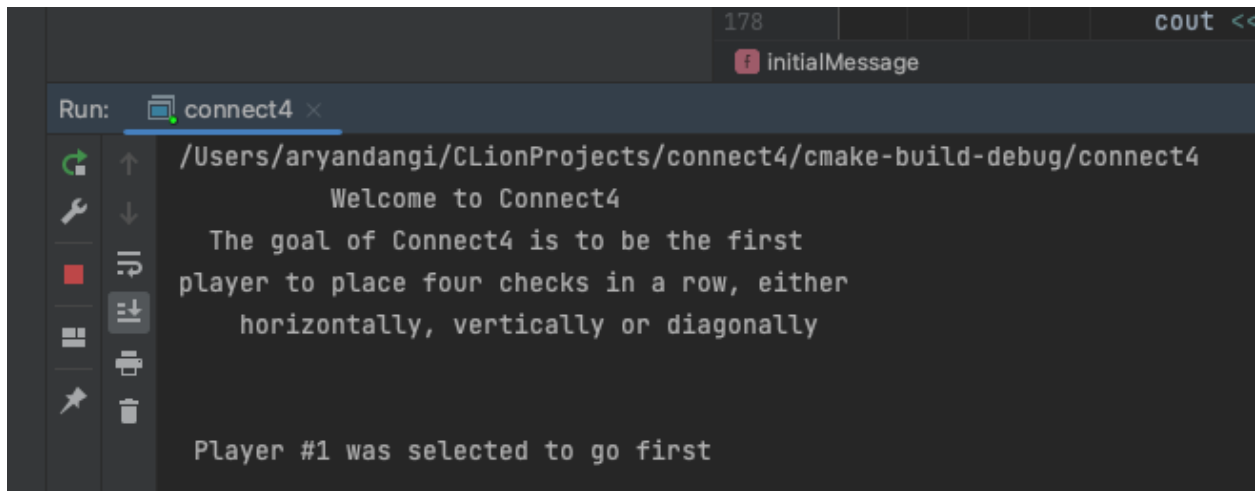


# CPET 121 - Computational Problem Solving

## Program Operation Document - Connect4 - C++

### 1. Introduction and player to go first screen



```
178 cout << endl;
initialMessage


Run: connect4 x
/Users/aryandangi/CLionProjects/connect4/cmake-build-debug/connect4
Welcome to Connect4

The goal of Connect4 is to be the first
player to place four checks in a row, either
horizontally, vertically or diagonally

Player #1 was selected to go first
```

This basically outputs the welcome message of Connect4 and randomly selects which player goes first.

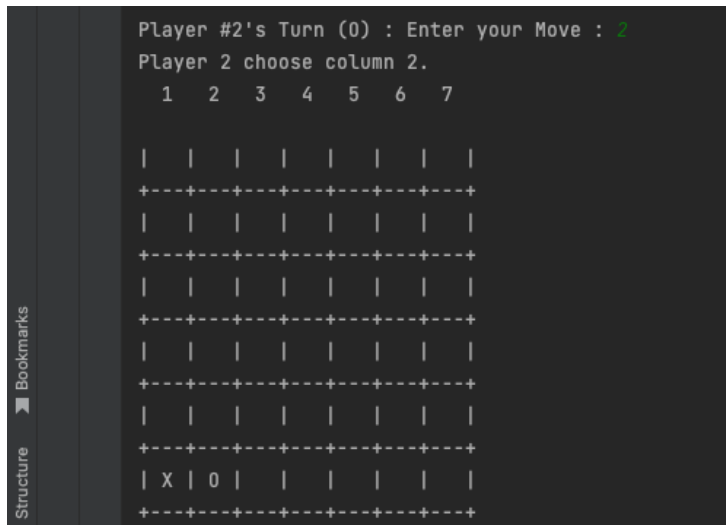
### 2. Player X input screen



```
Player #1's Turn (X) : Enter your Move : 1
Player 1 choose column 1.
 1  2  3  4  5  6  7
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
| X |  |  |  |  |  |
+---+---+---+---+---+---+
Player #2's Turn (O) : Enter your Move :
```

This screen outputs the game board and what column player wanted to put the X in. It also outputs the final grid when the player has added his X in a particular position for the next player to see the board and play his next move.

