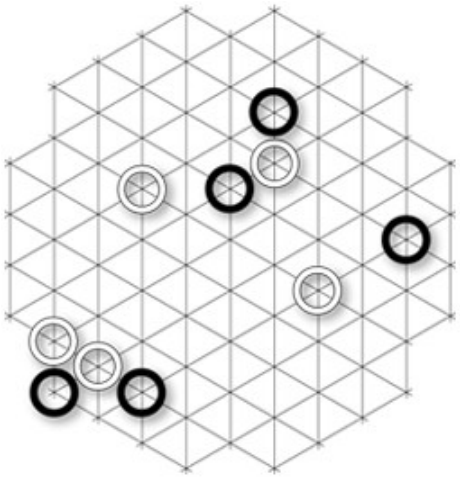
COL333

October 27, 2018

1 Goal

The goal of this assignment is to learn the adversarial search algorithms (minimax and alpha beta pruning), which arise in sequential deterministic adversarial situations.

2 The Game of Yinsh (`N`, `M`, `K`)



2.1 Objective

We will play three tournaments on Yinsh(`N`, `M`, `K`), with the three configurations of (`N`, `M`, `K`): (5,5,5), (6,6,5) and (6,6,6), where `N` is the **board size**, `M` is the **number of starting rings** and `K` is the **consecutive markers needed to remove a ring**. Each tournament will involve various game engines competing against each other. In each match, two games will be played with each player playing first once.

3 Tournament Structure & Scoring

1. **Round-Robin Stage**
   (a) Teams will be arranged into 16 groups (4-5 teams per group). Each team will play against each team in the group. The top two teams in each group will move on to the pre-pre-quarter finals stage.
   (b) If two teams, Team A and Team B, play against each other, then the total points of the match will be the individual sum of the two games played between the two. The points distribution is explained in the README of the provided Github repository.
   (c) The top team in a group will be decided as the team which has the maximum tournament points. If there is a tie, the team who won the match between the two tied teams (had a higher points sum for the match) will go the next round. In a rare event two teams have the same score then the team that had the best performance playing first will be used to tie break. If that also can't decide we will toss.
2. **Knock-Out Stage**
   (a) Teams will play against the assigned team at each level of the knock-out tree. The winner of the game will move on to the next stage.
   (b) If a game is tied in knockout round level with same margins then teams will compete against random players from round robin stage in a sudden death fashion. I.e., both teams in a tie will individually compete against a random player. Whichever team performs better against the latest random player will win. In case of tie, a new random player will be tried. Even after 5 random players if no decision is arrived, then a toss will be conducted.

4 Evaluation Scheme

1. We will run three tournaments in the following order. The time limit per game is also mentioned.
   (a) (`N`=5, `M`=5, `K`=5) - 2 minutes
   (b) (`N`=6, `M`=6, `K`=5) - 2.5 minutes
   (c) (`N`=6, `M`=6, `K`=6) - 3 minutes
2. Each tournament performance is worth 5 points. All teams which do not timeout and don't send invalid moves will get a score of at least 1.5 per tournament. The final scores will be as follows
   (a) Winning any tournament: 15 for the whole assignment. If you win a tournament, you can't play the remaining tournaments.
   (b) Losing in finals: 5

(c) Losing in semi-finals: 4.7

(d) Losing in quarter-finals: 4.4

(e) Losing in pre-quarter finals: 4.1

(f) Losing in pre-pre-quarter finals: 3.8

(g) Teams that do not qualify to the pre-pre-quarters (and have a functioning submission) will receive between 1.5 and 3.5 marks, based on the the number of wins or losses that they have. For instance, for a team in a group of 4, 0 wins gets the team a score of 1.5, 1 win gets the team a score of 2.5 and 2 wins gets the team a score of 3.5. For a team in a group of 5, 0, 1, 2 and 3 wins gets the team a score of 1.5, 2.166, 2.833 and 3.5 respectively. However, if a team timed out, or sent a wrong move, etc., they will not get their functionality credit of 1.5.

5 What is being provided?

Your assignment packet contains code for the game server in python and starter code for your player in Python that: (1) interacts with the game server, (2) allows you to manually send moves to the server. At the server end, you are provided a GUI which allows you to visualize the proceedings of the game.

