**A SUMMER TRAINING REPORT**

**ON JAVA APPLICATION DEVELOPMENT (GAME TIC TAC TOE) USING NETBEANS IDE.**

Submitted by

**Vaibhav Gupta 11014802716**

In partial fulfillment of Summer training for the award of the degree of

**Computer Science Engineering**

Under the supervision of

Tim Buchalka

Java, Android & Python Expert Developer



**Maharaja Agarsen Institute of Technology, PSP Area,**

**Sector-22, Rohini, New Delhi-110085**

# **ACKNOWLEDGEMENT**

First and foremost, I Thank the Almighty God for sustaining the enthusiasm with which I plunged into this endeavor.

I avail this Opportunity to express my profound sense of sincere and deep gratitude to many people who are responsible for the knowledge and experience I have gained during the Project Work.

I have great pleasure in expressing my deep sense of gratitude to guide

**Mr. Tim Buchalka (Udemy.com)** for his valuable and prompt guidance without which this project would not have been a successful one. I extend My overwhelming gratitude to **faculty members of CSE MAIT** for their valuable guidance and meticulous supervision during the preparation of this Project Report.

My hearty and inevitable thanks to all the respondents who helped me to bring out the project in a successful manner. Last but not the least I extend my gratitude towards my parents and friends who extended their wholehearted support towards the successful completion of this Project Work.

Thank you

Vaibhav Gupta

(11014802716)

**CERTIFICATE**

****

**DECLARATION GIVEN BY THE STUDENT**

I hereby declare that the project work entitled **“JAVA APPLICATION DEVELOPMENT (GAME TIC TAC TOE) USING NETBEANS IDE.”**  submitted to the **MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY**, is a record of an original work done by me under the guidance of Tim Buchlak.

Faculty Member and this project work has not performed the basis for the award of any Degree or diploma/ associateship/fellowship and similar project if any.

…………………………….

**Vaibhav Gupta** (11014802716)

# **PREFACE**

This is the report of the project **“JAVA APPLICATION DEVELOPMENT (GAME TIC TAC TOE) USING NETBEANS IDE”** developed by Vaibhav Gupta , pre-final year student of Bachelor Of Technology in Computer Science Engineering at Maharaja Agrasen Institute Of Technology, Delhi during his training at Udemy.com

The objective of this training is to gain practical knowledge in the field of Java development by making some cool real life mini projects that will help us to build a simple yet an effective software application for playing games.

As no learning is complete without a taste of the application. After the course completion, it has undergone a live project.

# **About Organization**

**Udemy.com** is an online learning platform. It is aimed at professional adults.Unlike academic MOOC programs driven by traditional collegiate coursework, Udemy provides a platform for experts of any kind to create courses which can be offered to the public, either at no charge or for a tuition fee.Udemy provides tools which enable users to create a course, promote it and earn money from student tuition charges.

No Udemy courses are currently credentialed for college credit; students take courses largely as a means of improving job-related skills.Some courses generate credit toward technical certification. Udemy has made a special effort to attract corporate trainers seeking to create coursework for employees of their company.

As of 2017, there are more than 55,000 courses on the website.

Udemy serves as a platform that allows instructors to build online courses on topics of their choosing. Using Udemy’s course development tools they can upload video, PowerPoint presentations, audio and live classes to create courses.Instructors can also engage and interact with users via online discussion boards.

Courses are offered across a breadth of categories, including business and entrepreneurship, academics, the arts, health and fitness, language, music, and technology.Most classes are in practical subjects such as Excel software.Udemy also offers Udemy for Business, enabling businesses access to a targeted suite of over 2,000 training courses on topics from digital marketing tactics to office productivity, design, management, programming, and more. With Udemy for Business, organizations can also create custom learning portals for corporate training.

Udemy offers paid and free courses, depending on the instructor.

# **INDEX**

|  |  |  |
| --- | --- | --- |
| S no. | Page heading | Page no. |
| 1. | List of figures. | 2 |
| 2. | Introduction to elements of trading | 3 |
| 3. | Mini Project | 7 |
| 4. | Making of the project | 12 |
| 5. | Screenshots of the project | 15 |
| 6. | Conclusion | 20 |
| 7. | References | 21 |

# **List of figures**

* Java.io
* Java.swing
* JFrame class
* JPanel class
  + Jlable class
  + JButton class
  + JRadioButton class
  + JTextArea class
  + OpenDialog Box.

# **Introduction of elements of trading**

JAVA

Java produces applets (browser-run programs), which facilitate graphical user interface (GUI) and object interaction by Internet users. Prior to Java applets, Web pages were typically static and non-interactive. Java applets have diminished in popularity with the release of competing products, such as Adobe Flash and Microsoft Silverlight.  
  
 Java applets run in a Web browser with Java Virtual Machine (JVM), which translates Java bytecode into native processor instructions and allows indirect OS or platform program execution. JVM provides the majority of components needed to run bytecode, which is usually smaller than executable programs written through other programming languages. Bytecode cannot run if a system lacks required JVM.  
  
 Java program development requires a Java software development kit (SDK) that typically includes a compiler, interpreter, documentation generator and other tools used to produce a complete application.   
  
 Development time may be accelerated through the use of integrated development environments (IDE) - such as JBuilder, Netbeans, Eclipse or JCreator. IDEs facilitate the development of GUIs, which include buttons, text boxes, panels, frames, scrollbars and other objects via drag-and-drop and point-and-click actions.  
  
 Java programs are found in desktops, servers, mobile devices, smart cards and Blu-ray Discs (BD).

Java is a programming language that produces software for multiple platforms. When a programmer writes a Java application, the compiled code (known as bytecode) runs on most operating systems (OS), including Windows, Linux and Mac OS. Java derives much of its syntax from the C and C++ programming languages.  
  
 Java was developed in the mid-1990s by James A. Gosling, a former computer scientist with Sun Microsystems.

Java is Object Oriented language. Object Oriented Programming is a programming concept that works on the principle that objects are the most important part of your program. It allows users create the objects that they want and then create methods to handle those objects. Manipulating these objects to get results is the goal of Object Oriented Programming.

Object Oriented Programming popularly known as OOP, is used in a modern programming language like Java

During my training period, I worked on the following technologies:

1. **Swing**

Java Swing is a lightweight Graphical User Interface (GUI) toolkit that includes a rich set of widgets. It includes package lets you make GUI components for your Java applications, and It is platform independent.

The Swing library is built on top of the Java Abstract Widget Toolkit (**AWT**), an older, platform dependent GUI toolkit. You can use the Java GUI components like button, textbox, etc. from the library and do not have to create the components from scratch.

Swing is a part of **JFC (Java Foundation Classes)**. Building Graphical User Interface in Java requires the use of Swings. Swing Framework contain a large set of components which allow high level of customization and provide rich functionalities, and is used to create window based applications.  
Java swing components are lightweight, platform independent, provide powerful components like tables, scroll panels, buttons, list, color chooser, etc.

1. **Container class**

Container classes are classes that can have other components on it. So for creating a GUI, we need at least one container object. There are 3 types of containers.

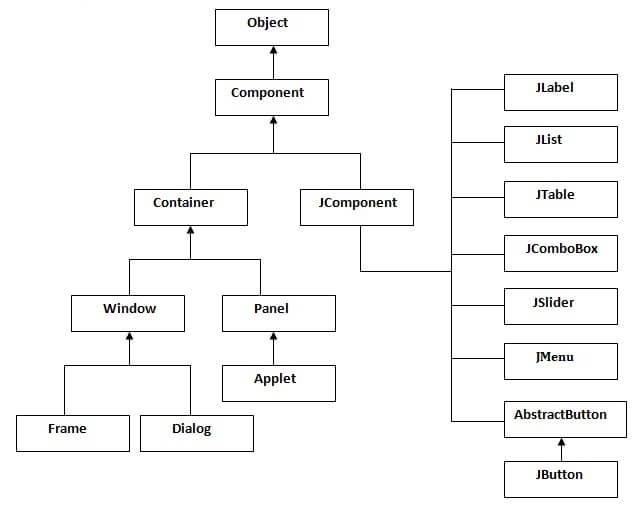
1. **Panel**: It is a pure container and is not a window in itself. The sole purpose of a Panel is to organize the components on to a window.
2. **Frame**: It is a fully functioning window with its title and icons.
3. **Dialog**: It can be thought of like a pop-up window that pops out when a message has to be displayed. It is not a fully functioning window like the Frame.
4. **JPanel**

JPanel, a part of Java Swing package, is a container that can store a group of components. The main task of JPanel is to organize components, various layouts can be set in JPanel which provide better organisation of components, however it does not have a title bar.

