**TIC TAC TOE**

### A Project Work

*Submitted in the partial fulfillment for the award of the degree of*

# BACHELOR OF ENGINEERING

### IN

### BIG DATA ANALYTICS

### Submitted by:

### KUNAL SHARMA (20BCS3999)

### Under the Supervision of:

### Ms. Monika



# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING APEX INSTITUE OF TECHNOLOGY

### CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

**PUNJAB**

#### JULY 2021

**DECLARATION**

I, **Kunal Sharma**, student of **‘Bachelor of Engineering in Computer Science’**, **Batch 2024**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**Tic Tac Toe’** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

|  |  |  |
| --- | --- | --- |
| Name ( team members) | UID | Signature |
| Kunal Sharma | 20BCS3999 | Kunal |

#### Date: 27 July 2021

**Place: Chandigarh University**

# Table of Contents

**page**

|  |  |  |
| --- | --- | --- |
|  | Abstract Acknowledgement  List of Figures  List of Tables | 4 |
| 5 |
| 6 |
| 7 |
|  |
|  |
|  |
| **1.** | **INTRODUCTION\*** | 8 |
|  | * 1. Problem Definition   2. Project Overview/Specifications\*   3. Hardware   1.3.1 Laptop or pc  1.4 Software Specification  1.4.1 Operating System: Windows  1.4.2 Toolkit: Dev C++  1.4.3 Platform: C++  1.4.4 Database: notepad | 8  8  9  9  9  9-10  10  10  11 |
| 1. **LITERATURE SURVEY**    1. Existing System    2. Proposed System    3. Feasibility Study 2. **PROBLEM FORMULATION**    1. project planning    2. Use case diagram    3. Sequence diagram    4. Package diagram    5. Class diagram | | 12 |
| 12  12  13 |
| 14  14-15  16-17  17-19  20-21    22  22  23  23  23-31    32  33 |
|  | |
| 1. **OBJECTIVES**   **4.1 Analysis**   1. **METHODOLOGY**   5.1 Project Design  5.2 Project snapshots   1. **RESULTS AND DISCUSSION** 2. **REFERENCES** | |

**Abstract**

Our Aim is to design and create a C++ project for the development of a Tic Tac Toe game . This game is developed to provide best entertaining services to the users and students. It is two player game, which enables the user to choose choices according to his will. This system also helps to promote responsible and interesting entertainment so that people can refresh themselves from their busy schedule. Tic Tac Toe is a very popular Game, we often play this game Classrooms and hostel while we are in student life. Tic Tac Toe Game is a very famous game in this game everyone wants to win so that each and every input of user we check winning condition if our condition satisfy then it will print the message according to Game that either player win or computer win or Game is draw. This game also asks the player using audio feature, whether he wants to choose ‘X’ or ‘O’ and declares the result using audio command.

**ACKNOWLEDGEMENT**

I take this opportunity to express my deep gratitude and most sincere thanks to my project mentor, Ms. Monika for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work. I would like to thank my mentor for guiding me. Last but not the least I'm grateful to my team member.

A special acknowledgement goes to my colleagues who helped me in completing the project by exchanging interesting ideas and sharing their experience.

Kunal Sharma

## List of Tables

##### Table Title page

***2.1*** *Table of comparison( between existing system and proposed system)* ***12***

.

## List of Figures

##### Figure Title page

1. *Use case diagram of Tic Tac Toe 15*
2. *Sequence diagram of Tic Tac Toe 17*
3. *Package diagram of Tic Tac Toe 19*
4. *Class diagram of Tic Tac Toe 21*
5. *Invalid username snapshot 24*
6. *Invalid username snapshot 24*
7. *Lose game scenario snapshots. 25-26*
8. *Win game scenario snapshots. 27-28*
9. *Tied game scenario snapshots. 29-31*

# INTRODUCTION

* 1. **Problem Definition**

Tic Tac Toe Game is a very famous game in this game everyone wants to win so that each and every input of user we check winning condition if our condition satisfy then it will print the message according to Game that either player wins or computer wins or the Game is draw. Tic-tac-toe is a very. mindful game. You have to be aware of every move of your opponent and even figure out which move you will make next. To beat your opponent, be it a person or a computer even, you need to plan out a strategy. The player who succeeds in placing three respective marks in horizontal, vertical, or diagonal row wins the game. Here, player will be notified about his turn using audio.

**1.2 Overview**

The main objective of this game project is to facilitate a user-friendly environment and reduce the manual effort. The purpose of this research is to provide a virtual image for the combination of both structured and unstructured information of my project “Tic Tac Toe”. This is a single-player strategy game on the Windows platform, where the first player is human and the second player is computer. The game is developed for full-time entertainment and enthusiasm. It is a simple game that flexes the basic concepts in programming. For example, a Tic Tac Toe program requires data structures for storing the board and conditional logic for knowing whose turn it is or if someone has won. Also it teaches the gamer to be alert at every situation he/she faces, because if the gamer is not fully alert and notice the saucer fire he/she must be hit by the saucer-bombs.

Tic Tac Toe's simple rules are very easy to decompose into small, discrete functions that we can put together to create a complete game. This process of decomposition and synthesis, breaking something down and putting it back together, is absolutely essential to programming. Here computer will use audio command to tell the player to start the game, chose his choice as well to declare the result of the game.

**1.3** **Hardware Requirements:**

**1.3.1 Laptop or PC**

A laptop, laptop computer, or notebook computer is a small, portable personal computer (PC) with a screen and alphanumeric keyboard. These typically have a "clamshell" form factor, typically having the screen mounted on the inside of the upper lid and the keyboard on the inside of the lower lid, although 2 in 1 Pcs with a detachable keyboard are often marketed as laptops or as having a "laptop mode." Laptops are folded shut for transportation, and thus are suitable for mobile use. Its name comes from lap, as it was deemed practical to be placed on a person's lap when being used. Today, laptops are the used in a variety of settings, such as at work, in education, for playing games, web browsing, for personal multimedia, and general home computer use.

**1.4 Software Requirements**:

**1.4.1 Operating System: Windows**

1.4.1.1 Windows 7

Windows 7 is the Microsoft Windows operating system (OS) released commercially in October 2009 as the successor to Windows Vista. Windows 7 is built on the Windows Vista kernel and was intended to be an update to the Vista OS. It uses the same Aero user interface (UI) that debuted in Windows Vista. As a result, to many end users, the biggest changes between Vista and Windows 7 were faster boot times, new UIs and the addition of Internet Explorer ([IE](https://searchenterprisedesktop.techtarget.com/definition/Internet-Explorer)) 8. The OS is widely available in three retail editions: Windows 7 Home Premium, Professional and Ultimate. Starter, Home Basic and Enterprise editions are available in some markets. In development, Windows 7 was known by the code names Blackcomb and Vienna.