The server code can be found at: `https://github.com/NikhilGupta1997/Yinsh-AI`

6 Interaction with the Game Server

6.1 Dependencies

Python dependencies : pip install selenium jinja2 numpy
Install chromedriver : `http://chromedriver.chromium.org/downloads`
Unzip and copy it to someplace in your PATH variable. Eg : /usr/bin

6.2 Running the server

The server will run via the command: `python server.py <port>`
The additional flags are:

1. **ip** - IP Address of server
2. **n** - Board Size (N)
3. **S** - Sequence Length (K)
4. **NC** - Number of Clients (fixed 2)
5. **TL** - Time Limit
6. **LOG** - Log File

Example `python server.py 10000 -n 5 -s 5 -NC 2 -TL 120 -LOG server.log`

6.3 Running the client

cd into the client folder and run the following:
`python client.py <port-no> <server-ip> <executable> -mode <GUI/CUI>`
Open the server and client from two different terminals. The server will allow 2 clients to connect to it.

7 What to Submit?

7.1 Submission Files

Submit your code for your game player. The code should be contained in zip file named in the format `<BotName>.zip`. Please ensure this matches your bot name in the google sheet originally floated. This will be case-sensitive. Make sure that when we unzip the following files are produced:

1. `compile.sh`
2. `run.sh`
3. `writeup.txt`

You will be penalized for any submissions that do not conform to this requirement. Your code must compile and run on our VMs. They run amd64 Linux version ubuntu 12.04. You are already provided information on the compilers on those VMs. These configurations are similar to GCL machines like 'todi' (have a RAM of 16 GB).
The `writeup.txt` should have two lines as follows

1. First line should be just a number between 1 and 3. Number 1 means C++. Number 2 means Java and Number 3 means Python.

2. Second line should mention names of all students you discussed/collaborated with (see guidelines on collaboration vs. cheating on the course home page). If you never discussed the assignment with anyone else say None.

After these first two lines you are welcome to write something about your code, though this is not necessary.

7.2 Code Verification

Your submission will be auto-graded. This means that it is absolutely essential to make sure that your code follows the input/output specifications of the assignment. Failure to follow any instruction will incur a significant penalty. You will be able to check if your submissions comply with the proper format and runs as expected in the test environment by submitting to Moodle. The auto-grading in Moodle will run your bot against the RandomPlayer.py in the same environment as the final evaluation and display the game output.

8 Deep Learning based Solutions

GPU's will not be available in the test environment

For Python, only Pytorch 0.4.0 will be allowed (Only Python 3 will be allowed) For C++, only Tensorflow will be allowed (version is not yet decided) For Java, we are unable to support any DL based libraries

If you need to use any library not provided by the default python installation, please post it on Piazza (along with the version number), for us to confirm if you can use it. (Pytorch has numpy as a pre-requisite, hence it will be present in the server.)

Multiple-versions of the same library cannot be supported. In case there are conflicting requirements from the students, the one with the latest version will be chosen.

This is the first time we are allowing DL based solutions, so this is new for us too and we will need your help in figuring out the details. So please be active on Piazza, and let us know if there are any issues.

9 What is Allowed / Not Allowed?

1. **THE LATE SUBMISSION OF THIS ASSIGNMENT IS NOT ALLOWED.**
2. You must use either the same partner as in Phase 1. Or you may work alone. In case your partner has withdrawn or has audited (and does not want to submit the assignment) you can find another such partner, but you must make sure that your code is not caught in plagiarism against other codes in this assignment.
3. While you are allowed one of three languages we recommend that you use C++ since it produces the most efficient code. This assignment requires you to play the best game within a given time constraint.
4. Your code must be your own. You are not to take guidance from any general purpose AI code or problem specific code meant to solve this or related problem.
5. It is preferable to develop your algorithm using your own efforts. However, we will not stop you from google searching.
6. You must not discuss this assignment with anyone outside the class. Make sure you mention the names in your write-up in case you discuss with anyone from within the class outside your team. Please read academic integrity guidelines on the course home page and follow them carefully.
7. You get a zero if your player does not match with the interaction guidelines in this document.
8. We will run plagiarism detection software. Any team found guilty will be awarded a suitable penalty as per IIT rules.