Algorithms & Data Structures Assignment

Tic Tac Toe Game

Module number: SET08122
Napier Number: 40337591
Long Darragh Chow

Introduction

The problem what I have facing in this Tic Tac Toe game is to save every game record in the history, and this is very difficult to achieve. And other problem on this coursework is try to achieve the other requirement such as Player play with the AI (Computer), unfortunately I don't have enough knowledge to implement the AI function to the work. For my coursework, Tic-Tac-Toe (Noughts & Crosses) game is using the C programming language to code and the coursework is basically using the Gun Compiler Collection (GCC) to compile. There are many features can be add into the game such as player has marked its position but he want to undo or redo the last move. And the game can set the game board size not just only 3x3, it can be 5x5 or 9x9. Furthermore, save each game winner in record is one of the feature too.

Design

Fig (1.0)

Fig (2.0)

Command Prompt - 40337591

```
        Tic Tac Toe
Player 1 (X)  -  Player 2 (0)


     1  |  2  |  3
     ___|_____|___
     4  |  X  |  6
     ___|_____|___
     7  |  8  |  9


Player 2, enter a number:
```
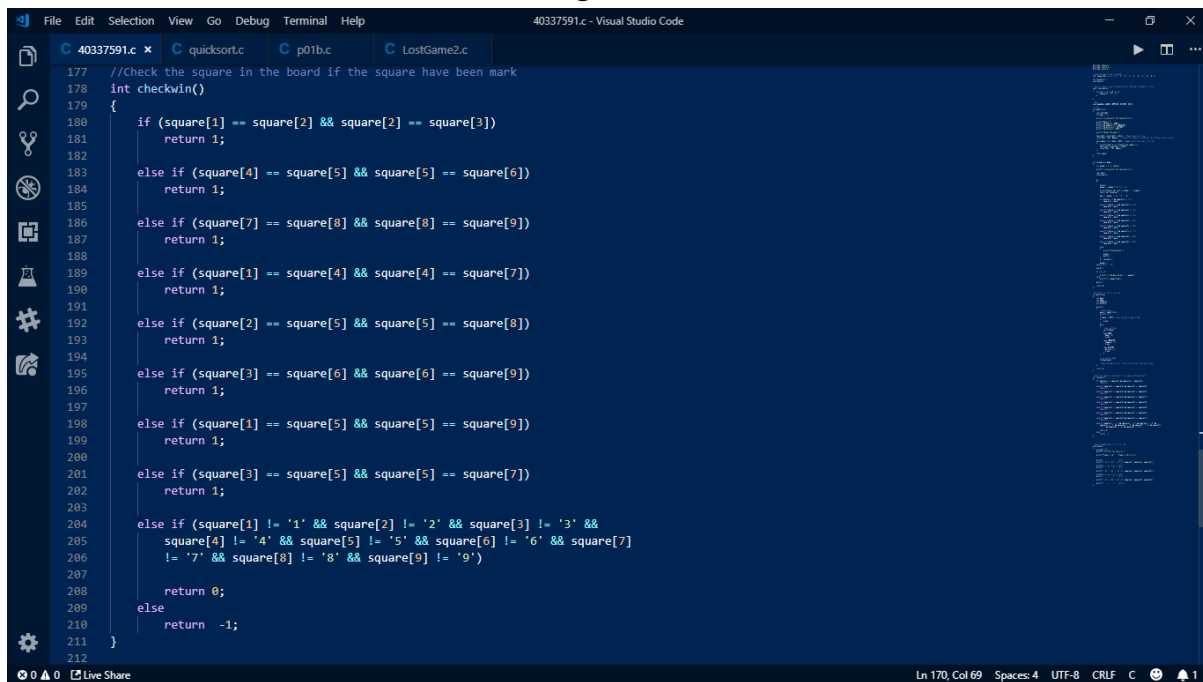
Fig (3.0)

```
54
55
56  int TacGame(int Game)
57  {
58      int player = 1, i, choice;
59
60      printf("\n\n×××Tic Tac Toe×××\n\n");
61
62      char mark;
63      clearsquare();
64
65      do
66      {
67
68          board();
69          player = (player % 2) ? 1 : 2;
70
71          printf("Player %d, enter a number:  ", player);
72          scanf("%d", &choice);
73
74          mark = (player == 1) ? 'X' : 'O';
75
76          if (choice == 1 && square[1] == '1')
77              square[1] = mark;
78
79          else if (choice == 2 && square[2] == '2')
80              square[2] = mark;
81
82          else if (choice == 3 && square[3] == '3')
83              square[3] = mark;
84
85          else if (choice == 4 && square[4] == '4')
86              square[4] = mark;
87
88          else if (choice == 5 && square[5] == '5')
89              square[5] = mark;
```

```
91          else if (choice == 6 && square[6] == '6')
92              square[6] = mark;
93
94          else if (choice == 7 && square[7] == '7')
95              square[7] = mark;
96
97          else if (choice == 8 && square[8] == '8')
98              square[8] = mark;
99
100         else if (choice == 9 && square[9] == '9')
101             square[9] = mark;
102
103         else
104         {
105             printf("Invalid move ");
106
107             player--;
108             getch();
109         }
110         i = checkwin();
111
112         player++;
113     }while (i ==  - 1);
114
115     board();
116
117     if (i == 1)
118
119         printf("==>\aPlayer %d win ", --player);
120     else
121         printf("==>\aGame draw");
122
123     getch();
124
125     return 0;
126
```

