Giorgi Solomnishvili;RedID: 822164480

TIC-TAC-TOE

12/2/2018

Content

(* - Mandatory)

1*. Description of the problem/method

TIC – TAC – TOE

I wrote a code for tic tac toe. Tic tac toe is a game. Players play on a three by three game board. The first player plays with X and the second - O. Players alternate placing Xs and Os on the game board until either opponent has three in a row or all nine squares are filled. The object of Tic Tac Toe is to get three in a row. This is the general idea of tic-tac-toe.

Now I am going to describe what my project does. When you run it, the menu will be printed on the screen. There are five options: 1 – Single Player; 2 – Two players; 3 – History; 4 – Help/Rules; 5 – Quit. The program asks you to choose one of those five options by entering the corresponding number. The first option allows you to play against computer; the second – against your friend; the third option asks you to enter the name of the player to see his history with this game (you will see how many games he/she has played, against whom he played, did he win or not, what was difficulty level, how long did the game last, and date of each game); the fourth option displays the rules for tic-tac-toe; the fifth option is simply Quit. (See Picture 1)

If you choose **the first option**, the program asks you to enter your name, then asks you to choose your character (X or 0). Then you will be asked to choose the difficulty level (Easy, Normal or Hard). If you choose easy, you will be able to defeat computer very easily. It is a little bit harder to defeat computer, but you can still manage to. If you choose hard, your chance of defeating computer is low. After the game is over, the program asks you if you want to play again. If you choose not, the game will return to the menu. If you choose yes, the program asks you if you want to switch your character and change the difficulty level. History of that game will be saved in the text file. (See pictures 2-5)

If you choose **the second option,** asks two players to enter their names. The first will be X the second will be 0. After the game is over, the program will ask players if they want to play again and if they want to switch characters. History of that game will be saved in the text file. (See pictures 6-7)

If you choose **the third option**, asks you to enter player's name and gives you his/her history. (See pictures 8-9).

If you choose **the fourth option**, you will see the rules and description of tic tac toe. (See picture 10).

This code can be compiled with code::Blocks on Windows.

2. Pseudocode (if required. Mandatory for the Lab assignments, starting from #5 and Projects)

I wrote 20 functions for this program, including main function:

I) gotoy(int y) – this function allows me to take curson down by y; it prints y new line character. I did not use conio.h

```
Void return-type function called gotoy take one integer parameter y.
    DECLARE in variable i;

    FOR i 0 to y
            PRINT new line character
    END FOR
```

ENF FUNCTION

<u>II) goto(int x)</u> <u>–</u> this function allows me to shift cursor left by x times; it prints space x times. I did not use conio.h

```
Void return type gotox takes one integer parameter x
    DECLARE i;
    FOR I from 0 to x
            PRINT space
    END FOR
END FUNCTION
```

<u>III)</u> **<u>print(char a[], char b[],char c[]) –</u>** this function prints tic tac toe table. This table has three rows and each row consists of three boxes. Each row is stored in each string (a,b and c).

```
Void return type function print takes three string parameter a, b and c
    DECLARE int variable k and SET k to 50
    DECLARE char array with 12 characters – lines[12] and SET 10 hyphens in it
    CALL gotox(k)//shift cursor leftwards 50 times
    PRINT a
    CALL gotox(k)//shift cursor leftwards 50 times
    PRINT lines
    CALL gotox(k)//shift cursor leftwards 50 times
    PRINT b
    CALL gotox(k)//shift cursor leftwards 50 times
    PRINT lines
    CALL gotox(k)//shift cursor leftwards 50 times
    PRINT c
END FUNCTION
```

<u>IV) main()</u> // DECLARES MOST OF THE VARIABLE, CALLS GAME FUNCTIONS

```
DECLARE char array with 12 characters – a[12] // this will store first row of tic tac toe table
DECLARE char array with 12 characters – b[12] // this will store second row of tic tac toe table
DECLARE char array with 12 characters – c[12] // this will store third row of tic tac toe table
DECLARE char variable userChoice and SET to '1' // this will store user's menu option choice
DECLARE char array with to variables – player1[50] // stores name of the first player
DECLARE char array with to variables – player2[50] // stores name of the second player
DECLARE char array with to variables – tmpName[50]// used to switch player1 and player2
DECLARE char variable userChar // this will be user's character (X or 0) if he/she chooses option 1
DECLARE char variable difficulty // here will be stored user's choice of difficulty level
DECLARE char array with 50 characters – nameInHistory[50] // stores player's name of option 4
DECLARE char option='y';//decides whether two players want to play again
DECLARE  char option1='n';//decides whether user want to change character or not
DECLARE   char option2='y';//decides whether the user wants to change difficulty
DECLARE six char array. //Each of them stores rules of the game

WHILE userChoice does not equal ‘5’
    CALL make_Empty_Table(a,b,c) //discussed elsewhere
    CALL printOriginalScreen(&userChoice) /*discussed elsewhere printOriginalScreen updates
userChoice. If updated value is more than 5 or less than 1, printOriginalScreen will be called again*/
    SWITCH (userChoice)
            Case ‘1’ // user chooses 1 from menu options
                    SET option to ‘y’
                    SET option1 to ‘n’
                    SET option2 to ‘y’
                    Use system command to clear screen
                    PRINT "Enter Your Name: "
                    READ user input into player1);

                    PRINT “nChoose your character (x/0): "
                    READ userinput into userChar
                    WHILE userChar is not x or 0
                            Clear screen with system command
                            PRINT "Enter Your Name: "
                            READ user input into player1);
                            PRINT “nChoose your character (x/0): "
                            READ userinput into userChar
                    END WHILE
                    WHILE option does not equal ‘n’
                            IF option1 equals ‘y’ THEN
                                    IF userChar is 0 THEN SET userChar to x
                                    ELSE SET userChar to 0
                            END IF
                            SET difficulty to ‘2’
                            IF option2 is ‘y’
                                    DO
                                            System clear
                                            CALL gotoy(3) and CALL gotox(40)
                                             PRINT "Choose the level of difficulty (1 2 or 3)\n"
                                            CALL gotoy(1) and CALL gotox(45)
                                            PRINT “1. EASY\n”
                                            CALL gotoy(1) and CALL gotox(45)
                                            PRINT “2. NORMAL\n”
                                            CALL gotoy(1) and CALL gotox(45)
                                            PRINT “3. HARD\n”
                                            READ user input in difficulty
                                            WHILE difficulty doesnot equal ‘1’, ‘2’ and ‘3’
                                    END DO-WHILE
```

```
                END IF
                IF userChar is 'x' CALL gamex(a,b,c,player1,difficulty)//Discussed elsewhere
                ELSE CALL game0(a,b,c,player1,difficulty)//Discussed elsewhere
                PRINT "Do you want to play again? (Y/N) "
                READ user input in variable option
                IF option is ‘n’ THEN BREAK
                PRINT “Do you want to change your character? (Y/N)”
                READ user input in option1
                PRINT “Do you want to change difficulty level?” (Y/N)
                READ user input in option2
                CALL make_Empty_Table(a,b,c)
                Clear screen with system command
        END WHILE
        BREAK
END CASE ‘1’

CASE ‘2’ //user chooses 2 from menu options
        SET option to ‘y’
        system("cls");
        PRINT "Enter the name of Player1 (He/She will be X): "
        Read user input in player1
        PRINT "Enter the name of Player2 (He/She will be 0): "
        Read user input in player2

        WHILE option does not equal to ‘n’ and ‘N’
                system("cls");
                CALL game(a,b,c,player1,player2) //Discussed elsewhere
                PRINT "Dear (player1) and (player2), do you want to play again? (Y/N) "
                READ user input in variable option
                IF option is ‘n’ or ‘N’ THEN BREAK
                PRINT “Do you want to exchange characters? (Y/N)”
                READ user input in variable option1
                CALL make_Empty_Table(a,b,c)
                System(“cls”)
        END WHILE
        BREAK
END CASE ‘2’

CASE ‘3’ //user chooses 3 from menu options
        SET option to ‘y’
        WHILE option does not equal to ‘n’ and ‘N’
                system("cls");
                PRINT "Enter Player's Name To See His History: "
               READ user input in nameInHistory
```

```
                CALL makeNameUpperCase(nameInHistory) //Discussed elsewhere
                CALL system("cls"); gotoy(1); gotox(48);
                PRINT "%s's HISTORY\n\n",nameInHistory
                CALL searchFile(nameInHistory);//Discussed elsewhere
                PRINT "Do you want to look for the other player? (Y/N) "
                READ user input in option
            END WHILE
            BREAK
        END CASE '3'

        CASE '4' //user chooses 4 from menu options
            SET option to 'y'
            WHILE option does not equal to 'b' and 'B'
                system("cls");
                gotoy(2); gotox(46);
                PRINT "D E S C R I P T I O N\n\n\n"
                CALL make_Empty_Table(a,b,c);
                display(a,b,c);
                PRINT  strings containing rules
                PRINT "Press B to return to MENU: "
                READ user input in option
            END WHILE
            Break
        END CASE '4'

        CASE '5' //user chooses 5 from menu options
            SET userChoice to '5';
            break;
        END CASE '5'
        END SWITCH
    END WHILE
    END MAIN
```

