Zohaib Saqib

Project Report

a) a description of the program.

The program starts by defining some data, including the grid for the game and various strings to be printed during the game. The main function then starts by calling the Board function to display the initial state of the grid. The program then prompts the player whose turn it is to enter a number corresponding to a cell on the grid. The player's choice is then checked to make sure it's a valid move, and if it is, the grid is updated accordingly. After each turn, the returnWin function is called to determine if either player has won or if the game is a draw. If the game is over, the appropriate message is displayed and the program exits. Otherwise, the loop continues with the next player's turn.

b) At first, we had issues dividing the work, but after discussing every procedure that might be required in a simple game of tic tac toe, we efficiently split the work among the group members, and if one member struggled, another would try to help. Then at the end we combined all the code and searched for errors and followed whoevers combined code ran with no errors.

c) This project has taught me how to divide work in a team and use resources to help finish a task you have been given. Even besides mips, this project has helped in communication and organization which would be beneficial in the professional world.

d) a discussion of algorithms and techniques used in the program, e.g. how does the program work?

1. Board Representation :

The game board is represented using a “grid” that holds the value of the game pieces “X”, and “O” as characters. This representation makes it easy to access and manipulate the game board using simple array indexing.

2. User Input :

The program takes user input through the console using the “syscall” instruction. The input is validated against the game board and any invalid input is rejected with an error message.

3. Game Logic :

The program checks for a winner by checking all possible winning combinations of the game board. The game board is checked after every move to see if a player has won, or if the game has ended in a draw.

4. Subroutine Calls :

The program uses subroutine calls to modularize the code and simplify the implementation. The “Board” subroutine is called to print the game board, while the “returnWin” subroutine is called to check the status of the game.

5. Branching Instructions :

The program uses branching instructions like “bne”, and “beq” to control the flow of the program based on certain conditions. This is used to implement the turn-based game logic and to check for invalid input.

Zohaib Saqib

e) Zohaib Saqib (me): Wrote code for invalid move checks and win conditions
Risheeka Mitra: Character select input, output of who won, and user manual
Alex Beattie: User and computer input and fixed linking errors in win conditions
Maanasa Aragula : board function, x/o loading, grid for user input and user moves, shared responses on doc