vim Driver.java

```java
/*

Jared Dyreson
CWID: 889546529
Driver.java -> Hacked together code to make it seem like we're playing a game of Tic Tac Toe

*/

import java.util.Scanner;
import java.text.MessageFormat;

public class Driver {
    public static void main(String[] args){
        // Auto generated with caffine and autovt@.service

        TicTacToe t = new TicTacToe();
        int weight = 0;
        boolean winner = false;

        Scanner stdin = new Scanner(System.in);
        int x_selection, y_selection = 0;
        Player player_one = new Player('X', false, 1);
        Player player_two = new Player('O', false, 2);

        // we run the program until the board is full
        do{
            do{

                System.out.print("[+] Player 1 (X), please provide your column number: ");
                y_selection = stdin.nextInt();

                System.out.print("[+] Player 1 (X), please provide your row number: ");
                x_selection = stdin.nextInt();
            } while((int)t.coordinate(x_selection, y_selection) > 10);


            t.insert(x_selection, y_selection, 'X');
            t.print_board();

            t.check_winner(player_one, player_two);
            weight = t.board_layout_weight();

            // these are the conditions that we can have to break out of the program
            // any winners or the board becomes full, because it is an odd amount of spaces, we need to add a check before second player puts stuff down
            if(player_one.get_winning_stat() || player_two.get_winning_stat() || weight == 9){ break; }

            do{
                System.out.print("[+] Player 2 (O), please provide your column number: ");
                y_selection = stdin.nextInt();

                System.out.print("[+] Player 2 (O), please provide your row number: ");
                x_selection = stdin.nextInt();
```

1,2 Top

```java
            } while((int)t.coordinate(x_selection, y_selection) > 10);


            t.insert(x_selection, y_selection, 'O');
            t.print_board();

            weight = t.board_layout_weight();

            t.check_winner(player_one, player_two);
            // check if there is a winner for a second time because they may have won in the previous turn
            if(player_one.get_winning_stat() || player_two.get_winning_stat()){ break; }

        }while(weight != 9);


        // proper message to display the winner based on who won
        if(player_one.get_winning_stat()){
            System.out.println("[+] Player 1 has won the match!");
        }
        else if(player_two.get_winning_stat()){
            System.out.println("[+] Player 2 has won the match!");
        }
        else{
            // no winner
            System.out.println("[-] This game has resulted in a cat's game, MEOW!");
        }
    }
}
```

80,1 Bot

vim Player.java

```java
/*

Jared Dyreson
CWID: 889546529
Player.java -> Tic Tac Toe (TTT) Player class that interacts with the TTT board.

*/

public class Player{
    // Auto generated with caffine and network-manager.service
    private char player_symbol = 'X';
    private boolean is_winner = false;
    private int player_number = 1;

    // a constructor for Player class, such wow
    public Player(char ps, boolean w, int pn){
        this.player_symbol = ps;
        this.is_winner = w;
        this.player_number = pn;
    }

    public void set_winning_stat(boolean value){
        // make them feel special
        is_winner = value;
    }
    public boolean get_winning_stat(){
        // did they win?
        return is_winner;
    }
}
```

vim TicTacToe.java

```java
/*

Jared Dyreson
CWID: 889546529
TicTacToe.java -> Tic Tac Toe (TTT) board class and all it's helper functions.

*/

import java.util.Arrays;
import java.io.CharArrayWriter;
import java.util.Scanner;

public class TicTacToe {
    // Auto generated with caffine and network-manager.service
    public char[][] board_layout = new char[3][3];

    public void insert(int x, int y, char content){
        // this allows us to plot points on the board like Cartesian positionals.
        board_layout[x][y] = content;
    }

    public char coordinate(int x, int y){
        // this allows us to interface with the board as if it was a Cartesian graph.
        return board_layout[x][y];
    }

    public void print_board(){

        // print the board without separators cause it looks much better like this in my opinion
        for (char[] a : board_layout) {
            for (char i : a) {

                int position = (int)i;
                if(position != 88 && position != 79){ System.out.print("_ \t"); }
                else{ System.out.print(i + "\t"); }
            }
            System.out.println("\n");
        }
    }
    public int array_summation(char arr[]){
        // determining the "weight" of a given char array
        // this allowed for it to be reused for both numbers, as we did all conditional checking outside the function and was not dependent on the content of the array.
        // similar to Python's sum() function
        int counter = 0;
        for(int i = 0; i < arr.length; ++i){
            if(arr[i] != ' '){ counter+=(int)arr[i]; }
        }
        return counter;
    }
    public void check_winner(Player one, Player two){
        // int value of X -> 88
        // int value of O -> 79
```

