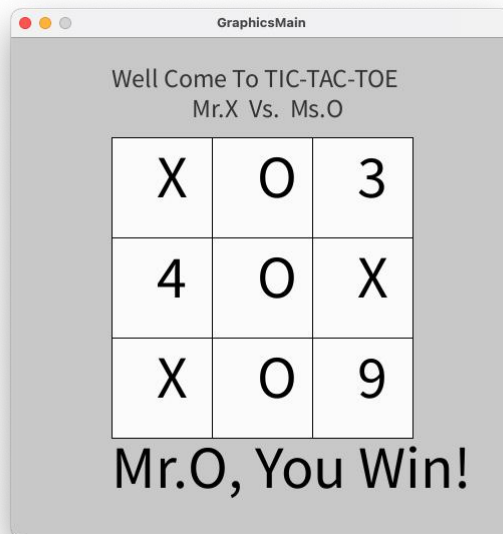
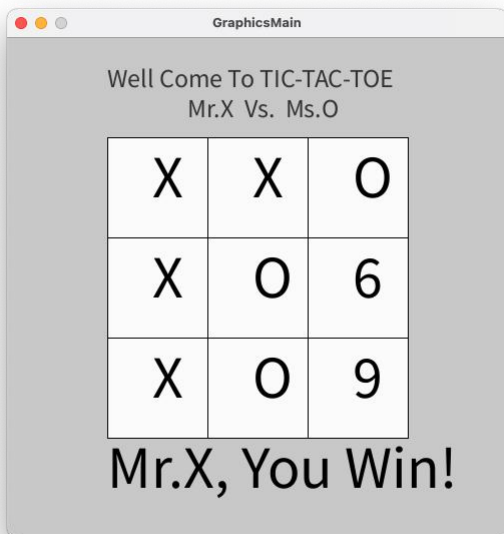
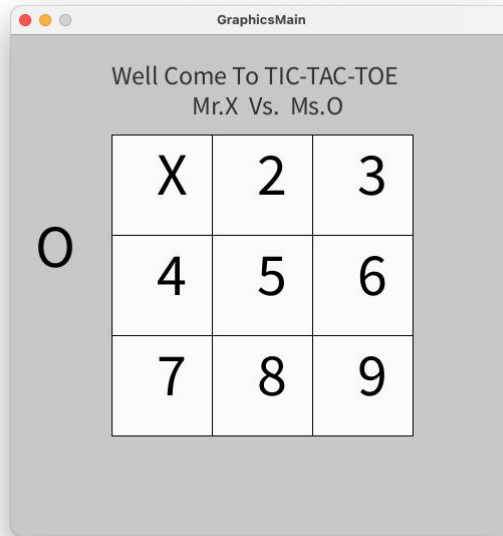
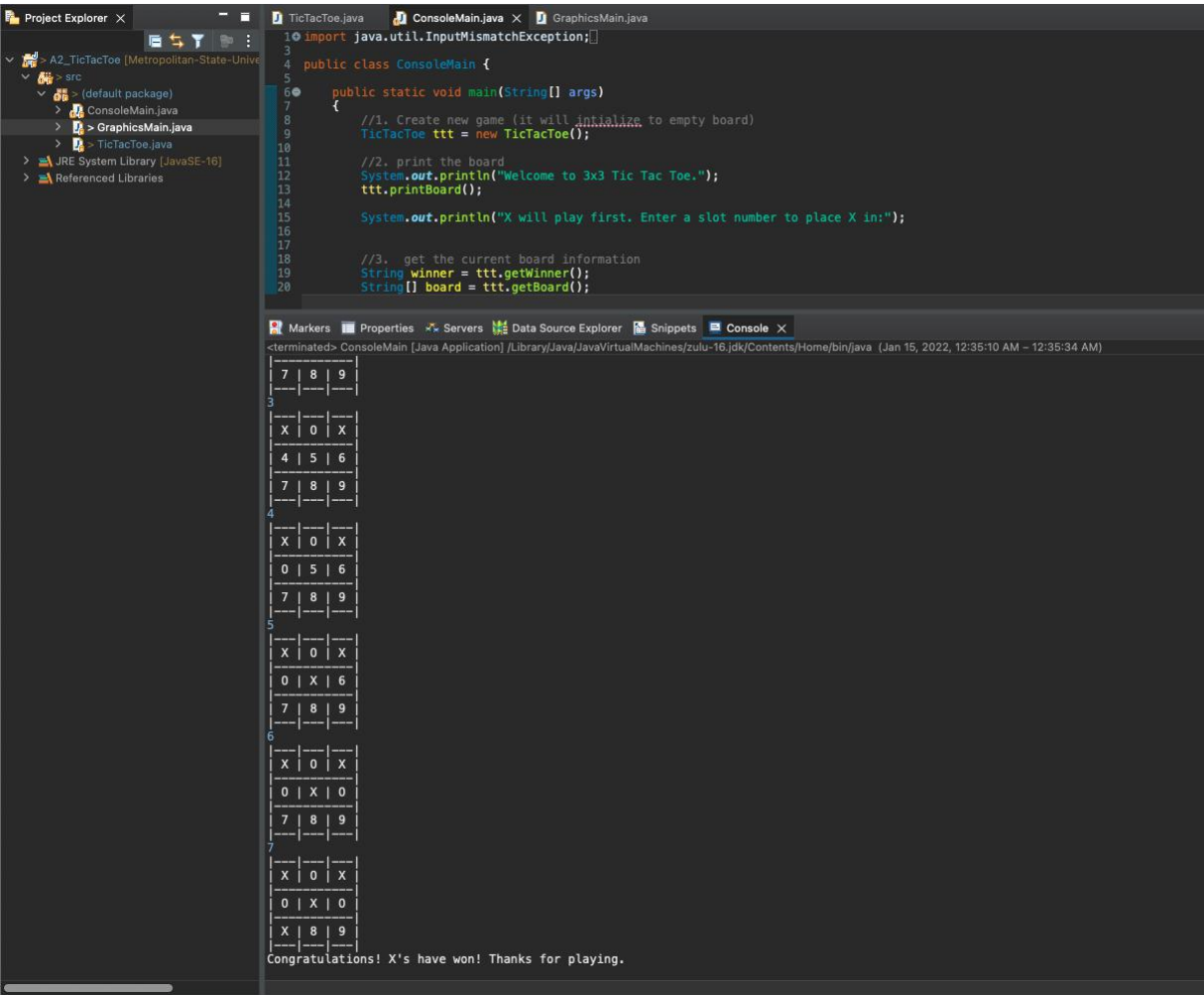
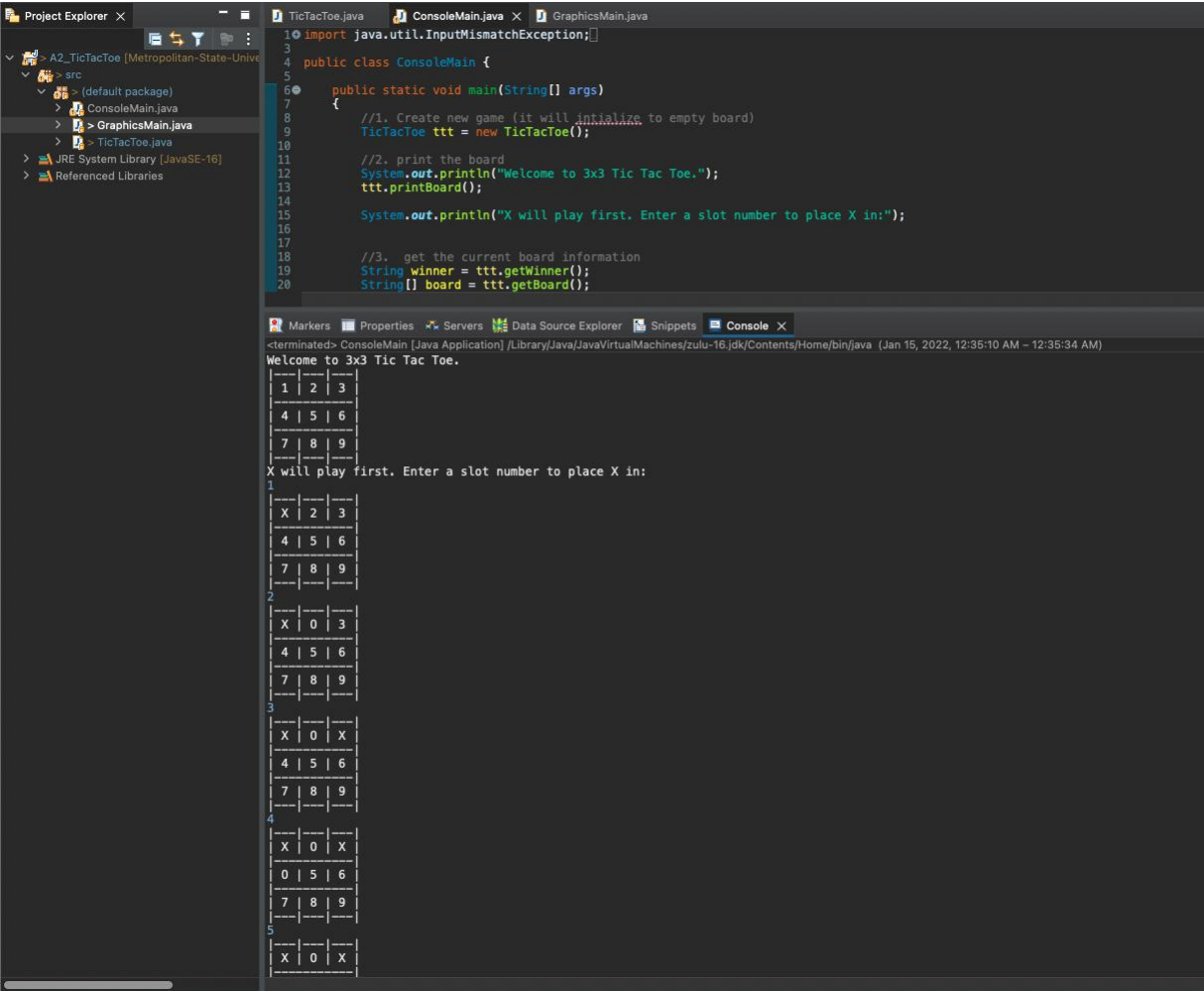


Assignment #1

The result of GraphicsMain.java:



The result of ConsoleMain.java:



GraphicsMain.java

```
Project Explorer X
A2_TicTacToe [Metropolitan-State-Univ]
  > src
    > (default package)
      > ConsoleMain.java
      > GraphicsMain.java
      > TicTacToe.java
    > JRE System Library [JavaSE-16]
    > Referenced Libraries

GraphicsMain.java X
1