Ln 64, Col 5   Spaces: 4   UTF-8   CRLF   C   🙂   🔔 1

Fig (4.0)

```c
//Check the square in the board if the square have been mark
int checkwin()
{
    if (square[1] == square[2] && square[2] == square[3])
        return 1;

    else if (square[4] == square[5] && square[5] == square[6])
        return 1;

    else if (square[7] == square[8] && square[8] == square[9])
        return 1;

    else if (square[1] == square[4] && square[4] == square[7])
        return 1;

    else if (square[2] == square[5] && square[5] == square[8])
        return 1;

    else if (square[3] == square[6] && square[6] == square[9])
        return 1;

    else if (square[1] == square[5] && square[5] == square[9])
        return 1;

    else if (square[3] == square[5] && square[5] == square[7])
        return 1;

    else if (square[1] != '1' && square[2] != '2' && square[3] != '3' &&
        square[4] != '4' && square[5] != '5' && square[6] != '6' && square[7]
        != '7' && square[8] != '8' && square[9] != '9')

        return 0;
    else
        return  -1;
}
```

On the start of the game, the menu appears on the screen. And there has 4
option, Game, Computer, history and Exit. Pressing '0' to start the game, it will
show board Fig(1.0), the game has been set Player1 will assign as 'X' , Player2
assign as 'O'. Player1 can select any positions between 1 to 9 first and turn to
Player2. If player1 select a position 5, position 5 will save as X move on the
board (Fig(2.0)). After player1 marked X as position 5, player2 cannot mark
positon 5 again. Every position have been would not be able to mark again. In
coding, every position is saved on the array (fig(0.3)). And fig(0.3) has been
shown the code implement the function for the game, it consist the player
move, check the position and win function (fig(0.4)). In the main function, it has
a menu has been implemented with a switch statement which can choose the
title they want. For example, if user input 0, it go to game, 1 to computer, 2 to
history and 3 to exit. Otherwise, if user choose the number out of 0 to 3, it will
show the text "Input error. Please enter again". Therefore in the code, every
function only work on one things, and that will be easier to change the code in
the future.

Enhancements

It would be few features to the game.

1. Player with AI (Computer): Allow player play with the AI. In the game menu, I already have an option – Computer, that's for the future when I complete the AI feature.
2. Choice of 'X' and 'O': Normally 'X' always is the first move for the game, but I would like to change the first move can be 'O'. It can add one more code like Player can choose 'X' or 'O' for the first move.
3. Improve the History: The only things I want to finish before submit the coursework is complete the recode the game history for every game.
4. More game board: Increase more game board not just only 3x3, it could be 5x5 or 9x9, and set the rule how to determine the winner.

I would improve the History on the menu, because the record every game result is the most important thing for every game. Also try to change the background to different design. Like the other famous game Mario, they designed a good menu and background, also with the background music, these elements are very important for making more player interest to the game.

Critical Evaluation

The most I felt do very well is the MENU, because I have an experience to make maze game, and it has included to use the MENU, and I'm using my experience in to the coursework, that make me used least time to code the game. That's the reason I made a MENU in this coursework and feel doing well. The poor feature of my work is the check the winner in this game. It was harder than print the game board. Even I know how to set up the player move but I have no idea how to check the winner in the final. And I found out some people they put the check winner function into the main function or separate it in different part, but I decided to separate the function in different part because it make people easy to check and easy to understand my code.

Personal Evaluation

I have not learn any programming language before I start my university. I am completely zero knowledge about how to code, even I have learnt Java or C# in last year or last trimester. But when I first touch on C. It easier to understand than the code language I have learnt before. Therefore I am very enjoyable to learn the C language. whatever how I enjoy the C language, but I still have to face different problem such as I have to figure it out how to use the example function or method to my coursework, before I completed the basic requirement, the only thing I can do is keep debugging the code until there's no error, or even there have no error but it do not show the result I want. That's the harder challenge I faced in coursework. Finally, what I have learn in totally is Coding do not have correct answer, the only things I can do is keep learning and coding to achieve my work.