Software Engineering Principles using Android

A PROJECT REPORT

ON

Tic-Tac-Toe

GROUP MEMBERS

1.Sonali Dixit	-	1641012248
2.Srestha Epari	-	1641012409
3.Ritika Kumari	-	1641012427
4.Yash Jhamnani	-	1641012476

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Institute of Technical Education and Research

SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY

Bhubaneswar, Odisha, India

ACKNOWLEDGEMENT

It is matter of great pleasure for us to get this opportunity expressing our sincere sense of gratitude to Siksha‘O’Anusandhan Deemed to be University. Firstly, we would like to express our heartily thanks to Institute of Technical Education and Research for providing lab facility and other relevant facilities. My guide **Dr.Barnali Sahu** was the main force behind all these efforts. Because of her/his valuable suggestions and proper guidance for this project. We express our sincere thanks to the Computer Science & Engineering department HOD **Dr.Debahuti Mishra** who had allowed us to use facilities of the institute. We are also thankful to all those who have helped us in this endeavor either directly or indirectly especially all our teachers. At last we would like to express a big thank you to all friends and all known & unknown person who had helped us directly or indirectly.

Signature of students:

Place: Odisha, Bhubaneswar

CERTIFICATE

This is to certify that the project report titled “**Tic-Tac-Toe**” being submitted by **Sonali Dixit, Srestha Epari, Ritika Kumari, Yash Jhamnani** of CSE section H to the Institute of Technical Education and Research, Siksha‘O’Anusandhan Deemed to be University, Bhubaneswar for the partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering is a record of original confide work carried out by them under my/our supervision and guidance. The project work, in my/our opinion, has reached the requisite standard fulfilling the requirements for the degree of Bachelor of Technology.

The application developed for this project work have not been submitted in part or full to any other University or Institute for the award of any degree or diploma.

Dr. Barnali Sahu

Computer Science and Engineering

ITER, SOA Deemed to be University

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of students with regd. Numbers

Date: 06 Dec 2019

CONTENTS

1. SYNOPSYS

Tic-tac-toe (also known as **noughts and crosses** or **X’s and O’s**) 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.

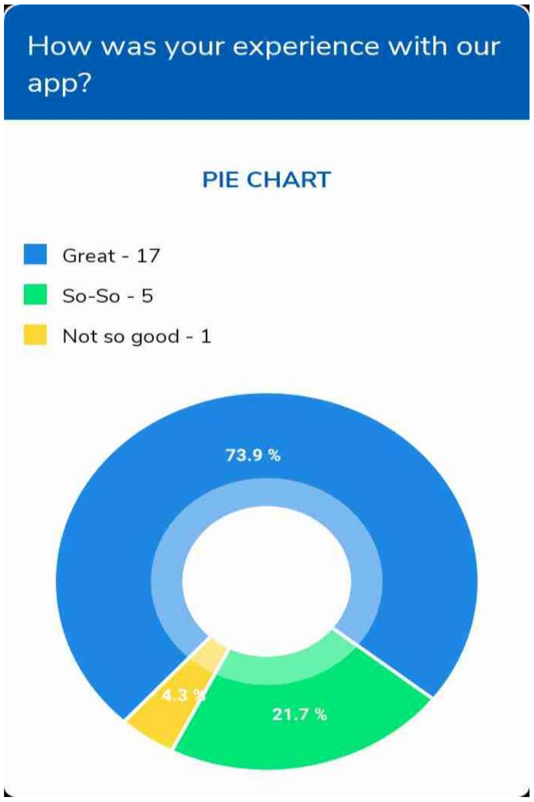
In order to solve Tic Tac Toe, we need to go deeper than just to think about it as a game where two players place X’s and O’s on the board. Formally speaking, Tic Tac Toe is a zero-sum and perfect information game. It means that each participant’s gain is equal to the other participants’ losses and we know everything about the current game state.

