Option A: 2D Tic Tac Toe Game

Project Overview:

The purpose of this project is to develop a simple, console-based 2D Tic-Tac-Toe game for two players. This game will allow users to input moves using row and column numbers and will display the board after each move. The project focuses on clear structure, basic game logic, and user interaction through the console.

Objectives are to:

1. Provide a functional two player game
2. Allow two players to take turns placing X and O through console input
3. Display the board with row and column labels
4. Prevent invalid or duplicate moves
5. Simulate two turns (four in total moves) as required

Scope of the project:

The game will use 4x4 array internally, where the playable area is 3x3 grid. The program will not include win detection, score tracking, or replay functionality. The project will focus only on simulating turns and displaying the board currently.

Deliverables:

1. Java source code file (.java)
2. Pseudocode documentation
3. Project plan documentation

Timeline:

1. Planning and design
2. Pseudocode development
3. Coding and testing
4. Final review and submission

Milestones:

1. Plan the game structure and board layout
2. Write pseudocode for the game logic
3. Implement the Java program
4. Test move validation and board display
5. Final review and submission

PseudoCode:

```
//Instance Variables

DECLARE board: ARRAY [0:3, 0:3] OF CHAR

DECLARE whoseturn: INTEGER

DECLARE players: ARRAY [0: 1] OF STRING

DECLARE pieces: ARRAY [0: 1] OF CHAR ← [‘X’, ‘O’]

//Constructor

CONSTRUCTOR TicTacToe (player1: String, player2: String)

    board ← NEW ARRAY [0:3, 0:3] OF CHAR

    players ← NEW ARRAY [0:1] OF STRING

    players[0] ← player1

    players[1] ← player2

    whoseturn ← 0

    FOR i ← 0 TO 3

        FOR j ← 0 to 3

            board[i][j] ← ‘_’

        NEXT j
```

```
    NEXT i
ENDCONSTRUCTOR
//Check if move is within bounds
FUNCTION inbounds (row: INTEGER, column: INTEGER) RETURNS
BOOLEAN
    IF (row >= 1 AND row <= 3 AND column >= 1 AND column <=
3) THEN
        RETURN TRUE
    ELSE
        RETURN FALSE
    ENDIF
ENDFUNCTION
//Place piece of board
FUNCTION Move (row: INTEGER, column: INTEGER) RETURNS
BOOLEAN
    IF NOT inbounds (row, column) THEN
        OUTPUT "ERROR: THE MOVE IS OUT OF BOUNDS!"
        RETURN FALSE
```

```
    ELSE IF board[row][column] <> '_' THEN
        OUTPUT "ERROR: POSITION IS ALREADY OCCUPIED!"
        RETURN FALSE
    ENDIF
    board[row][column] ← pieces[whoseturn]
    whoseturn ← (whoseturn + 1) MOD 2
    RETURN TRUE
ENDFUNCTION
// Display the game board
PROCEDURE printBoard()
    FOR p ← 1 TO 3
        OUTPUT p
    NEXT p
    OUTPUT NEWLINE
    FOR b ← 1 TO 3
        OUTPUT b
        FOR m ← 1 TO 3
```

```
            OUTPUT board[b][m]
        NEXT m
        OUTPUT NEWLINE
    NEXT b
ENDPROCEDURE
// Main simulation
OUTPUT "—WELCOME TO TIC-TAC-TOE GAME—"
OUTPUT "Enter Player 1's name for (X): "
INPUT p1
OUTPUT "Enter Player 2's name for (O): "
INPUT p2
game ← NEW TicTacToe(p1, p2)
CALL game.printBoard()
FOR i ← 0 TO 3
    OUTPUT game.players[game.whoseturn], ", who plays for
(", game.pieces[game.whoseturn], ")"
    OUTPUT "Please enter row (1-3): "
    INPUT row
```

```
    OUTPUT "Please enter column (1-3): "
    INPUT column
    IF game.Move(row, column) THEN
        CALL game.printBoard()
    ELSE
        CALL game.printBoard()
        i ← i -1
    ENDIF
NEXT i
```