### 3. Player O input screen



```
Player #2's Turn (0) : Enter your Move : 2
Player 2 choose column 2.
  1  2  3  4  5  6  7
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
| X | O |  |  |  |  |
+---+---+---+---+---+---+
```

This screen outputs the game board after the second player has played his moves. It shows where player 2 places the O and the final board from which player 1 can again see and play his move.

### 4. Illegal move screen



```

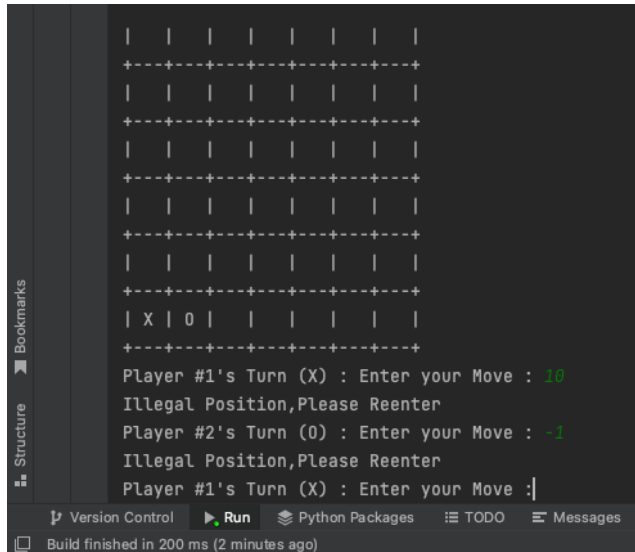
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
|  |  |  |  |  |  |
+---+---+---+---+---+---+
| X | O |  |  |  |  |
+---+---+---+---+---+---+

Player #1's Turn (X) : Enter your Move : 10
Illegal Position, Please Reenter
Player #2's Turn (O) : Enter your Move : -1
Illegal Position, Please Reenter
Player #1's Turn (X) : Enter your Move :|

Version Control  Run  Python Packages  TODO  Messages
Build finished in 200 ms (2 minutes ago)
```

This screen shows what happens when a player enters an illegal column number in which the code displays an error message and asks the user to input values again.

## 5. Loose turn screen



```
| | | | | | |
+---+---+---+---+---+
| | | | | | |
+---+---+---+---+---+
| | | | | | |
+---+---+---+---+---+
| | | | | | |
+---+---+---+---+---+
| X | O | | | | |
+---+---+---+---+---+
Player #1's Turn (X) : Enter your Move : 10
Illegal Position,Please Reenter
Player #2's Turn (O) : Enter your Move : -1
Illegal Position,Please Reenter
Player #1's Turn (X) : Enter your Move : |
```

This screen also shows when a player enters an illegal column number, the player loses their turn from which the other player gets a chance again to play.

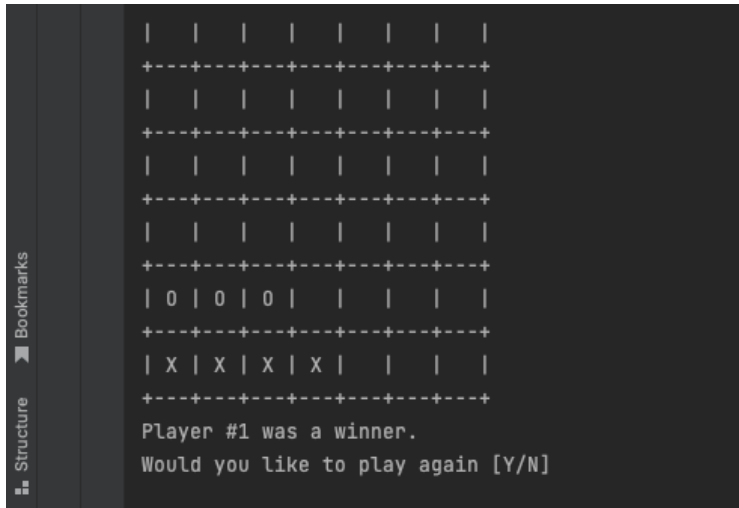
## 6. Vertical Win Screen



```
| | | | | | |
+---+---+---+---+---+
| | | | | | |
+---+---+---+---+---+
| X | | | | | |
+---+---+---+---+---+
| X | O | | | | |
+---+---+---+---+---+
| X | O | | | | |
+---+---+---+---+---+
| X | O | | | | |
+---+---+---+---+---+
Player #1 was a winner.
Would you like to play again [Y/N]
```

This screen shows the winner output when a player inputs four characters vertically.