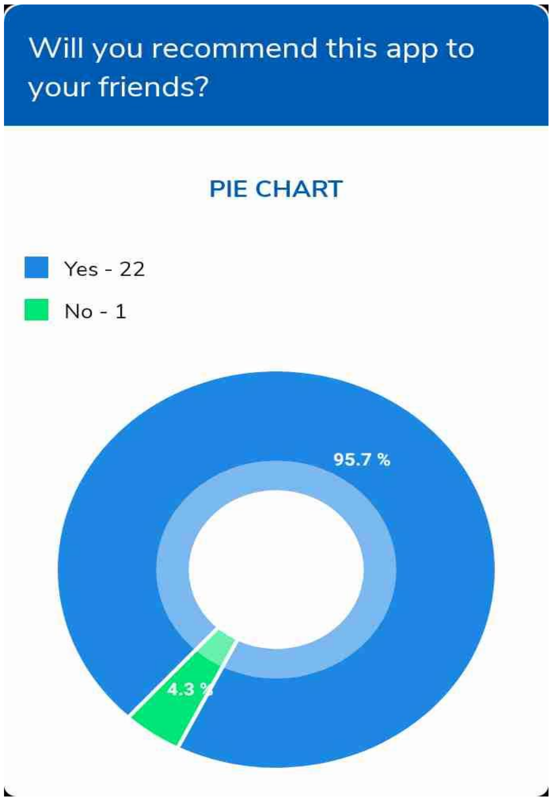
2. QUESTIONNAIRE

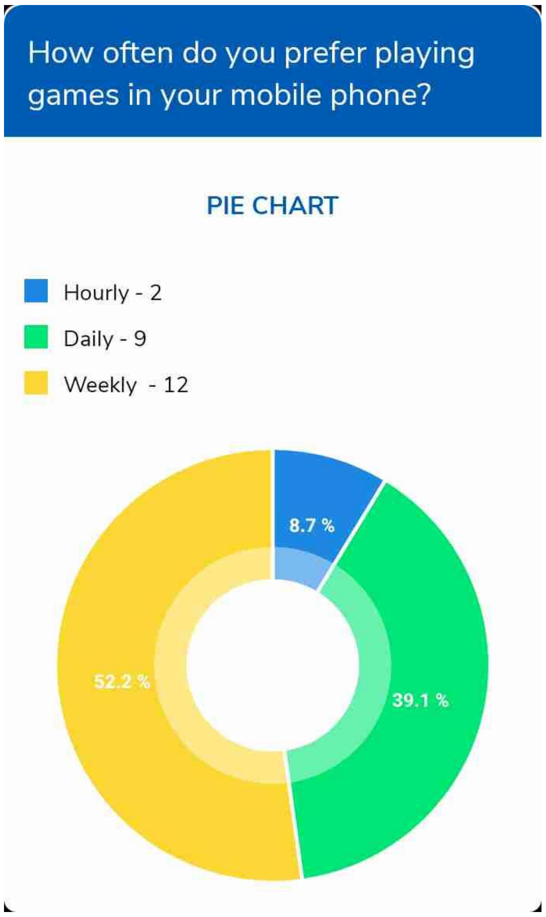
Q.No	Description
01.	How often do you prefer playing games in your mobile phone? a) Hourly b) Daily c) Weekly
02.	Do you like playing Tic-Tac-Toe? a) Yes b) No
03.	How was your experience with our app? a) Amazing b) So So c) Not Good
04.	Will you recommend this app to your friends? a) Yes b) No c) May be

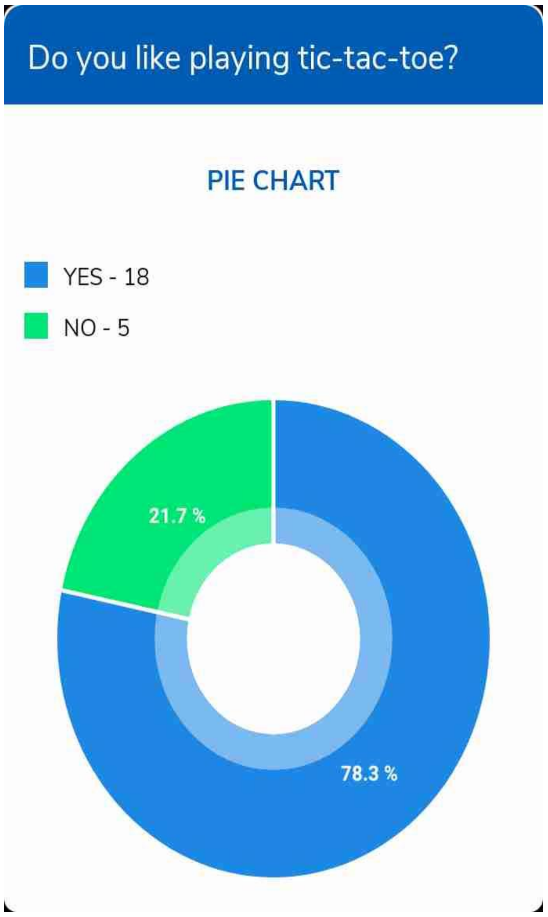
3. ANALYSIS OF THE SURVEY

The participants of the survey were quite impressed by the application presented. Showed active participation during the questionnaire and the project team was able to collect insightful information which was helpful in making the application more user-friendly. Based on the Survey, we got few excellent reviews for our app. The reviews expressed proper functioning, good user interface and no error like corrupt file was met. Moreover, people are also interested in recommending our app to their friends.






How often do you prefer playing games in your mobile phone?
PIE CHART
Hourly - 2
Daily - 9
Weekly - 12
8.7 %
39.1 %
52.2 %


Do you like playing tic-tac-toe?
PIE CHART
YES - 18
NO - 5
21.7 %
78.3 %

4. SRS DOCUMENT

4.1 INTRODUCTION

There has been interest in designing computer algorithms to play common games since the early advent of the modern digital computer. With a few exceptions, these efforts have relied on domain-specific information programmed into an algorithm in the form of weighted features that were believed to be important for assessing the relative worth of alternative positions in the game. That is, the programs relied on human expertise to defeat human expertise. Every “item” of knowledge was pre-programmed. In some cases, the programs even tuned to defeat particular human opponents, indicating their brittle nature. The fact that they require human expertise a priori is testament to the limitation of this approach.

