Saad Reprot

by Muhammad Saad

Submission date: 20-Nov-2025 08:21AM (UTC+0500)
Submission ID: 2821568141
File name: connectFour_PF_Lab_-_Final_Project_Report.docx (1.74M)
Word count: 789
Character count: 4369

A very advanced, console-based implementation of the classic 'Connect Four' game.

connectFour+

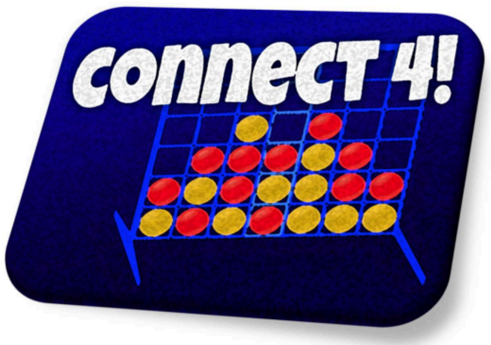
PF Lab – Final Project Report

Muhammad Saad (25K-0604)

FINAL PROJECT REPORT

PROGRAMMING FUNDAMENTALS

University Name:	FAST Nuces KHI
Department:	Department of Computer Science
Course:	Programming Fundamentals
Project Title:	connectFour+
Submitted By:	Muhammad Saad (25K-0604)
Submitted To:	Sir Sheeraz Iqbal
Semester:	Fall 2025
Date:	21 November, 2025



CONTENTS

1. INTRODUCTION

connectFour+ is an advanced console-based version of the famous “Connect Four” game. The game consists of a grid-like gameboard with openings on the top of each column. Players take turns to drop their pieces (red or blue usually) into one of the available columns. The objective of the game is to make a consecutive row of 4 with only 1 types of piece; this could be horizontally, vertically, or diagonally. Whichever player manages to perform this, wins.

This implementation of the game consists of various matchModes, gameModes, gameboard-sizes, players, persistence, gameHistory, leaderBoards, an about section, as well as many other features. All of these options and features are neatly arranged in menus which could easily be accessed and used by the user.

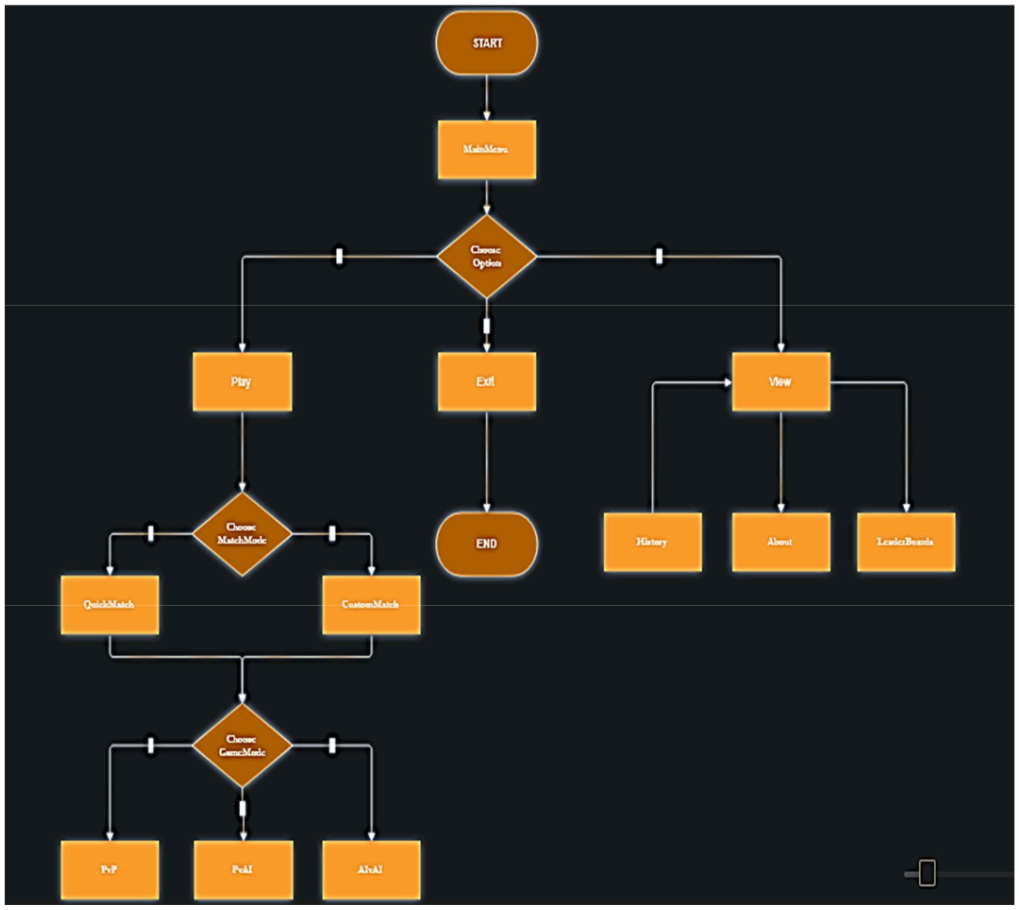
The project took almost a full month to make (almost the complete month of October, 2025) and consists of 2600-3200 loc (range due to different measures of counting).