1.4.1.2 Windows 8

The successor of Windows 7 will support beside of the 32 bit and 64-bit Intel and AMD SoC (system-on-a-chip) architectures also ARM for a widest possible range of form factors on the market. NVIDIA, Qualcomm and Texas Instruments are working on different SoC designs based on the ARM architecture:  
 - Snapdragon ARM system from Qualcomm  
 - OMAP ARM system from Texas Instruments

- Tegra ARM system from NVIDIA with main focus to high-performance graphic  
Microsoft is porting basic-applications like the Internet Explorer and Office applications to the new platform. They are native ARM applications for Windows 8.

1.4.1.3 Windows 10

Windows 10 is a major version of the Microsoft [Windows](https://techterms.com/definition/windows) operating system that was released on July 29, 2015. It is built on the Windows NT [kernel](https://techterms.com/definition/kernel) and follows windows 8

Part of the reason Microsoft decided to name the 2015 release "Windows 10" (and skipped "Windows 9") is because the [operating system](https://techterms.com/definition/operating_system) is designed to be a new direction for Microsoft. One of the primary aims of Windows 10 is to unify the Windows experience across multiple devices, such [desktop computers](https://techterms.com/definition/desktop_computer), [tablets](https://techterms.com/definition/tablet), and [smartphones](https://techterms.com/definition/smartphone). As part of this effort, Microsoft developed Windows 10 Mobile alongside Windows 10 to replaces Windows Phone – Microsoft's previous mobile OS.

**1.4.2. Toolkit: Dev C++**

Dev-C++ is a fully featured graphical IDE (Integrated Development Environment) that uses the MinGw compiler system to create Windows as well as Console based C/C++ applications. It can also be used with any other GCC-based compiler like Cygwin. Dev-C++ is free software and is distributed under the GNU General Public License. Thus, we can distribute or modify the IDE freely. It was originally developed by “Bloodshed Software”. It has been forked by Orwell after it was abandoned by Bloodshed in 2006

**1.4.3. Platform: C++**

C++ is an object-oriented computer language created by notable computer scientist Bjorne Stroustrop as part of the evolution of the C family of languages. C++ is pronounced "see-plus-plus." It was developed as a cross-platform improvement of C to provide developers with a higher degree of control over memory and system resources.

Some call C++ “C with classes” because it introduces object-oriented programming principles, including the use of defined classes, to the C programming language framework. Over time, C++ has remained a very useful language not only in computer programming itself, but in teaching new programmers about how object-oriented programming works. However, it does not support only object-oriented, but also procedural and functional. Thanks to its high

flexibility and scalability, C++ can be used to develop a broad range of software, applications, browsers, Graphical User Interfaces (GUIs), operating systems, and games.[1]

**1.4.4. Database: notepad**

Notepad is a basic text editor that's built into Windows. It is excellent for writing relatively short text documents that you want to save as plain text. However, that's not all you can do with it. If you have not used Notepad much, or ever, you may be surprised by how easy it is to work with. Notepad looks and works mostly the same in Windows 10 and older versions of Microsoft's operating systems such as Windows 7.

**1.4.5**  **Toolkit- C sharp**

C# (pronounced "C-sharp") is an object-oriented programming language from Microsoft that aims to combine the computing power of C++ with the programming ease of Visual Basic. C# is based on C++ and contains features similar to those of Java. C# is designed to work with Microsoft's .NET platform. Microsoft's aim is to facilitate the exchange of information and services over the Web, and to enable developers to build highly portable applications.

C# supports strongly, implicitly typed variable declarations with the keyword var, and implicitly typed arrays with the keyword new[] followed by a collection initializer. C# supports a strict Boolean data type, bool. Statements that take conditions, such as while and if, require an expression of a type that implements the true operator, such as the Boolean type. While C++ also has a Boolean type, it can be freely converted to and from integers, and expressions such as if (a) require only that a is convertible to bool, allowing a to be an int, or a pointer. C# disallows this "integer meaning true or false" approach, on the grounds that forcing programmers to use expressions that return exactly bool can prevent

**1.4.6. Visual studio**

Visual Studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++

Visual Studio (like any other IDE) includes a code editor that supports syntax highlighting and code completion using IntelliSense for variables, functions, methods, loops, and LINQ queries. IntelliSense is supported for the included languages, as well as for XML, Cascading Style Sheets, and JavaScript when developing web sites and web applications.

1. **LITERATURE REVIEW**

**2.1 Existing System**

The moves taken by the computer and the human are chosen randomly, rand() function is used for this. The program is in not played optimally by both sides because the moves are chosen randomly. The program can be easily modified so that both players play optimally (which will fall under the category of Artificial Intelligence). Also the program can be modified such that the user himself gives the input (using scanf() or cin).

Winning Strategy- If both the players play optimally then it is destined that you will never lose (“although the match can still be drawn”). It doesn’t matter whether you play first or second.

**2.2 Proposed System**

In our program the user's moves are taken as input and moves taken by the system are dependent on user's moves. The game is played optimally by both sides because the moves are not chosen randomly. A grid of 3x3 will be displayed in which each cell will be alloted with a cell number, and the user will make his move by giving the cell number as input. The system will make its move with the help of some conditions, designed in such a way that the system wins most of the time. Also the user will be given a choice of whether he wants 'X' or 'O' for his move.

**2.3 Feasibility study**

A feasibility analysis usually involves a thorough assessment of the operational (need), financial. and technical aspects of a proposal. Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and. whether it can be developed with the given budgetary constraints. A feasibility study should be. relatively cheap and done at the earliest possible time. Depending on the study, the decision is made whether to go ahead with a more detailed analysis. When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration.

Facts considered in the feasibility analysis were

1. Technical Feasibility

Technical feasibility includes whether the technology is available in the market for development and its availability. The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc, in order to give an introduction of technical system. Considering our project it is technical feasible

1. Economic Feasibility

This feasibility study present tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service.

1. Behavioral Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept broad audience from around the world

Table.1

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Existing System** | **Proposed** **System** |
| Method | Manual | Automatic |
| Time | more | less |
| Accuracy | accurate | more accurate |
| complexity | Less complex | Less complex |
| Player | Two player | Single player |

# PROBLEM FORMULATION

**3.1 Project planning and scheduling is a part of project management.**

The project planning stage requires several inputs, including conceptual proposals, project schedules. The development of this project is not successfully done without proper planning and scheduling. Project planning and scheduling is very important stage for us.

(a) **Analysis:**

The maximum time for analysis phase of this project is 3 days.

(b) **Design:**

The maximum time for design phase of this project is 5 days.

(c**) Implementation:**

The maximum time for implementation phase of this project is 7 days.

(d) **Testing:**

The maximum time for testing phase of this project is 2 days.

**3.2 Use Case Diagram**

A use case diagramat its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

3.2.1 Actor

An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances.

3.2.2 Association

An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type.

3.2.3 System

If a subject (or system boundary) is displayed, the use case ellipse is visually located inside the system boundary rectangle. Note that this does not necessarily mean that the subject classifier owns the contained use cases, but merely that the use case applies to that classified.