<u>V) printOriginalScreen(char* userChoice)</u> – prints the menu

```
Char return type function printOriginalScreen takes one char pointer variable userChoice
    system("color 0E");
    CALL gotoy(2); gotox(50);
    PRINT "Tic Tac Toe\n"
    CAL gotoy(1); gotox(49);
    I have struct defined which has 1 sub data – char array. Name of this struct is poem.
    I have found poem about tic tac toe and displayed it above the menu.
```

“Love is a game
of tic tac toe
Constantly waiting
for the next X and O”
LANG LEAV

I print this “poem” from my struct poem

```
        gotox(43); PRINT "------------------------------\n"
        gotoy(2); gotox(45);
        PRINT "Enter corresponding Number\n"
        gotoy(1); gotox(45);
        PRINT "1. Single Player\n"
        gotox(45);
        PRINT "2. Two Players\n"
        gotox(45);
        PRINT "3. History\n"
        gotox(45);
        PRINT "4. Help/Rules\n"
        gotox(45);
        PRINT "5. Quit\n\n"
        gotox(45);

        PRINT "Enter Your Number:> "
        READ user input in userChoice);
        system("cls");
        return *userChoice;
END FUNCTION
```

<u>VI) make_Empty_Table(char a[], char b[],char c[])</u> – it makes tic-tac-toe table empty after each game

```
Void return type function make_Empty_Table takes three char array variables a, b, c.
    The function uses strcpy to copy "   |   |   " in each string (a,b,c).
END FUNCTION
```

<u>VII) display(char a[], char b[],char c[])</u> – it displays tic-tac-toe table. All nine boxes of the table are numbered from 1-9

```
Void return type function display takes three char array variables a, b, c.
    DECLARES three char array – aa[12], bb[12], cc[12]
    Using strcpy copies a, b and c, to aa, bb and cc
    aa[1]='1'; aa[5]='2'; aa[9]='3';        //  1 | 2 | 3
    bb[1]='4'; bb[5]='5'; bb[9]='6';        //  4 | 5 | 6
    cc[1]='7'; cc[5]='8'; cc[9]='9';        //  7 | 8 | 9
```

```
        CALL print(aa,bb,cc)
END FUNCTION
```

<u>If user chooses to play against computer, he can choose difficulty level</u>

VIII) easy() - EASY level

Int return type function easy takes no parameter

srand((int)time(0));

return random number between 1-9 // this number is the number of box which will be filled with computer's character

<u>IX) normalX(char a[], char b[],char c[])</u> – normal level if computer is x

Int return type function normalx takes three strings as parameter. These strings make tic-tac toe table

This function checks the first row. If user can win by placing 0 in any box from the first row, the computer puts x there and prevents user from winning

ELSE returns random number between 1-9 // this number is the number of box which will be filled with computer's character

X) hardX(char a[], char b[],char c[]) – hard level if computer is x

```
1 | 2 | 3
4 | 5 | 6
7 | 8 | 9
```

Int return type function hardx takes three strings as parameter. These strings make tic-tac toe table

Since computer is x, the best way to start is to put x in the middle box, so at first function returns 5.

Next function checks if computer can win by placing x in the first or second or third row,

If not, then checks if computer can win by placing x in the first or the second or the third column

If not, then checks if computer can win by placing x in the first or the second diagonal

If not, then checks if the user can win by placing 0 in the first or the second or the third column. If not, then checks if user can win by placing 0 in the first or the second diagonal. If not, then checks if user can win by placing 0 in the first or second or third row

If computer can win by placing x in the nth box, function returns n
Else if user can win by placing 0 in the mth box, function returns m
Else function returns random number from 1-9

<u>XI) normal0(char a[], char b[],char c[])</u> – normal level if computer is 0

Int return type function normal0 takes three strings as parameter. These strings make tic-tac toe table

This function checks the first row. If user can win by placing x in any box from the first row, the computer puts 0 there and prevents user from winning
ELSE returns random number between 1-9 // this number is the number of box which will be filled with computer's character

XII) hard0(char a[], char b[],char c[]) – hard level if computer is 0

```
1 | 2 | 3
4 | 5 | 6
7 | 8 | 9
```

Int return type function hard0 takes three strings as parameter. These strings make tic-tac toe table
Since computer is 0, the best way to start is to put 0 in the middle box, so at first function returns 5 (providing 5th box is empty). If the 5th box is filled, the second best option is to put 0 in corner so function returns 1, 3, 7 or 9.
Next function checks if computer can win by placing 0 in the first or second or third row,
If not, then checks if computer can win by placing 0 in the first or the second or the third column
If not, then checks if computer can win by placing 0 in the first or the second diagonal
If not, then checks if the user can win by placing x in the first or the second or the third column. If not, then checks if user can win by placing x in the first or the second diagonal. If not, then checks if user can win by placing x in the first or second or third row

If computer can win by placing 0 in the nth box, function returns n
Else if user can win by placing x in the mth box, function returns m
Else function returns random number from 1-9

XIII) go(char a[], char b[],char c[], int i,char sign) – this puts sign in the tic-tac-toe table in ith box if it is not already filled. Sign is either 'x' or '0'
Int return type function go takes 5 parameters: three strings (tic tac toe table), one int i and char sign
DECLARE int variable n and SET n to 0
i is the number of box where sign should be placed. This function uses if else statements and checks if ith box is empty. If it is empty, function puts sign in that box and increments n.
If ith box is already filled, nothing happens, n is still 0.
The function returns n. If it returned 0, this means that placement of sign did not happened

XIV) int checkWinner(char a[], char b[],char c[]) – checks if someone has won.

