CSE1502 – Intro to Software Dev. with C++

Final Project Report
Spring 2025

By

Russell Ramroop

Mechanical Engineering

rramroop2023@my.fit.edu

April 30, 2025

.

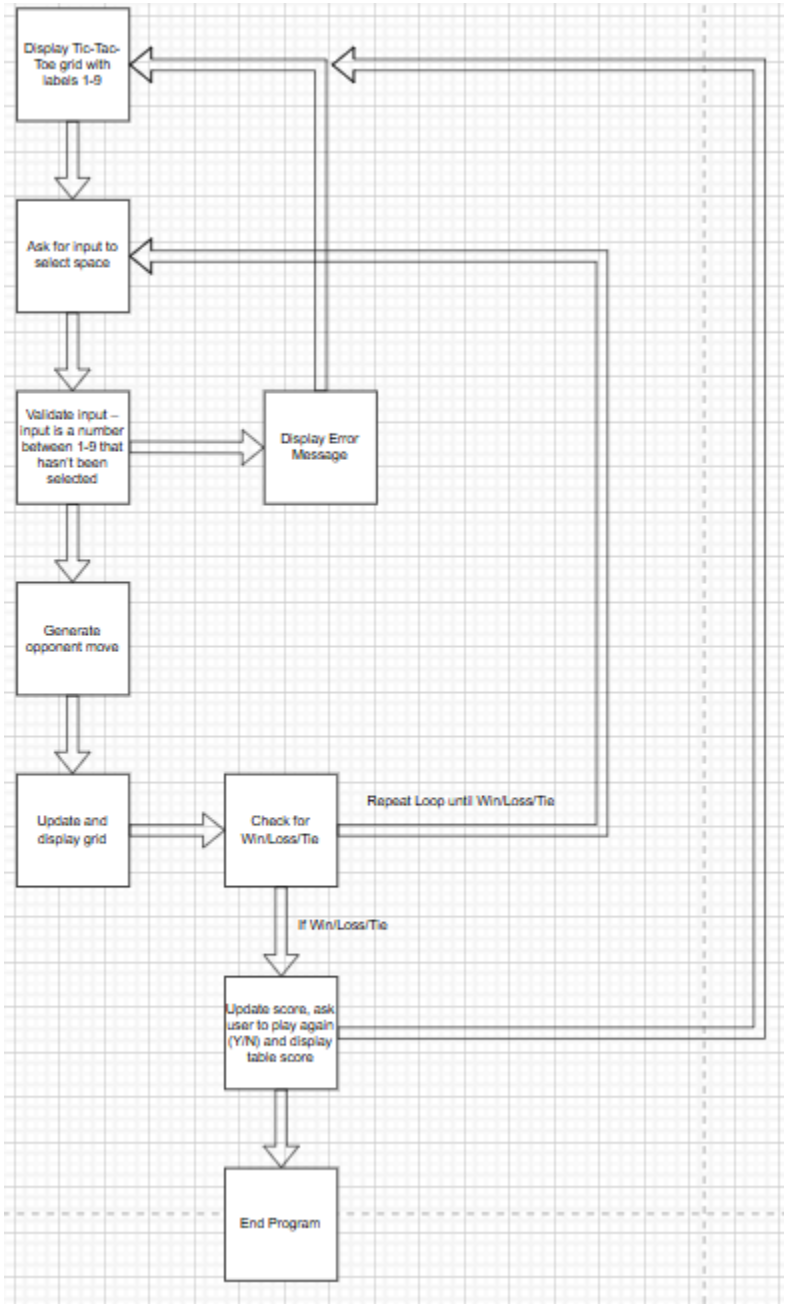
1. Background & Problem Statement

A tic-tac-toe game with the user and opponent. The grid is set up with the nine boxes labeled 1-9, and the user inputs a number corresponding to that box each time. The opponent will be a random number between 1-9 and constantly update with each selected box. Once a winner or tie has happened, the user can play again or end the program with a console output displaying all wins, losses, and ties.

