Name haris Farooq

Class bscs

Section 3d

Code

#include<iostream>

#include<windows.h>

class TicTacToe {

private:

char board[3][3];

char currentPlayer;

public:

TicTacToe() : currentPlayer('X') {

// Initialize the board with empty spaces

for (int i = 0; i < 3; ++i) {

for (int j = 0; j < 3; ++j) {

board[i][j] = ' ';

}

}

}

void printBoard() {

for (int i = 0; i < 3; ++i) {

for (int j = 0; j < 3; ++j) {

std::cout << board[i][j] << " | ";

}

std::cout << std::endl << "---------" << std::endl;

}

}

bool makeMove(int row, int col) {

if (row < 0 || row >= 3 || col < 0 || col >= 3 || board[row][col] != ' ') {

return false; // Invalid move

}

board[row][col] = currentPlayer;

currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';

return true; // Valid move

}

bool checkWin() {

// Check rows, columns, and diagonals for a win

for (int i = 0; i < 3; ++i) {

if (board[i][0] != ' ' && board[i][0] == board[i][1] && board[i][1] == board[i][2]) {

return true; // Row win

}

if (board[0][i] != ' ' && board[0][i] == board[1][i] && board[1][i] == board[2][i]) {

return true; // Column win

}

}

if (board[0][0] != ' ' && board[0][0] == board[1][1] && board[1][1] == board[2][2]) {

return true; // Diagonal win

}

if (board[0][2] != ' ' && board[0][2] == board[1][1] && board[1][1] == board[2][0]) {

return true; // Diagonal win

}

return false; // No win yet

}

bool isBoardFull() {

for (int i = 0; i < 3; ++i) {

for (int j = 0; j < 3; ++j) {

if (board[i][j] == ' ') {

return false; // Board is not full

}

}

}

return true; // Board is full

}

char getCurrentPlayer() const {

return currentPlayer;

}

};

int main() {

using namespace std;

system("color 6");

TicTacToe game;

while (true) {

// Print the current state of the board

game.printBoard();

// Get the current player's move

int row, col;

cout << "Player " << game.getCurrentPlayer() << ", enter your move (row and column): ";

cin >> row >> col;

// Make the move and check for a win

if (game.makeMove(row, col)) {

if (game.checkWin()) {

cout << "Player " << game.getCurrentPlayer() << " wins!" << endl;

break;

}

else if (game.isBoardFull()) {

cout << "It's a draw!" << endl;

break;

}

}

else {

cout << "Invalid move. Try again." << endl;

}

}

return 0;

}

Result