Int return type function checkWinner take three strings for parameters (the table)
In order to win tic tac toe, three similar sign (x or 0) should be in a row, column or diagonal. If X satisfies this condition, the function returns 1 (meaning player1 is a

winner), else if 0 satisfies this condition, the function returns 2 (meaning player2 is a winner). Else the function returns 0, meaning no one is a winner.

XV) makeNameUpperCase(char* name) – this makes all letters from name upper

```
DECLARE int variable i; SET i to 0;
WHILE *(name+i) does not equal to null terminating character
    *(name+i)=toupper(*(name+i))
    Increment i
END WHILE
```

XVI) searchFile(char player[50]) – searches history of player

```
Void return type function searchFile takes one string parameter
    DECLARE two char arrays: line[151] and name[50]
    DECLARE int i=0
    DECLARE *FILE inFile

    FOPEN History.txt file in reading mode
    While feof(inFile) is false
        READ first line from infile to line //Use fgets
        READ first word from line into name //use sscanf
        IF name and player are identical THEN
            PRINT line
            Increment i
        END IF
    SET first characters from name and line to \0
    IF i is 0
        PRINT "There is no such player in database"
    END IF
END function
```

XVII) appendFile(char player1[50],char player2[50],int duration,int winner,char difficulty)

```
This function creates file
*FILE outFile
Fopen History.txt in append mode
CALL makeUppercase(player1)
CALL makeUppercase(player2)
Using fprintf this function fills each players history according to the following structure:
```

```
                              Player's History
Player1 | VS | Player2 | Outcome | Difficulty | Sign | Duration | Date
(name)         (name)   (win/lose)             (X/0)  (seconds)   (date)
```

If the user was playing against computer, name of player2 will be computer, difficulty will be easy, normal or hard.

If the user was playing against another human being, "difficulty" will be that person

XVIII) gamex(char a[], char b[],char c[],char player1[],char difficulty) – allows one game, user is x

```
Function gamex takes four strings (table and name of player),  and one char parameter
DECLARE int j,i,k,winner=0;
DECLARE int l=0;
DECLARE int startTime=time(0);

WHILE l does not equal to 5
    Increment l
    Set k to 0
    WHILE k does not equal 1
        READ j from user
        Set k to go(a,b,c,j,'x')
        IF j is 0 THEN SET l to 6; BREAK
        Set winner to checkWinner(a,b,c)
    END WHILE
    IF winner does not equal 0 THEN break;
    IF l is equal or greater than 5 THEN
        Winner to checkWinner(a,b,c)
        Break
    END IF
    SET k to 0
    WHILE k does not equal 1
        SWITCH difficulty
            CASE '1'
                i=easy()
            CASE '2'
                i=normal0(a,b,c)
            CASE '3'
                i=hard0(a,b,c)
        Set k to go(a,b,c,i,'0')
    END WHILE
    Set winner to checkWinner(a,b,c)
    IF winner does not equal 0 THEN break;
END WHILE
IF l is not 6
    If winner is 1 PRINT player is vicrotious
    Else If winner is 2 PRINT computer is vicrotious
    ELSE PRINT it is a TIE
    CALL appendFile(player1,player2,duration,winner,difficulty);
ELSE system("cls")
```
END FUNCTION

XIX) game0(char a[], char b[],char c[],char player1[],char difficulty) – allows one game, user is 0

This function has the same logic as the previous one. The only difference is that user is 0 and computer is x

<u>XX) game(char a[], char b[],char c[],char player1[],char player2[]) -</u> allows two player game

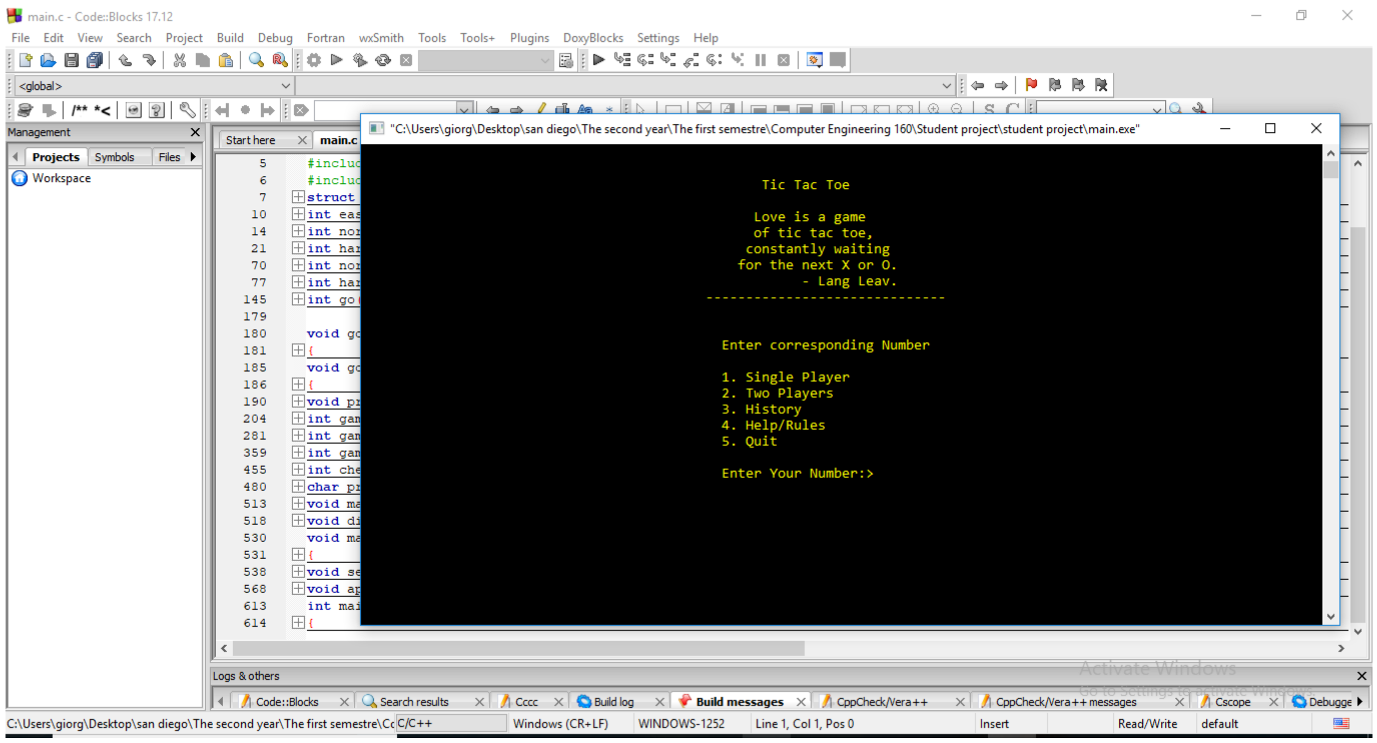
```
DECLARE int j,i,k,winner=0;
DECLARE int l=0;
DECLARE int startTime=time(0);

WHILE l does not equal to 5
    Increment l
    Set k to 0
    WHILE k does not equal 1
        READ j from user
        Set k to go(a,b,c,j,'x')
        Set winner to checkWinner(a,b,c)
    END WHILE
    IF winner does not equal 0 THEN break;
    IF l is 5 THEN
        Winner to checkWinner(a,b,c)
        Break
    END IF

    Set k to 0
    WHILE k does not equal 1
        READ i from user
        IF i is 0 THEN SET l to 6; BREAK
        Set k to go(a,b,c,i,0x')
        Set winner to checkWinner(a,b,c)
    END WHILE
    IF winner does not equal 0 THEN break;
END WHILE
IF l is not 6 THEN
    If winner is 1 PRINT player1 is vicrotious
    Else If winner is 2 PRINT player2 is vicrotious
    ELSE PRINT it is a TIE
    CALL appendFile(player1,player2,duration,winner,difficulty);
ELSE system("cls")
END FUNCTION
```