1. **JFrame**

JFrame is a class of javax.swing package extended by java.awt.frame, it adds support for JFC/SWING component architecture. It is the top level window, with border and a title bar. JFrame class has many methods which can be used to customize it.

### **Hierarchy of Java Swing classes**

The hierarchy of java swing API is given below.

# **MINI PROJECT**

**Tic Tac Toe**

**Tic-tac-toe** is a game for two players, *X* and *O*, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.

This project was made by using UI feature of JAVA language. JAVA offers users with a variety of libraries and pre-defined constructors.

The project displays a 3X3 grid and allows the user to cycle between ‘X’, ‘O’, and ‘ ’ by cycle clicks on the grid. The intent of the project is to incorporate the logic of the game through programming. The project also includes a button to reset the grid.

This game is developed using **NETBEANS IDE** which allows the users to use variety of features example the design view in which we can make the front end of the application using variety of tools as JFrame, JButton, JPanel, Jlabel, etc.

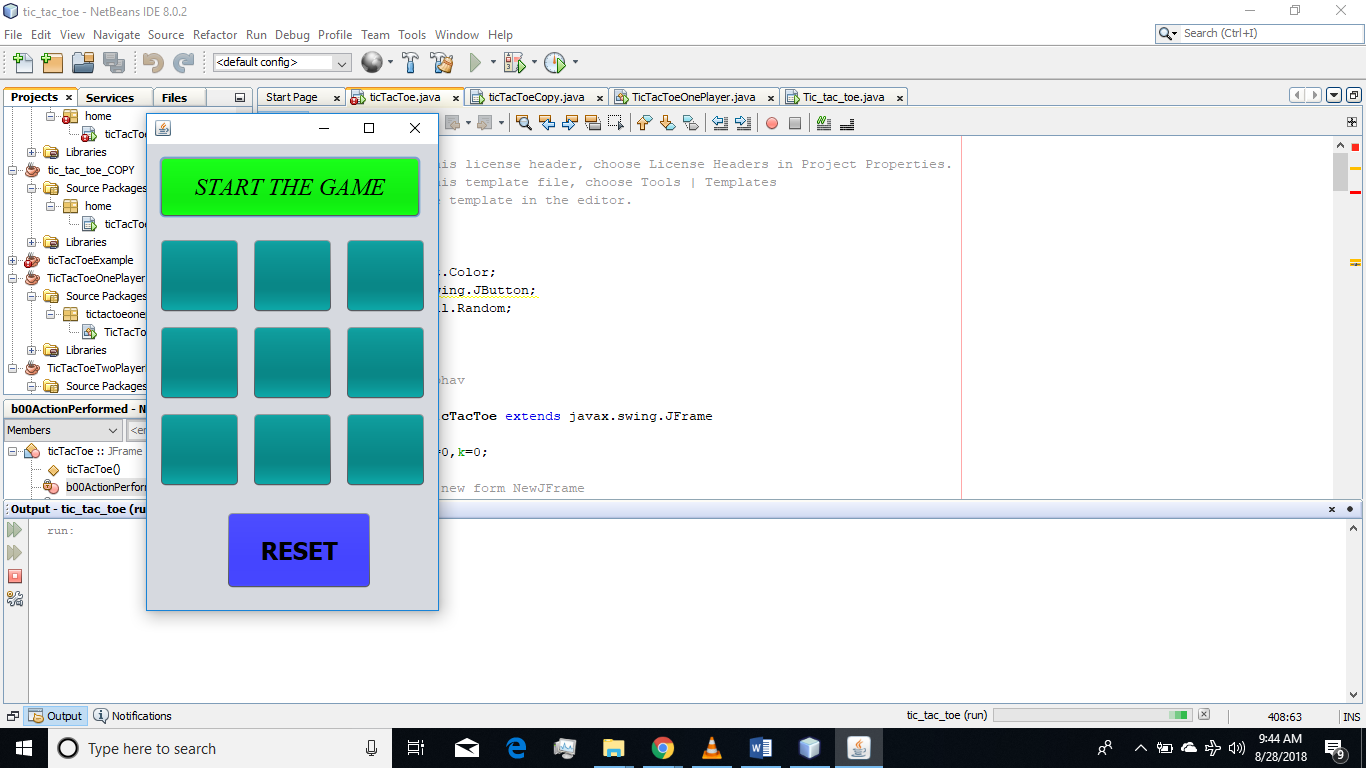
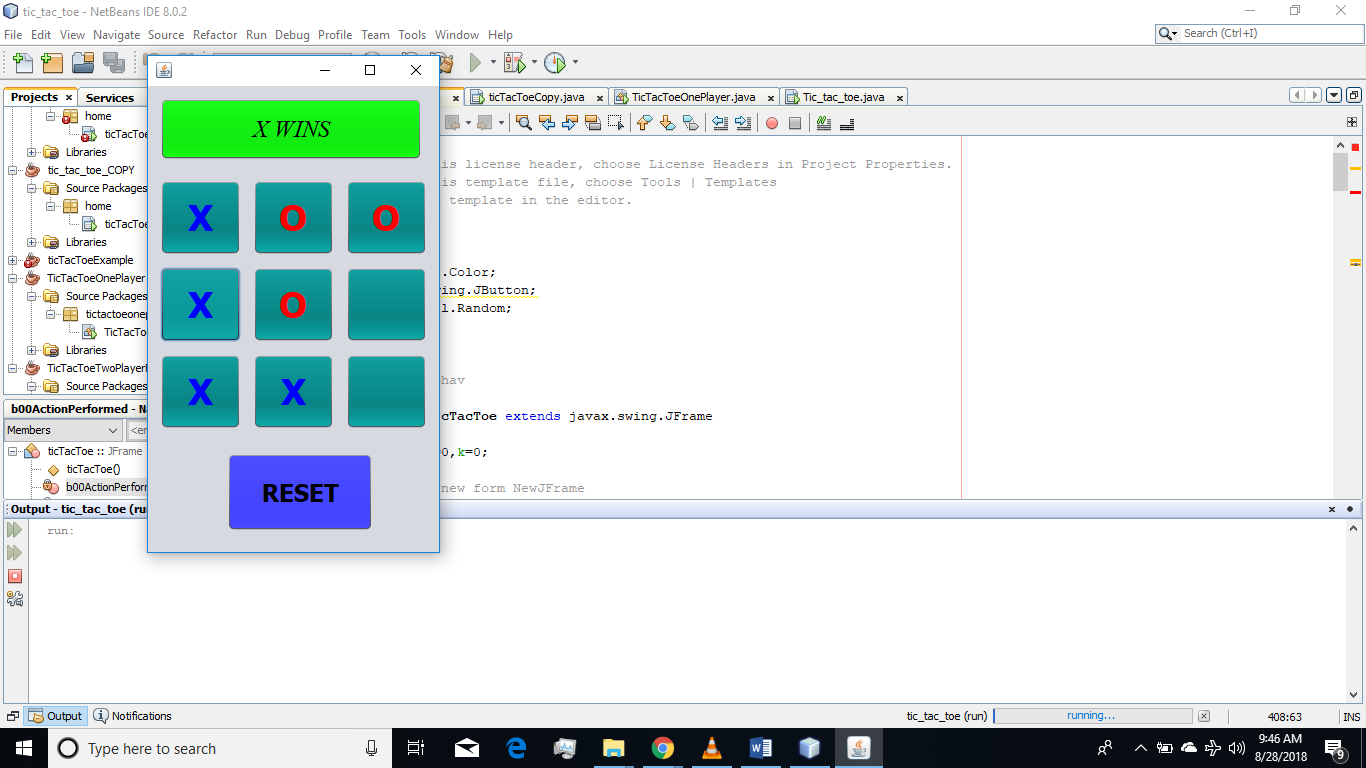
 