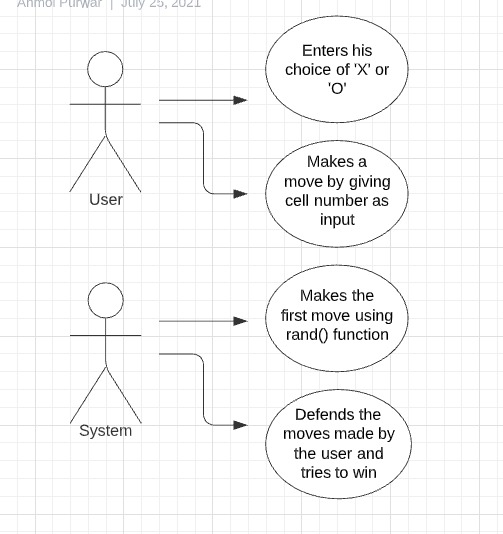


Figure 1.Use Case Diagram

**3.3 Sequence Diagram**

The Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case. With the advanced visual modeling capability, you can create complex sequence diagram in few clicks. Besides, Visual Paradigm can generate sequence diagram from the flow of events which you have defined in the use case description.

3.3.1 Actor

An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances. Since an actor is external to the subject, it is typically defined in the same classifier or package that incorporates the subject classifier

3.3.2 Call Message

A message defines a particular communication between Lifelines of an Interaction. Call message is a kind of message that represents an invocation of operation of target life line . Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent

from an object that will not wait for a response from the receiver before continuing its

tasks.

3.3.3 Lifelines  
 Life line are vertical dashed lines that indicate the object's presence over time.

3.3.4 **Destroying Objects**  
 Object can be terminated early using an arrow labelled "<< destroy >>" that points to an X.

This object is removed from memory. When that object's lifeline ends, you can place an X

at the end of its lifeline to denote a destruction occurrence.

**3.3.5 Loops**  
 A repetition or loop within a sequence diagram is depicted as a rectangle. Place the

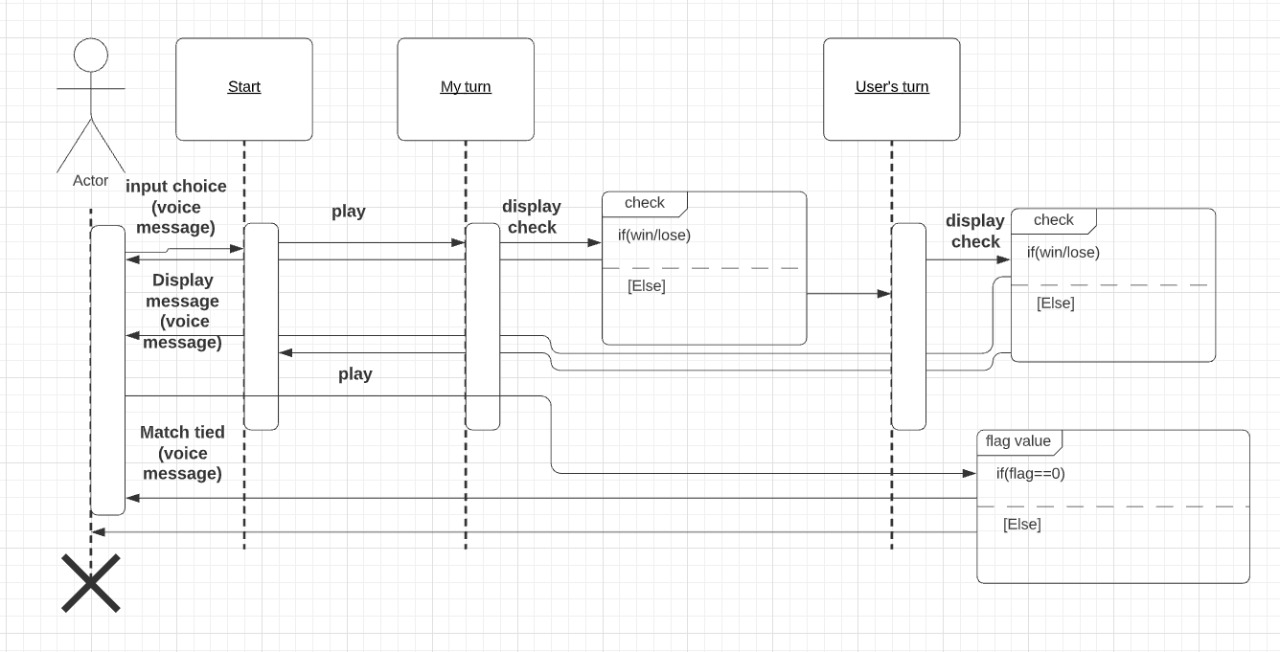
 condition for exiting the loop at the bottom left corner in square brackets [ ].

Figure 2.Sequence diagram

**3.4 Package Diagram**

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages. A package is a grouping of related UML Elements such as diagrams, documents, classes, or even other packages. Each element is nested within the package, which is depicted as a file folder within the diagram, then arranged hierarchically within the diagram. Package diagrams are most commonly used to provide a visual organization of the layered architecture within any UML classifier, such as a software system.

3.4.1 **Package**

A namespace used to group together logically related elements within a system.

Each element contained within the package should be a packageable element and

have a unique name.

* + 1. **Packageable element**

A named element, possibly owned directly by a package. These can include events, components, use cases, and packages themselves. Packageable elements can also be rendered as a rectangle within a package, labeled with the appropriate name.

* + 1. **Dependencies**

A visual representation of how one element (or set of elements) depends on or influences another. Dependencies are divided into two groups: access and import dependencies.

* + 1. **Element import**

A directed relationship between an importing namespace and an imported packageable element. This is used to import select individual elements without resorting to a package import and without making it public within the namespace.

* + 1. **Package import**

A directed relationship between and importing namespace and an imported package. This type of directed relationship adds the names of the members of the imported package to its own namespace

* + 1. **Package merge**

A directed relationship in which the contents of one package are extended by the contents of another. Essentially, the content of two packages are combined to produce a new package.