3*. TIME SPENT

It was estimated that I would spend 48 hours on that project. I did spent two days (48 hours) on that project

4. I have uploaded project report, .DOC file that contains code for my project (source code) and .C file. Source code is written in 5 font size, so that it will not be modified.
   You can compile and run uploaded .C file in code::blocks. My project does not require any additional file to compile. You just need .C file which is uploaded.

5*. Screen capture of the code and the resulting display(s)


main.c - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
Projects Symbols Files
Workspace
Start here
main.c
Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger

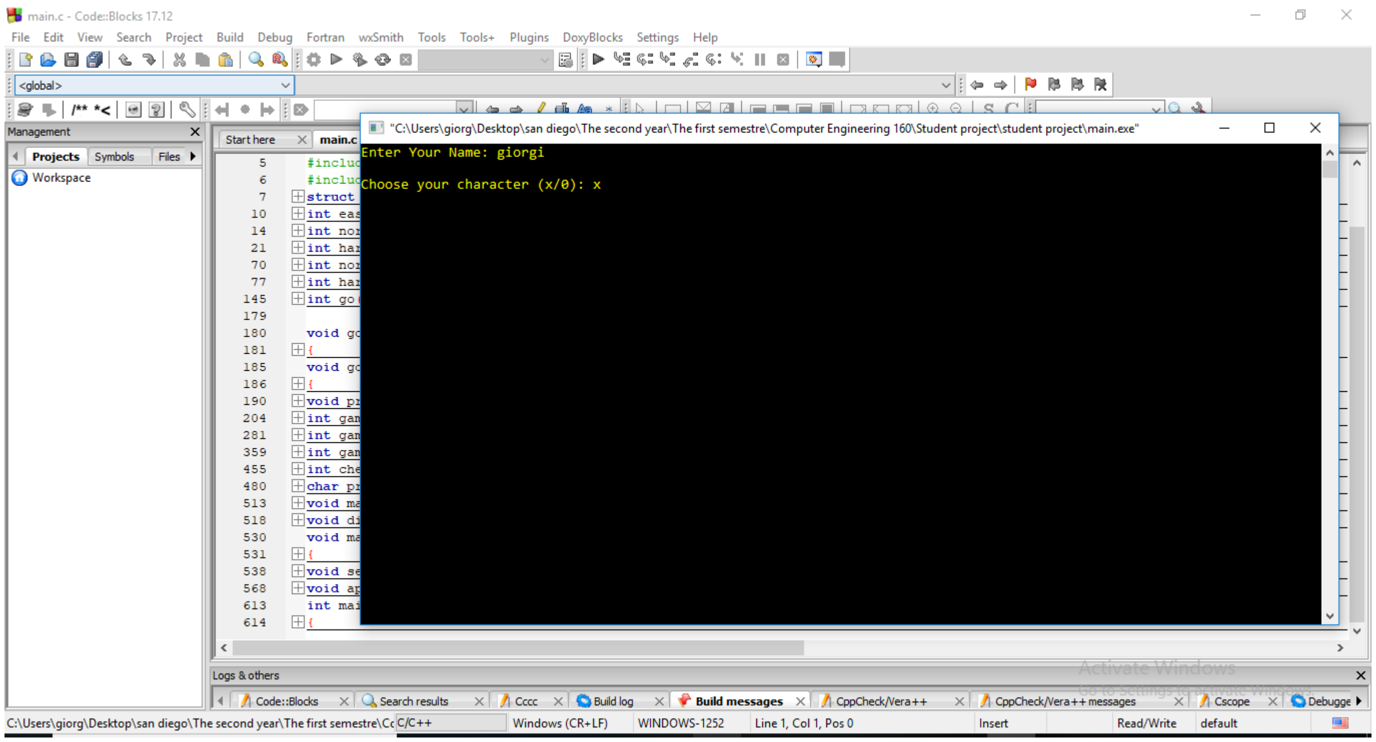
```
#incluc
#incluc
struct
int eas
int nor
int har
int nor
int har
int go
void gc
{
void gc
{
void pr
int gan
int gan
int gan
int che
char pr
void ma
void di
void ma
{
void se
void ap
int mai
{
```

"C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Computer Engineering 160\Student project\student project\main.exe"

```
                    Tic Tac Toe

                   Love is a game
                  of tic tac toe,
                 constantly waiting
                for the next X or O.
                        - Lang Leav.
            ------------------------------


              Enter corresponding Number

              1. Single Player
              2. Two Players
              3. History
              4. Help/Rules
              5. Quit

              Enter Your Number:>
```

C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Cc C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default