Figure 1 TTT Game Figure 2 TTT Win in

Board game

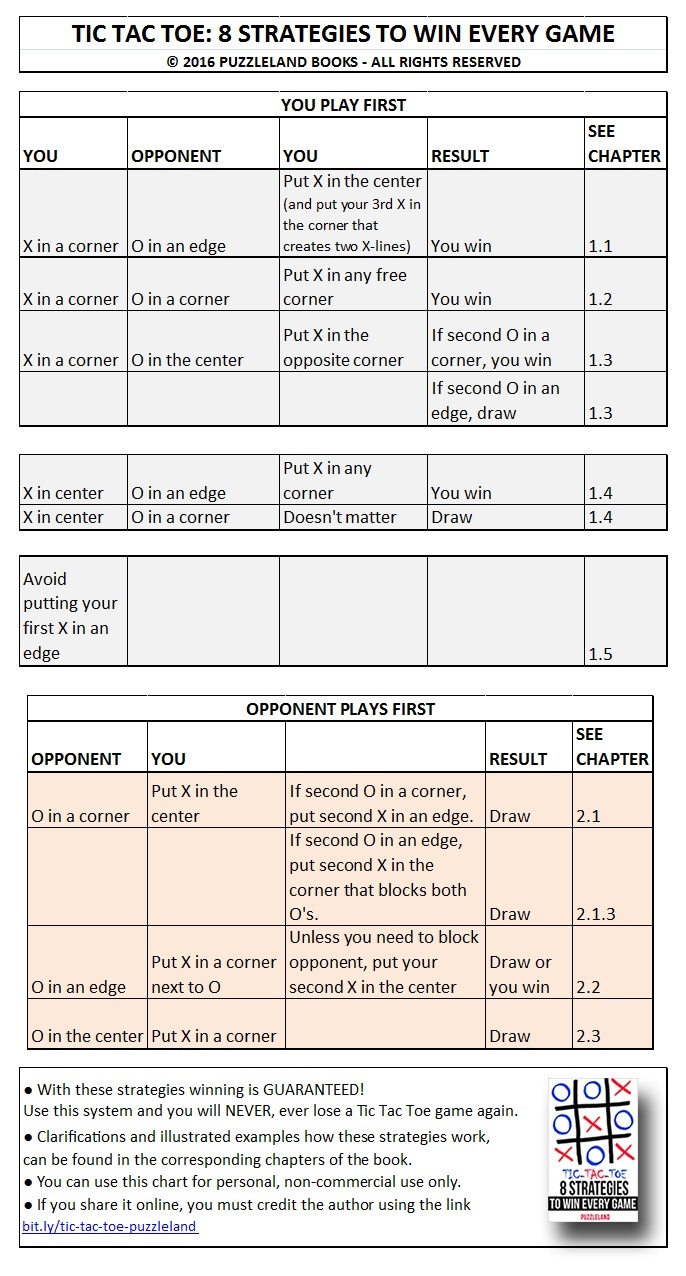
**Tic Tac Toe** (also known as **Noughts and crosses** or **Xs and Os**) is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.



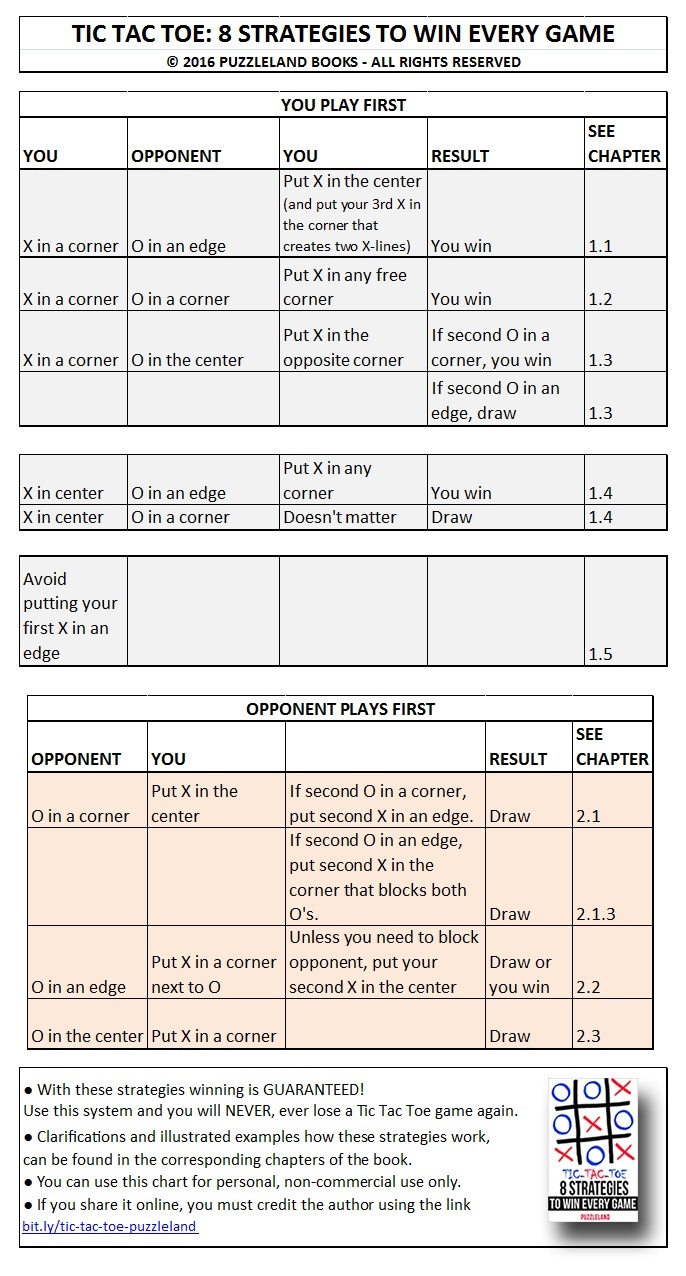
Tic-Tac-Toe Game is a very popular game played by two participants on the grid of 3 by 3. A special symbol (X or O) is assigned to each participant to indicate that the slot is covered by the respective participant. The winner of the game is the participant who first cover a horizontal, vertical or diagonal row of the board having only their symbols. This project proposed a winning strategy of Tic-Tac-Toe game. This algorithm is designed for computer as a player in which computer act according to the intelligence of model to maximize the chances of success. The human player can makes its own choices. Any of the player can play first by their choice. The computation rules ensures selection of best slot for computer that will lead to win or prevent opponent to make a winning move. This is extended work of the paper “The Winner Decision Model of Tic-Tac-Toe Game by using Multi-Tape Turing Machine”. The contribution of this work is to design a strategy to play Tic-Tac-Toe game in which computer will never lose.

TECHNIQUE USED IN BUILDING THE LOGIC OF THE GAME.

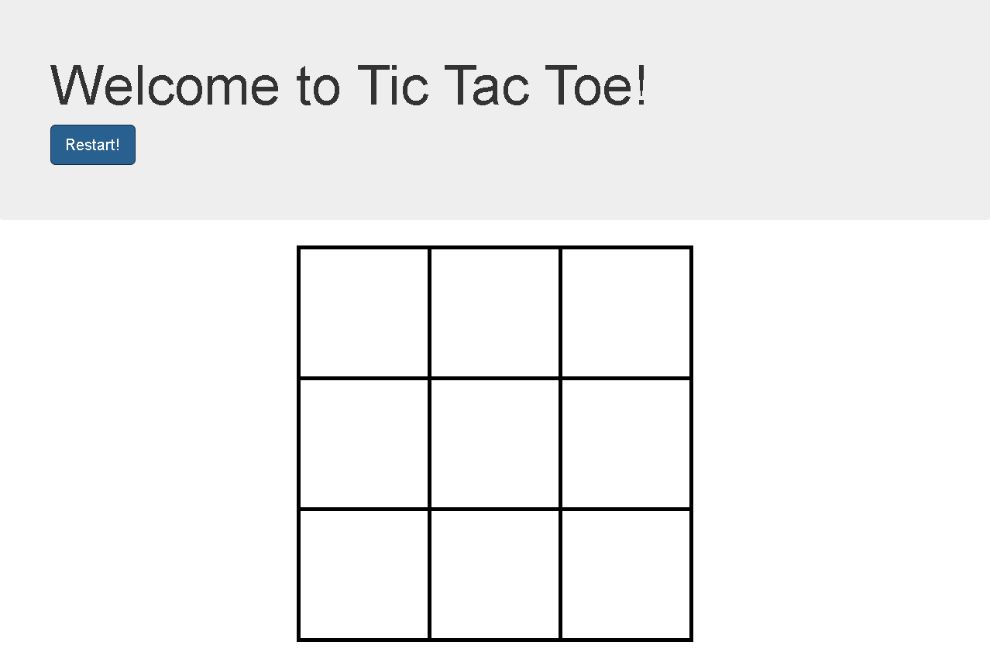
These are the steps followed by the plyer or computer to win (or at least draw the game of Tic Tac Toe).



