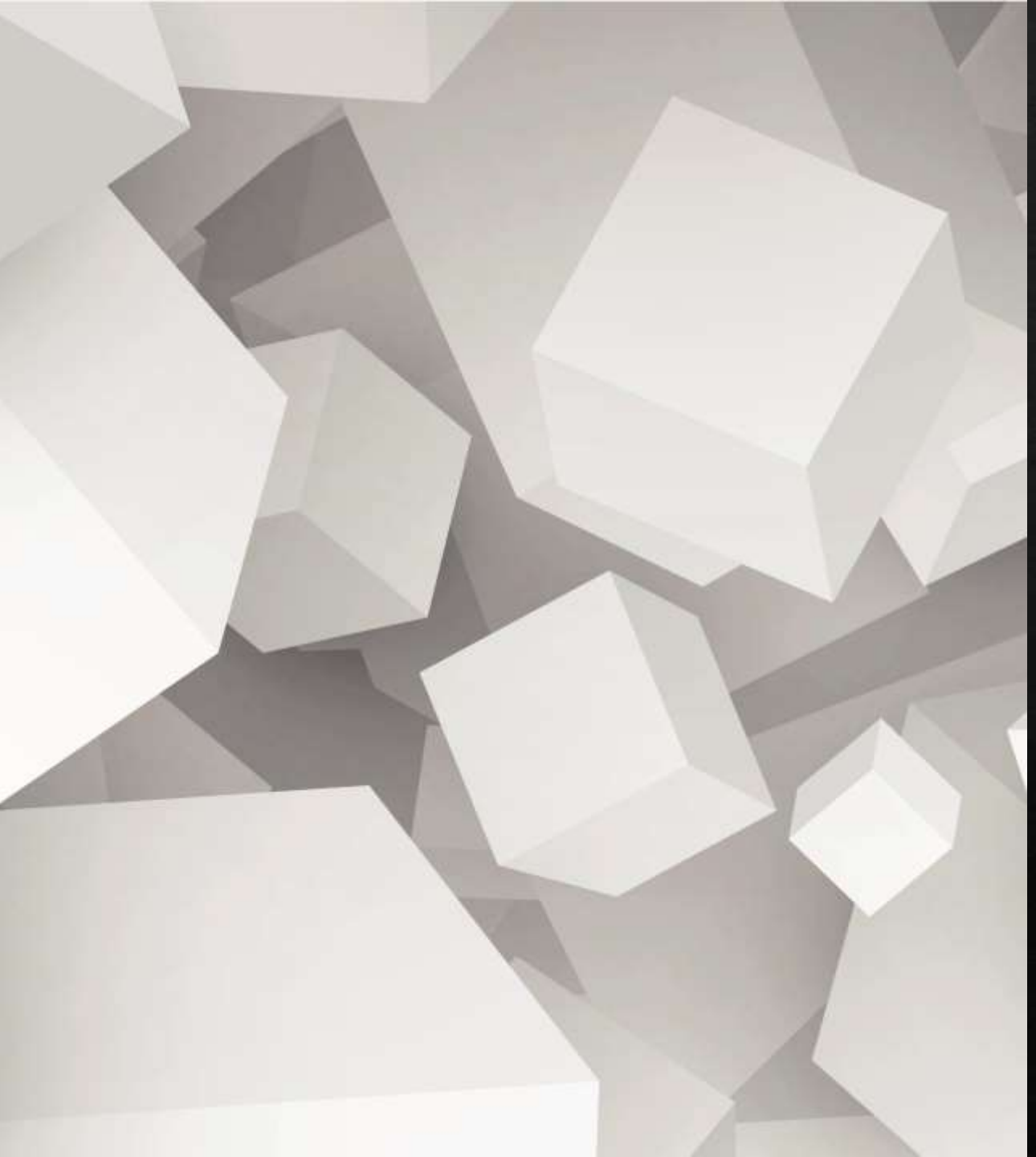
GARDEN CITY UNIVERSITY
EMPHASIS ON LIFE

TIC TAC TOE GAME IMPLEMENTATION IN C++

23BTRE118	Indu Chowdary M
23BTRE121	Karol Jerome
23BTRE120	Jeevan
23BTRE	Ughandhar Reddy
23BTRE	A



CONTENTS

INTRODUCTION

- Tic Tac Toe is a classic game that most of us have played at some point. The objective of this presentation is to walk you through the process of developing this game in C++ while highlighting the essential functions and future enhancements.
- Objective: Develop a simple Tic Tac Toe game using C++.
- Two-player console-based game.

