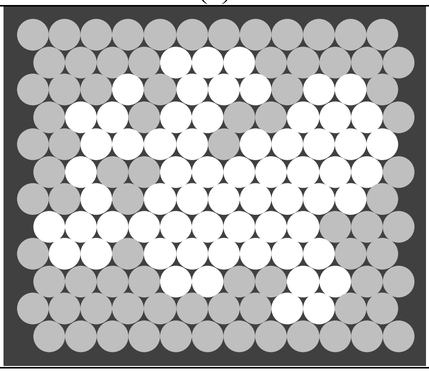
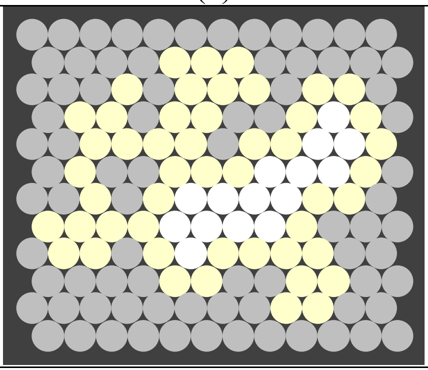
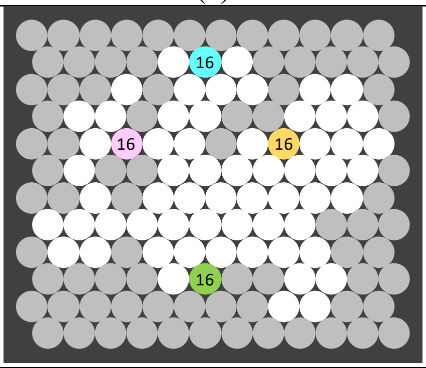
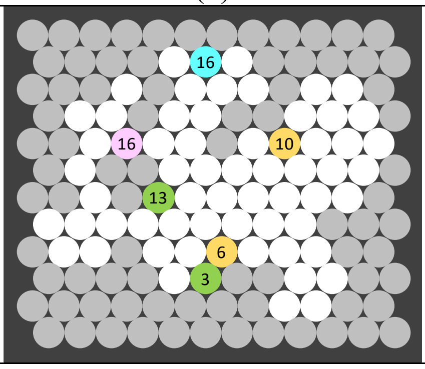
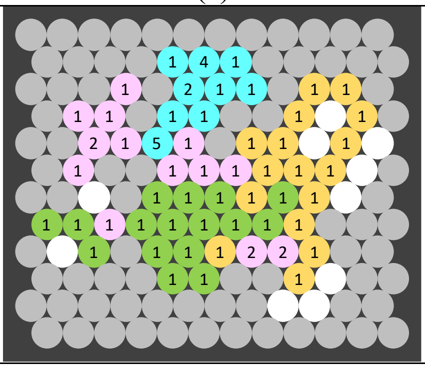
Undergraduate AI Capstone NYCU Spr2022 Group Game Project #1 Due 4/26/2022

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

<u>Description and rules of the game:</u>

- Figure (a) is an example 12x12 board. The white circles (64 in total) are the actual playing field, which is randomly initialized for each game. The playing field always forms a connected region.
- Figure (b) is the example playing field with the border cells (cells in the playing field that are at the board boundary or adjacent to non-playing-field (gray) cells) colored in yellow.
- Each player is allocated 16 sheep. The first move of a player is to place all the 16 sheep in a single border cell. An example board after the first round is in Figure (c).
- The color of a cell indicates its occupying player. The number of a cell indicates the number of the player's sheep in that cell.
- In each move, a player
  - Selects a cell with more than one sheep,
  - Splits those sheep into two non-empty groups,
  - Moves one group along a line of unoccupied cells <u>as far as it can go</u> (no stopping until it cannot move any further, and no hopping over occupied cells or non-playing-field cells), and leaves the other group at the original place.
- Example moves of two of the players are shown in Figure (d).
- A player has to "pass" (skip the move) if no valid move exists.
- Score of a player = (# occupied cells) * 3 + (size of largest connected region of a player's cells).
- Figure (e) is an example terminal state of the game, with the resulting scores given in Figure (f).

(a)	(b)	(c)
		
(d)	**(e)**	**(f)**
		🟡 = 16 * 3 + 15 = 63 🟢 = 16 * 3 + 13 = 61 🟣 = 13 * 3 + 6 = 45 🔵 = 9 * 3 + 9 = 36

<u>Regarding the algorithm and implementation:</u>

- 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. Since there are 4 players in the game, your game tree will grow such that each round consists of four layers. The decisions / evaluations made at each layer is based on that particular player's perspective.

- 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++ 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.

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.

The tournament:

- For each game, the players with the 1st, 2nd, 3rd, and 4th highest scores receive 5, 3, 2, and 1 tournament points, respectively. When there are ties, the tournament points are evenly distributed among those players with tied scores.
- Each team's program will be played for the same number of games, with the same number of games in each of the 4 starting positions. The player combination of the games will be arranged randomly.
- Ranking is based on the total tournament points of the teams. Total game scores are used as tie-breakers for teams having the same total tournament points.
- The ranking will factor into 25% 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 (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.