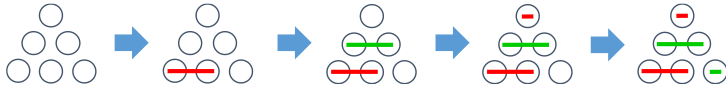


The objective of this assignment is for you to design a game-playing agent. This is an individual project on a fairly simple game used to be played by children. The "game board" is just a set of circles arranged hexagonally.

The basic game rules:

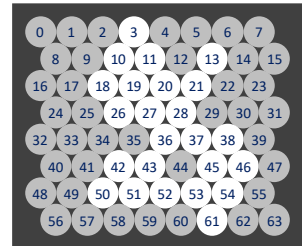
- Players take turn crossing out the cells.
- In each move, a player can choose to cross out one, two, or three remaining cells. The selected cells have to be contiguous and along a straight line.
- The player that crosses out the last remaining cell loses the game.

Below is an example sequence where the second player (green) loses:



Environment:

- The game board (and example is shown to the right) is an 8x8 region, with the numbers representing cell indices. However, only a subset of the cells (white cells) are actually used in a game (24 in the example). The subset of cells used is generated randomly for each game and always form a connected region. (The size of the subset will fall in the range of 12~32.)



Regarding the algorithm and implementation:

- 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. Endgame databases might be useful as well.
- It is required that you 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 all the 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.
- You can only implement the program in C/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 and simple client programs (to act as your opponents) for you to use during the development of your program.
- There will be a time limit for each move. If there is no response from your program within the time limit, a random valid move will be selected for you by the server program.

The tournament:

- A number of boards will be generated for the tournament. Games will be played twice (two players taking turns to start) for each pair of submissions on each board. The winner of a game gets one point. The final ranking will be based on your total points.
- The ranking, after being scaled to 60~100, will factor into one third of your grade for this project.

Submission:

- The submission is through New E3. No late submission is accepted for this project. You should submit your program source code (following instructions from the TAs) and a report file separately. The report (PDF file, maximum 5 pages single-spaced) should describe how your game AI works, your experiments and your experiences. The TAs will announce later whether you need to submit executables. Use your ID as the file name for both the report and the code.