

**Department of Computer Science**

**CSCL2201 – Computer Organization and Assembly Language**

**Tic Tac Toe**

**Project Report**

**Semester: Fall 2024**

**Group members:**

1. **Laiba Jabbar (2312394)**
2. **Hira Niaz malik (2312113)**
3. **Alisha Anwar (2312105)**
4. **Areeba (2312393)**
5. **Aiman (2312129)**

***Submitted to:***

**Sir Saqib Sadiq**

***Lab Engineer***

**Sir Hurrair**

**Table of Content:**

* Objective
* Problem Description
* Concepts Implemented
* Project Features
* Solution and Application Area
* Milestone
* Feasibility Study
* Conclusion
* References

**Objective:** The objective of this project is to implement a simple text-based Tic-Tac-Toe game using Assembly language (8086).

**Problem Description:** The Tic-Tac-Toe game is a classic two-player game where each player takes turns marking a cell on a 3x3 grid. The player who succeeds in placing three of their marks (either "X" or "O") in a horizontal, vertical, or diagonal row wins the game. The implementation is in Assembly language (8086), providing a low-level view of game development.

**Concepts Implemented:**

**Assembly (8086):** The entire game is written in Assembly language, specifically targeting the 8086 architectures.

**Text-Based Interface:** The game features a text-based interface, utilizing the DOS interrupt calls for input/output and cursor control.

**Game Logic:** The program includes the logic for checking win conditions, handling player turns, and managing the game board.

**User Input Handling:** The code captures user input and validates it, ensuring it corresponds to a valid move in the game

.

**Graphics (ASCII Art):** ASCII art is used to create a simple Tic-Tac-Toe logo on the title screen.

**Project Features:**

**Game Rules:** A section explains the rules of the game to the players before starting.

**Board Display:** The program dynamically updates and displays the Tic-Tac-Toe board with each move.

**Player Interaction:** Players take turns entering their moves by selecting a cell number.

**Winning Condition**: The program checks for the winning condition after each move and declares the winner when applicable.

**Draw Condition:** If no player wins and the board is filled, the game declares a draw.

**Play Again Option**: After each game, players are prompted to play again, providing flexibility for multiple rounds.

**Project Features:**

**Console Interface:** The game utilizes the console for interaction, providing a simple and user-friendly interface.

**Player vs. Player:** Two players can take turns to play the game, selecting positions on the 3x3 Tic Tac Toe grid.

**Winning Conditions**: Detection of winning conditions, such as three symbols in a row, column, or diagonal

**Solution and Application Area of the Tic-Tac-Toe Game**

**Solution:**

* **Educational Tool**: Helps students learn Assembly language and low-level programming.
* **Programming Practice**: Offers hands-on experience with efficient coding and debugging.
* **Game Development Basics**: Demonstrates fundamental game logic and user interaction in a resource-constrained environment.
* **Problem-Solving Skills**: Enhances skills by working with constraints and optimizing code.

**Application Areas:**

* **Educational Institutions**: Used as a teaching aid and student project in computer science courses.
* **Retro Gaming Enthusiasts**: Acts as a base for developing more complex games and learning about early computing systems.
* **Embedded Systems**: Applies to programming microcontrollers and embedded devices with limited resources.
* **Software Development**: Aids in understanding code optimization for performance and memory usage.
* **Historical Computing**: Provides insights into legacy systems and preserves historical programming techniques.

**Milestones**

* **Project Initiation**
  + Define scope and objectives.
  + Form the team and assign roles.
* **Requirements Gathering**
  + Outline game rules and mechanics.
  + Identify target platform and tools.
* **Initial Design**
  + Basic design of game structure and user interface.
* **Development Phase 1**
  + Implement game board display.
  + Handle user input and validate moves.
* **Development Phase 2**
  + Develop win condition logic.
  + Manage player turns.
* **Testing and Debugging**
  + Test for bugs and logical errors.
* **Final Development**
  + Add features like ASCII art.
  + Implement play again option.
* **Documentation**
  + Prepare project report and documentation.
* **Review and Feedback**
  + Present project for feedback and revise.
* **Project Submission**
  + Submit final report and source code

**Feasibility Study for Tic-Tac-Toe**

**Technical Feasibility**

* Compatible with 8086 architectures.
* Requires knowledge of Assembly language.
* Tools and resources available for development.

**Operational Feasibility**

* Simple text-based interface.
* Straightforward game logic and rules.

**Economic Feasibility**

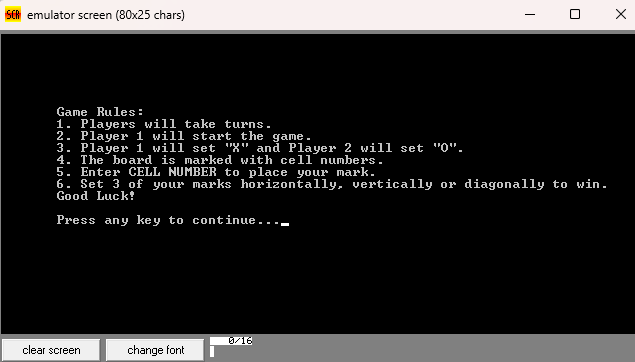
* Low development and maintenance costs.
* Minimal additional resources required.

**Schedule Feasibility**

* Achievable within the semester.
* Clear milestones and deadlines.

References

Internet and AI

**Rules**

**Game Result**

****

**Conclusion:**

In conclusion, this Assembly (8086) implementation of Tic-Tac-Toe demonstrates the utilization of low-level programming to create a text-based interactive game. The project successfully integrates game logic, user input handling, and ASCII art to provide a classic gaming experience. Enhancements and optimizations can be explored for future iterations, but the current implementation serves as a foundation for understanding Assembly language programming and game development.