

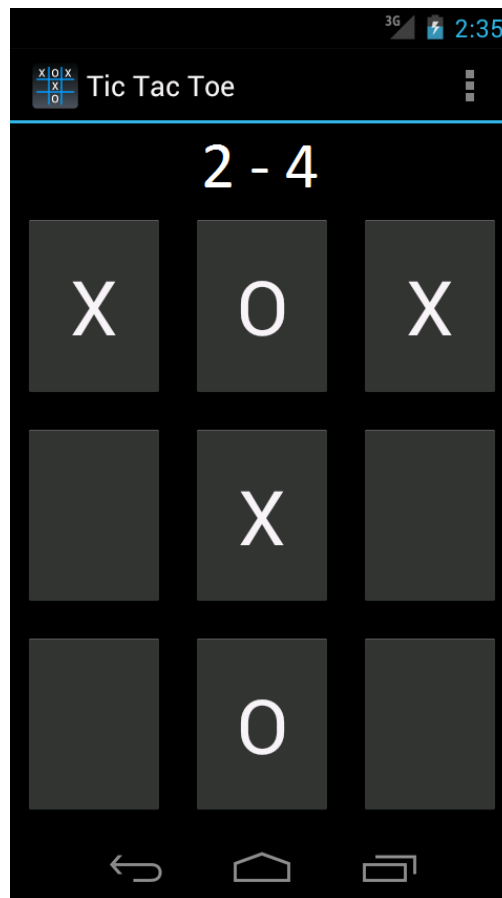
Using a 3x3 Grid Layout (or any layout which will correctly and dynamically resize on different screen resolutions) containing 9 buttons, implement a simple game of Tic-Tac-Toe.

The buttons can be in 2 states:

- **Empty**: the button was not pressed, it does not have any character (**X** or **O**) on it and is clickable

- **Full**: the button was pressed and either **X** or **O** appears on it. The button is henceforth until the end of the match disabled and nothing happens if the user presses it again.

As it is a turn based game, you will have to implement the logic behind switching between **X** and **O**'s turn. After an **X** is placed on the board, the next button clicked will have a **O** on itself and vice-versa. You will also have to keep track of the score for player 1 (which always starts first) and player 2. The scores will be displayed above the board like in the picture shown below:



Once a game finishes, a *Toast* with the result of the match is shown (e.g. “*The game ended in a draw*” or “*X wins!*”) The score will also be updated accordingly. The left side score represents X’s wins and the right side represents O’s wins. **X always starts first.**

Twist 1:

On the action bar, implement the following actions:

- Reset match
- Reset score
- Exit application

Twist 2:

Implement an AI and on game start, pop a dialog (maybe using *AlertDialog*) asking the player whether or not they wish to play against a computer. If the user presses “yes” then the game will be played against a computer and if the user presses “no”, the game will be a turn-based two player game. Add an action to the action bar with the name “*Switch game mods*” which switches between *vs. Computer* and *vs. Other Player* mode. When pressed, the action bar option will pop a confirmation dialog asking the user if he is sure he wants to switch game modes as this will reset the score and current match. (“*Are you sure you want to switch game modes? The existing match and scores will be reset!*”) If the user presses “yes” then the game mode is switch, else, the dialog is closed and nothing happens. At this stage, the AI can be imperfect (it doesn’t have to make the best move at all times)

Twist 3:

Save the current state of the game (board data and score) on orientation change (whenever the phone is turned from landscape to portrait or vice-versa)

Twist 4:

Save the score in *Shared Preferences* and always fetch it from there. Once the *Reset match* option in the action bar is clicked, the data in *Shared Preferences* is reset.

Twist 5:

Implement a perfect AI for the second twist. Read more about it [here](#) and [here](#)!

Feel free to design the game in any way you want. You can even use Image Buttons for the buttons instead of normal buttons and add some image to them (either **X** or **O**). Also, feel free to design the background in any way you want – use your imagination! However, it is fine if you simply implement it just like in the picture.