In our project, we demonstrate the neural evolutionary strategy capable of playing tic- tac-toe under the recommendations and extend these results by employing partitioned neural network architecture, the mixtures of expert paradigm. In the following parts, section 2 introduces the neural evolutionary strategy capable of playing tic-tac-toe.

4.1.1 PURPOSE

Tic-Tac-Toe is a two player game that takes turn marking spaces on a 3 by 3 grid,and the objective of the game is to place three connecting marks in a horizontal, vertical, or diagonal row. Tic Tac Toe, also known as "Noughts and Crosses" or "X's and O's", is a solved game. This means there is a known, mathematically proven strategy to follow for the best result each game. In Tic Tac Toe, two players who follow the right strategy will always tie, with neither player winning. The motive of the game is to entertain oneself in free time. Technology is replacing manual work with every new day. Similarly, our leisure time pass is also getting digitised. Our app is a representation of new tech.

4.1.2 SCOPE

The game play will be simple. There will be a simple square game board divided into nine tiles or grid spaces. When the player clicks on one of the grid spaces, it will be assigned either an "X" or an "O". The game is over when one player claims 3 grid spaces in a row or there are no moves left. The game will have a small amount of polish to make it complete. At the start of the game, the board will not be active until the first player has chosen whether they are to play "X" or "O". A panel will indicate whose turn it is. When the game is over, a banner will display the winner or announce a draw if no one wins. A restart button will be displayed when the game is over, returning the game to the starting state when clicked.

The game will need a few basic elements.

- A background providing a backdrop for the entire game.
- An element that will be our game board.
- An element, or set of elements, that breaks the game board up into nine areas in an even grid.
- Nine tiles that can be assigned either an "X" or and "O", but once assigned these values will persist and not be changeable by the players - either the current player or the opponent.
- Logic to change sides when a player takes their turn.
- Logic to check for a "Win" condition, allowing for draws where no one wins.
- A panel that displays who is the winner when the game is over.

4.1.3 DEFINITION, ACRONYM AND ABBREVIATIONS1

SRS:	Software Requirement Specification
API:	Application Programming Interface
UML:	Unified modeling Language
XML:	Extensible Markup Language
GUI:	Graphic User Interface
JDK:	Java Development Kit

4.2 THE OVERALL DESCRIPTION

Tic-tac-toe is a game between two players, **O** and **X**, who alternate in marking the spaces in a 3×3 board. A player wins by getting three of their own marks in a horizontal, vertical or diagonal row. A game is drawn if all nine squares contain a mark and no player has three marks in a row.

A Tic Tac Toe variation conforms to the following general pattern:

- The Board is composed of a 3 x 3 board
- There are two players, **X** and **O**. **X** always has the first turn

- A turn consists of a move that updates the board state in some manner [including "no change"]
- After each move, the game result is evaluated. Once the game result is decided (either X wins, O wins, or draw) no more moves are allowed.

4.2.1 PRODUCT PERSPECTIVE

The application will be an android based, self-contained and independent software product.

4.2.2 Hardware Interfaces

System: Intel Dual Core

Hard Disk: 40 GB

Ram: 8GB

I/O devices: Laptop

Android phone

Software Interfaces

Operating System: Window 10

Software Tools: JDK 1.7, Eclipse, SDK, Android Studio

4.2.3 Product Functions

Tic-tac-toe is a gaming application for two players where one player wins or there is a draw.

• Start Game

• Player moves

• Reset Game

• Background Music

• End Game

4.2.4 User Characteristics

The user should be familiar with using android device. The user is expected handle hardware and software specifications.

4.3 REQUIREMENTS

4.3.1 User Interface

The users interact with the app through the screen, opening which the game board is displayed. Other than this, the name of the players, scores, reset button and the music in the background also starts to play.

4.3.2 Hardware Interfaces

System: Intel Dual Core

Hard Disk: 40 GB

Ram: 8GB

I/O devices: Laptop

Android phone

4.3.3 Software Interfaces

Operating System: Window 10

Software Tools: JDK 1.7, Eclipse, SDK, Android Studio

4.3.4 Functional Requirements

Platform Independence – The program will use java and is platform is independent. Start ,Exit and Minimize the application – The user invokes the application from the os with the aid of JVM. If the user exit from file the program will terminates. If the user minimize, the window will be minimized.

4.3.5 Non-functional Requirements

Performance -The game performs well without any lag. It will calculate the efficiency based on the code and give the output.

Usability - The game is easy even for the new players. The game is easily understandable how to navigate through interface built.

Reliability - The application is developed in java and is hence robust.

Supportability - The application can be developed in any operating system as java is platform independence.

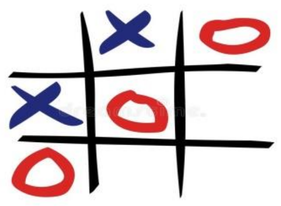
4.3.6 Performance Requirements

The only way in which systems will meet their performance targets is for them to be specified clearly and unambiguously. It is a simple fact that if performance is not a stated criterion of the system requirements then the system designers will generally not consider performance issues. While loose or incorrectly defined performance specifications can lead to disputes between clients and suppliers. In many cases performance requirements are never ridged as system that does not fully meet its defined performance requirements may still be released as other consideration such as time to market.

