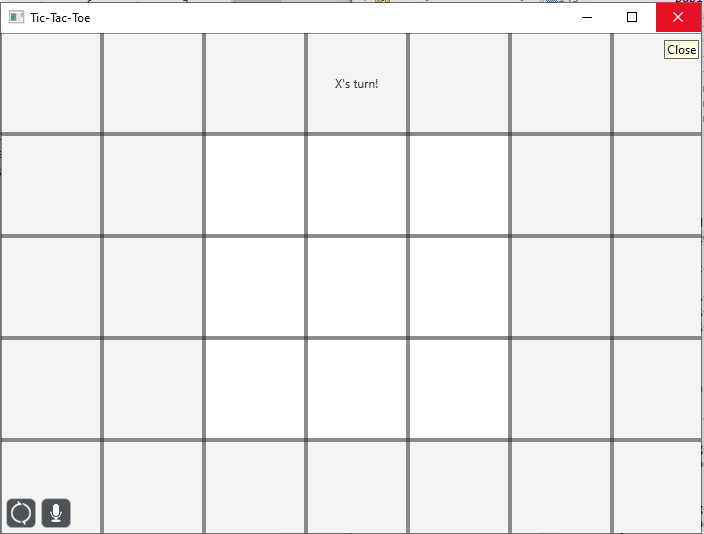
# Solution Approach

My solution to the Tic-Tac-Toe game was to use ImageViews to display the interactable game tiles. They at first would be rendered as a blank jpeg, but after clicking it would display the X or O picture. This is how I setup the game tiles. These game tiles are then inside a 7 column by 5 row grid pane. This allow me to have the nine needed tiles center nicely in the window. The GUI construction was pretty straight forward from here. Create an ImageView for each column and row index being used by our game and set it to blank. Handle the controller for each to interact with the TicTacToe class based on their location of the gameboard.



## Handling Game Logic

The onMouseClicked function for each ImageView is similar in following the same pattern. This was to make sure the game was not over, confirm the tile being clicked is open, and see if it is X or O’s turn based on turn. If turn int is odd then it is X, if even then O. After updating the image corresponding to the ImageView it will again check to see if the game is over. If it is it will then make the reset button available and announce the winner.

When this code is being executed inside TicTacToe it is using the game rules to determine its return value. I validated these game rules by creating a 2D string array which would be my tic tac toe game board. This would be the backend providing information to the frontend GUI on what to display.

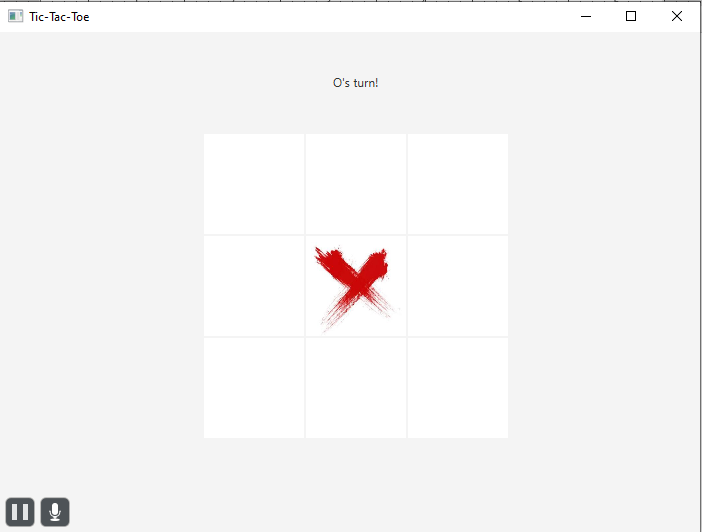
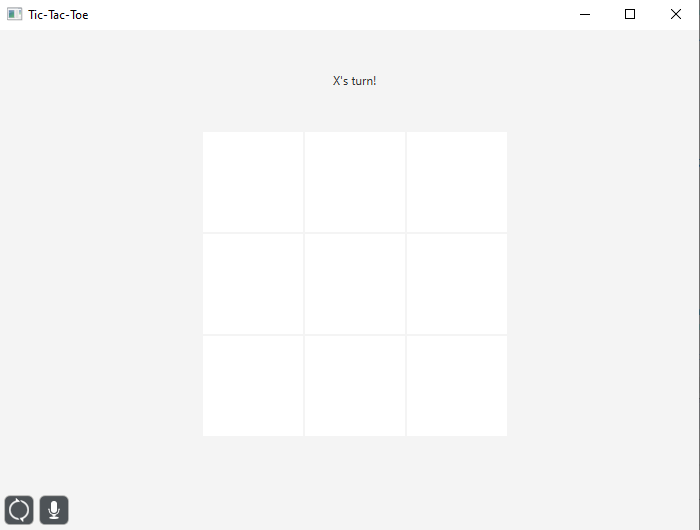
#### GetIsOver