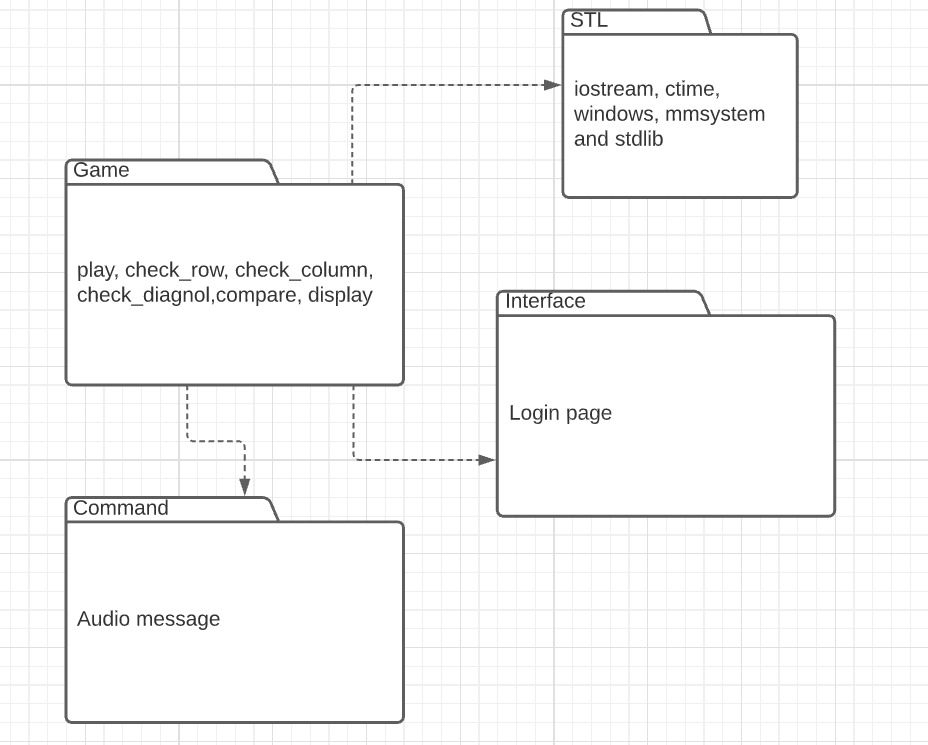


Figure 3.package diagram

**3.5 Class Diagram**

A template for creating objects and implementing behavior in a system. In UML, a class represents an object or a set of objects that share a common structure and behavior. They're represented by a rectangle that includes rows of the class name, its attributes, and its operations. When you draw a class in a class diagram, you're only required to fill out the top row—the others are optional if you'd like to provide more detail. Depending on the context, classes in a class diagram can represent the main objects, interactions in the application, or classes to be programmed.

3.5.1 Name

The first row in a class shape.

3.5.2 **Attributes**

The second row in a class shape. Each attribute of the class is displayed on a

separate line. Attributes are shown in the second partition. The

attribute type is shown after the colon. Attributes map onto member

variables (data members) in code.

* + 1. **Methods**

The third row in a class shape. Also known as operations, methods are displayed in

list format with each operation on its own line. Operations are shown in the third

partition. They are services the class provides. The return type of a method is

shown after the colon at the end of the method signature. The return type of

method parameters is shown after the colon following the parameter name.

Operations map onto class methods in code

* + 1. Generalization

Generalization is a relationship between two classes: a general class and a

special class:

3.5.5 Association

An association represents a relationship between two classes. An association

indicates that objects of one class have a relationship with objects of another class,

in which this connection has a specifically defined meaning (for example, “is flown

with”).

3.5.6 Multiplicity

A multiplicity allows for statements about the number of objects that are

involved in an association.

3.5.7 Aggregation

An aggregation is a special case of an association (see above) meaning “consists

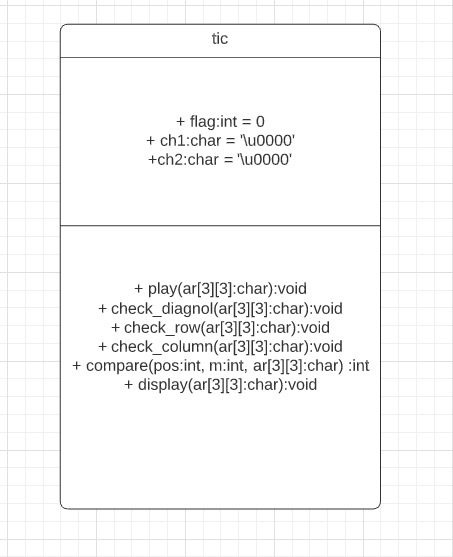
 Of”.

Figure 4.class diagram

# RESEARCH OBJECTIVES

The proposed research is aimed to carry out work leading to the development of a tic tac toe game using C++ language. The proposed aim will be achieved by dividing the work into following objectives:

* 1. To understand and explore various types of C++ language vulnerabilities existing in opensource software.
  2. To study and analyze various techniques that are suitable for vulnerable tic tac toe code discovery.
  3. To design and develop the technique for code and login page.
  4. To analyze and design various methods to add audio files in the code.
  5. To verify and validate the proposed system.

**4.1 Analysis**

Analysis is detailed study of the various operations performed by a system and their relationship

between within and outside is collected on the available files decision and transaction handled by the present system. All the logical aspect of the system is conversed in the phase. Analysis is the most important phase in the system of a system. In analysis phase one has to study the existing system in detail and also collect necessary information regarding the system to be designed. Hence in this phase Uml diagrams are made indicating the data flow in the system, and then only can a system be made correct otherwise it will be incorrect. Analysis is conducted with the following objective in mind:

* Identify the player needs.
* Evaluate the system concept for feasibility
* Perform economic and technical analysis.

• Identify the customer need.

• Evaluate the system concept for feasibility.

• Perform economic and technical analysis.

• Technical Feasibility

• Economic Feasibility

• Behavioral Feasibility

# METHODOLOGY

The following methodology will be followed to achieve the objectives defined for proposed research work:

1. In order to start with C++ project one needs to install Dev C++ IDE in which the code will be written.
2. We can install Dev C++ from sourceforge.net website with any version at least 5.11.
3. We need to install MinGW version for making our project work as it comes with all the library that are needed to perform the functions.
4. We need to add some text in linker option in compiler like -lingw and many more
5. Header files used in this C++ code are –