In order to assess the performance of a system the following must be clearly specified:
•Response-Time
•Workload
•Scalability
• Platform

4.3.7 Design Constraints

A move is guaranteed to be valid and is placed on an empty block.
Once a winning condition is reached, no more moves is allowed.
A player who succeeds in placing n of their marks in a horizontal, vertical, or diagonal row wins the game.



4.3.8 Security

The user who installs the app can play the game. Only the developers have the authority to make any change in the app.

4.3.9 Maintainability

When developing a website one aspect that should not be overlooked is the ease of maintenance of that website. There are always bound to be design changes, bug fixes and display issues that arise at some point. The time taken to fix these issues often has some relation to the already existing codebase.

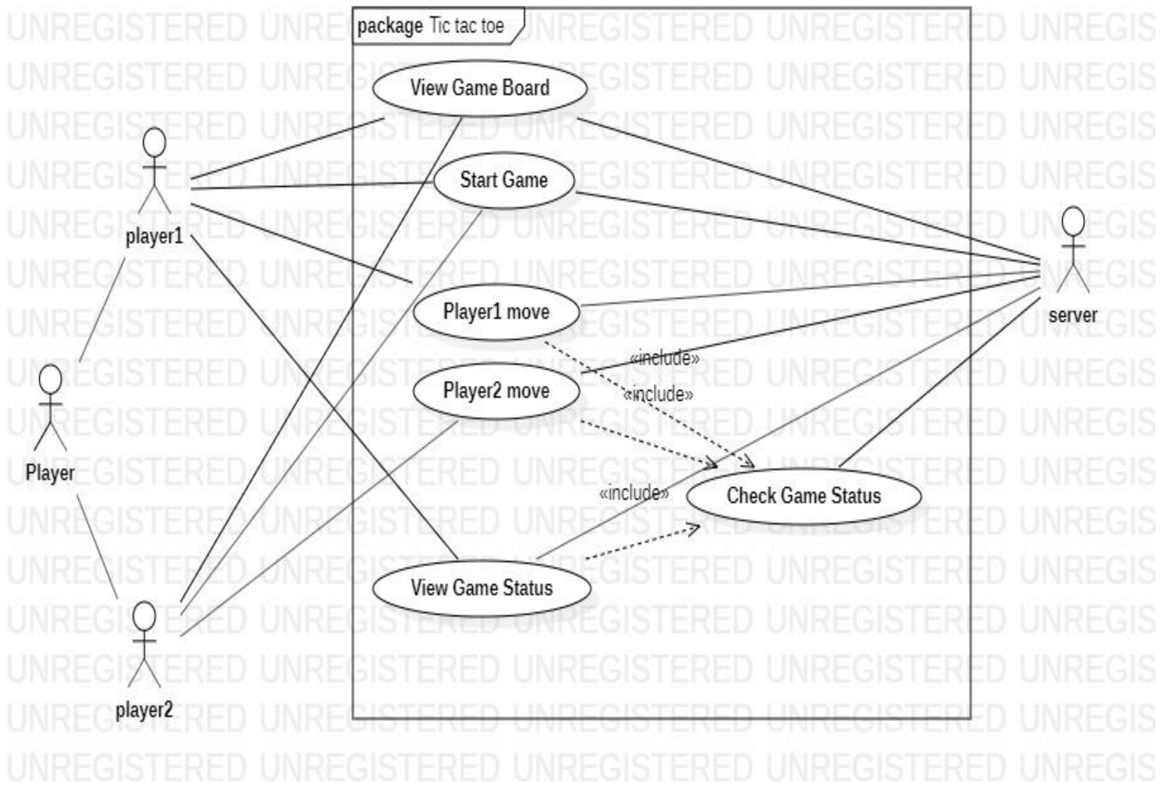
While studying at university I went to see a guest speaker who said as soon as you write a line of code that line becomes legacy code and will probably need to be maintained. It is an interesting idea and really drives the point to write clean maintainable code. Over the years of developers have come up with a variety of systems, models and technologies to solve this issue.

