Math 121 # Project 2 instructions Fall 2022

Four-in-a-row Tic Tac Toe

- Due by: Monday Dec. 19th 11:59pm, on Gradescope.
- four-in-row.py (If I run this file, it should implement your project)
- (whatever py files your main file imports, except graphics.py)

Students may work by themselves or with one other person. If working in a group of two, just one person needs to submit on Gradescope and you can tag your partner in your submission.

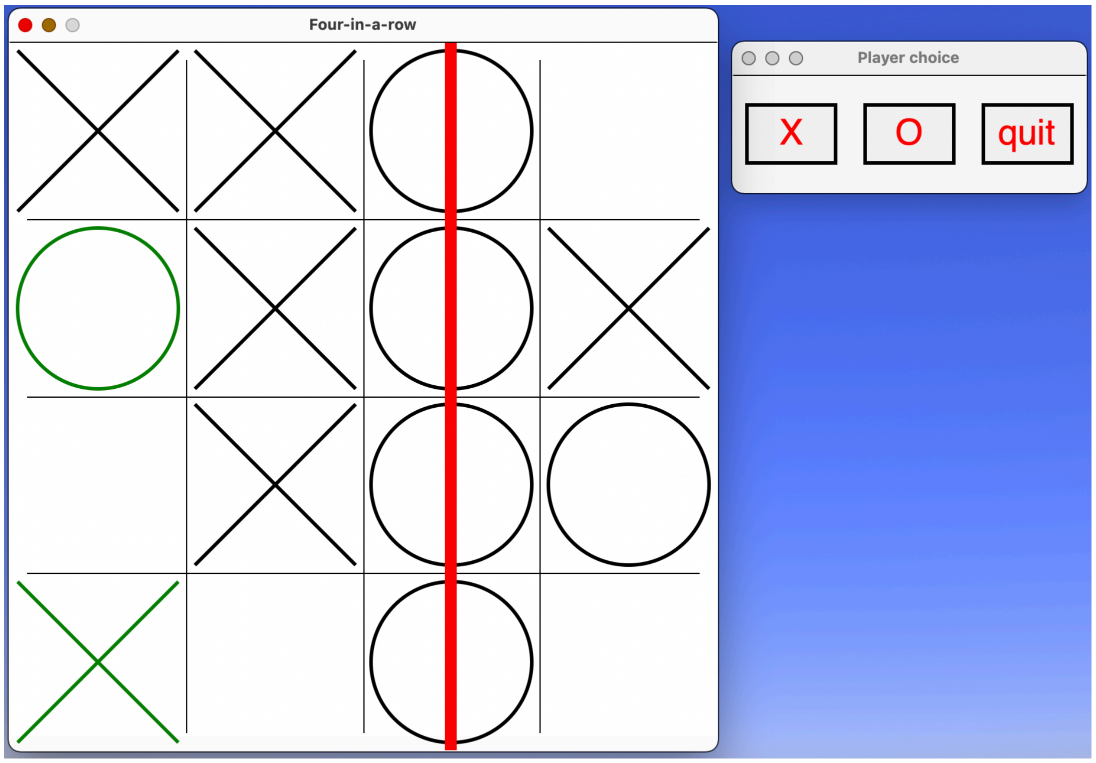
Objective

Create a two-player version of a four-in-a-row Tic Tac Toe game. Please see the assessment below for basic requirements, but feel free to implement them in your own, creative ways. You may also refer to the tic-tac-toe.py file for 3x3 on Slack posted with this project.

The Game

In a 4-in-a-row Tic Tac Toe, two players start out with a 4-by-4 grid. Just like in classic Tic Tac Toe, player X goes first. In 4-in-a-row version:

a. A player wins if they have four of their markers in one row, or one column, or one diagonal;
b. Players only have six markers. Whenever a player already has six markers on the board: the player specifies an empty square, and the oldest marker gets moved to the new location.



If you decide to implement a bot player, I strongly recommend first implementing a two-human-players version of the game, before attempting to implement a bot as one of the players.

Assessment

Feature	Description	Points
Basic prototype	The program has at least a working basic prototype: there is a 4-by-4 grid, and clicking on an empty square on the grid results in a drawing that resembles either an X or an O. X and O take turns. Clicking on an already claimed square should not result in anything.	40
6 marker limit	If a player already has six markers on the board: once the player clicks on an empty square, the oldest marker gets moved there. (The square where that marker used to be is now empty, and can be claimed in the future.)	15
Winner recognition	If a player makes a winning move (4 markers in a row, 4 markers in a column, or 4 markers in a diagonal), the game correctly recognizes the winner.	10
Winning line	If a player makes a winning move, it gets correctly marked by a thick line.	5
Ability to continue playing	After some player won the game, the program graphically presents the user with a choice to play again, or to quit the game entirely.	5
BONUS: One player is a (somewhat) intelligent bot	An intelligent bot would do at least the following moves: (1) If it's possible for the bot to win on this turn, do so; (2) If the human player will win on their turn, prevent it; (3) else: randomly choose an empty square.	+10

Code	Description	Points
Basic code requirements	Introductory comment specifying author and date. All import statements happen right after the introductory comment.	5
Code clarity	Code is clear and easy to follow. Uses reasonably descriptive variable and function names and informative comments.	10
Code encapsulation	Good use of self-defined functions (or methods in self-defined classes) to keep each function's (or each class method's) code easy to understand and at reasonable length.	10
	Total	110