1,1 Top

```java
        // winning score  can be either 264 or 237 for horizontal/vertical when transposed
        // as much as I would have liked to create an algorithm to determine a diagonal win, I was forced to just use hard coded positionals.
        // although they are hard coded, they still follow the same methodology for the vertical and horizontal counters. The values are staggered by k-1 instead of being a sequential array, making it much harder to develop a reliable solution.

        int counter = 0;
        int vertical_counter = 0;
        int diagonal_counter_lr = 0;
        int diagonal_counter_rl = 0;

        // iterate from the bottom up

        for(int y = 0; y < 3; ++y){
            for(int x = 2;  x >= 0; --x){
                int coordinate_point = (int)coordinate(x, y);
                if(coordinate_point == 88 || coordinate_point == 79){
                    vertical_counter+=coordinate_point;
                }
            }
            // this is so we are not stuck with relying on the last iteration of the loop as the vertical counter is reset every cycle
            if(vertical_counter == 237 || vertical_counter == 264){ break; }
            vertical_counter = 0;
        }

        for(int i = 0; i < board_layout.length; ++i){

            // attempt to find a diagonal going top left to bottom right
            diagonal_counter_lr = (int)coordinate(0, 0)+(int)coordinate(1, 1)+(int)coordinate(2, 2);

            // attempt to find a diagonal from top right to bottom left

            diagonal_counter_rl = (int)coordinate(2, 0)+(int)coordinate(1, 1)+(int)coordinate(0, 2);


            // what is the summation of all values inside the inner array
            counter = array_summation(board_layout[i]);


            // since horizontal and vertical (when transposed) are the same, we can use them interchangeably
            // check if there is a winner for player one
            if(counter == 264 || vertical_counter == 264 || diagonal_counter_lr == 264 || diagonal_counter_rl == 264){
                one.set_winning_stat(true);
            }
            // check if there is a winner for player two
            else if(counter == 237 || vertical_counter == 237 || diagonal_counter_lr == 237 || diagonal_counter_rl == 237){
                two.set_winning_stat(true);
            }

            // after every iteration through a row we need to reset the values
            counter = 0;
            diagonal_counter_lr = 0;
            diagonal_counter_rl = 0;
```

102,9 76%

```java
        }
    }
    public int board_layout_weight(){
        // see how many elements are actually on the playing board
        // blank spaces don't count towards the weight and are ignored. Only values X and O are counted, which is not entirely needed. This is to filter out results that may taint the board, such as arbitrary chars not intended to be there
        int weight = 0;
        for(int i = 0; i < board_layout.length; ++i){
            for(int j = 0; j < board_layout.length; ++j){
                int position = (int)coordinate(i, j);
                if(position == 88 || position == 79){ weight+=1; }
            }
        }
        return weight;
    }
}
```