## 7. Horizontal win screen



This screen shows the winner output when a player manages to output four characters horizontally.

## 8. Right Diagonal win screen



This screen shows the winner output when a player manages to output four consecutive characters in a right diagonal.

## 9. Left Diagonal win screen

```
| | | | | | | |
+---+---+---+---+---+---+
| | | | | | | |
+---+---+---+---+---+---+
| | | X | | | |
+---+---+---+---+---+---+
| | | O | X | | |
+---+---+---+---+---+---+
| | | O | O | X | |
+---+---+---+---+---+---+
| | X | X | O | O | X |
+---+---+---+---+---+---+
Player #1 was a winner.
Would you like to play again [Y/N]
|
```

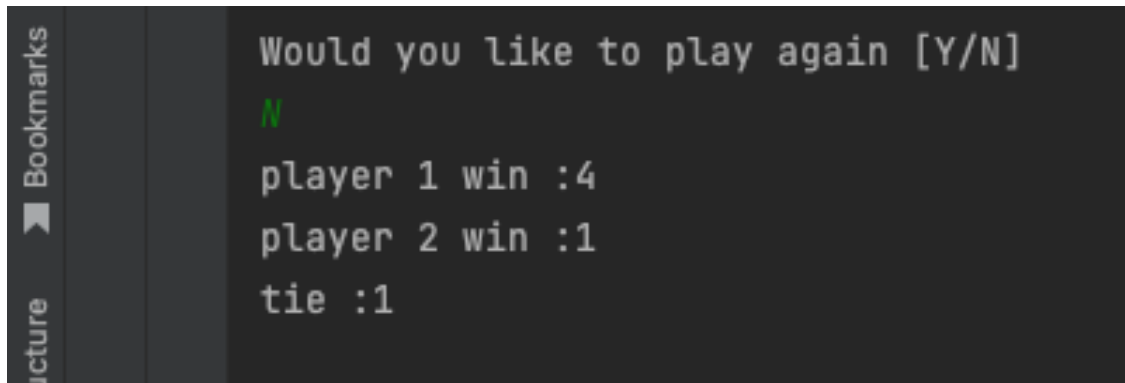
This screen shows the winner output when a player manages to put four consecutive characters in a left diagonal or something called an anti diagonal.

## 10. Tie game screen

```
1 2 3 4 5 6 7
| X | O | X | O | X | O | X |
+---+---+---+---+---+---+
| O | X | O | X | O | X | O |
+---+---+---+---+---+---+
| O | X | X | O | X | O | X |
+---+---+---+---+---+---+
| O | O | X | O | X | O | X |
+---+---+---+---+---+---+
| X | O | X | O | X | O | X |
+---+---+---+---+---+---+
| O | X | O | X | O | X | O |
+---+---+---+---+---+---+
The Game Ended In A Tie
Would you like to play again [Y/N]
```

This screen shows the output when the board is tied. It means that when no player has managed to get four characters consecutively in any pattern and the spaces of the board have run out, the game is said to be tied and the “play again” statement is outputted if the players want to try again.

## 11. Game tally screen

A screenshot of a terminal window showing a game tally screen. On the left side, there is a vertical sidebar with the text 'Bookmarks' and 'Jcture'. The main area of the terminal displays the text 'Would you like to play again [Y/N]' in a light gray font. Below this, the character 'N' is shown in a green font, indicating the user's input. Further down, the tally is displayed in a light gray font: 'player 1 win :4', 'player 2 win :1', and 'tie :1'.

This screen shows what happens when the players no longer want to play the game. If the character “N” is selected, the game exits and shows a tally of which player won and how many games were tied.

### Testing Philosophy

- The code was tested in blocks because there were many elements to consider, test and see if they were working well.
- Each function was tested to see if they were working well by calling them in the main and printing each output before the final output.
- The game board was tested with different values which included values not present in the acceptance frame to see whether the code has the ability to read the values and give the error.
- All tests of horizontal, vertically, diagonal and anti-diagonal were taken and seen if the code is output the winner when four consecutive characters are the patterns given.