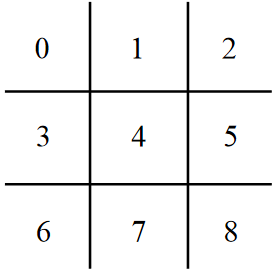
Tic-Tac-Toe Network 6.4



* Use this [IPConnect.java](https://drive.google.com/file/d/1bibVGUvvIA7lX3rlVdNlfkyI6ImhBaTM/view?usp=sharing) file which is needed to establish a connection over the network.
* Use [TicTacToe.java](https://drive.google.com/file/d/1T_X6KEnIomXiyb5tuwgJlbb5m-snv1X0/view?usp=sharing) as a template to build a network solution. Follow the comments provided within the file.
* Use port 8000 to set up the server hosting port
* IPConnect methods:
  + connect.hostSocket(int port) : creates a server socket bounded to a specific port
  + connect.connectSocket(int port, String ip\_address) : connects to the specified port number on the host server defined by the IP Addresst
  + connect.getMessage()
  + connect.sendMessage(String msg)

1. Implement **winnerExists()** and **isBoardFull()** methods
2. Within the main method:
   1. for host: use **Scanner** object and **connect.hostSocket(port)** to set up the server hosting port (8000)
   2. for client: use **Scanner** object and **connect.connectSocket(port, ip\_address)** to connect to the server
   3. host and client: Write logic for the game loop to determine if the game should continue or not. Use **winnerExists()** and **isBoardFull()** in your solution.

Hint: The **nextInt()** and **nextLine()** scanner methods are useful in reading user input

1. makeMove() method:
   1. Get a valid board position for the user’s next move. Make sure the position is not occupied and is between 0-8.
   2. Update the game board with the user’s move
   3. Send new user position over network to opponent using **connect.sendMessage(String msg)**
   4. Determine if the user’s move results in a win or a draw.
      1. If the game is over, display the appropriate outcome of the match and have makeMove() return true so the game loop can terminate.
      2. If the game is not over after the user’s move, makeMove() will return false

Hint: use **Integer.toString()** to convert int to String

1. waitMove() method:
   1. Read the board position of the opponent’s next move using **connect.getMessage()**
   2. Update the game board with the opponent’s move
   3. Determine if the opponent’s move results in a loss or a draw.
      1. If the game is over, display the appropriate outcome of the match and have waitMove() return true so the game loop can terminate.
      2. If the game is not over after the opponent’s move, waitMove() will return false

Hint: use **Integer.parseInt()** to convert String to int

1. Display the state of the board after initializing the board and after each update/play using the **drawBoard()** method
2. Test your code and play multiple rounds to ensure the code works without error. One game should produce a winner/loser and the other game should produce a draw.
   1. Navigate to your BlueJ Tic Tac Toe Project Directory on Terminal/Command Prompt
   2. enter: *java TicTacToe* to run the program.