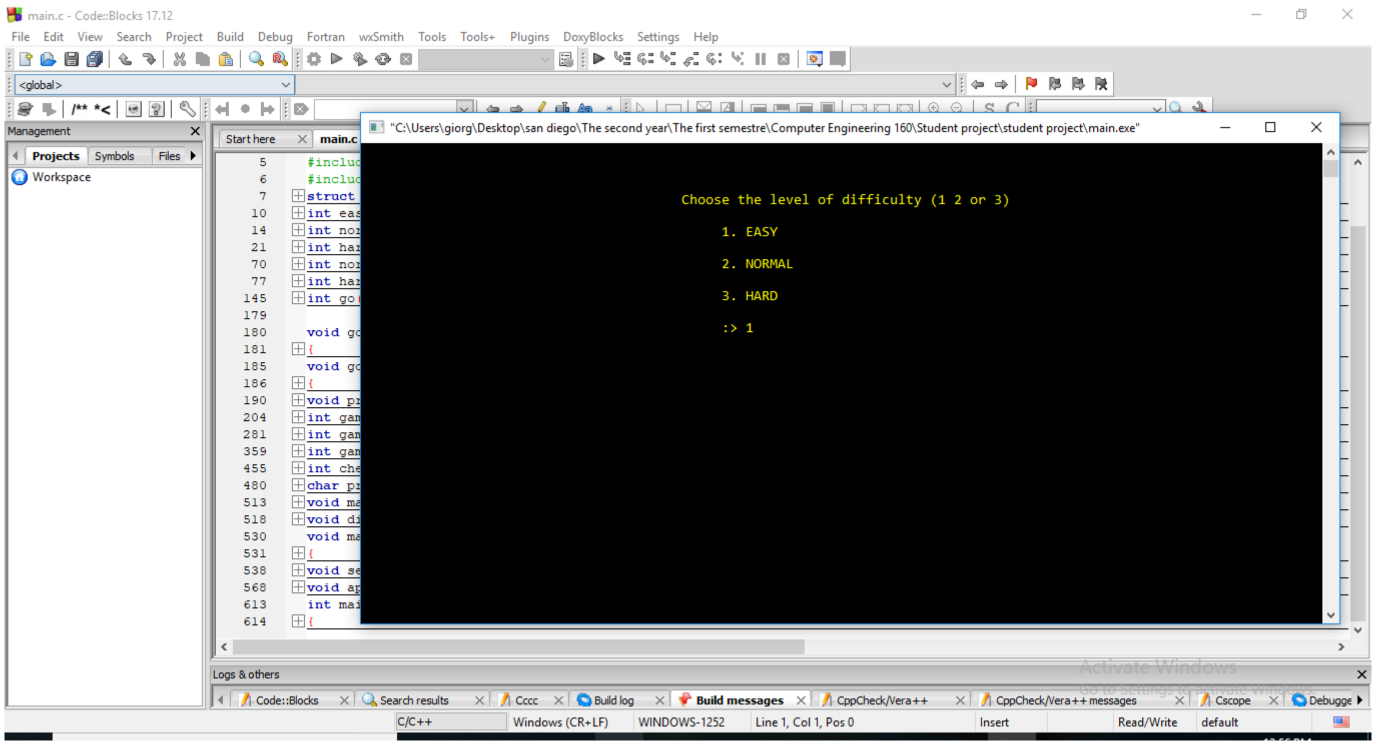
Picture 1


main.c - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols Files
Workspace
Start here main.c
Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger

"C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Computer Engineering 160\Student project\student project\main.exe"

```
Enter Your Name: giorgi

Choose your character (x/0): x
```

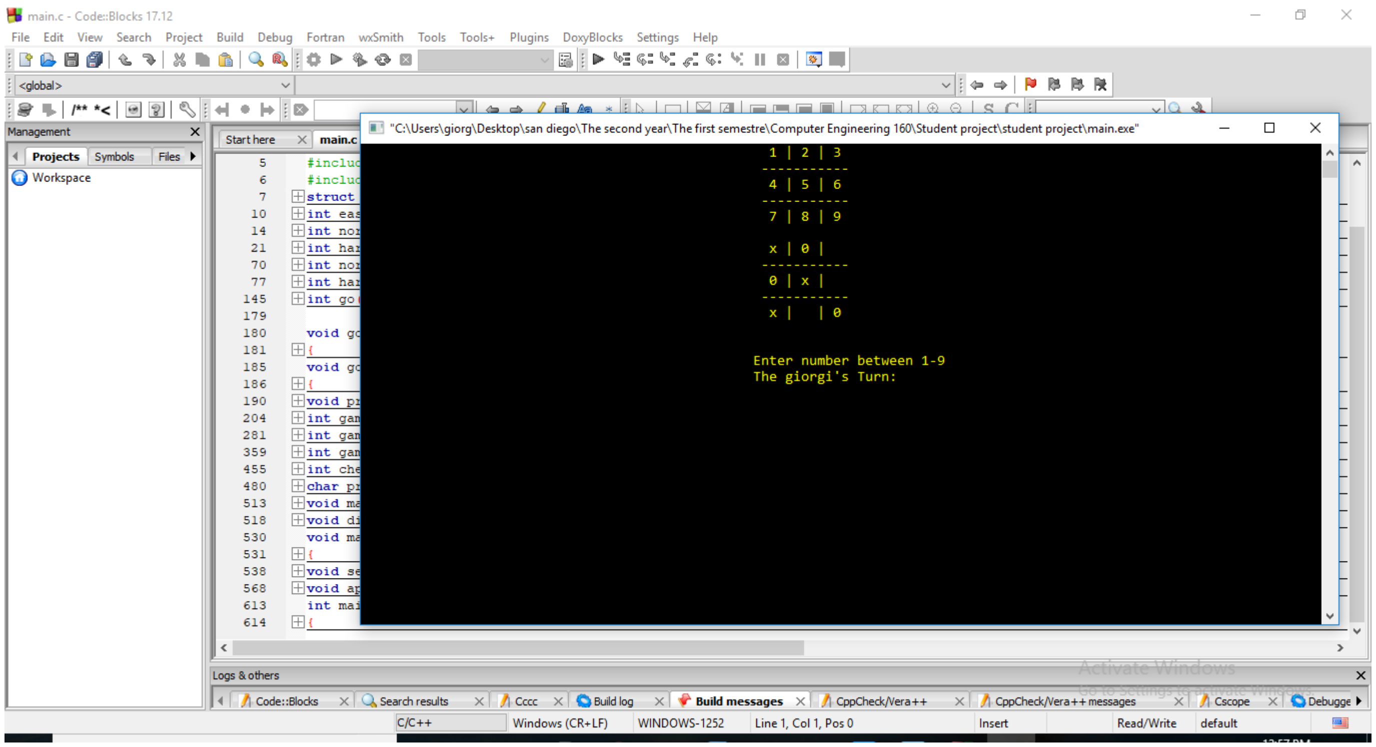
Picture 2


main.c - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols Files
Workspace
Start here main.c
Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger

"C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Computer Engineering 160\Student project\student project\main.exe"

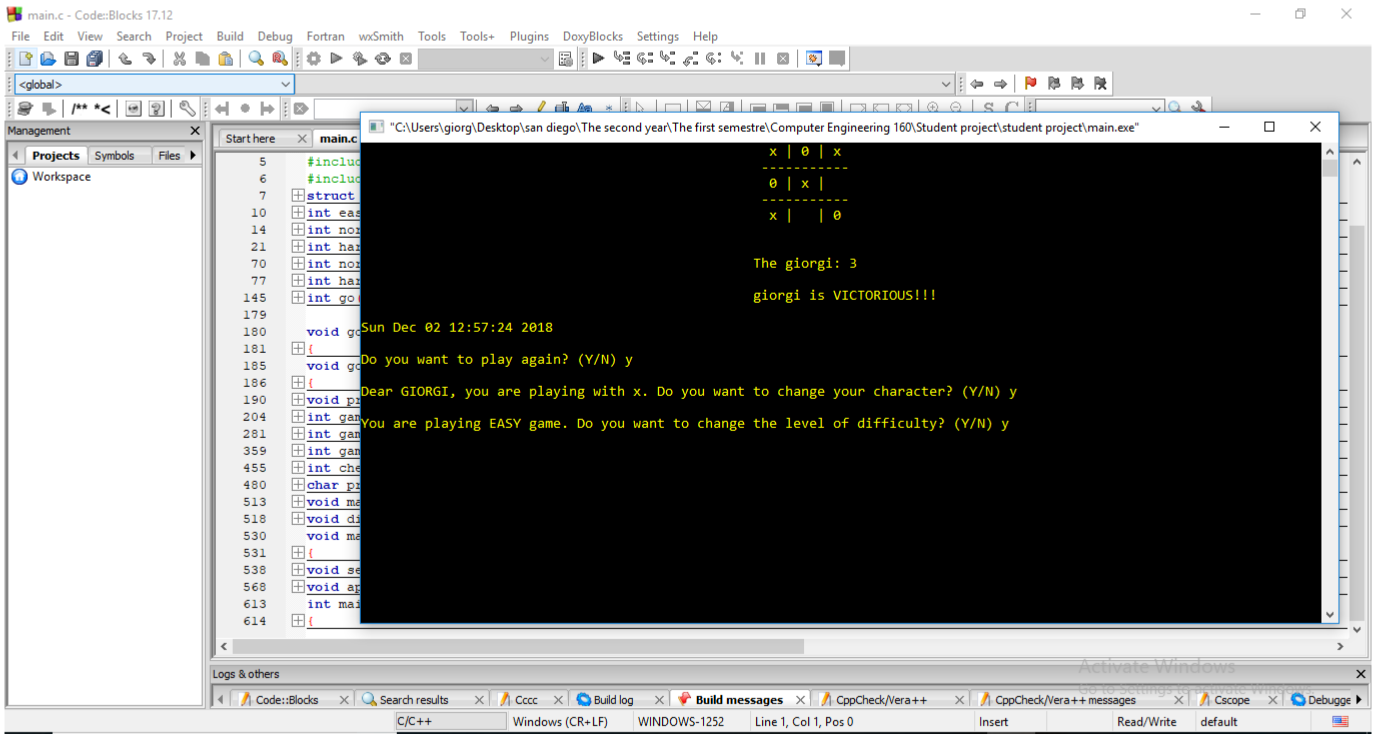
```
                    Choose the level of difficulty (1 2 or 3)

                         1. EASY

                         2. NORMAL

                         3. HARD

                         :> 1
```

