High-Level plan for our program

1. First, we will create 3 things outside the main method or any other method. These include the board, playerSymbol and comSymbol.

* Board – This will be our “Board for the programmers. We plan to use this to call the graphics, so we know where to place everything. This board will be a 2-dimensional 3x3 board simulating a tic tac toe board. We plan to make this of char type as it will be easy for us to read the variable with our drawstring Methods without having the black bar asking for the player to enter the number and then hit enter which seems like a thing from the 1900’s. It is much more convenient to just press a key on the board and instantly the computer goes to work.

{{ , , }

{ , , }

{ , , }}

* playerSymbol – This will be our variable for the player. It is a final variable as it cannot be changed as we plan to give the player a definite choice of being the ‘X’.
* comSymbol – This will be our variable for the computer. It is a final variable as it cannot be changed as the computer will always be ‘O’.

1. Next is the main method. This will be the place where everything will be called and run. This is the place for the entire game.
2. Background method – The method where the colour of the background will be set. This is because Ready Java initially doesn’t give us a method for setting the background. This method will not have any parameters as we don’t need any. The background will always be the same. Also, this method will not have a return type as we will draw directly on the output page.
3. drawSymbol\_O method – This is the method that will be called on to draw the ‘O’ onto any place on the output page just by giving the x and y value for the top left-hand corner. Also, this method will not have a return type as we will draw directly on the output page.
4. drawSymbol\_x method – Similarly we have a method to draw the ‘X’ onto any place on the output page. Also, this method will not have a return type as we will draw directly on the output page.
5. introPage method – This method gives an intro to our game. It begins by displaying the title and asks the player to press any key to continue. This also doesn’t have a return type as we do not plan to return anything and will just display directly on the output page.
6. gameType method – This method asks the player what intensity of the game does he/she want to play. They have 3 options: easy, medium or hard. This method doesn’t have any parameters as we don’t need anything, but it gives back a value 1, 2 or 3 depending on what the player chose.
7. drawBoard method – This method draws the board whenever the player chooses to play the game, may it be in the beginning or after a round has finished.
8. firstTurn method – This method asks the player if he/she would like to go first or should the computer go first. Then it returns a value of 1 (meaning they want to go) or 2 (let the computer go first).
9. clearInput method – This method clears the input of the text at the bottom on the output page.
10. playerMove method – This method asks the player to make a move by entering a number in type char and returning it in an integer value.
11. convertToPos method – This method takes the inputted position (a parameter) and calculates where exactly the location is in rows and columns. We needed this because we ask the player to only input a number 1-9 as it will save time as the player won’t have to find the row and column and enter 2 numbers. We feel this because we ourselves sometimes forget which one is row and which one is column. So, we decided to give numbers 1-9 to the 9 boxes. It is also convenient and time consuming as only one number must be entered.
12. changeTurn method – This method takes the inputted value of whose turn it is and determines whose turn it will be next.
13. comRandomMove – This method is for the computer to make a random move for the easy level only. It returns a int[] array with the rows and columns. This form of the game is very easy as the computer is not using any algorithm to determine a good move.
14. makeOptimalMove – This method is for the computer to make a decent move that will not make the computer lose easily. It calls 2 methods called checkComWin and checkPlayerWin to see if the move will make the computer win or the player win or none.
15. makePerfectMove – This method is like the optimal move method, but this is very precise and will make sure there is no way for the computer to lose.
16. checkComWin – This method uses the board array and determines if there is any possible move that will make the computer directly win in the columns, rows and the 2 diagonals. Then it returns an array with the info of the row and column that we can put to win.
17. checkPlayerWin – This method uses the board array and finds out if there are any possible moves for the player that he/she may make and win. Then it returns an array with the info including where they will place the move to win.
18. gameOverChecker – This method checks if the game is over and returns a type Boolean (true or false).
19. tieChecker – This method checks if the game is a tie or not and returns a type Boolean (if yes then true or if no then false).
20. clearBoard – This method clears the board by drawing a rectangle over the previous one.
21. displayResult – This method displays the result of who won by taking an input of char type and drawing strings on the output page.
22. playAgain – This method asks the player if he/she wants to play again and returns a Boolean value (true meaning yes and false meaning no).
23. exitScreen – This method shows a thanks for playing sign.