**Table 1.1**



* **Counter-Attack -**Making a move that blocks your opponent
* **Center -**The square in the middle surrounded by all the other squares.
* **Edge -**A piece bordering the center.
* **Corner -**A piece bordered by two edge squares.



If any of the following cases arises when the opponent player is about to win then comes into play is the **defensive move** in which the computer will move to block the way for the player to win i.e. not to complete a complete line of three elements.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **O** | **O** |  |  |  |  |  |  |  |  |  |  |  | **O** |  |  |
|  |  |  |  |  |  |  | **O** |  |  | **O** |  |  |  |  |  |  |
|  |  |  |  |  |  |  | **O** |  | **O** |  |  |  |  | **O** |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **O** | **O** |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | **O** |  |  |  |  | **O** |  |  | **O** |  | **O** |  |
|  |  |  |  |  | **O** |  |  |  |  |  | **O** |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | **O** |  |  |  |  |  | **O** |  |  |  | **O** |  |
|  |  |  |  |  | **O** |  |  |  |  | **O** |  |  |  |  |  |  |
|  |  | **O** | **O** |  |  |  |  |  |  |  |  |  | **O** |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | **O** |  | **O** |  |  |  | **O** |  |  |  |
|  |  |  |  |  |  |  | **O** |  |  | **O** |  |  |  |  |  |  |
|  | **O** | **O** |  |  |  |  |  |  |  |  |  |  |  |  | **O** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **O** |  | **O** |  |  |  |  |  |  |  | **O** |  |  |  |  |  |
|  |  |  |  |  |  | **O** |  |  |  |  |  |  | **O** | **O** |  |  |
|  |  |  |  |  |  | **O** |  |  |  |  | **O** |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | **O** |  |  | **O** |  |  |  |  |  |  |  |
|  |  |  |  |  |  | **O** |  |  |  |  |  |  |  | **O** | **O** |  |
|  | **O** |  | **O** |  |  |  |  |  | **O** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Table 1.2

This will guarantee that the opponent will not win the game but it will not guarantee the winning scenario of the computer. Instead these strategies

Will guarantee draw of the game .

# **Making of the project**

DECLARATION OF LIBRARIES AND BUILT IN CONSTRUCTORS IN PROJECT

Import javax.swing.JButton;

Import javax.swing.JLabel;

Import javax.swing.JTextArea;

Import javax.swing.JFrame;

Import javax.swing.JPanel;

Import javax.swing.JToggleButton;

Import javax.swing.JDialog;

Import java.awt.Color;

Import java.util.Random;

OBJECTS PROJECT

THESE ARE FOR THE 3\*3 GRID

private javax.swing.JButton b00;

private javax.swing.JButton b01;

private javax.swing.JButton b02;

private javax.swing.JButton b10;

private javax.swing.JButton b11;

private javax.swing.JButton b12;

private javax.swing.JButton b20;

private javax.swing.JButton b21;

private javax.swing.JButton b22;

REST ARE HERE

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JButton oneplayer;

private javax.swing.JButton reset;

private javax.swing.JDialog select;

private javax.swing.JToggleButton start;

private javax.swing.JButton twoplayer;

MEMORY ALLOCATION TO THE OBJECTS

bindingGroup = new org.jdesktop.beansbinding.BindingGroup();

select = new javax.swing.JDialog();

jPanel3 = new javax.swing.JPanel();

oneplayer = new javax.swing.JButton();

twoplayer = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jPanel1 = new javax.swing.JPanel();

b00 = new javax.swing.JButton();

b01 = new javax.swing.JButton();

b02 = new javax.swing.JButton();

b22 = new javax.swing.JButton();

b11 = new javax.swing.JButton();

b12 = new javax.swing.JButton();

b21 = new javax.swing.JButton();

b20 = new javax.swing.JButton();

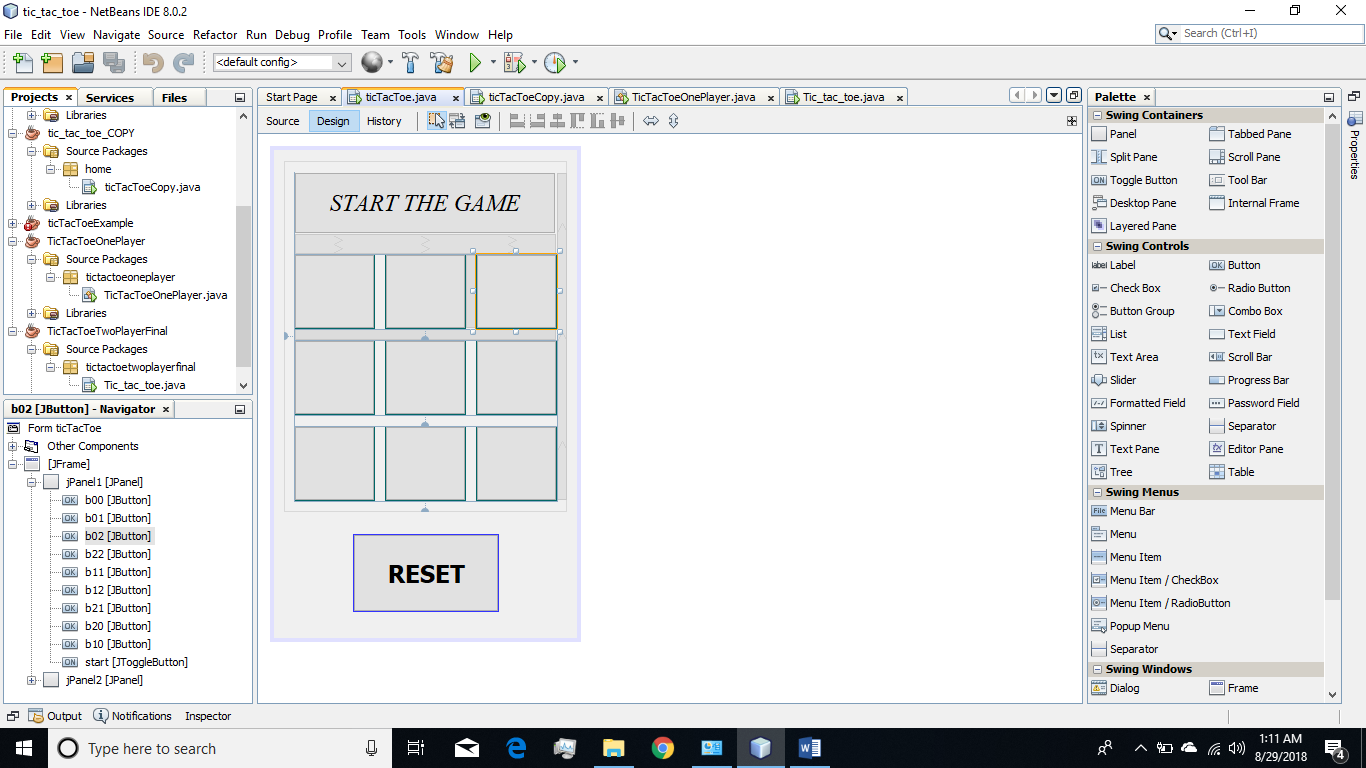
b10 = new javax.swing.JButton();

start = new javax.swing.JToggleButton();

jPanel2 = new javax.swing.JPanel();