2. OBJECTIVES

1) To allow users to play the connectFour game
2) To provide the user with a menu with 3 options: Play, View, & Exit
3) To provide the user with 3 gameModes: PvP, PvAI, AIvAI
4) To allow the user to configure the settings (like gameboard size and the chosen AI)
5) To animate each move played
6) To display the results of the game in the end
7) To allow the user to see the complete game history, and to replay each individual game later on
8) To track the rankings of each player and allow the user to see the leaderBoards in a tabulized manner
9) To allow the user to get help about the game (both brief and detailed)
10) To allow the user to exit the program whenever he/she wishes to do so

3. SYSTEM DESIGN

SYSTEM OVERVIEW/FLOW



4. IMPLEMENTATION

Language: C

Compiler/IDE: VS Code / Dev C++ / GCC

KEY FEATURES

- **LOCs**
  - 2596 Lines of Pure-Code
  - 3007 loc after formatting with C/C++ extension in VSCode
  - +210 loc if ABOUT included
- **Menu**
- **QuickMatch & CustomMatch options**
- **Multiple GameModes**
- **Multiple GameBoard sizes**
- **Animated printing of each move**
- **Persistence (Game History)**
- **Replay past games**
- **LeaderBoards**
- **Help (About)**

CODE SNIPPET

```c
void PvAI()                                                                    //
player vs AI mode
{
    setGame();                                                                 //
initializing game. globals
    if (game.quickMatch)                                                       //
quickMatch() mode
    {
        choosePlayer(0);                                                       //
initially did this:  strcpy(game.player1Name, "Player 1");
        strcpy(game.player2Name, "Thalith");                                   //
defaulting ai with thalith
        game.rowCount = 6;                                                     //
Classic (7 x 6) gameBoard
        game.colCount = 7;
    }
    else                                                                       //
customMatch() mode
```

```
    {
        setPlayers();                                                                   //
choose players
        setGameBoard();                                                                 //
choose gameBoard size
    }
    if (!program.randomSeeded){ srand(time(NULL)); program.randomSeeded = true; }      //
seeding rand() with time(NULL) once per program
    startGame();                                                                        //
printing the gameBoard & a little pause before starting game

    while (game.playGame)                                                               //
infinite gameLoop until a terminal state is reached
    {
        switchActivePlayer();                                                           //
switches game.activePlayer
        if  (game.activePlayer == 1) { getPlayerMove(); }                               //
gets move based on game.activePlayer
        else                         { getAIMove();     }                               //
getting AI's move when game.activePlayer is even (2 + 2n)
        updateGameBoard();                                                              //
updates the gameBoard row by row & calls printGameBoard in between them
        updateGameState();                                                              //
updates game.gameState based upon the 5 checking functions
        evaluateGameBoard();                                                            //
evaluating game based upon game.gameState
        saveGameBoardDetails();                                                         //
updates the gameBoardHistory.txt file with the current playerMove
    }

    saveGameDetails();
    updateLeaderBoards();                                                               //
leaderBoards will only be updated if the mode is not AIvAI (is PvP or PvAI)
    pressEnterToContinue();                                                             //
waits for user input before clearing the screen & reprinting mainMenu
}
```

5. CONCLUSION, LIMITATIONS & REFERENCES

CONCLUSION

Hence the report demonstrates that this implementation of the connectFour game is very good. :)

https://github.com/The-Saad-El/Connect-Four

LIMITATIONS

1) Lack of a GUI
2) Lack of an advanced AI algorithm (miniMax)
3) Lack of a resumeGame feature
4) Lack of multiplayer capability using networking
5) Major blinking when printing gameboard
6) Used global variables and functions without any parameters

FUTURE ENHANCEMENTS

1) Add a GUI layer using RAYLIB & RAYGUI
2) Implement MiniMax & LookAhead AIs (with depth 5+)
3) Implement resumeGame feature using filing
4) Implement multiplayer feature using socket-networking (client-peer model)
5) Disable cursor when printing the gameboard (using \033 escape sequences)
6) Pass arguments to functions when called (so to properly see program-flow)

REFERENCES

- ❖ Let Us C by Yashavant P. Kanetkar
- ❖ https://www.programiz.com/c-programming
- ❖ https://www.geeksforgeeks.org/c-programming-language/

Saad Reprot

ORIGINALITY REPORT

3%	3%	0%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	louisdl.louislibraries.org Internet Source	2%
2	www.coursehero.com Internet Source	1%

Exclude quotes	On	Exclude matches	< 3 words
Exclude bibliography	On		