#include<iostream>

#include<ctime>

#include<stdlib.h>

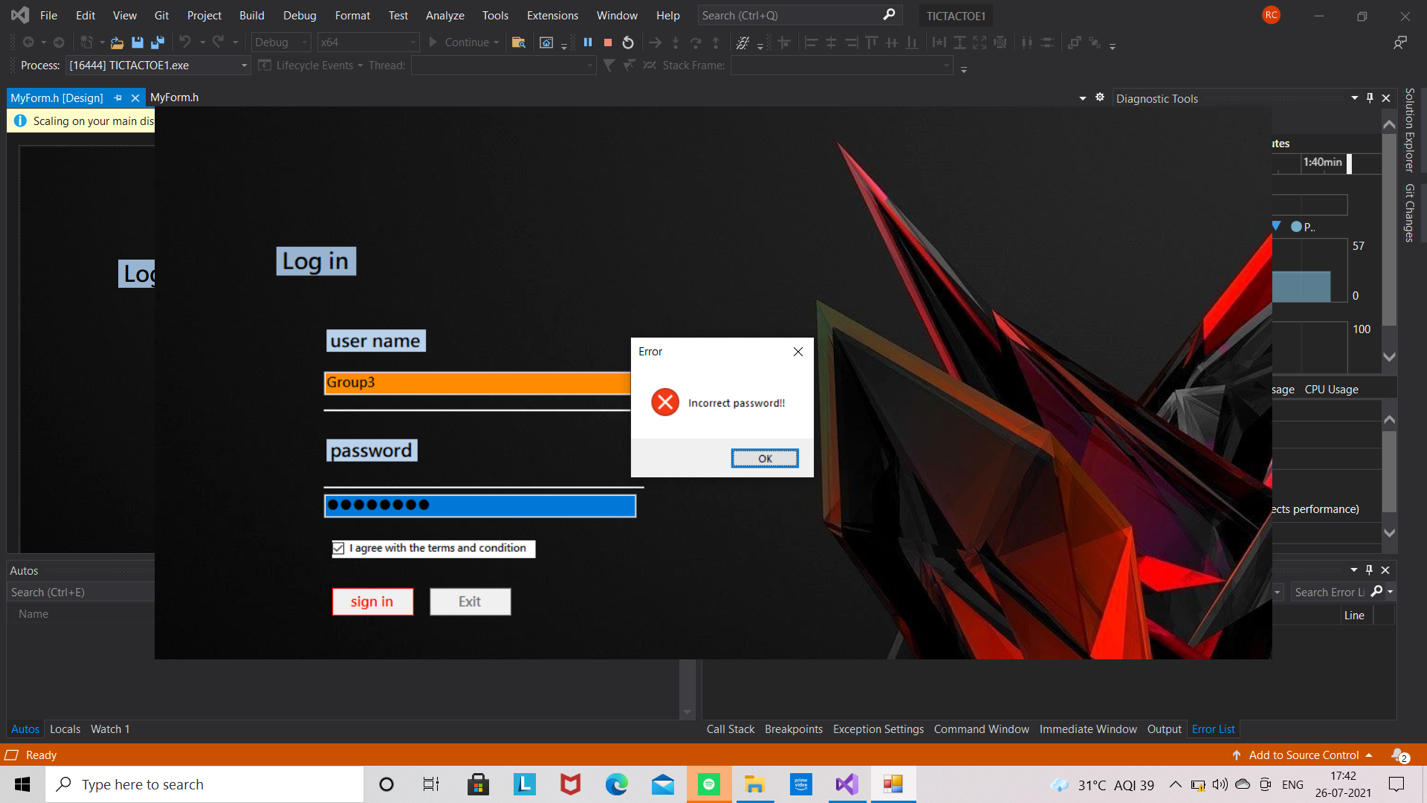
#include<windows.h>

#include<mmsystem.h>

**5.1 Project design**

We planned the project over a period of 17 days and divided it into four iterations. We planned the first iteration for analysis, second iteration for game design, third iteration for coding and the final iteration for the product.

Setting off for a journey usually requires planning -- for instance, it is good to know what destination you want to reach and what stops you will make along the way. We do the same in programming. Before jumping into writing code, we pseudo-code: we write steps we will take and goals we want to achieve. The general procedure for Tic Tac Toe looks like:

* Create a login page to grant access to the user to play the game.
* Create a structure to store and represent the state of the board.
* Ask the user for his choice (with help of voice command) of whether he wants to use X or O for his turn.
* Get the user's move by taking the cell number as input.
* Make sure their move is valid.
* Check to see if the game has been won/lost or came to a draw.
* Change the player turn and let the computer make a move.
* Repeat until the game is over.
  1. **PROJECT SNAPSHOTS**
     1. **Login Page**