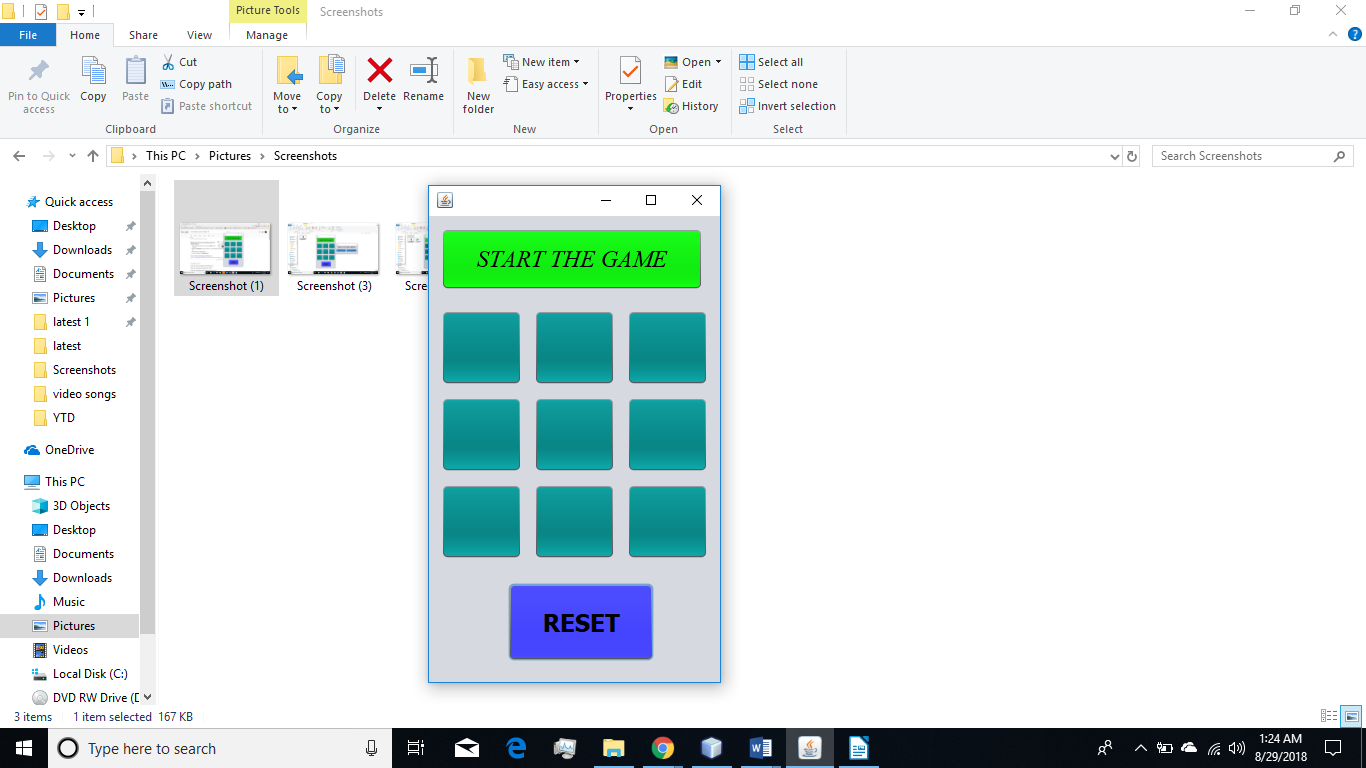
reset = new javax.swing.JButton();

# **Screenshots of the project**



Original blueprint of the application window.

Figure 2.1

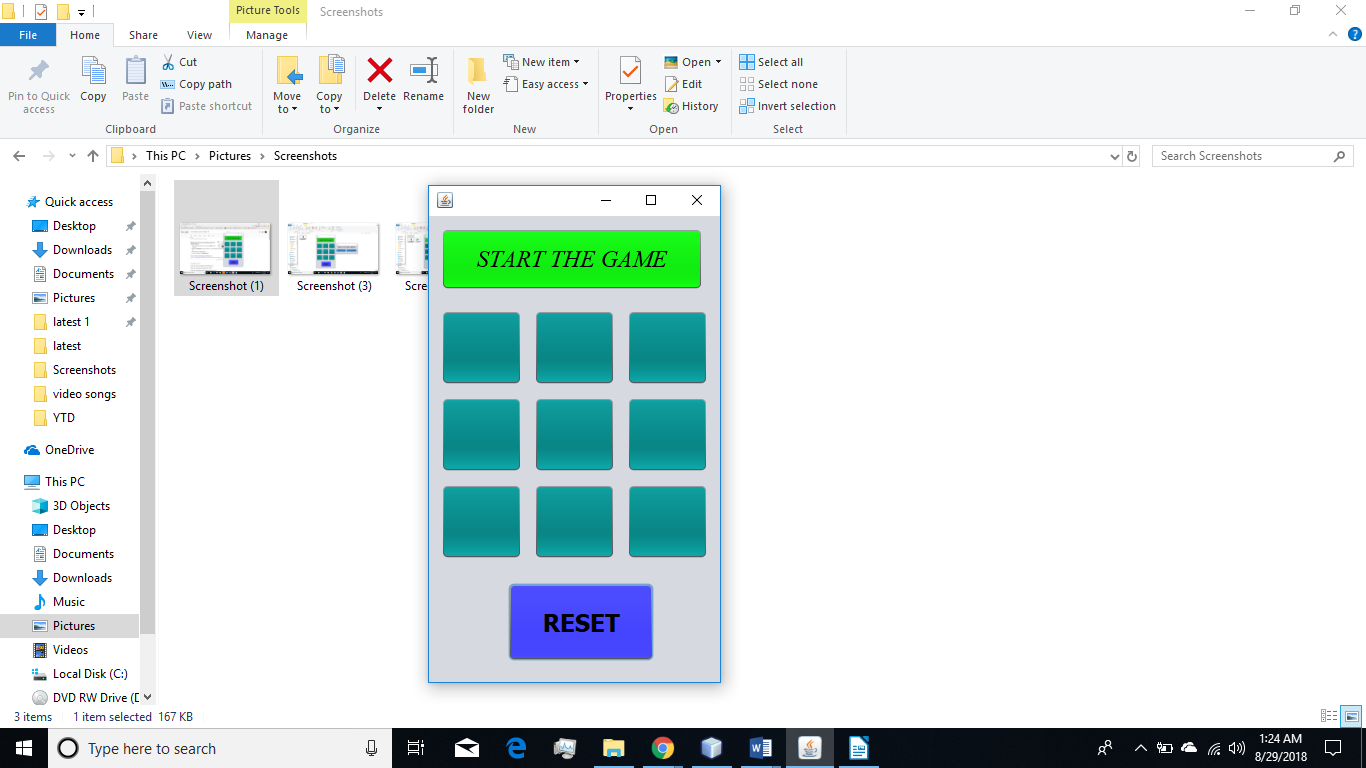


GUI representation of the opening screen of the game.

START THE GAME button will start the game.

RESET will reset the board.