2 import processing.core.PApplet;
3
4 * Edit and Update By Nalonsone.Randankh
11 * The GraphicsMain class will extend PApplet. It should have the methods
19 public class GraphicsMain extends PApplet {
20
21 // declare any variables you need
22 TicTacToe ticTacToe;
23 Box[] boxes = new Box[9];
24 int clickControls[] = new int[9];
25
26 public static void main(String[] args) {}
27
28
29
30 public void settings() {
31   size(500, 500);
32 }
33
34 public void setup() {
35   // TODO: create the tic tac toe board.
36   // create and initialize every box by fixed size.
37   for (int i = 0; i < 9; i++) {
38     boxes[i] = new Box(100 + 100 * (i % 3), 100 + 100 * (i / 3));
39   }
40   // Create new game (it will initialize to empty board)
41   ticTacToe = new TicTacToe();
42   textSize(26);
43   fill(50);
44   text("Well Come To TIC-TAC-TOE", 100, 50);
45 }
46
47 public void draw() {
48   // TODO: handle the events
49   background(200);
50   textSize(26);
51   fill(50);
52   text("Well Come To TIC-TAC-TOE", 100, 50);
53   text("Mr.X Vs. Ms.O", 180, 80);
54   for (int i = 0; i < 9; i++) {
55     // build, update and display every box.
