# Design and implementation Document

**I decided to use the C language with procedural paradigms to implement the modified tic-tac-toe programme. Below is the pseudocode of the overall of the modified tic-tac-toe programme.**

A screenshot of a computer program

Description automatically generated

Figure 1- Pseudocode of tic-tac-toe

**In Figure 1 I have shown the overall pseudocode of tic-tac-toe, so I have written pseudocode for each function below:**

1. Display Board Function

A screenshot of a computer program

Description automatically generated

Figure - Pseudocode of display board function

1. Winner Function

A screenshot of a computer program

Description automatically generated

Figure - pseudocode of winner function

1. Players Move Function

A screenshot of a computer program

Description automatically generated

Figure - pseudocode of Player Move function

1. Game Function

A screenshot of a computer game

Description automatically generated

Figure - pseudocode of game function

1. Main Function

A black text on a white background

Description automatically generated

Figure - pseudocode of main function

**Follow the pseudocode function, I have written the code for each function below:**

1. Display Board Function

A computer code on a dark background

Description automatically generated

Figure 7- Displays Board function

1. Winner Function

A screenshot of a computer program

Description automatically generated

Figure 8-Winner function

1. Players Move Function

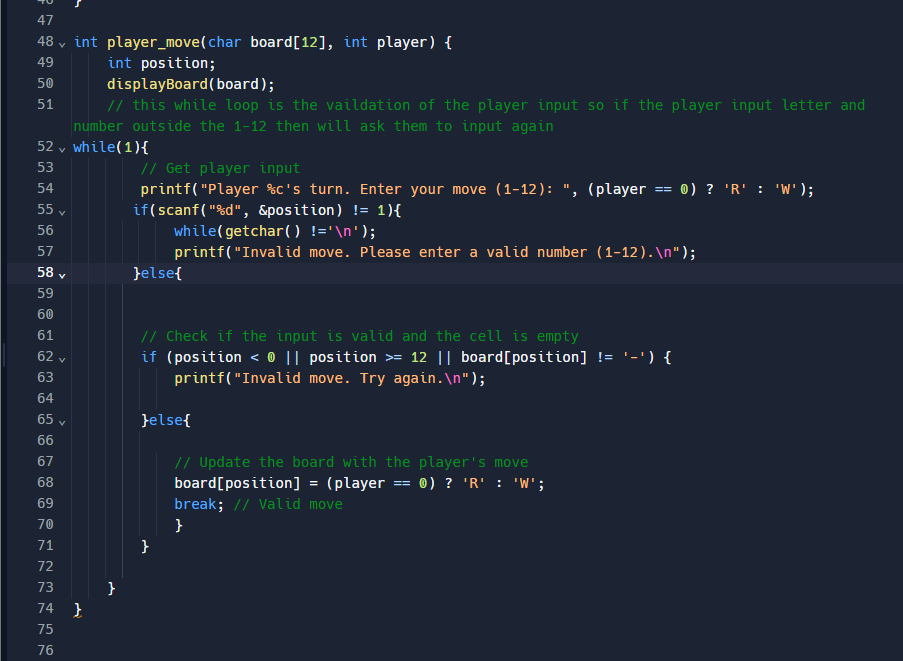


Figure 9- Players move function

1. Game Function

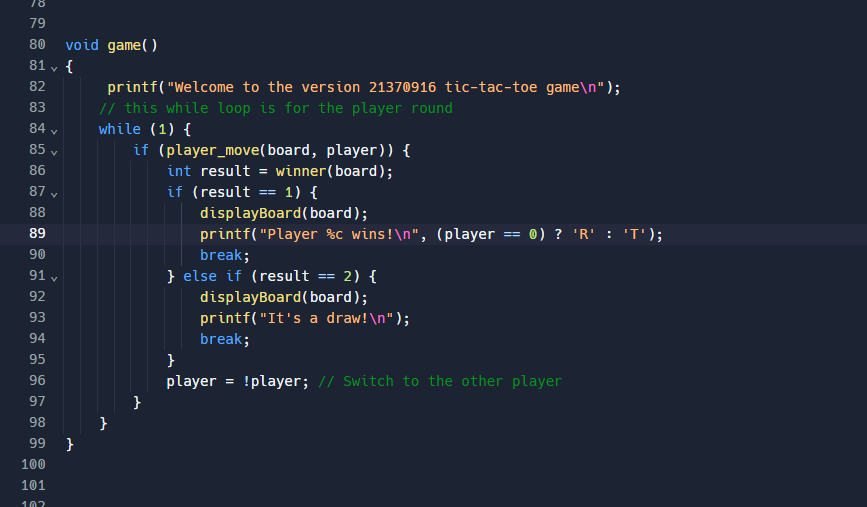


Figure 10- Game function

1. Main Function

A screen shot of a computer

Description automatically generated