Figure 2.2



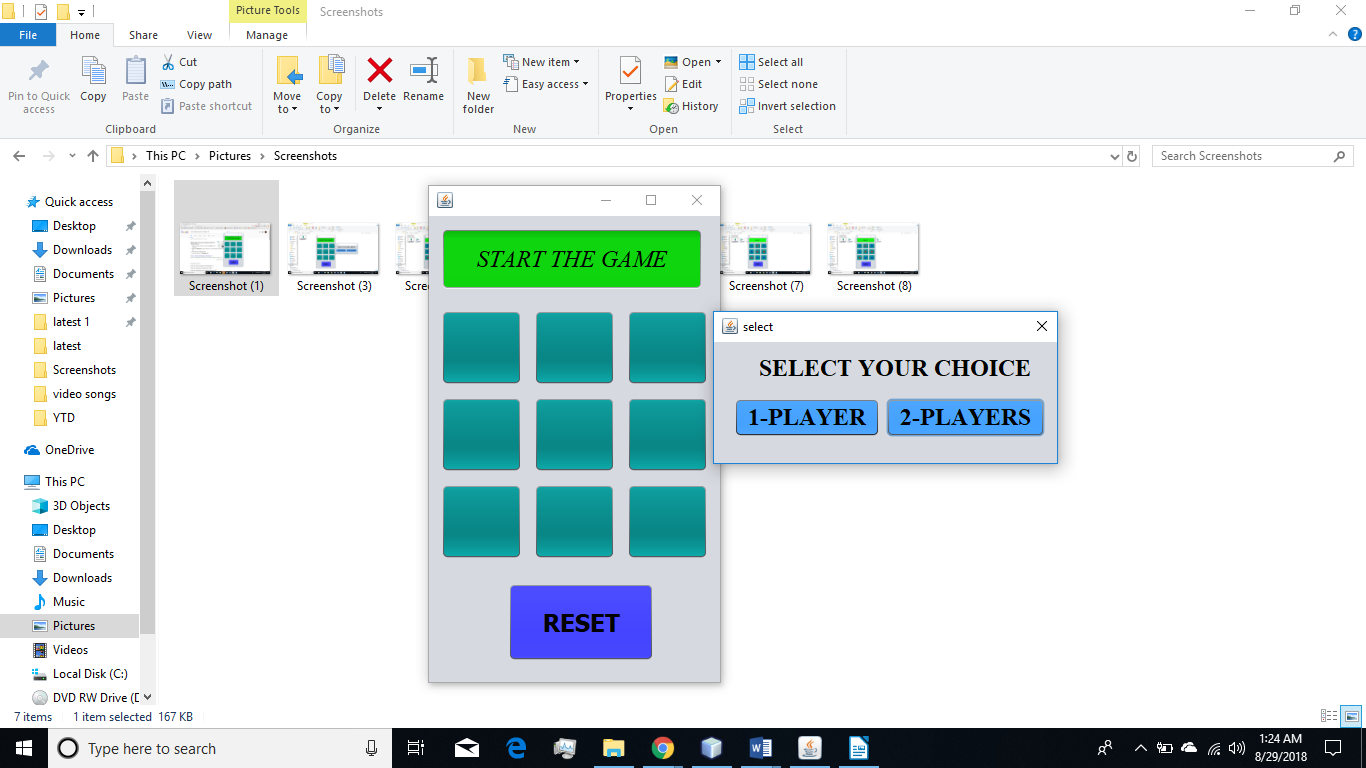
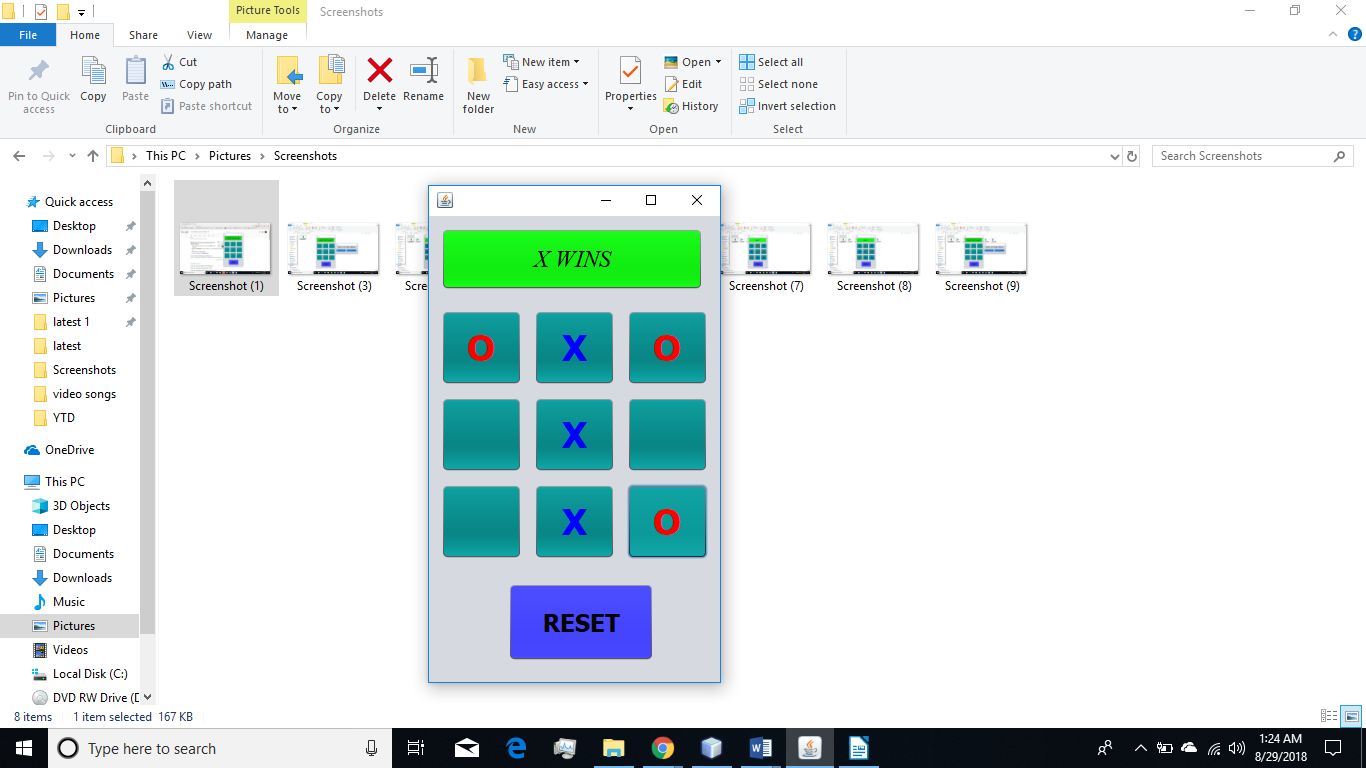


Figure 2.4

After clicking at the “START THE GAME” button this dialogue box will appear which will ask the player weather he want to play with computer or with an additional player.