56     boxes[i].drawBox(mouseX, mouseY);
57     boxes[i].updateBox(i);
58   }
59   // Keep checking for the winner to display a message
60   // X or O or draw
61   // if the game get winner or done all box.
62   // stop the click event and print the result.
63   String checking = ticTacToe.checkWinner();
64   if (!checking.equals("None")) {
65     for (int j = 0; j < 9; j++) {
66       clickControls[j] = 10000;
67     }
68     if (checking.equals("draw")) {
69       text("Game Over!!!", 100, 450);
70     } else if (checking.equals("X")) {
71       text("Mr.X, You Win!", 100, 450);
72     } else if (checking.equals("O")) {
73       text("Mr.O, You Win!", 100, 450);
74     }
75   } else {
76     text(ticTacToe.getTurn(), mouseX, mouseY);
77   }
78 }
79
80 // based on where the user clicked, update the display
81 // place X or O on the board
82 public void mouseClicked() {
83   // click each mouse clicked position to handle the event and update the Board
84 }
```

```
Project Explorer X
A2_TicTacToe [Metropolitan-State-Univ]
  > src
    > (default package)
      > ConsoleMain.java
      > GraphicsMain.java
      > TicTacToe.java
    > JRE System Library [JavaSE-16]
    > Referenced Libraries

GraphicsMain.java X
62 // stop the click event and print the result.
63 String checking = ticTacToe.checkWinner();
64 if (!checking.equals("None")) {
65   for (int j = 0; j < 9; j++) {
66     clickControls[j] = 10000;
67   }
68   if (checking.equals("draw")) {
69     text("Game Over!!!", 100, 450);
70   } else if (checking.equals("X")) {
71     text("Mr.X, You Win!", 100, 450);
72   } else if (checking.equals("O")) {
73     text("Mr.O, You Win!", 100, 450);
74   }
75 } else {
76   text(ticTacToe.getTurn(), mouseX, mouseY);
77 }
78
79 // based on where the user clicked, update the display
80 // place X or O on the board
81
82 public void mouseClicked() {
83   // click each mouse clicked position to handle the event and update the Board
84   // then turn user to another.
85   for (int i = 0; i < 9; i++) {
86     if ((mouseX > clickControls[i] + 100 + (100 * (i % 3)) && mouseX < clickControls[i] + 200 + (100 * (i % 3)))
87         && (mouseY > clickControls[i] + 100 + (100 * (i / 3))
88             && mouseY < clickControls[i] + 200 + (100 * (i / 3)))) {
89       ticTacToe.getBoard()[i] = ticTacToe.getTurn();
90       if (ticTacToe.getTurn().equals("X")) {
91         ticTacToe.setTurn("O");
92       } else {
93         ticTacToe.setTurn("X");
94       }
95       clickControls[i] = -10000;
96     }
97   }
98 }
99
100
101 private class Box {
102   /**
103    * Box for display each number and O or X .
