**24F-0040**

**Laiba**

**Lab 06 class task**

**Code:**

#include <iostream>

#include <stack>

#include <string>

using namespace std;

// ====================== Game History ======================

struct GameResult {

string winnerName;

int score;

int tries;

};

stack<GameResult> history;

// ====================== Tic Tac Toe Core ======================

char board[3][3];

char currentMarker;

int currentPlayer;

void resetBoard() {

char start = '1';

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

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

board[i][j] = start++;

}

}

}

void drawBoard() {

cout << "\n";

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

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

cout << " " << board[i][j];

if (j < 2) cout << " |";

}

cout << "\n";

if (i < 2) cout << "-----------\n";

}

cout << "\n";

}

bool placeMarker(int slot) {

int row = (slot - 1) / 3;

int col = (slot - 1) % 3;

if (board[row][col] != 'X' && board[row][col] != 'O') {

board[row][col] = currentMarker;

return true;

}

else {

return false;

}

}

// return empty string if no winner

string checkWinner(string name1, string name2) {

// Rows

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

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

return (currentPlayer == 1 ? name1 : name2);

}

}

// Columns

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

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

return (currentPlayer == 1 ? name1 : name2);

}

}

// Diagonals

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

return (currentPlayer == 1 ? name1 : name2);

}

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

return (currentPlayer == 1 ? name1 : name2);

}

return "";

}

void swapPlayerAndMarker(char markerP1, char markerP2) {

if (currentPlayer == 1) {

currentPlayer = 2;

currentMarker = markerP2;

}

else {

currentPlayer = 1;

currentMarker = markerP1;

}

}

void game(string name1, string name2) {

cout << "Player 1, enter your name: ";

cin >> name1;

cout << "Player 2, enter your name: ";

cin >> name2;

resetBoard();

cout << name1 << ", choose your marker (X or O): ";

char markerP1, markerP2;

cin >> markerP1;

if (markerP1 == 'x' || markerP1 == 'X') {

markerP1 = 'X';

markerP2 = 'O';

}

else {

markerP1 = 'O';

markerP2 = 'X';

}

currentPlayer = 1;

currentMarker = markerP1;

drawBoard();

string playerWon;

for (int i = 0; i < 9; i++) {

if (currentPlayer == 1) {

cout << name1 << ", enter your slot (1-9): ";

}

else {

cout << name2 << ", enter your slot (1-9): ";

}

int slot;

cin >> slot;

if (slot < 1 || slot > 9) {

cout << "Invalid! Try again.\n";

i--;

continue;

}

if (!placeMarker(slot)) {

cout << "Slot already occupied! Try again.\n";

i--;

continue;

}

drawBoard();

playerWon = checkWinner(name1, name2);

if (playerWon != "") {

cout << playerWon << " wins!\n";

// Save result in stack

GameResult result;

result.winnerName = playerWon;

result.score = 1;

result.tries = history.size() + 1;

history.push(result);

return;

}

swapPlayerAndMarker(markerP1, markerP2);

}

cout << "It's a draw!\n";

// Save draw result

GameResult result;

result.winnerName = "draw";

result.score = 0;

result.tries = history.size() + 1;

history.push(result);

}

// ====================== History ======================

void showHistory() {

cout << "\n=== Game History ===\n";

cout << "Name\tScore\tTry\n";

cout << "-------------------------\n";

if (history.empty()) {

cout << "No games played yet.\n";

return;

}

stack<GameResult> temp = history; // copy stack

while (!temp.empty()) {

GameResult r = temp.top();

cout << r.winnerName << "\t" << r.score << "\t" << r.tries << "\n";

temp.pop();

}

cout << "-------------------------\n";

}

// ====================== Main Menu ======================

int main() {

string name1, name2;

int choice;

do {

cout << "\n=== Tic Tac Toe Menu ===\n";

cout << "1. Play Game\n2. Show History\n3. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

if (choice == 1) game(name1, name2);

else if (choice == 2) showHistory();

else if (choice == 3) cout << "Goodbye!\n";

else cout << "Invalid choice, try again.\n";

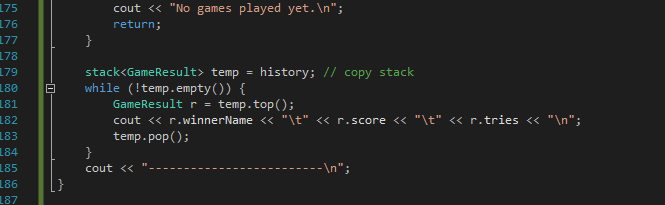
} while (choice != 3);

system("Pause");

return 0;

}

Stack pic:



**Code Output pic:**