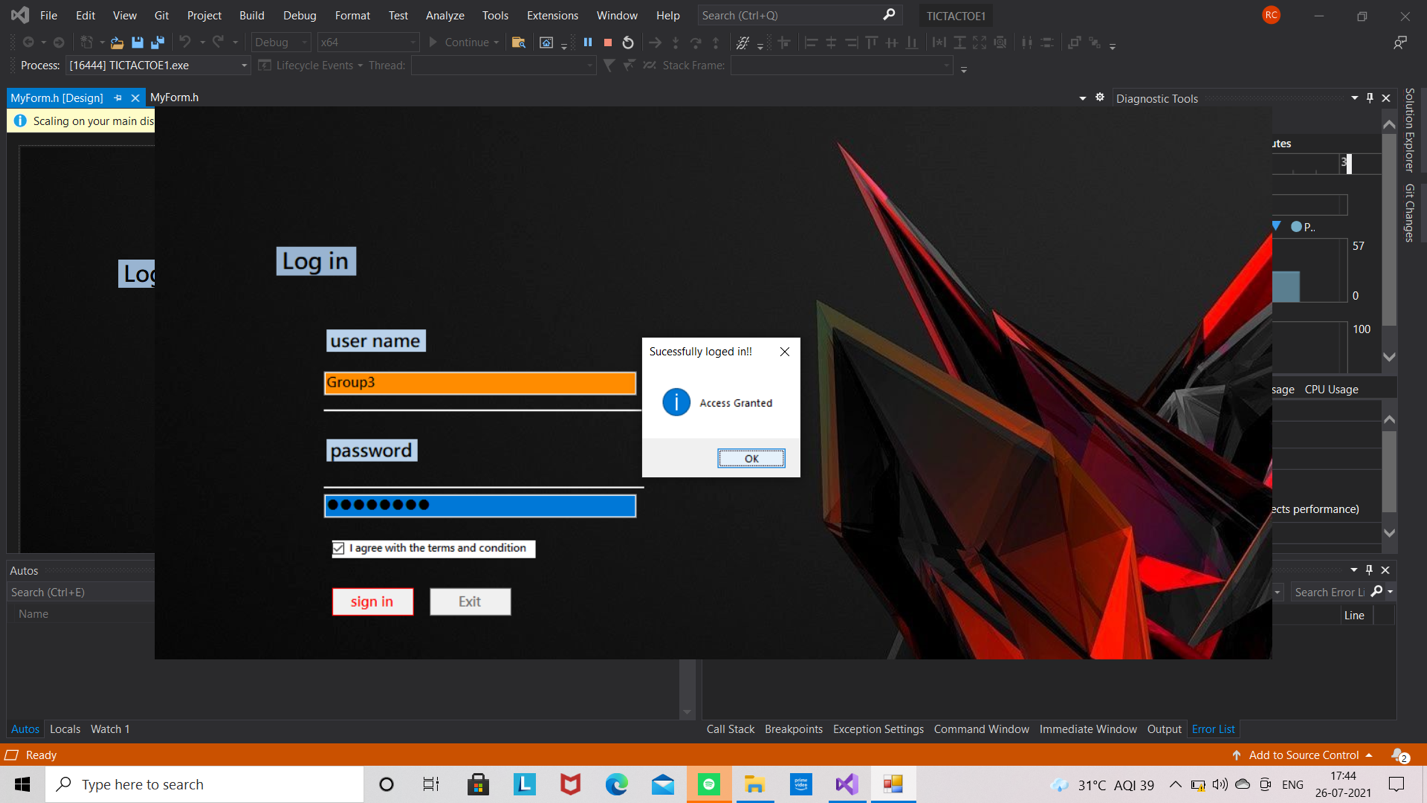
**** Figure 5.invalid username

Figure 6.valid username

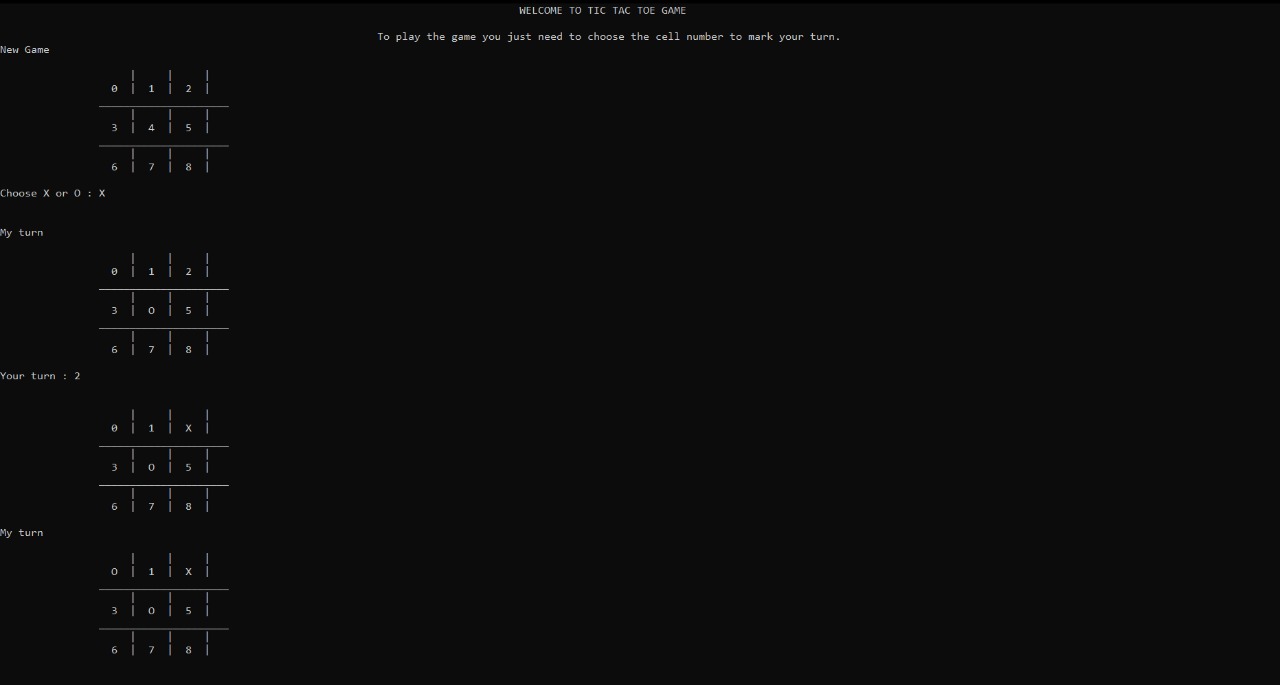
 **5.2.2 Lose scenario**

Figure 7.1.lose scenario

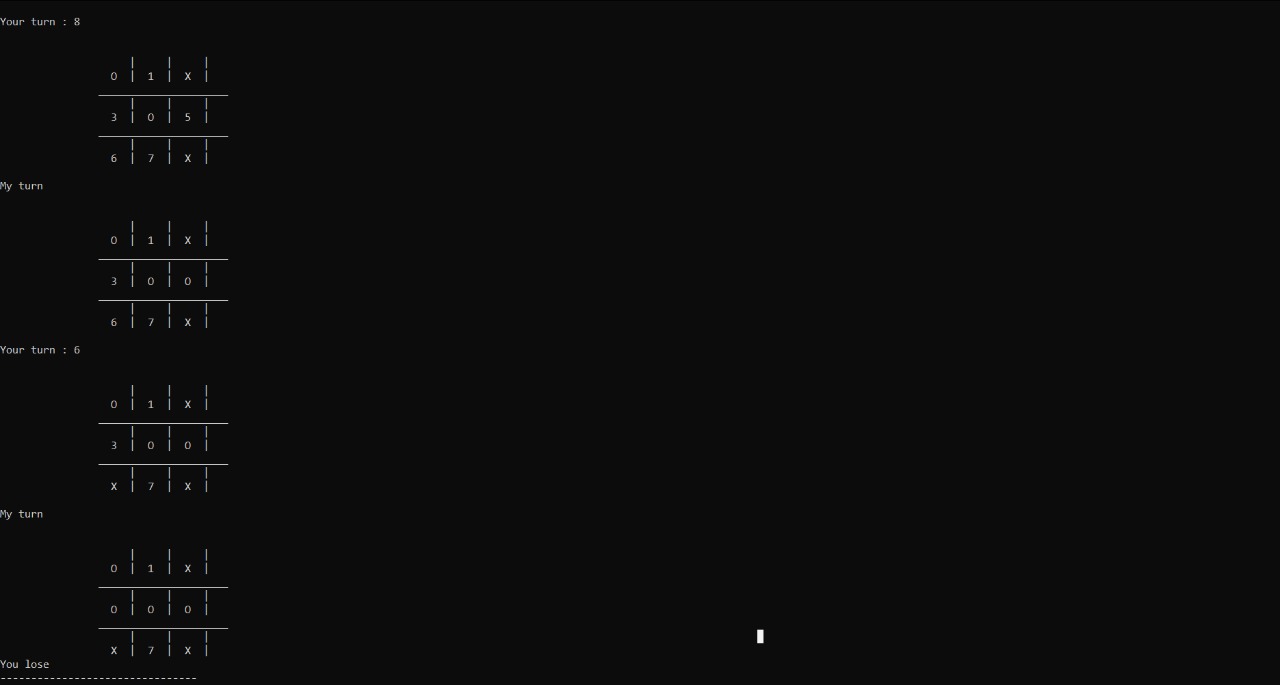


Figure 7.2.lose scenario

5.2.3 Win Scenario

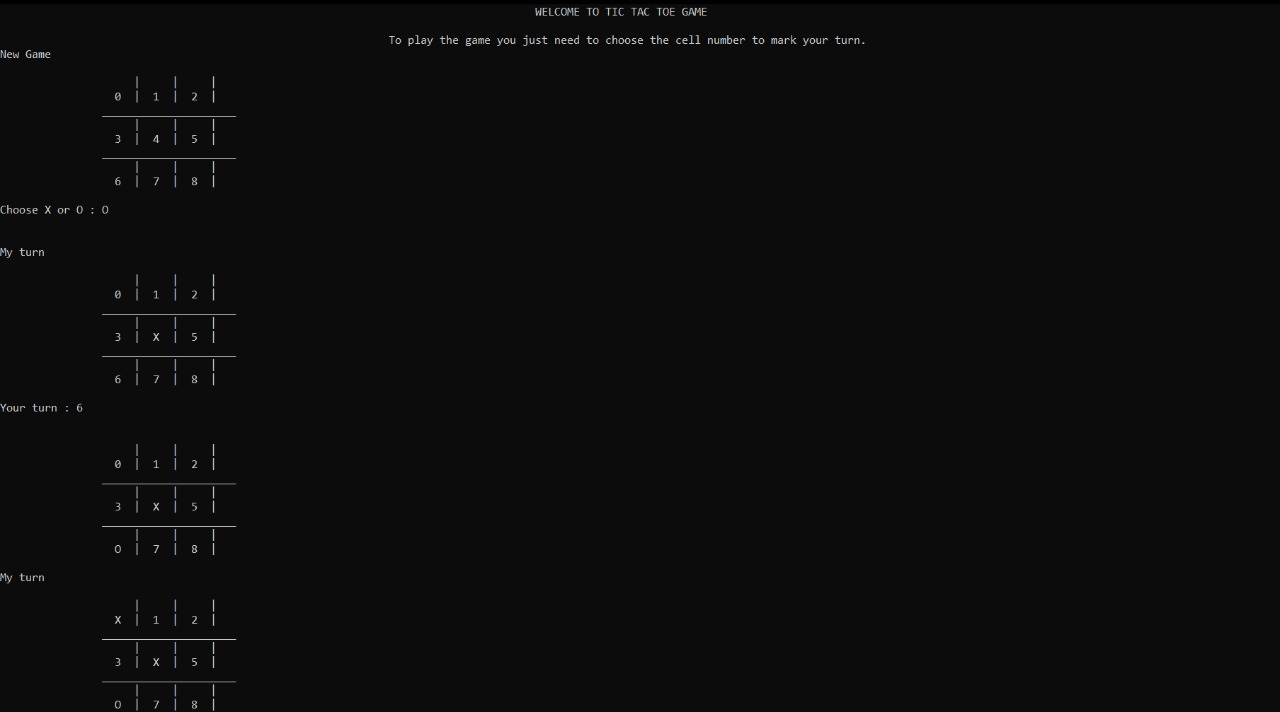


Figure 8.1Win Scenario

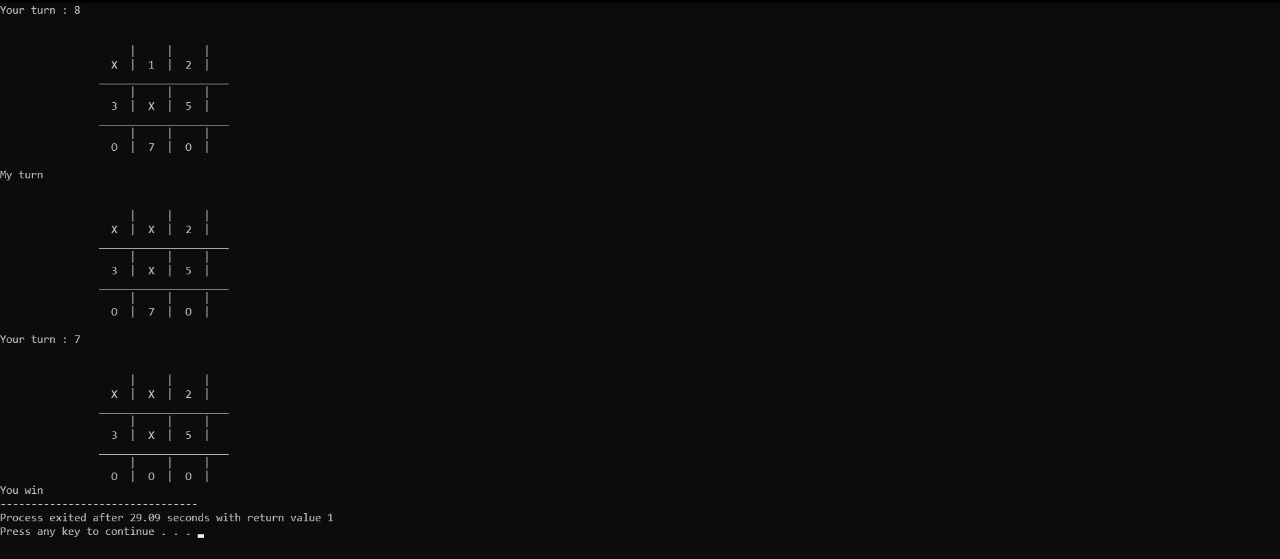


Figure 8.2.Win scenario

5.2.4 Match Tied Scenario

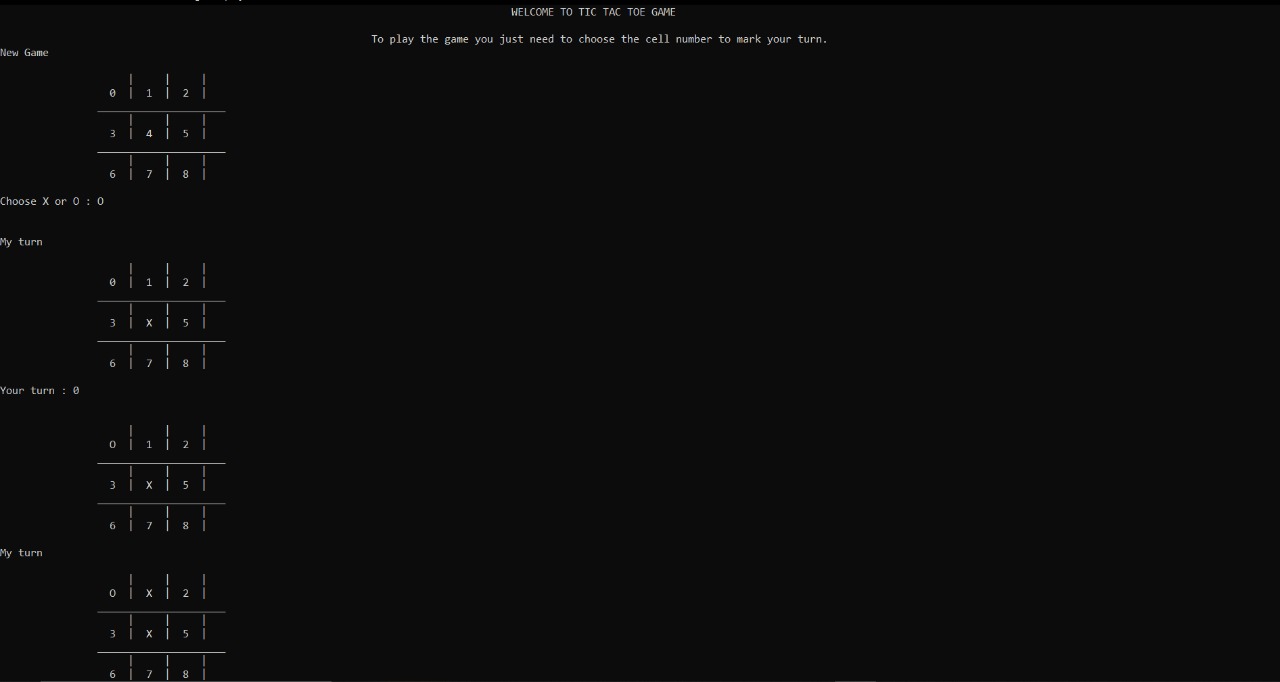


Figure 9.1.Match Tied Scenario

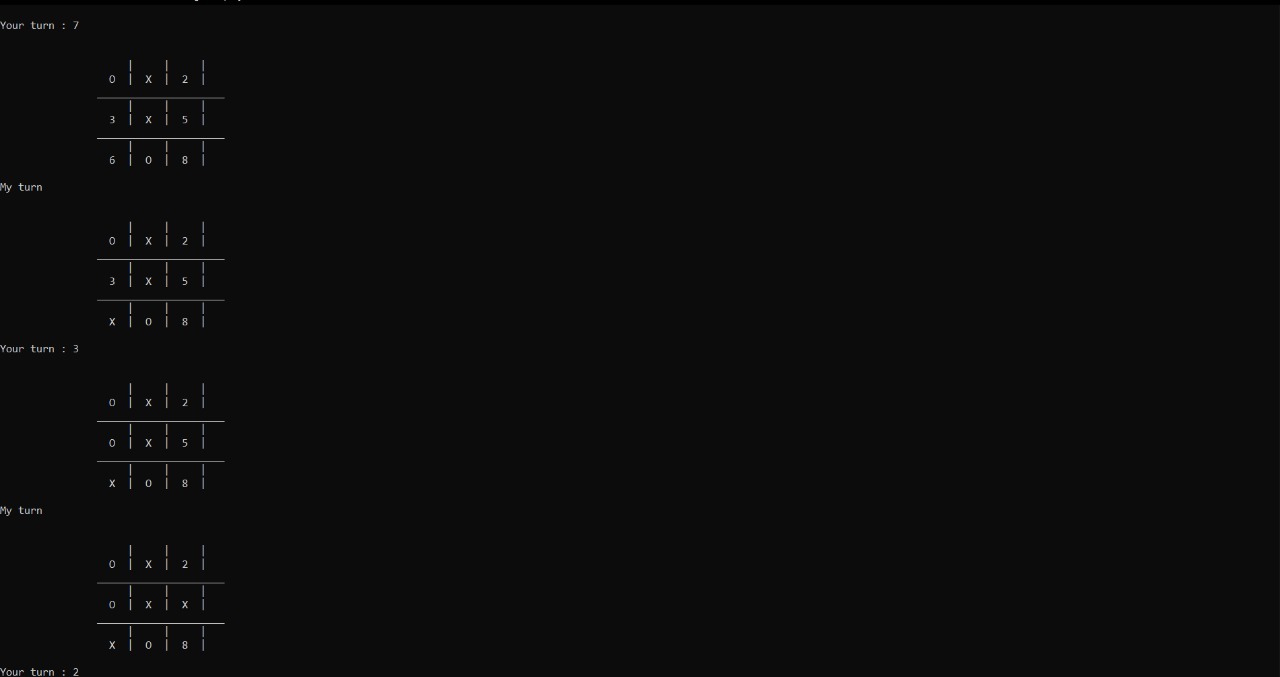


Figure 9.2.Match tied cenario

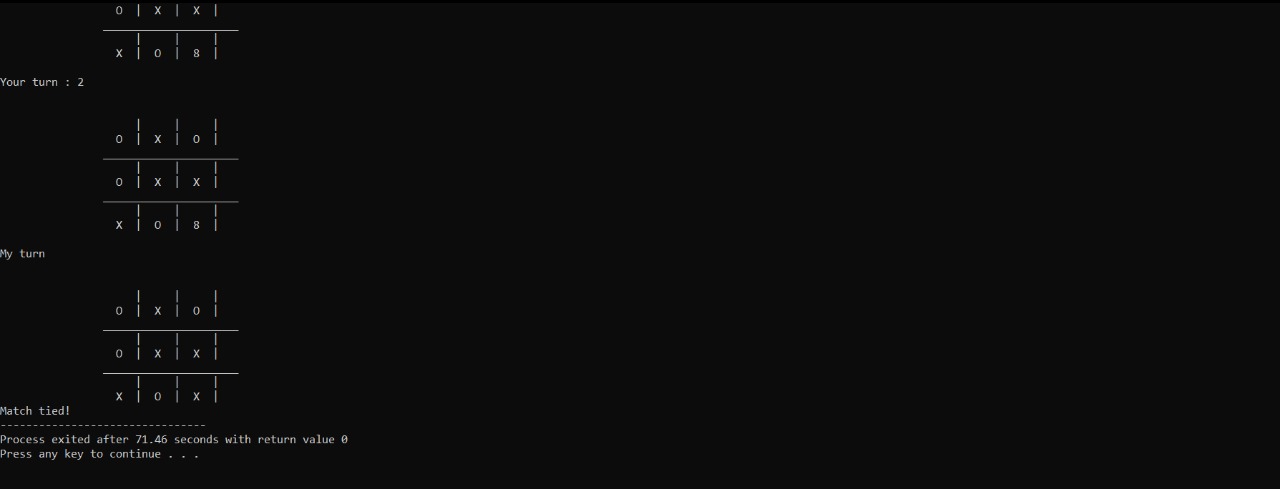


Figure 9.3.Match tied case

1. **RESULTS AND DISCUSSION**

The Tic Tac Toe game is most familiar among all the age groups. Intelligence can be a property of any purpose-driven decision maker. This basic idea has been suggested many times. An algorithm of playing Tic Tac Toe has been presented and tested that works in efficient way. Overall, the system works without any bugs. Our project is only a humble venture to satisfy the needs to manage their project work. Several user-friendly coding has also been adopted. The objective of the software planning is to provide a framework with a limited project completion time frame at the beginning of the project and should be updated on a regular basis.