104    */
105   int rectX, rectY; // Position of square Box
106   int rectSize = 100;
107   int rectHighlight;
108
109   public Box(int X, int Y) {
110     rectHighlight = color(251);
111     rectX = X;
112     rectY = Y;
113   }
114
115   public void drawBox(int mouseXBox, int mouseYBox) {
116     fill(rectHighlight);
117     rect(rectX, rectY, rectSize, rectSize);
118   }
119
120   public void updateBox(int i) {
121     fill(0);
122     textSize(60);
123     String str = ticTacToe.getBoard()[i];
124     text(str, rectX + 45, rectY + 60);
125   }
126 }
127
128 }
```

ConsoleMain.java

```
Project Explorer x
A2_TicTacToe [Metropolitan-State-Univ]
  src
    (default package)
      ConsoleMain.java
      GraphicsMain.java
      TicTacToe.java
  JRE System Library [JavaSE-16]
  Referenced Libraries

ConsoleMain.java x
1 import java.util.InputMismatchException;
2
3
4 public class ConsoleMain {
5
6     public static void main(String[] args)
7     {
8         //1. Create new game (it will initialize to empty board)
9         TicTacToe ttt = new TicTacToe();
10
11         //2. print the board
12         System.out.println("Welcome to 3x3 Tic Tac Toe.");
13         ttt.printBoard();
14
15         System.out.println("X will play first. Enter a slot number to place X in:");
16
17
18         //3. get the current board information
19         String winner = ttt.getWinner();
20         String[] board = ttt.getBoard();
21
22
23         //4. As long as the winner is "None", we should keep playing the game
24         // The game stops when winner = X or winner = Y or winner = draw
25         Scanner in = new Scanner(System.in);
26         while (winner.equals("None")) {
27             String turn = ttt.getTurn();
28
29             int numInput;
30
31             // Exception handling.
32             // numInput will take input from user like from 1 to 9.
33             // If it is not in range from 1 to 9.
34             // then it will show you an error "Invalid input."
35             try
36             {
37                 numInput = in.nextInt();
38                 if (!(numInput > 0 && numInput <= 9)) {
39                     System.out.println("Invalid input; re-enter slot number:");
40                     continue;
41                 }
42             }
43             catch (InputMismatchException e) {
44                 System.out.println("Invalid input; re-enter slot number:");
45                 continue;
46             }
47
48             String current_value = board[numInput - 1];
49             String given_value = String.valueOf(numInput);
50             boolean current_and_given_are_equal = current_value.equals(given_value);
51
52
53             // This game has two player x and 0.
54             // Here is the logic to decide the turn.
55             if (current_and_given_are_equal) {
56                 // update the board position with X or 0
57                 board[numInput - 1] = turn;
58
59                 if (turn.equals("X")) {
60                     ttt.setTurn("0");
61                 }
62                 else {
63                     ttt.setTurn("X");
64                 }
65
66                 ttt.printBoard();
67                 winner = ttt.checkWinner();
68             }
69         }
70     }
71 }
```

```
Project Explorer x
A2_TicTacToe [Metropolitan-State-Univ]
  src
    (default package)
      ConsoleMain.java
      GraphicsMain.java
      TicTacToe.java
  JRE System Library [JavaSE-16]
  Referenced Libraries

ConsoleMain.java x
20 String[] board = ttt.getBoard();
21
22
23 //4. As long as the winner is "None", we should keep playing the game
24 // The game stops when winner = X or winner = Y or winner = draw
25 Scanner in = new Scanner(System.in);
26 while (winner.equals("None")) {
27     String turn = ttt.getTurn();
28
29     int numInput;
30
31     // Exception handling.
32     // numInput will take input from user like from 1 to 9.
33     // If it is not in range from 1 to 9.
34     // then it will show you an error "Invalid input."
35     try
36     {
37         numInput = in.nextInt();
38         if (!(numInput > 0 && numInput <= 9)) {
39             System.out.println("Invalid input; re-enter slot number:");
40             continue;
41         }
42     }
43     catch (InputMismatchException e) {
44         System.out.println("Invalid input; re-enter slot number:");
45         continue;
46     }
47
48     String current_value = board[numInput - 1];
49     String given_value = String.valueOf(numInput);
50     boolean current_and_given_are_equal = current_value.equals(given_value);
51
52
53     // This game has two player x and 0.
54     // Here is the logic to decide the turn.
55     if (current_and_given_are_equal) {
56         // update the board position with X or 0
57         board[numInput - 1] = turn;
58
59         if (turn.equals("X")) {
60             ttt.setTurn("0");
61         }
62         else {
63             ttt.setTurn("X");
64         }
65
66         ttt.printBoard();
67         winner = ttt.checkWinner();
68     }
69     else {
70         System.out.println("Slot already taken; re-enter slot number:");
71     }
72 }
73
74 // If no one win or lose from both player x and 0.