119,0-1 Bot

```
[+] Player 1 (X), please provide your column number: 0
[+] Player 1 (X), please provide your row number: 0
X       _       _

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 1
[+] Player 2 (O), please provide your row number: 0
X       O       _

_       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 0
[+] Player 1 (X), please provide your row number: 1
X       O       _

X       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 2
[+] Player 2 (O), please provide your row number: 0
X       O       O

X       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 0
[+] Player 1 (X), please provide your row number: 2
X       O       O

X       _       _

X       _       _

[+] Player 1 has won the match!

Press ENTER or type command to continue_
[+] Player 1 (X), please provide your column number: 0
[+] Player 1 (X), please provide your row number: 0
X       _       _

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 1
X       _       _

O       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 0
X       X       _

O       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 2
X       X       _

O       _       _

O       _       _

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 0
X       X       X

O       _       _

O       _       _

[+] Player 1 has won the match!

Press ENTER or type command to continue

[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 0
_       X       _

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 0
O       X       _

_       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 0
O       X       X

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 1
O       X       X

O       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 1
O       X       X

O       X       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 2
O       X       X

O       X       _

O       _       _

[+] Player 2 has won the match!

Press ENTER or type command to continue
[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 1
_       _       _

_       X       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 0
O       _       _

_       X       _

_       _       _

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 2
O       _       _

_       X       _

_       _       X

[+] Player 2 (O), please provide your column number: 2
[+] Player 2 (O), please provide your row number: 0
O       _       O

_       X       _

_       _       X

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 1
O       _       O

_       X       X

_       _       X

[+] Player 2 (O), please provide your column number: 2
[+] Player 2 (O), please provide your row number: 1
[+] Player 2 (O), please provide your column number: 1
[+] Player 2 (O), please provide your row number: 0
O       O       O

_       X       X

_       _       X

[+] Player 2 has won the match!

Press ENTER or type command to continue_
[+] Player 1 (X), please provide your column number: 0
[+] Player 1 (X), please provide your row number: 0
X       _       _

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 1
[+] Player 2 (O), please provide your row number: 0
X       O       _

_       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 1
X       O       _

_       X       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 2
X       O       _

_       X       _

O       _       _

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 2
X       O       _

_       X       _

O       _       X

[+] Player 1 has won the match!

Press ENTER or type command to continue
[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 0
_       _       X

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 0
O       _       X

_       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 1
O       _       X

_       X       _

_       _       _

[+] Player 2 (O), please provide your column number: 1
[+] Player 2 (O), please provide your row number: 0
O       O       X

_       X       _

_       _       _

[+] Player 1 (X), please provide your column number: 0
[+] Player 1 (X), please provide your row number: 2
O       O       X

_       X       _

X       _       _

[+] Player 1 has won the match!

Press ENTER or type command to continue
[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 0
_       X       _

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 0
O       X       _

_       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 0
[+] Player 1 (X), please provide your row number: 2
O       X       _

_       _       _

X       _       _

[+] Player 2 (O), please provide your column number: 1
[+] Player 2 (O), please provide your row number: 1
O       X       _

_       O       _

X       _       _

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 0
O       X       X

_       O       _

X       _       _

[+] Player 2 (O), please provide your column number: 2
[+] Player 2 (O), please provide your row number: 2
O       X       X

_       O       _

X       _       O

[+] Player 2 has won the match!

Press ENTER or type command to continue_

[+] Player 1 (X), please provide your column number: 0
[+] Player 1 (X), please provide your row number: 0
X       _       _

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 2
[+] Player 2 (O), please provide your row number: 0
X       _       O

_       _       _

_       _       _

[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 0
X       X       O

_       _       _

_       _       _

[+] Player 2 (O), please provide your column number: 1
[+] Player 2 (O), please provide your row number: 1
X       X       O

_       O       _

_       _       _

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 2
X       X       O

_       O       _

_       _       X

[+] Player 2 (O), please provide your column number: 0
[+] Player 2 (O), please provide your row number: 2
X       X       O

_       O       _

O       _       X

[+] Player 2 has won the match!

Press ENTER or type command to continue

[+] Player 2 (O), please provide your column number: 1
[+] Player 2 (O), please provide your row number: 1
X       _       _

O       O       _

X       _       _

[+] Player 1 (X), please provide your column number: 1
[+] Player 1 (X), please provide your row number: 0
X       X       _

O       O       _

X       _       _

[+] Player 2 (O), please provide your column number: 2
[+] Player 2 (O), please provide your row number: 0
X       X       O

O       O       _

X       _       _

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 2
X       X       O

O       O       _

X       _       X

[+] Player 2 (O), please provide your column number: 2
[+] Player 2 (O), please provide your row number: 2
[+] Player 2 (O), please provide your column number: 1
[+] Player 2 (O), please provide your row number: 2
X       X       O

O       O       _

X       O       X

[+] Player 1 (X), please provide your column number: 2
[+] Player 1 (X), please provide your row number: 1
X       X       O

O       O       X

X       O       X

[-] This game has resulted in a cat's game, MEOW!

Press ENTER or type command to continue
```