Picture 3



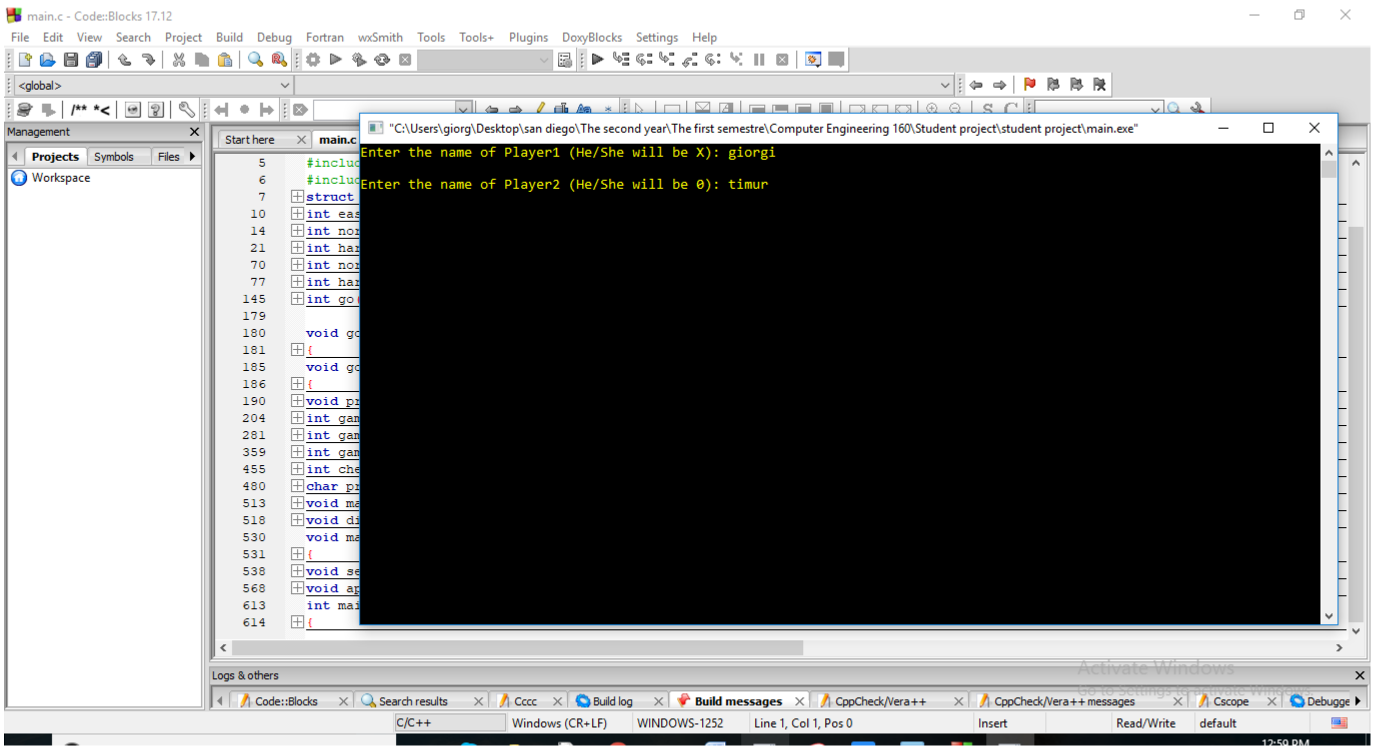
```
"C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Computer Engineering 160\Student project\student project\main.exe"
 1 | 2 | 3
-----------
 4 | 5 | 6
-----------
 7 | 8 | 9

 x | 0 |
-----------
 0 | x |
-----------
 x |   | 0


Enter number between 1-9
The giorgi's Turn:
```

Picture 4



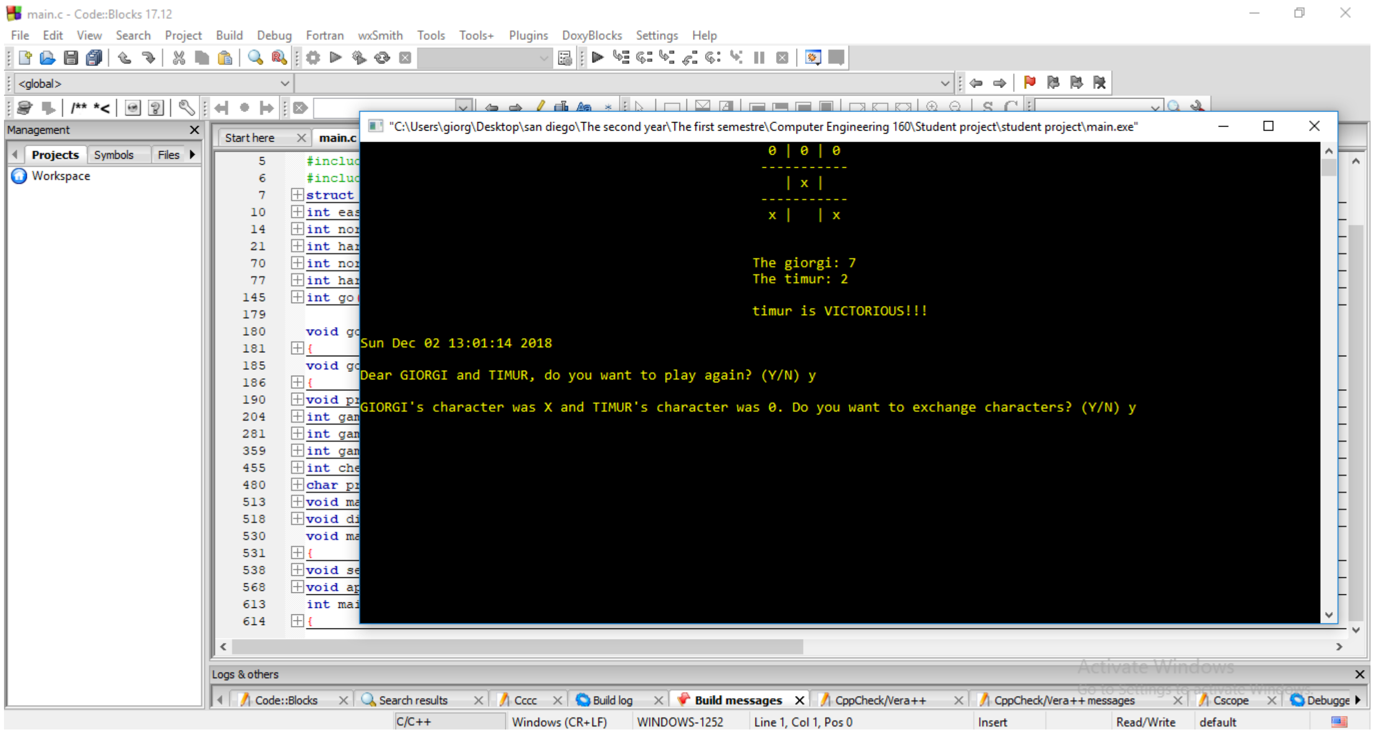
```
"C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Computer Engineering 160\Student project\student project\main.exe"
 x | 0 | x
-----------
 0 | x |
-----------
 x |   | 0


The giorgi: 3

giorgi is VICTORIOUS!!!

Sun Dec 02 12:57:24 2018

Do you want to play again? (Y/N) y

Dear GIORGI, you are playing with x. Do you want to change your character? (Y/N) y

You are playing EASY game. Do you want to change the level of difficulty? (Y/N) y
```