Figure 11- Main function

**In the iteration code of my implementation process, I wrote the following code:**

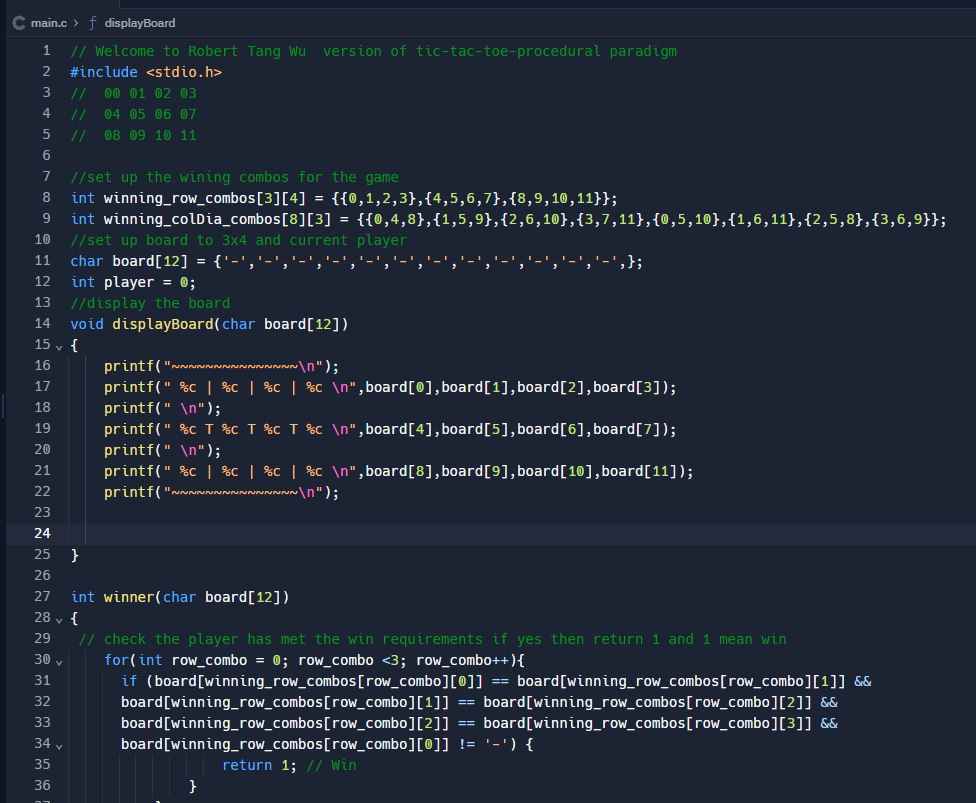


Figure 12- Iteration code part 1

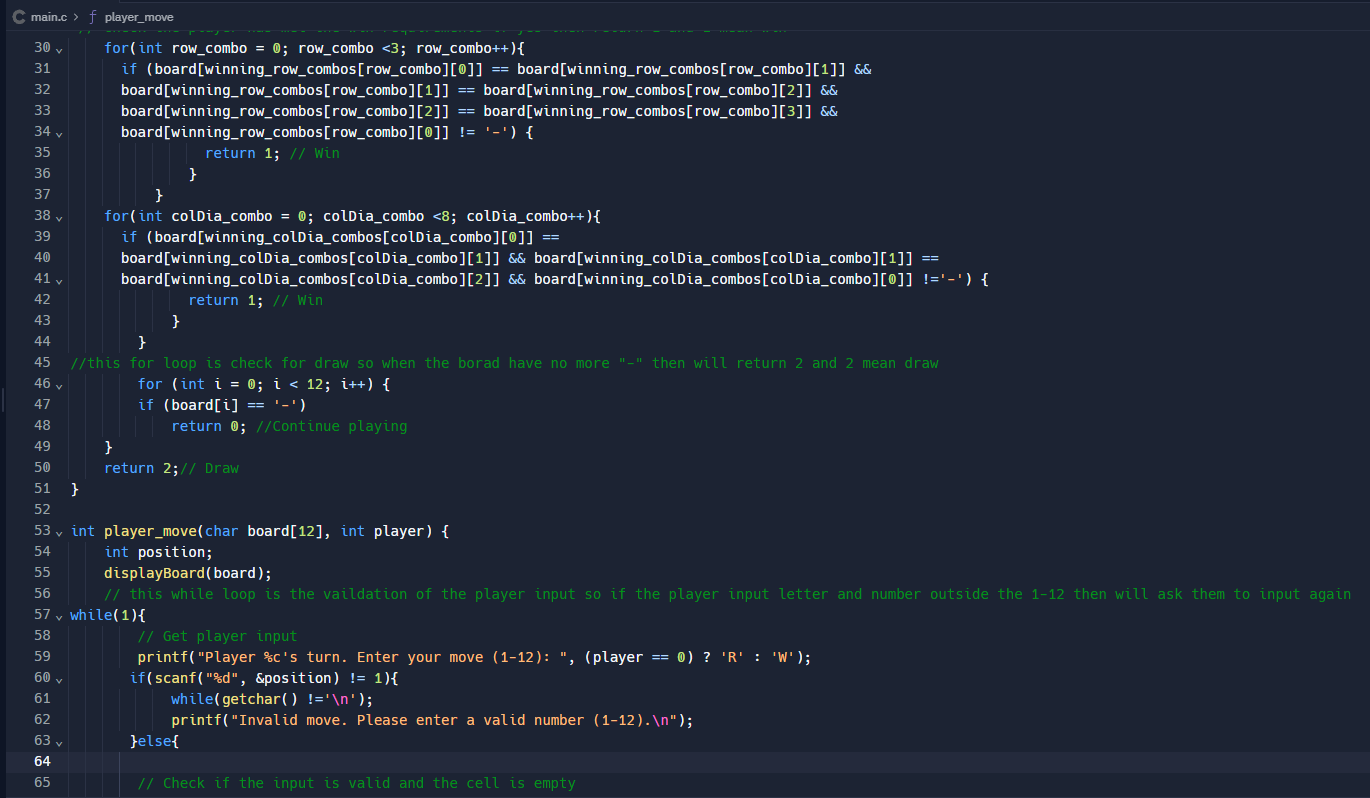


Figure - Iteration code part 2

A screenshot of a computer program

Description automatically generated

Figure 14- Iteration code part 3

A computer screen shot of a program code

Description automatically generated

Figure 15- Iteration code part 4

**In the iteration of my implementation, I have found some error, so I have updated the code**

**A screenshot of a computer

Description automatically generated**

Figure 16-Error of the code when input number 0 the” R” symbol is input to the board

A screenshot of a computer program

Description automatically generated

Figure 17- Error of the code when input number 12 to the board is not shown “R” symbol in last cell of the board

A screenshot of a computer

Description automatically generated

Figure 18-The error of the code when W win is shown the "T" win the game

A screen shot of a computer program

Description automatically generated

Figure 19- The code haven’t set up number start 1-12 instead start 0-11

A screen shot of a computer program

Description automatically generated

Figure 20- This causes the winning result error because set up the "T" win not "W"

A screen shot of a computer program

Description automatically generated

Figure 21- Update code add the position-- to the code, now the range of the number will start from 1-12 not 0-11

A screenshot of a computer

Description automatically generated

Figure 22-Result of the redesign code to allow the player to input 1-12

A computer screen shot of a program code

