Tic Tac Toe

The goal of this project is to practice more with classes, graphics, and game simulation.

Deliverables: your solution file named `user1_user2.py`. Don't forget to include a comment at the top of your .py file with both authors' names! In addition to all of the usual coding standard, you should use actual doc strings for each of your functions (comments in triple quotes on the line immediately after your function definition).

Your task

Implement the game Tic Tac Toe using one of the graphics packages provided (Zelle or Pygame). You may choose either graphics system, the expectations/grading will be the same for both. You may not use any other graphics options for this assignment; to run your program I should only need pygame installed and/or graphics.py in the correct folder. Your program should allow 2 users to play a basic version of the game, each clicking in turn on an empty square until one of them wins or the board is full, at which point an end-screen will state the winner if there is one and show the winning row/col/diag.

Basic Game Play

You should display an initial empty 3x3 grid to start your game. The first user may then click on any of the 9 squares and their letter (X or O) will appear in that square. Each turn alternates between players clicking squares and their letter showing. After each click you should

1. Check if the click was valid - within the bounds of the grid and in a currently empty square. If the click is invalid you should display the "invalid" end-screen.
2. If click is valid, display the correct letter (X or O) in the clicked square.
3. Check if game is over - either a player has won with 3-in-a-row, or all squares are full without any winner. If game is not over, go back to step 1 and repeat for other player. If game is over, display correct end-screen for reason game ended, as described below.

End Screens

When the game ends, you should display a specific end screen and no longer accept any user clicks except for the upper left x to close the window. The options are

1. Invalid click - if one of the player made an invalid click, the endscreen should display the game grid as it was when the invalid click occurred, but also show with a red circle where the user actually clicked, and text somewhere stating "invalid click".
2. Game Over - all squares full with no winner. The end screen should display the game grid as it was when the last square was filled, except change the X or O in the last square to be red, showing the last move that was made. It should also display text somewhere stating "Tie Game".
3. Game Won - a player got 3 in a row. The end screen should display the game grid as it was when the 3 in a row was completed, except change the X's or O's in the 3 in a row to be red, showing where the winning 3 were. It should also display text somewhere stating "Player * wins!" where * is either X or O as appropriate.

Bonus

Add an option for a player to play against a very simple AI instead of another player. When the game first loads display 2 "buttons" - square outlines with text inside, one that says "1-player" and one that says "2-player". If the user clicks inside the "2-player" square the game continues as normal, but if they click "1-player" they will play against the computer. Your AI should apply the following rules, in the order given.

1. If the computer has 2-in-a-row anywhere on the board and the 3rd square to complete the triple is empty, play in that square to win!
2. If the computer can't win, check to see if the user-player has 2-in-a-row anywhere on the board and the 3rd square to complete the triple is empty. If so, play a "block" in the square that would allow the user to win. If the user has multiple 2-in-a-rows, it doesn't matter which you block, just block at least one of them.
3. If neither of rules 1 or 2 is triggered, just play in a random empty square.