2. Requirements

1. *Display Tic-Tac-Toe grid with boxes labeled 1-9*
2. *Ask for user input to select a grid box*
3. *Validate user input is unoccupied and within 1-9*
4. *Randomly generate a number that is not already selected*
5. *Update the grid display with each user and opponent move*
6. *Check for the winning condition after each move*
7. *Check for tie condition if all boxes are filled*
8. *Display the result of the game*
9. *Ask the user to play again or end the program*
10. *Track and update score (wins/losses/ties)*
11. *Display the score when the user ends the program*

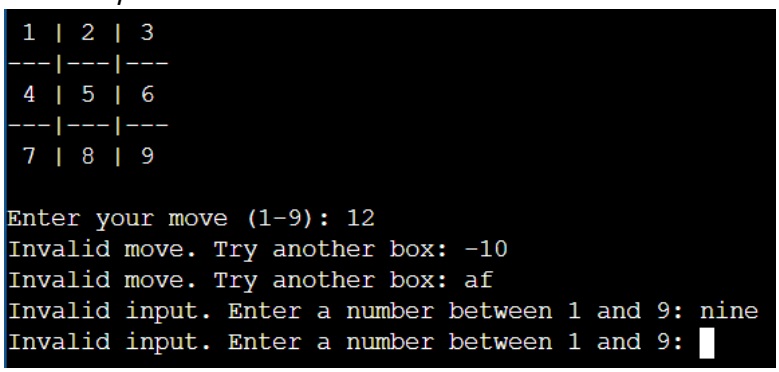
3. Software Construction (Diagrams & C++ Code)



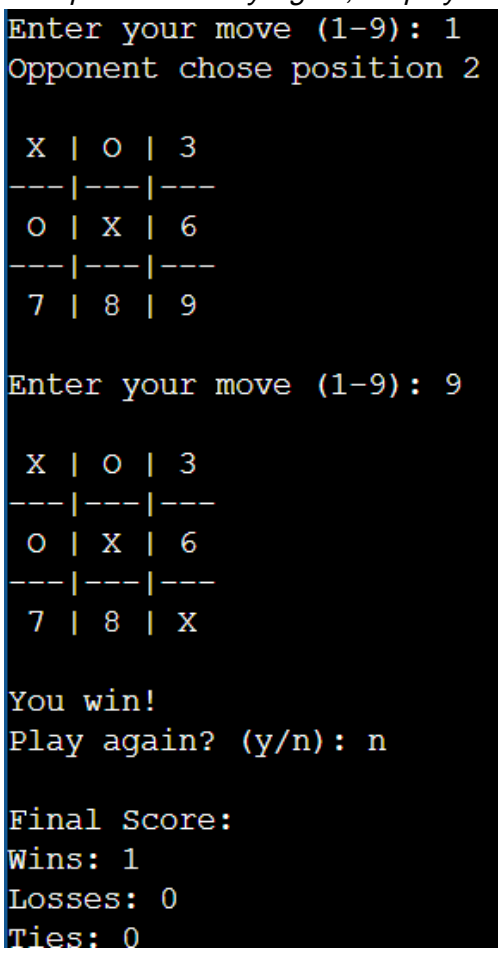
C++ Code is found in TicTacToe.cpp

4. Software Testing

Invalid Inputs:



Generate Opponent Move, Display Updated Grid, Check for Win/Loss/Tie, Update Score, Prompt User to Play Again, Display Table Score if the user ends program:



5. Conclusions & Future Directions

This project was slightly more challenging than expected because of how the inputs should be validated and checked. Instead of labeling the values as integers, they had to be defined as characters because it's easier to validate incorrect user input. Some future expansions for this project would be giving the opponent an algorithm to try to win instead of using a random number generator to generate its moves. This was a similar project done in a previous semester with significant changes in how the program operates.

6. References