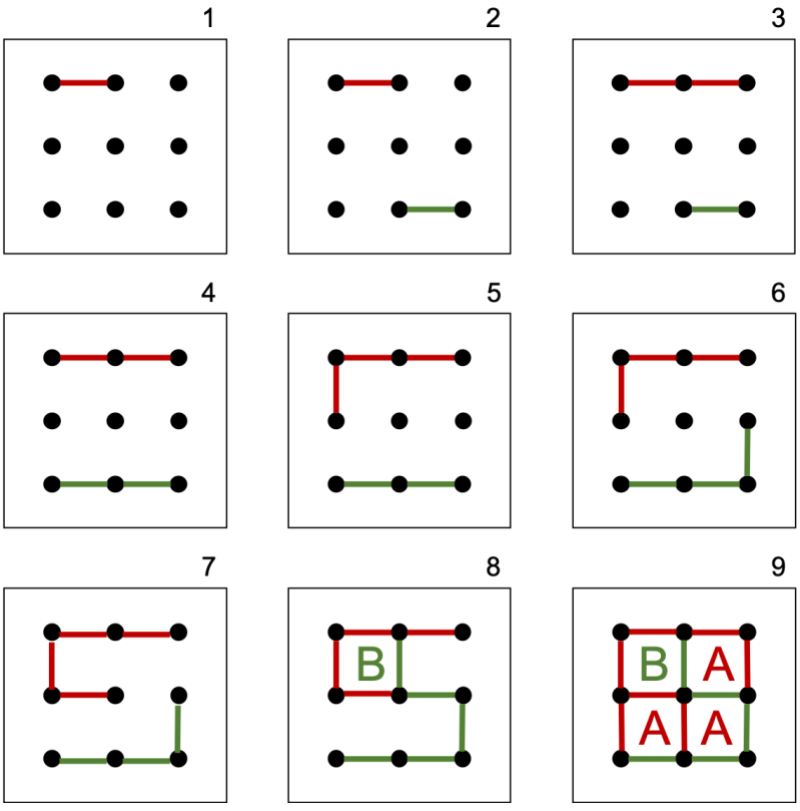
CS102A Course Project, Spring 2019: Dots and Boxes

Descriptions

Dots and Boxes is a game for two players. The game starts with an empty grid of dots. Usually two players take turns adding a single horizontal or vertical line between two unjoined adjacent dots. A player who completes the fourth side of a 1×1 box earns one point and continues with another turn until he couldn't get more points. A point is typically recorded by placing a mark that identifies the player in the box, such as an initial of the player's name. The game ends when no more lines can be placed. The winner is the player with the most points.

The diagram below shows a game being played on a 2×2 board (3×3 dots). The player A starts the game and adds a line in Step 1 (the lines added by A are in red color). Then the player B adds a line in Step 2 (the lines added by B are in green color). B earns one point in Step 8 by completing the first box and adds another line. However, this is a bad move. A is able to earn three points in Step 9. The winner is A. You can play an example game at http://www.papg.com/dab.html to get familiar with the rules.



Source: https://en.wikipedia.org/wiki/Dots_and_Boxes)

Project Requirements

- Please design and implement a Java program to simulate the Dots and Boxes game for two players. Note that the only programming language you can use in this project is Java.
- The size of board should be customizable. Your program should support at least the following sizes: 3×3 dots, 4×4 dots, and 5×5 dots.
- Your program should use different kinds of lines (for example, in different shape or colors) to identify the moves made by different players.
- When a player earns one point, a character (or characters) representing the player must be displayed inside the completed 1×1 box.
- When a game is over, the result must be displayed on screen. If there is a winner, the winner's name should be displayed. If it is a draw, your program can simply display a message "Game over, no winner".
- There are five tasks descried below. Two-member groups should finish the first three tasks. Three-member groups should finish all five tasks.

Task 1. Machine vs. Machine Mode (30 points for two-member groups, 20 points for three-member groups)

- Your program should be able to simulate a game between two machine players.
- There is no requirement for implementing strategies. The machine players can randomly select a possible move during a game.
- Your program should display the status of the game for each step. Displaying only a final result is not acceptable.

Task 2. Human vs. Human Mode (30 points for two-member groups, 20 points for three-member groups)

- Your program should be able to simulate a game between two human players.
- Your program should prompt a human player to make a move when it is his/her turn. Your program should handle invalid moves by human players, for example adding a line that already exists.
- If your program only has a command-line interface, human players can input the coordinate of two points to represent the line he/she wants to add.
- If your program has a graphical user interface, you should design a more natural way to let the human players connect dots (for example, by mouse clicks on the possible lines).

Task 3. Human vs. Machine Mode (20 points for two-member groups, 15 points for three-member groups)

- Following the requirements in Task 1 & Task 2, you should further make your program support games between a human player and a machine.
- Your program should allow the human player to choose whether he or she wants to make the first move.

Task 4. Game Status Saving (10 points for three-member groups)

- This is a requirement for three-member groups.
- Your program should be able to save the status of an ongoing game to disk. You can design a data format to save the game status to a file. When users start the game, they should be able to load a saved game and continue to play it. You don't need to implement the functionality that a user can only load the games saved by himself/herself.
- Your program should be able to store the score of different human players and display the leaderboard.

Task 5. Graphical User Interface (15 points for three-member groups)

- This is a requirement for three-member groups.
- Your program should have a graphical user interface (GUI). An easy way to implement the GUI is to use the `StdDraw` class provided by us.

Bonus

If your program satisfies all the above basic requirements, you will get 80 points. The remaining 20 points will be given as bonus. You are highly encouraged to go beyond our requirements. Below are some possible ways to get bonus.

- The game provides more interesting kinds of boards
- Make the machine player smart (you can find various strategies online)
- When loading games, the players can only load games saved by themselves
- The game supports undo function (a human player may regret after making a bad move and wish to retract the move)

Be creative and have fun ☺