Description automatically generated

Figure 23- Update code change the "T" to "W" to ensure the not "T" win when the "W" win

A screen shot of a computer

Description automatically generated

Figure 24- Result of the redesign code- when W won then print player W wins!

**When the whole program working, and I have tested the programming. These are the tests that I conducted:**

1. Run to the end with R player winning

A computer screen shot of a computer code

Description automatically generated

Figure 25- Test R player wining (col)

A screenshot of a computer

Description automatically generated

Figure 26-Test R player wining (diagonal)

A screen shot of a computer

Description automatically generated

Figure 27-Test R player wining (row)

1. Run to the end with W player winning

A screen shot of a computer

Description automatically generated

Figure 28- Test W player wining (col)

A screenshot of a computer

Description automatically generated

Figure 29- Test W player wining (diagonal)

A screenshot of a computer

Description automatically generated

Figure 30-Test W player wining (row)

1. Run to the end with a draw.

A screenshot of a computer

Description automatically generated

Figure 31- Test the game draw

1. Validation of input the out-of-range number

A screenshot of a computer program

Description automatically generated

Figure 32- Test input out of range number

1. Validation of input a letter

A screenshot of a computer program

Description automatically generated

Figure 33- Test input a letter

1. Validation of input of the symbol to the cell that already exists

A screenshot of a computer program

Description automatically generated

Figure 34- Test input of the symbol to the cell that already exists