75 // then here is the logic to print "draw".
76 if (winner.equalsIgnoreCase("draw")) {
77     System.out.println("It's a draw! Thanks for playing.");
78 }
79 // For winner -to display Congratulations! message.
80 else {
81     System.out.println("Congratulations! " + winner + "'s have won! Thanks for playing.");
82 }
83 }
84 }
85 }
86 }
87 }
```


TicTacToe.java

```
Project Explorer x
A2_TicTacToe [Metropolitan-State-University]
  src
    (default package)
      ConsoleMain.java
      GraphicsMain.java
      TicTacToe.java
    JRE System Library [JavaSE-16]
    Referenced Libraries

TicTacToe.java x
1 import java.util.Arrays;
2
3 public class TicTacToe {
4
5     // for holding the players turn
6     private String turn;
7
8     // for representing the board
9     private String[] board;
10
11     // for holding the winner information (X or O or draw or none)
12     private String winner;
13
14     /**
15      * Default Constructor For TicTacToe Class
16      */
17     public TicTacToe() {
18         this.setTurn("X");
19         this.setWinner("None");
20
21         String[] board = new String[9];
22         for (int a = 0; a < 9; a++) {
23             board[a] = String.valueOf(a + 1);
24         }
25         this.setBoard(board);
26     }
27
28
29     /**
30      * Overloaded Constructor For TicTacToe Class
31      */
32     public TicTacToe(String a_turn, String[] a_board, String a_winner) {
33         turn = a_turn;
34         board = a_board;
35         winner = a_winner;
36     }
37
38     /**
39      * Set method for the variable turn
40      */
41     public void setTurn(String a_turn) {
42         turn = a_turn;
43     }
44
45     /**
46      * Set method for the variable board
47      */
48     public void setBoard(String[] a_board) {
49         board = a_board;
50     }
51
52     /**
53      * Set method for the variable winner
54      */
55     public void setWinner(String a_winner) {
56         winner = a_winner;
57     }
58
59     /**
60      * Get method for the variable turn
61      */
62     public String getTurn() {
63         return turn;
64     }
65
66     /**
67      * Get method for the variable board
68      */
69     public String[] getBoard() {
70         return board;
71     }
72
73     /**
74      * Get method for the variable winner
75      */
76     public String getWinner() {
77         return winner;
78     }
79
80     /**
81      * method for getting the winner return value can be X or O or 'draw' or 'null'
82      *
83      * @return
84      */
85     public String checkWinner() {
86         for (int a = 0; a < 9; a++) {
87             String line = null;
88
89             switch (a) {
90                 case 0:
91                     line = board[0] + board[1] + board[2];
92                     break;
93                 case 1:
94                     line = board[3] + board[4] + board[5];
95                     break;
96                 case 2:
97                     line = board[6] + board[7] + board[8];
98                     break;
99                 case 3:
100                     line = board[0] + board[3] + board[6];
101                     break;
102                 case 4:
103                     line = board[1] + board[4] + board[7];
104                     break;
105                 case 5:
106                     line = board[2] + board[5] + board[8];
107                     break;
108                 case 6:
109                     line = board[0] + board[4] + board[8];
110                     break;
111                 case 7:
112                     line = board[2] + board[4] + board[6];
113                     break;
114             }
115             // For X winner
116             if (line.equals("XXX")) {
117                 return "X";
118             }
119             // For O winner
120             else if (line.equals("000")) {
121                 return "O";
122             }
123         }
124
125         // check if all the slots are full and if there is no winner
126         // return draw
127         for (int a = 0; a < 9; a++) {
128             if (Arrays.asList(board).contains(String.valueOf(a + 1))) {
129                 break;