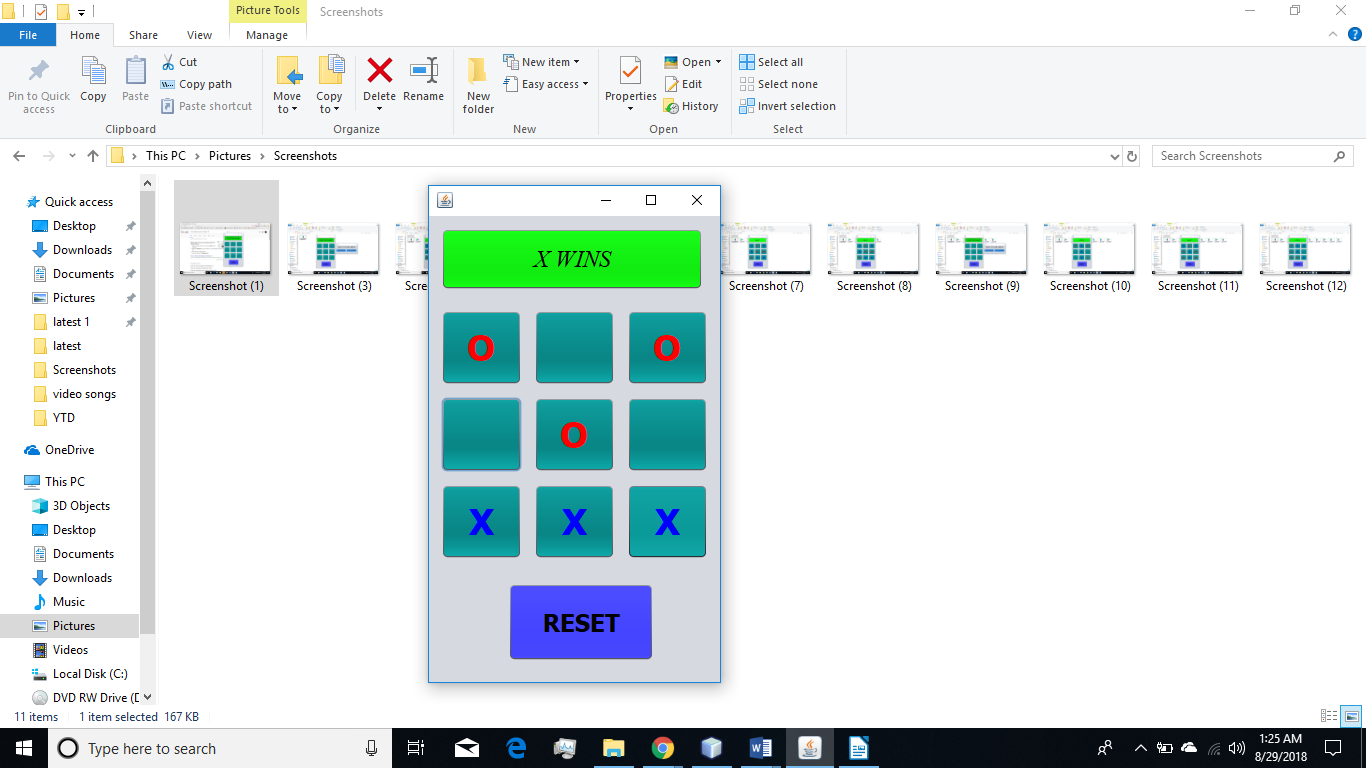
Figure 2.3

PLAYING WITH A HUMAN PLAYER



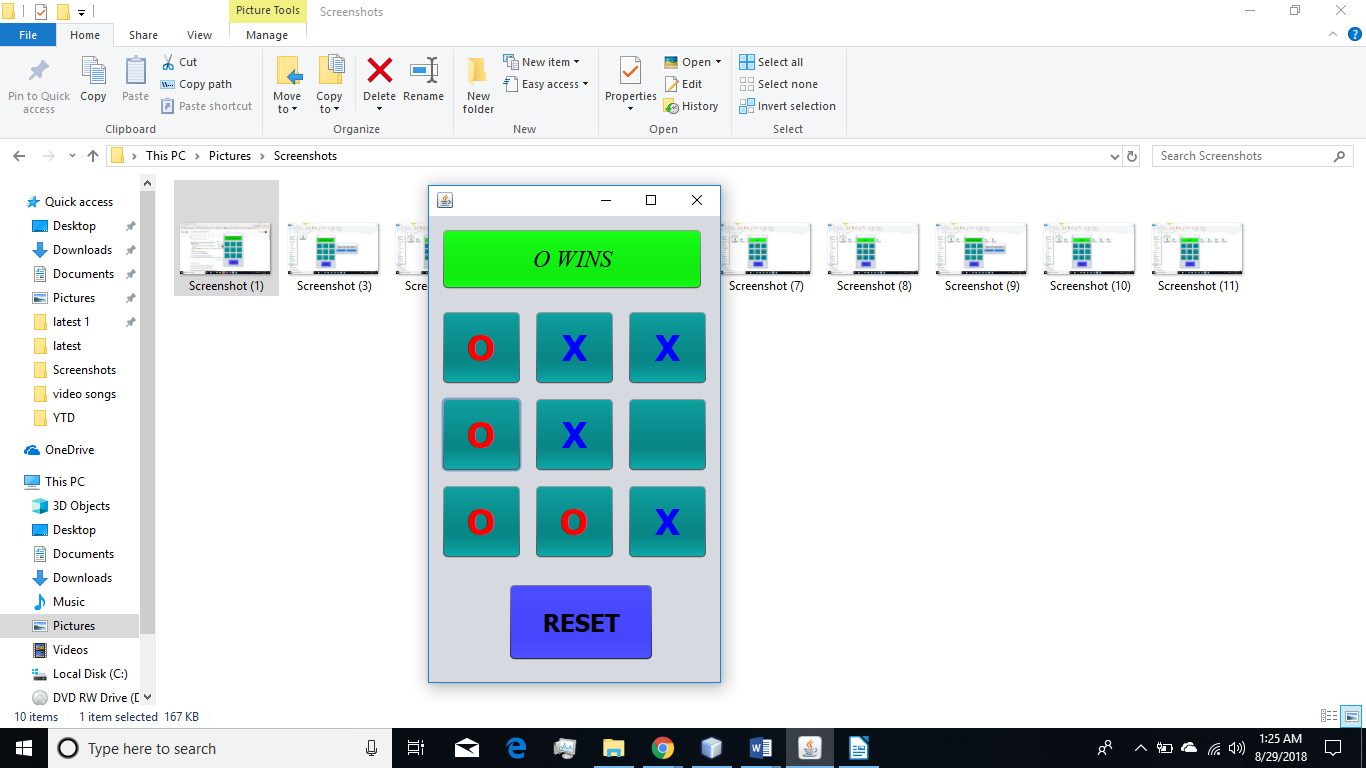
When one player X wins the game.

Figure 2.5



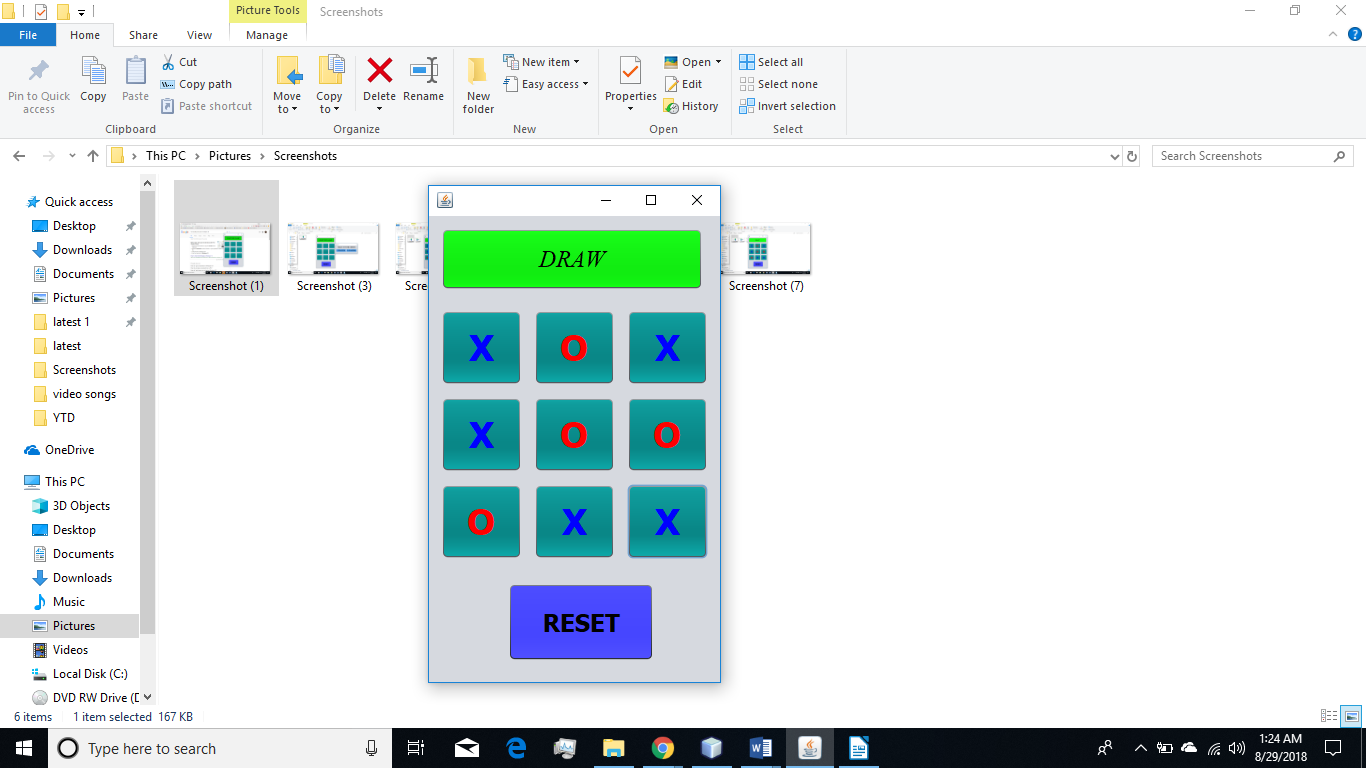
When a player W wins the game.

