Nao Kaitou

Project title:

Tic-tac-toe

course:

Computer organization and assembly language

**Date**: december 21,2020

Prepared for

**Sir Ahsan azhar**

prepared by

**ayyan BABAR (19L-0903)**

**nafeesa arif (19L-1212)**

**tayyaba shahid (19L-2205)**

TABLE OF CONTENTS

1.abstract 3

project Statement 3

summary 3

2.description 4

Introduction 4

objectives 4

functionalities 4

components 5

Screen Components 5

code Components 5

3.implementation 5

Programming methodology 5

special features 6

Sound 6

optimization 6

Readability 6

Manageability 6

Reusability 6

difficulties encountered 6

4.experimental evaluation 7

features and techniques used 7

Table 01 7

5.summary 7

conclusion 7

outcomes 7

Improvements 7

**abstract**

**Project Statement**

One of the very common games is tic tac toe, but how would we implement it to a very basic level of computer programming? The project gives a deep insight to how this simple old game is implemented using various features and techniques of Assembly Language x86.

The report is easily comprehensible and can be understood by layman. Moreover, the project report also aims at highlighting the difficulties that may be encountered and how they were solved on experimentation basis. The report also validates the project’s successful implementation.

**Summary of project**

* **Description:** 
  + **Introduction.**  The game is 2 player 3x3 game, with each player having a distinct mark of X or O. The player who gets 3 in a straight line in any direction wins.
  + **Objectives.** To successfully implement Graphics criteria, game using Assembly language according to the rules.
  + **Functionalities.** A graphical user interface, ease of control and different colors are added for the user.
  + **Components.** There are two types of components: screen and code components with their own unique usage
* **Implementation:**
  + **Programming Methodology.** Logical Programing Bottom-Up approach is used which makes use of smaller blocks to implement a bigger purpose
  + **Special Features.** The special feature of Sound effects is used to enhance the game experience
  + **Optimization.**  The optimization is done based upon the code components by Readability, Maintainability and Reusability.
  + **Difficulties Encountered.**  The difficulties of screen mark allocation, usage of mouse and change of colors of winning combination were encountered.
* **Experimental Evaluation.** 
  + **Features and Techniques used.** The project was evaluated using different techniques and features given in Table 01.
* **Summary:** 
  + **Conclusion.** To summarize, the game was successfully implemented using various features and techniques.
  + **Outcome.** The game project was illustrating the positive effects of usage of Assembly Language
  + **Improvements.** The projectcan be improved if multiplayer feature is added, however, it passes all the mentioned criteria for a project.

**Description**

**Introduction**

The game is a 3X3 2 player game with each player having the symbol X and O. Each player has one turn and cannot place a symbol in already occupied box. The game ends in two conditions:

1. there are not enough spaces i.e. the game draws
2. either of the player wins: one of the players creates the winning combination i.e. a line that cuts the mark horizontally, vertically or diagonally.

**Objectives**

The objectives of this project are as follows:

* + To successfully implement the graphic layout criteria
  + To achieve the goal of creation of the game by using special features of assembly language and maximum optimization
  + To successfully create a game that can be played according to the rules

**Functionalities**

1. The game provides a graphic user interface to allow user to interact with the game effectively
2. The layout allows different colors to differentiate among the symbols, play area, background and messages
3. The game provides the easy use of mouse to play in the Play area.

*Shown is the figure to illustrate the screen components. Fig. 01*

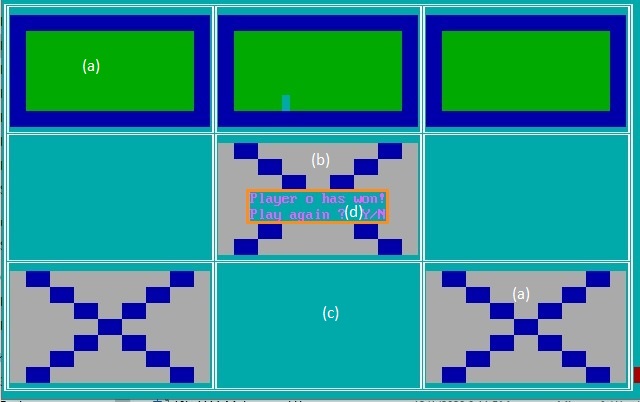
**Components**

There are two categories of components in the report:

* Screen components

Screen components include the following:

1. Symbols (O, X)
2. Foreground (play area)
3. Background
4. Message Prompts (turns, winner/draw, play again)



* Code components

The code component includes the Assembly language:

1. Subroutines (specific instructions/

functions to perform a specific task)

1. Variables (used to store and execute instructions)
2. Stack (for effective use of storage registers)
3. Video Memory (Graphics)
4. Special Instructions (used to operate the instructions)
5. Mouse Control (To allow user to interact using mouse)

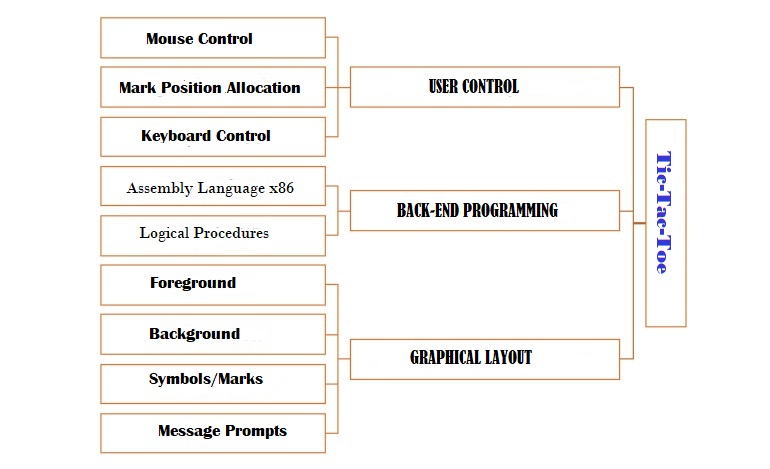
**implementation**

Implementation of the project aims at the programming methodology, special features, optimization and difficulties encountered in making the project.

Furthermore, the implementation briefly hints at the logical programming methodology, usage of sound effects; readability, manageability and reusability of the coding segment; along with the difficulties encountered while implementing the design.

**Programming methodology**

The programming methodology is based upon the code components of the project as described earlier. Thus, the **Logical Programming** is used, in which a problem is broken down into smaller units to fulfill a bigger task.

**

*Fig.02 Bottom-Up Approach Figure*

**Special features**

The game uses a unique feature of ***sound*** usage.

A unique sound is played each time there’s a click on the screen. Also, a distinct sound is generated when the game begins.

**optimization**

The game was optimized based upon its code components i.e. the code was optimized based on:

* **Readability:** The code was easy to be read.
* **Manageability**: The code is easy to be maintained i.e. if a small change is to be occurred, it can be done easily.
* **Reusability**: Several segments of code for a specific purpose (subroutines) were used to shorten the longevity of the code. Also, the special purpose subroutines were congregated into few subroutines which take inputs.

**Difficulties encountered**

In fulfilling the task, following such difficulties were encountered:

1. How would we print X and O in the desired position when the user clicks?
2. How the winning three would make a line of distinct color than the foreground?
3. How the board is to be drawn on the whole of the screen?

**experimental evaluation**

The project was evaluated using different machines and different combination of programming schemes, thus the problems encountered were solved accordingly:

**Features and techniques used**

|  |  |  |
| --- | --- | --- |
| **Features** | **Techniques** | **Feasibility** |
| Opcodes | Using arrays | Very much |
| Subroutines | By taking parameters | Highly |
| Mouse Control | Interrupts and Services | Quite |
| Jumps | Stack | Highly |

*Table 1: Features and Techniques of Assembly Language Used*

**summary**

To summarize, we provide with the conclusions, outcomes and further improvements that may be done in the project.

**conclusion**

The project, a simple game of Tic-Tac can be easily implemented in Assembly Language x86 using logical techniques and features.

**Outcomes**

The outcomes were based on various aspects. Following are the outcomes:

1. Assembly Language is suitable for interrupt service routines and memory residing programs such as the game mentioned, using less memory and execution time
2. The project can be seen as an exquisite example of how Assembly Language can be viewed as a potential tool to implement the very basic of generally played games.

**improvements**

Overall, the project satisfies the criteria of a well-made game, however, it can be improved if it allows a multiplayer option.