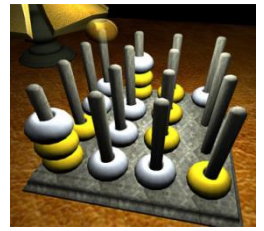


Undergrad AI Capstone NYCU Spr2021 Group Game Project #1 Due 5/4/2021

The objective of this assignment is for you to design a game-playing agent. The game to be played is a variation of 3-d connect-4. (A picture illustrating the original game is shown to the right.) This is a turn-taking, deterministic, perfect-information, and zero-sum game.



The game rules:

- The game board is also shown to the right. It has 24 cells; the cell indices are given.
- Game pieces can be stacked, so each cell can hold multiple pieces. The maximum number of game pieces in a cell is 6.
- The objective is to form four-in-a-lines. Four-in-a-lines can be along any direction, and they can have shared pieces. Game points are award for four-in-a-lines formed in the game. The k^{th} four-in-a-line formed gets $\lfloor 100/k \rfloor$ points.
- In each move, a player selects one cell that is not full yet to "drop" a game piece in. The new game piece will be at the top of the stack in that cell.
- Each player has 32 game pieces, so the depth of the game is 64 moves in total.
- The player who accumulates more game points wins the game.

		1	2		
	3	4	5	6	
7	8	9	10	11	12
13	14	15	16	17	18
	19	20	21	22	
		23	24		

The tournament:

- Each pair of teams will play two games, with each of the two teams starting the game once. A win, draw, and loss count as two, one, and zero tournament points, respectively.
- Ranking is based on the total tournament points of the teams. Total game point differentials are used as tie-breakers for teams having the same total tournament points.
- The ranking will factor into about 25% of your grade for this project.

Regarding the algorithm:

You have a lot of flexibility in designing your game agent. It can be as simple as a set of rules. You can try the classical method of minimax search, possibly with alpha-beta pruning. You can also try to implement MCTS, or to train your agent using reinforcement learning. The requirement is that you need to implement the algorithms yourself; you cannot use modules/libraries developed by others for game playing. When the TAs run the tournament, the game server and both player programs will run on the same computer. There will be no outside connectivity and no GPU support, and there will be a time limit for each move.

Regarding the implementation:

- You can only implement the program in C++ or Python 3. The environments will be posted by the TAs.
- The communication between the game server and your program, running as a client, is via TCP. The TAs will provide instructions and sample codes on including the communication capabilities in your program.
- The TAs will provide a sample server program for you to use during the development of your program.

Teams:

- The students should form teams of 1-3 members.
- Once teams are formed, provide the team information (team name, members' names and IDs, and team leader) on a discussion board in New E3.
- A team ID will be assigned to each team. You need to include this ID in both your filename and within the code.

Submission:

The submission is through New E3. No late submission is accepted for this project. You should submit your program source code (all in a single file if possible) and a report file separately. The report (maximum 5 pages single-spaced) should describe how your game AI works, your experiments and experiences, and contributions of individual team members. The TAs will announce later whether you need to submit executables. Only one submission is required for each team. Remember to list the team name, ID, and members in both the source code and the report.