Picture 5



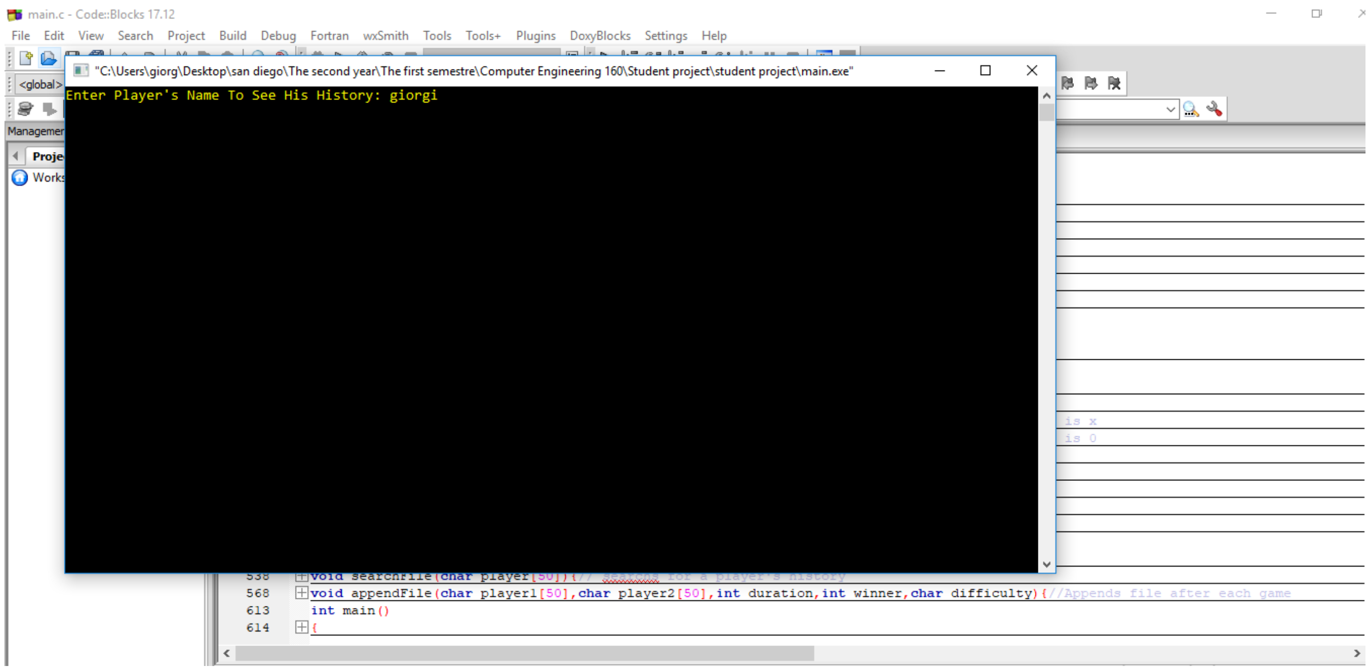
```
Enter the name of Player1 (He/She will be X): giorgi

Enter the name of Player2 (He/She will be 0): timur
```

Picture 6



```
                                        0 | 0 | 0
                                       -----------
                                          | x |
                                       -----------
                                        x |   | x


                                      The giorgi: 7
                                      The timur: 2

                                      timur is VICTORIOUS!!!

Sun Dec 02 13:01:14 2018

Dear GIORGI and TIMUR, do you want to play again? (Y/N) y

GIORGI's character was X and TIMUR's character was 0. Do you want to exchange characters? (Y/N) y
```