4.3.10 Portability

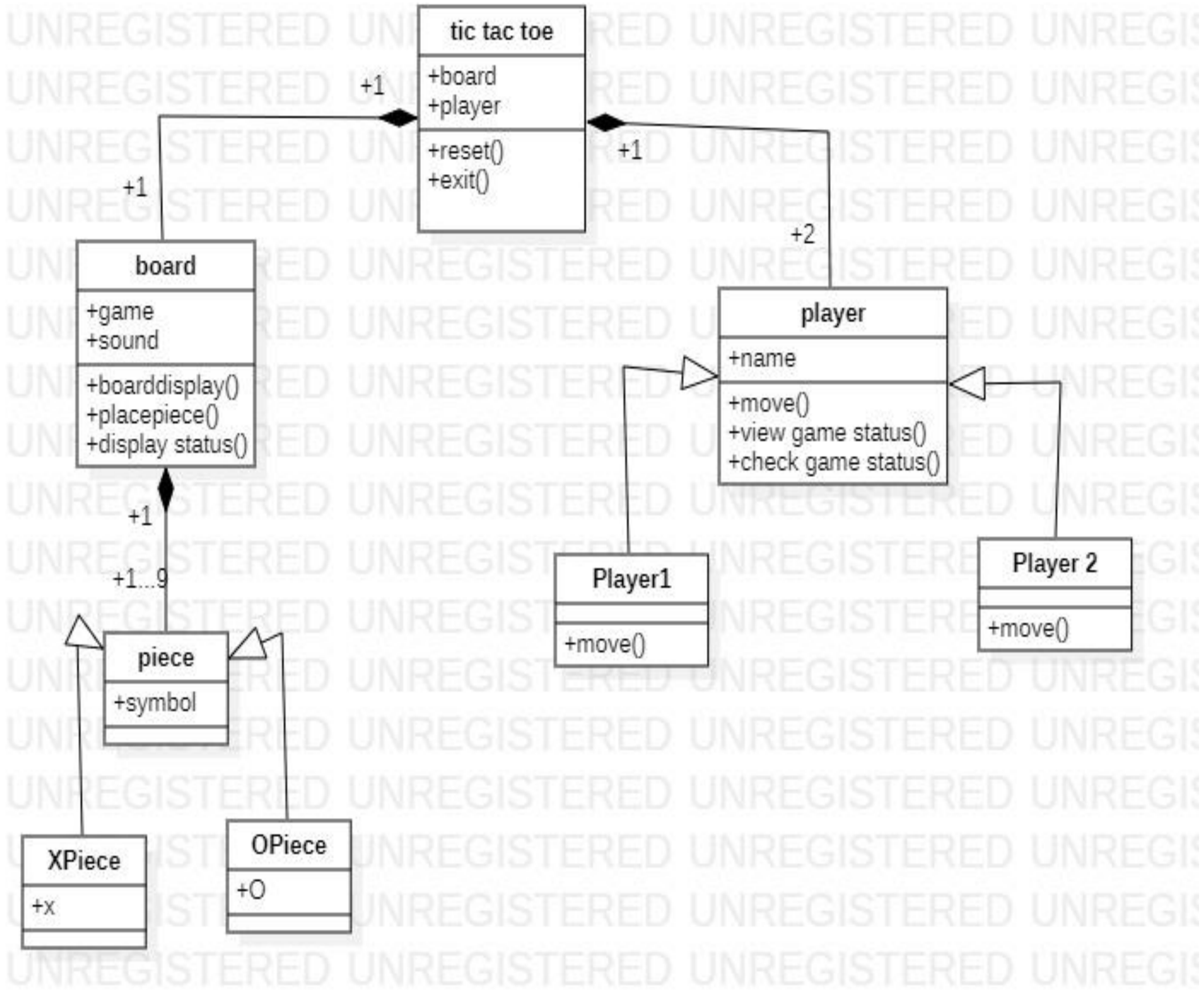
Portability has some work when moving an application program to another operating system. Recently, the Java programming language and runtime environment has made it possible to have programs that run on any operating system that supports the Java standard (from Sun Microsystems) without any porting work. Java applets in the form of precompiled byte code can be sent from a server program in one operating system to a client program (your Web browser) in another operating system without change.