These results suggest that both importance and predictability play a role in the acoustic realization of a word. Duration is longer and pitch movement is greater for non-predictable words while intensity is greater for important words.

One potential concern is that the effects of intensity may have been the result of paralinguistic factors such as emotional excitement related to winning the game. How one might distinguish between the linguistic and paralinguistic factors that drive prominence is a question of considerable debate. However, it is important to keep in mind that important trials did not consist of only emotionally exciting wins, but also the relatively more routine cases where a win was blocked. Moreover, the speaker’s personal reaction to the importance of their utterance is not inconsistent with the proposal that importance drives acoustic prominence.

**6.1 future work**

* To create a dashboard for the game
* To create a two-player game
* Multiple levels of difficulty
* Maintaining a score (for both players)

# REFERENCES

1. Object-Oriented programming using C++ Robert Lafore 4th Edition Sams Publishers
2. C++Primer Stanley B.Lippmann and Josee Lajore 4th Edition Addison Wesley.
3. C++: The Complete Reference Herbert shieldt 5th Edition\_Tata McGraw Hill Publication.
4. Balaguruswamy, E. (2008). *Object Oriented Programming with C++.* Tata McGraw Hill Publications.
5. Lippman. (2006). *C++ Primer* (3rd ed.). Pearson Education.
6. Robert, L. (2006). *Object Oriented Programming in C++* (3rd ed.). Galgotia Publications References.
7. Schildt, H., & Kanetkar, Y. (2010). *C++ completer* (1st ed.). Tata McGraw Hill.
8. Strousstrup. (2005). *The C++ Programming Language* (3rd ed.). Pearson Publications.
9. Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The unified modeling language user guide (2nd ed.). Upper Saddle River, NJ: Addison-Wesley.
10. https://cppcodetips.wordpress.com/2013/12/23/uml-class-diagram-explained-with-c-samples/
11. https://icarus.cs.weber.edu/~dab/cs1410/textbook/9.Classes\_And\_Objects/uml.html
12. www.cs.wcupa.edu/~zjiang/01-09-67.pdf
13. UML Explained, [Kendall Scott, 2001]
14. "The C++ Programming Language", Bjarne Stroustrup, 4th Edition, Addison-Wesley