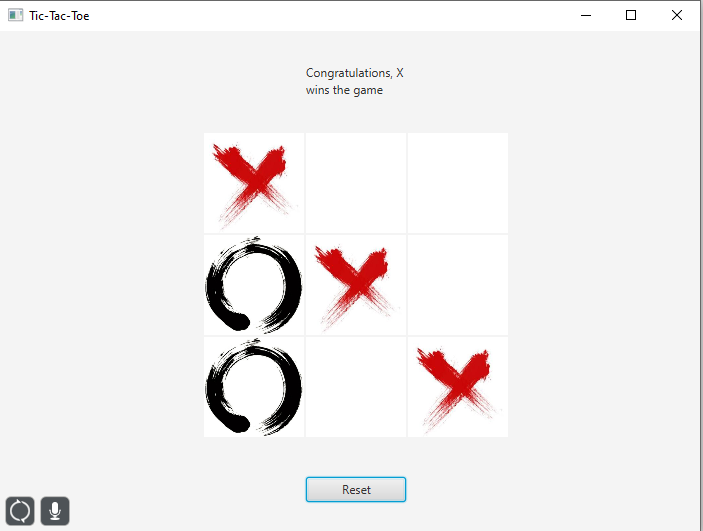
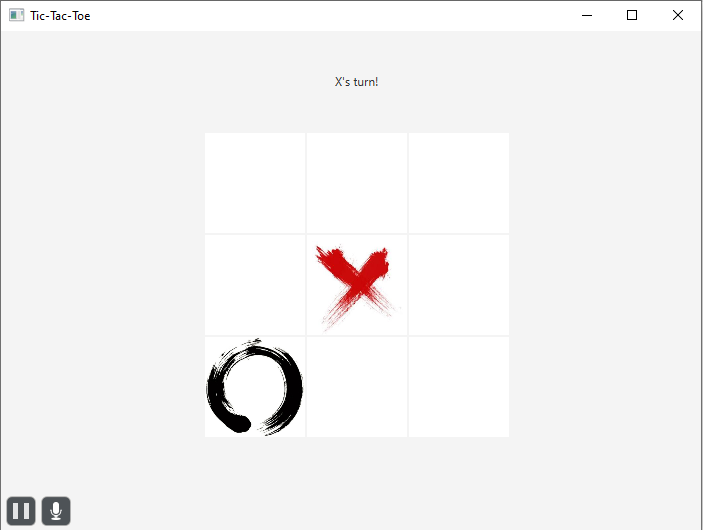
To check if a given game is over all vertical, horizontal, and diagonals must be checked each turn for 3 in a row. Also, if 10 turns have passed the game is not over, it now is a draw.

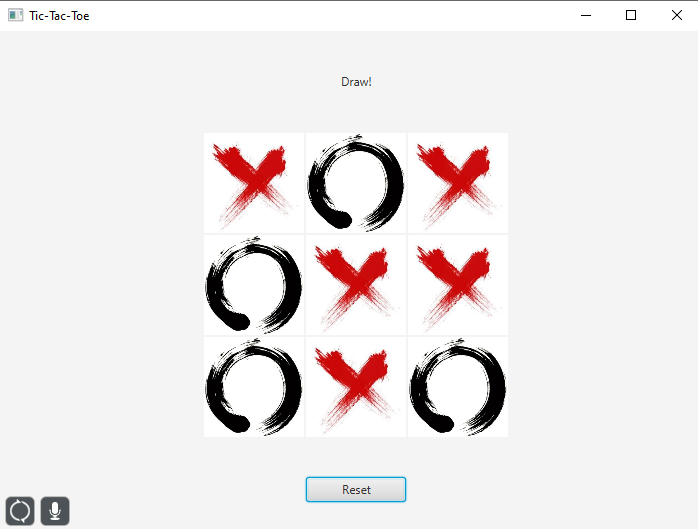
#### IsOpen

This method is to verify the tile you are interacting with has not already been set. It does this by checking the 2D string array to verify it is not an “X” or “O”.

# Screenshots of Check list Requirements







# Project Structure