GAME MECHANICS

- Players take turns to place their marker (X or O).
- The first player to align three markers horizontally, vertically, or diagonally wins.
- If all nine slots are filled without a winner, the game is a tie.

```
#include <iostream>
using namespace std;

char board[3][3] = {{'1','2','3'}, {'4','5','6'}, {'7','8','9'}};
char current_marker;
int current_player;

void drawBoard() {
    cout << " " << board[0][0] << " | " << board[0][1] << " | " << board[0][2] << endl;
    cout << "~~|~~|~~" << endl;
    cout << " " << board[1][0] << " | " << board[1][1] << " | " << board[1][2] << endl;
    cout << "~~|~~|~~" << endl;
    cout << " " << board[2][0] << " | " << board[2][1] << " | " << board[2][2] << endl;
}

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] = current_marker;
        return true;
    } else {
        return false;
    }
}

int winner() {
    for(int i = 0; i < 3; i++) {
        if(board[i][0] == board[i][1] && board[i][1] == board[i][2]) return current_player;
        if(board[0][i] == board[1][i] && board[1][i] == board[2][i]) return current_player;
    }
    if(board[0][0] == board[1][1] && board[1][1] == board[2][2]) return current_player;
    if(board[0][2] == board[1][1] && board[1][1] == board[2][0]) return current_player;
    return 0;
}

void swapPlayerAndMarker() {
    if(current_marker == 'X') current_marker = 'O';
    else current_marker = 'X';

    if(current_player == 1) current_player = 2;
    else current_player = 1;
}

void game() {
    cout << "Player 1, choose your marker: ";
    char marker_p1;
    cin >> marker_p1;

    current_player = 1;
    current_marker = marker_p1;

    drawBoard();

    int player_won;

    for(int i = 0; i < 9; i++) {
        cout << "It's player " << current_player << "'s turn. Enter your slot: ";
        int slot;
        cin >> slot;

        if(slot < 1 || slot > 9) {
            cout << "Invalid slot! Try again." << endl;
            i--;
            continue;
        }

        if(!placeMarker(slot)) {
            cout << "Slot already occupied! Try again." << endl;
            i--;
            continue;
        }
```

```
        drawBoard();

        player_won = winner();

        if(player_won == 1) {
            cout << "Player 1 won! Congratulations!" << endl;
            break;
        }
        if(player_won == 2) {
            cout << "Player 2 won! Congratulations!" << endl;
            break;
        }

        swapPlayerAndMarker();
    }

    if(player_won == 0) {
        cout << "It's a tie!" << endl;
    }
}
```

Functions Overview

- **Function Name:** drawboard
- **Description:** Prints the current state of the game board.

```
void drawBoard() {
    cout << " " << board[0][0] << " | " << board[0][1] << " | " << board[0][2] << endl;
    cout << "---|---|---" << endl;
    cout << " " << board[1][0] << " | " << board[1][1] << " | " << board[1][2] << endl;
    cout << "---|---|---" << endl;
    cout << " " << board[2][0] << " | " << board[2][1] << " | " << board[2][2] << endl;
}
```

- **Function Name:** placeMarker

- **Description:** Places the current player's marker at the chosen slot if it’s not already occupied.

```
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] = current_marker;
        return true;
    } else {
        return false;
    }
}
```

- **Function Name:** winner

- **Description:** Checks if the current player has won the game.

```
int winner() {
    for(int i = 0; i < 3; i++) {
        if(board[i][0] == board[i][1] && board[i][1] == board[i][2]) return current_player;
        if(board[0][i] == board[1][i] && board[1][i] == board[2][i]) return current_player;
    }
    if(board[0][0] == board[1][1] && board[1][1] == board[2][2]) return current_player;
    if(board[0][2] == board[1][1] && board[1][1] == board[2][0]) return current_player;
    return 0;
}
```

- **Function Name:** swapPlayerAndMarker
- **Description:** Switches the current player and marker for the next turn.

```
void swapPlayerAndMarker() {
    if(current_marker == 'X') current_marker = 'O';
    else current_marker = 'X';

    if(current_player == 1) current_player = 2;
    else current_player = 1;
}
```

- **Function Name:** game
- **Description:** Manages the game loop, handles player inputs,and checks for winners or ties.

```
void game() {
    cout << "Player 1, choose your marker: ";
    char marker_p1;
    cin >> marker_p1;

    current_player = 1;
    current_marker = marker_p1;

    drawBoard();

    int player_won;

    for(int i = 0; i < 9; i++) {
        cout << "It's player " << current_player << "'s turn. Enter your slot: ";
        int slot;
        cin >> slot;

        if(slot < 1 || slot > 9) {
            cout << "Invalid slot! Try again." << endl;
            i--;
            continue;
        }

        if(!placeMarker(slot)) {
            cout << "Slot already occupied! Try again." << endl;
            i--;
            continue;
        }
        drawBoard();

        player_won = winner();

        if(player_won == 1) {
            cout << "Player 1 won! Congratulations!" << endl;
            break;
        }
        if(player_won == 2) {
            cout << "Player 2 won! Congratulations!" << endl;
            break;
        }

        swapPlayerAndMarker();
    }

    if(player_won == 0) {
        cout << "It's a tie!" << endl;
    }
}
```

Future Enhancements

- Add AI for single-player mode.
- Implement a graphical user interface (GUI).
- Allow players to choose the board size (e.g., 4x4, 5x5).
- Implement a scoring system to track multiple games.

Conclusion

- The Tic Tac Toe game in C++ demonstrates basic game development concepts.
- Key functions: drawBoard, placeMarker, winner, swapPlayerAndMarker, game.
- Future enhancements can expand functionality and improve user experience.
- Thank you!

THANK YOU