

Programming Design Worksheet - Redfield for CS1310 (programs 2-7) and CS1311 (programs 1-6)

Copy this file. Type and past images to create new documents for designs.

Print it for class (if you must miss, submit one file to Designs).
First name **Davide** Last name **Russillo**
Design for program name **Tic-Tac-Toe**

DATA

Variables needed in WORDS for main and globally

GLOBAL

Board array
Player x or y
row number
column number
move counter

MAIN

input character

C **DECLARATIONS** for main & global

char ttt[3][3];
char player = 'X';
int row;
int col;
int moves;

char input;

(STARTING TicTacToe:put image; or draw: Insert, Drawing; or put at end of the file)

draw in RAM with possible values

ttt[0][0]	ttt[0][1]	ttt[0][2]
ttt[1][0]	ttt[1][1]	ttt[1][2]
ttt[2][0]	ttt[2][1]	ttt[2][2]

Algorithm to PSEUDOCODE level for each function

(remember to indent under if, switch, while, do-while, for)

main:

print welcome to tictactoe! X goes first. press enter to start

input newline character into input

clear_board()

while check_win() returns 0

 print clearpage

 draw_board_options()

 draw_board(0)

 take_turn()

reassign_player()

reassign_player()

print clearpage

draw_board_options

draw_board(check_if_win())

other functions (bold the names): (put them before main in the program!)

```
void draw_board_options(void)
```

```
    print _1_|_2_|_3_  
          _4_|_5_|_6_  
          7 | 8 | 9
```

```
void draw_board(int winstate)
```

```
    int i
```

```
    int wintype = winstate / 10
```

```
    int wincoord = winstate - winstate2 / 10 * 10
```

```
    for each row
```

```
        for each of the five ascii rows
```

```
            for each column
```

```
                if ttt[row][col] is empty
```

```
                    print empty cell
```

```
            else
```

```
                switch i
```

```
                    case first row
```

```
                        if player is X
```

```
                            if wintype is 1
```

```
                                print type one X ascii art
```

```
                            else if wintype is 2
```

```
                                print type two X ascii art
```

```
                                ...
```

```
                                ...
```

```
                            else print empty line
```

```
                    else
```

```
                        if wintype is 1
```

```
                            print type one Y ascii art
```

```
                    case second row
```

```
                        ...
```

```
                        ...
```

```
                if win is on row print =
```

```
                else if diagonal add // or \
```

```
                else add | unless last column
```

```
            print newline
```

```
    if tie print tie
```

```
    else if win print player won!
```

```

void clear_board(void)
    for each row
        for each column
            set ttt[row][col] = empty

void take_turn(void)
    int position
    int position_valid = 0
    int x
    int y

    while the position is invalid
        print its player's turn, choose an empty field
        input number into position
        position_valid = 1
        switch position
            case 1
                set x to 0, y to 0
            case 2
                set x to 0, y to 1
            ...
            case 4 set x to 1, y to 0
            ...
            ...
            default
                position_valid = 0
        if ttt[x][y] is empty and position_valid is 1
            ttt[x][y] = player
        else
            position_valid = 0
            print clearpage
            draw_board_options()
            draw_board(0)
            print Invalid input try again
    increment moves

void reassign_player(void)
    if player is X
        set player to 0
    else
        set player to X

```

```

int check_if_win(void)
    int i

    for i from 0 to 2
        if ttt[i][0] is not empty is equal to ttt[i][1] and ttt[i][2]
            return 10 + i
        else if ttt[0][i] is not empty and is equal to ttt[1][i] and
            ttt[2][i]
            return 20 + i
    if ttt[0][0] is not empty and is equal to ttt[1][1] and ttt[2][2]
        return 30
    else if ttt[0][2] is not empty and is equal to ttt[1][1] and
        ttt[2][0]
        return 40
    else if moves equals 9
        return 50
    return 0

```

OTHER part of the design (see assignment - *input or sample output*)

```

_1_|_2_|_3_
_4_|_5_|_6_
_7_|_8_|_9_

```

X X	0 0	X X
X	0 0	X
X X	0 0	X X
<hr/>		
X X	0 0	
X	0 0	
X X	0 0	
<hr/>		
	0 0	
	0 0	
	0 0	

0 won!