130             }
131         }
132         return "draw";
133     }
134 }
```

```
Project Explorer x
A2_TicTacToe [Metropolitan-State-University]
  src
    (default package)
      ConsoleMain.java
      GraphicsMain.java
      TicTacToe.java
    JRE System Library [JavaSE-16]
    Referenced Libraries

TicTacToe.java x
64 }
65
66 /**
67  * Get method for the variable board
68  */
69 public String[] getBoard() {
70     return board;
71 }
72
73 /**
74  * Get method for the variable winner
75  */
76 public String getWinner() {
77     return winner;
78 }
79
80 /**
81  * method for getting the winner return value can be X or O or 'draw' or 'null'
82  *
83  * @return
84  */
85 public String checkWinner() {
86     for (int a = 0; a < 9; a++) {
87         String line = null;
88
89         switch (a) {
90             case 0:
91                 line = board[0] + board[1] + board[2];
92                 break;
93             case 1:
94                 line = board[3] + board[4] + board[5];
95                 break;
96             case 2:
97                 line = board[6] + board[7] + board[8];
98                 break;
99             case 3:
100                 line = board[0] + board[3] + board[6];
101                 break;
102             case 4:
103                 line = board[1] + board[4] + board[7];
104                 break;
105             case 5:
106                 line = board[2] + board[5] + board[8];
107                 break;
108             case 6:
109                 line = board[0] + board[4] + board[8];
110                 break;
111             case 7:
112                 line = board[2] + board[4] + board[6];
113                 break;
114         }
115         // For X winner
116         if (line.equals("XXX")) {
117             return "X";
118         }
119         // For O winner
120         else if (line.equals("000")) {
121             return "O";
122         }
123     }
124
125     // check if all the slots are full and if there is no winner
126     // return draw
127     for (int a = 0; a < 9; a++) {
128         if (Arrays.asList(board).contains(String.valueOf(a + 1))) {
129             break;
130         }
131     }
132     return "draw";
133 }
```

```
Project Explorer x
A2_TicTacToe (Metropolitan-State-Univ
  > src
  > (default package)
  > ConsoleMain.java
  > GraphicsMain.java
  > TicTacToe.java
  > JRE System Library [JavaSE-16]
  > Referenced Libraries

TicTacToe.java x
98 line = board[0] + board[1] + board[2];
99 break;
100 case 3:
101 line = board[0] + board[3] + board[6];
102 break;
103 case 4:
104 line = board[1] + board[4] + board[7];
105 break;
106 case 5:
107 line = board[2] + board[5] + board[8];
108 break;
109 case 6:
110 line = board[0] + board[4] + board[8];
111 break;
112 case 7:
113 line = board[2] + board[4] + board[6];
114 break;
115 }
116 // For X winner
117 if (line.equals("XXX")) {
118 return "X";
119 }
120 // For O winner
121 else if (line.equals("000")) {
122 return "O";
123 }
124 }
125
126 // check if all the slots are full and if there is no winner
127 // return draw
128 for (int a = 0; a < 9; a++) {
129 if (Arrays.asList(board).contains(String.valueOf(a + 1))) {
130 break;
131 } else if (a == 8) {
132 return "draw";
133 }
134 }
135
136 // To enter the X Or O at the exact place on board.
137 // System.out.println( turn + "'s turn; enter a slot number to place " + turn + " in:");
138 return "None";
139 }
140
141 /**
142 * For printing the board
143 */
144 public void printBoard() {
145 System.out.println(" |---|---|");
146 System.out.println(" | " + board[0] + " | " + board[1] + " | " + board[2] + " |");
147 System.out.println(" |---|---|");
148 System.out.println(" | " + board[3] + " | " + board[4] + " | " + board[5] + " |");
149 System.out.println(" |---|---|");
150 System.out.println(" | " + board[6] + " | " + board[7] + " | " + board[8] + " |");
151 System.out.println(" |---|---|");
152 }
153
154 /**
155 * Returns the String representation of TicTacToe object
156 */
157 public String toString() {
158 System.out.println("Winner: " + this.winner);
159 System.out.println("Current Turn: " + this.turn);
160 this.printBoard();
161 return "";
162 }
163
164 }
```