Figure 2.7



When a player O wins the game.

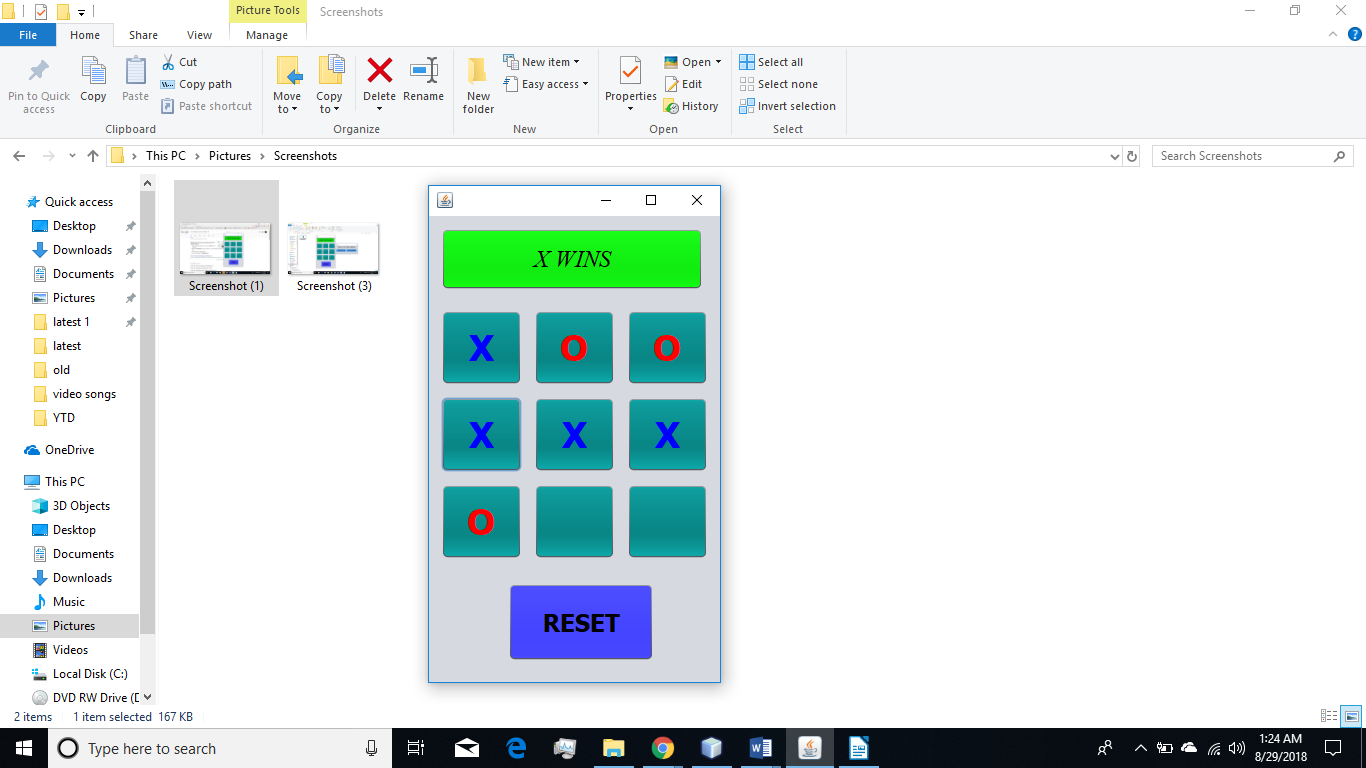
Figure 2.8



When no player wins the game i.e. the game draws.

Figure 2.9

PLAYING AGAINST THE COMPUTER.

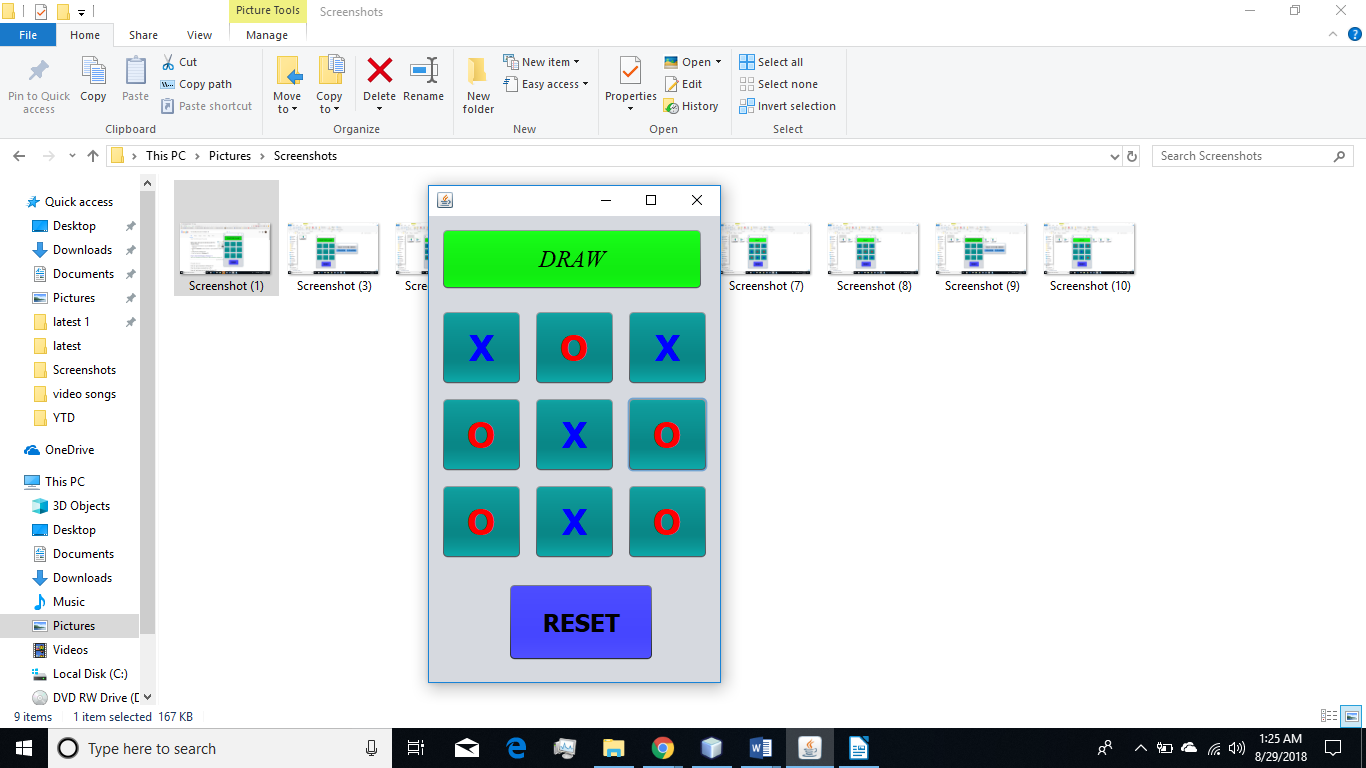


When the Computer wins the game

against a player O.And O here

moves first.

Figure 2.10



No win scenario when Computer

and the player game draws.

Figure 2.11

# **Conclusion**

* I learnt a new language (Java) and learnt how to build mini projects on the same.
* Java have many built in libraries and constructors which can be used to create applications for desktop.
* I learnt about IDE (Integrated Development Environments) mainly about NETBEANS IDE.
* Java GUI library (JAVAX.SWING) is easy to use and have many tools which can be used to make practical projects.
* Learnt how to remove the errors from the program using debugging.

# **References**

For making this report following references were used:

1. <https://docs.oracle.com/javase/tutorial/uiswing/components/textfield.html>
2. <https://www.geeksforgeeks.org/java-swing-internal-frame-with-examples/>
3. https://www.quora.com/Is-there-a-way-to-never-lose-at-Tic-Tac-Toe
4. <https://ieeexplore.ieee.org/document/8003944/?reload=true>
5. <https://en.wikipedia.org/wiki/Swing_(Java)>
6. <https://www.fromdev.com/2014/10/most-widely-used-java-libraries.html>