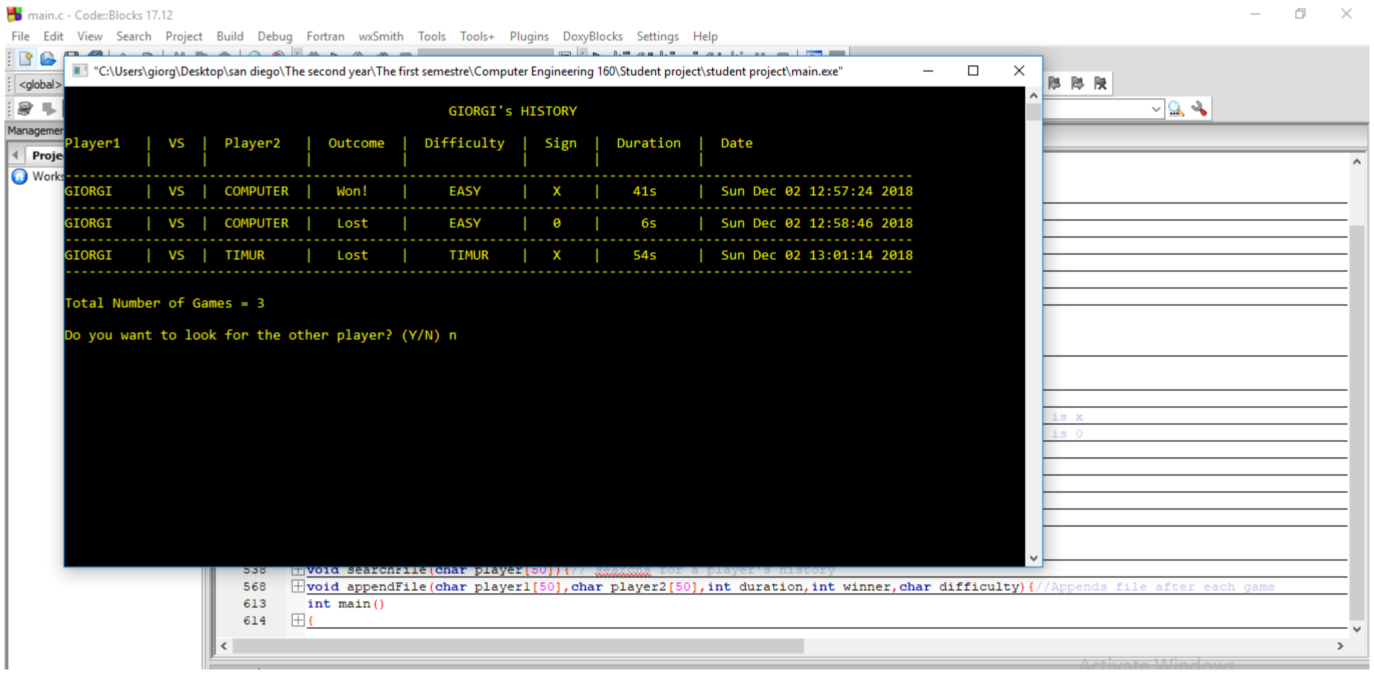
Picture 7


main.c - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
"C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Computer Engineering 160\Student project\student project\main.exe"

```
Enter Player's Name To See His History: giorgi
```

```
void searchFile(char player[50]){// searchs for a player's history
void appendFile(char player1[50],char player2[50],int duration,int winner,char difficulty){//Appends file after each game
int main()
{
```

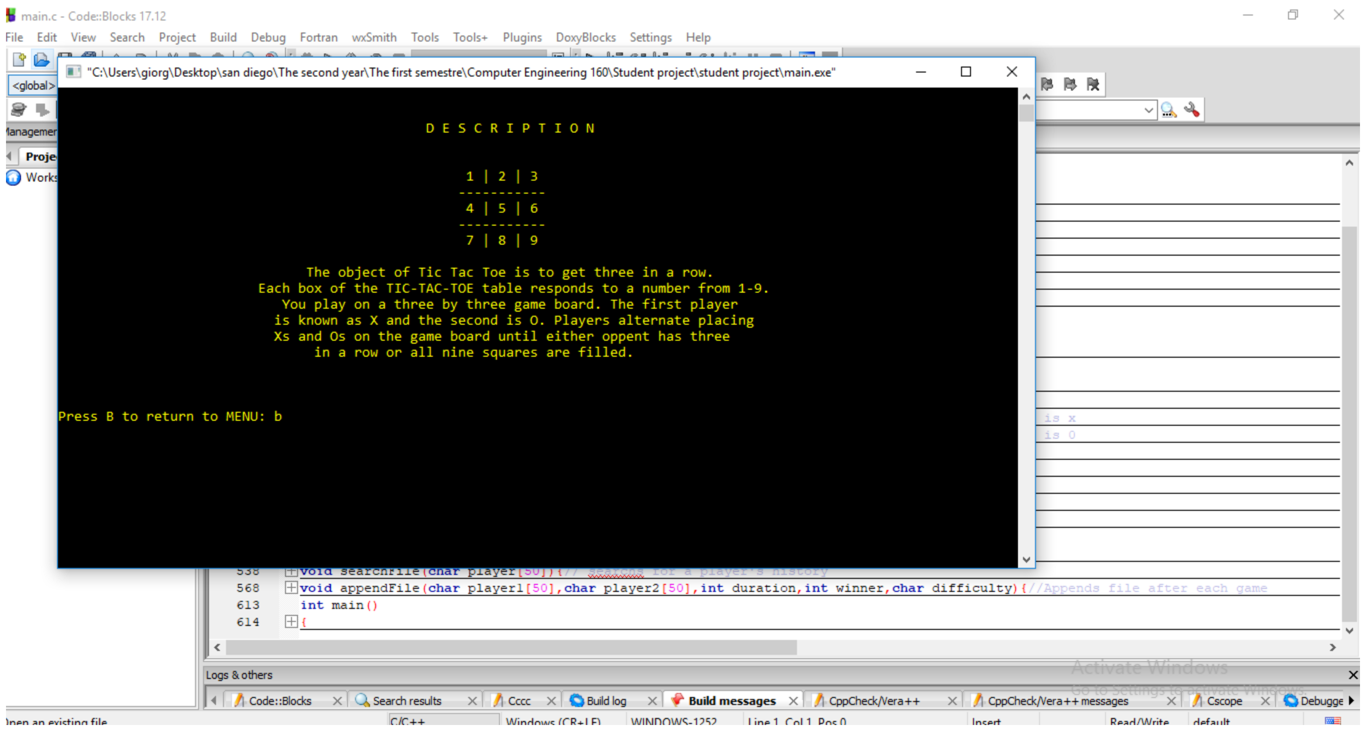
Picture 8


main.c - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
"C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Computer Engineering 160\Student project\student project\main.exe"

```
                                        GIORGI's HISTORY

Player1    |  VS  |  Player2   |  Outcome  |  Difficulty  |  Sign  |  Duration   |  Date
           |      |            |           |              |        |             |
--------------------------------------------------------------------------------------------------
GIORGI     |  VS  |  COMPUTER  |   Won!    |     EASY     |   X    |    41s      |  Sun Dec 02 12:57:24 2018
--------------------------------------------------------------------------------------------------
GIORGI     |  VS  |  COMPUTER  |   Lost    |     EASY     |   0    |     6s      |  Sun Dec 02 12:58:46 2018
--------------------------------------------------------------------------------------------------
GIORGI     |  VS  |  TIMUR     |   Lost    |     TIMUR    |   X    |    54s      |  Sun Dec 02 13:01:14 2018
--------------------------------------------------------------------------------------------------

Total Number of Games = 3

Do you want to look for the other player? (Y/N) n
```

```
void searchFile(char player[50]){// searchs for a player's history
void appendFile(char player1[50],char player2[50],int duration,int winner,char difficulty){//Appends file after each game
int main()
{
```

Picture 9


"C:\Users\giorg\Desktop\san diego\The second year\The first semestre\Computer Engineering 160\Student project\student project\main.exe"

```
                         D E S C R I P T I O N


                              1 | 2 | 3
                             -----------
                              4 | 5 | 6
                             -----------
                              7 | 8 | 9


              The object of Tic Tac Toe is to get three in a row.
        Each box of the TIC-TAC-TOE table responds to a number from 1-9.
           You play on a three by three game board. The first player
          is known as X and the second is O. Players alternate placing
          Xs and Os on the game board until either oppent has three
               in a row or all nine squares are filled.



Press B to return to MENU: b
```

```
void searchFile(char player[50]){// searchs for a player's history
void appendFile(char player1[50],char player2[50],int duration,int winner,char difficulty){//Appends file after each game
int main()
{
```

Picture 10

6,7) Platform and tools

I used code::Blocks and its standard library functions. Functions written by me are in main.c

This program runs on windows. I used system command to make characters color yellow and to clear screen occasionally.

References.

Code::Blocks as a compiler: Downloaded from:
https://sourceforge.net/projects/codeblocks/files/Binaries/13.12/Windows/codeblocks-13.12mingwsetup.exe/download

Basics”, (2018) In zybook. Retrieved from:
https://learn.zybooks.com/zybook/SDSUGeorgiaCompE160Fall2018/