5. UML Diagram

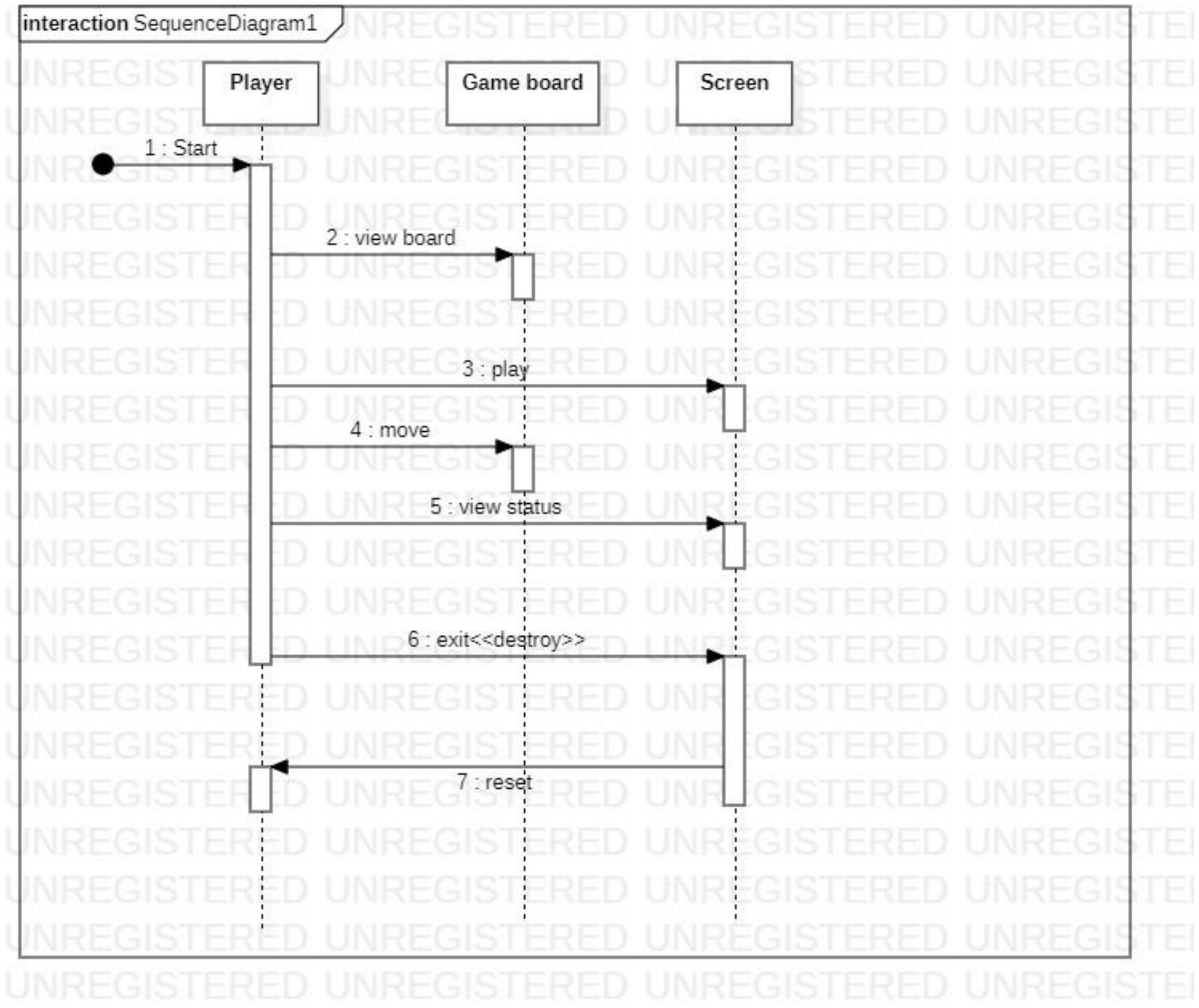
5.1 Use Case Diagram:



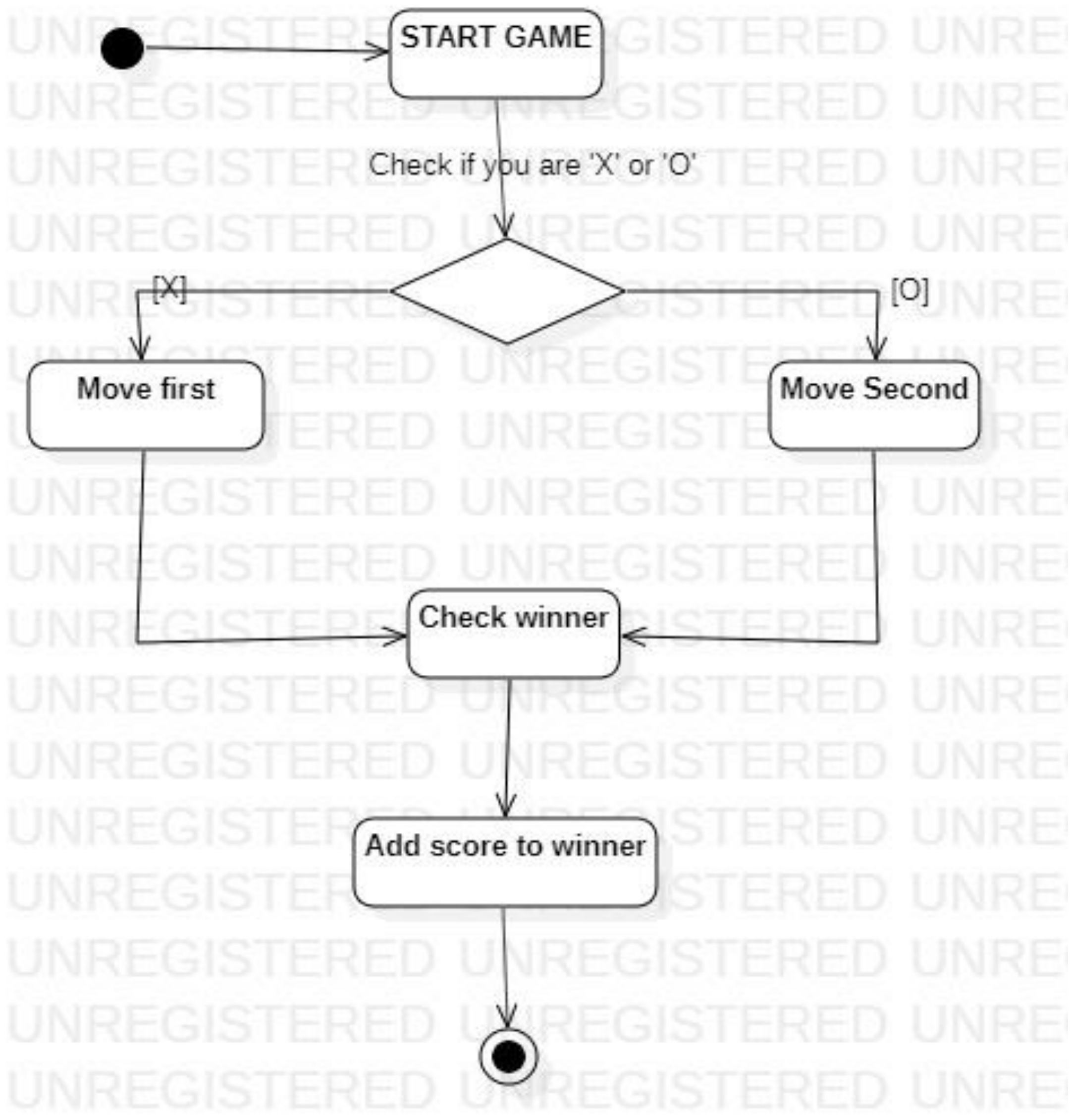
5.2 Class Diagram:



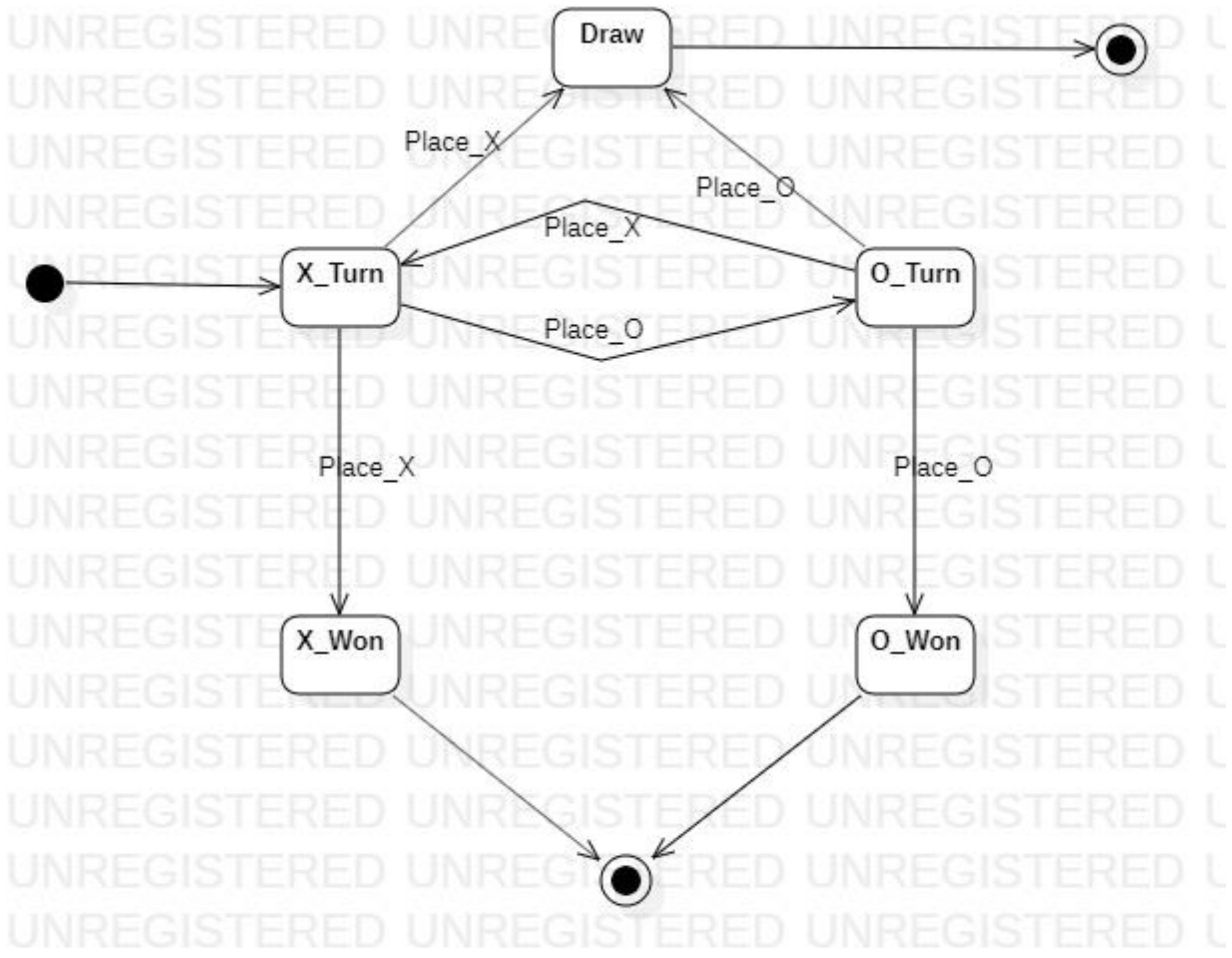
5.3 Sequence Diagram:



5.4 Activity Diagram:



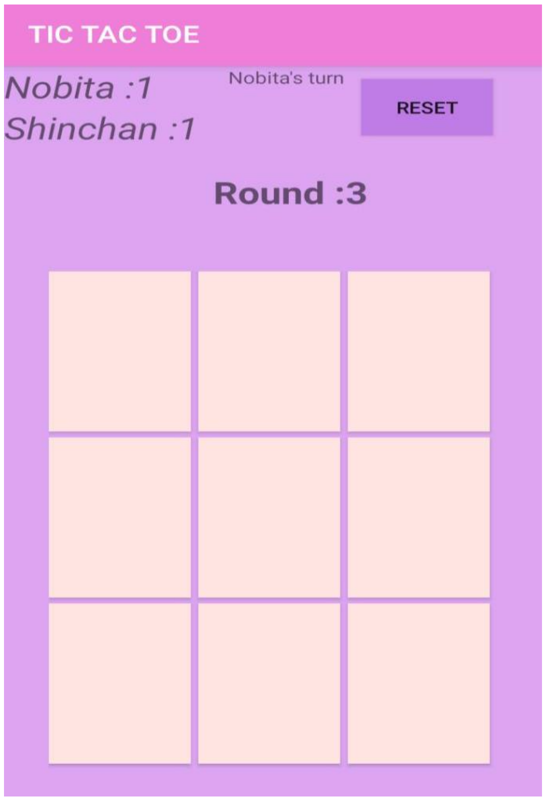
5.5 State Diagram:



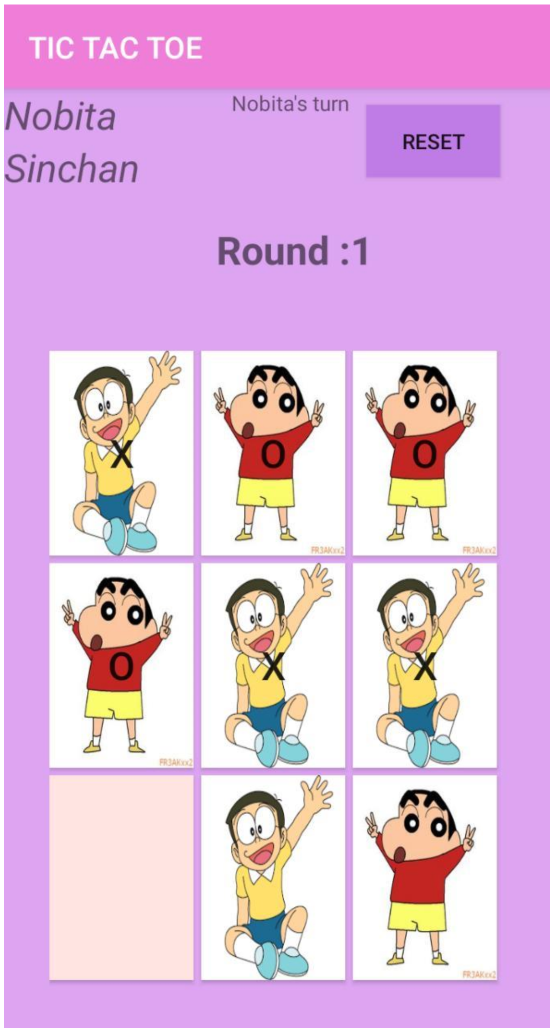
6. UI (user-interface)

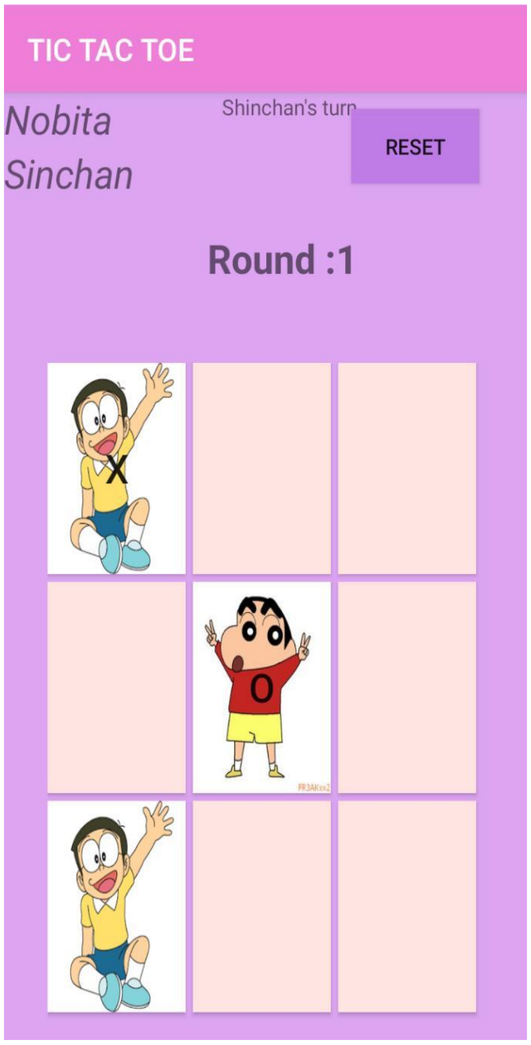
6.1 Layout Designs

Home screen:

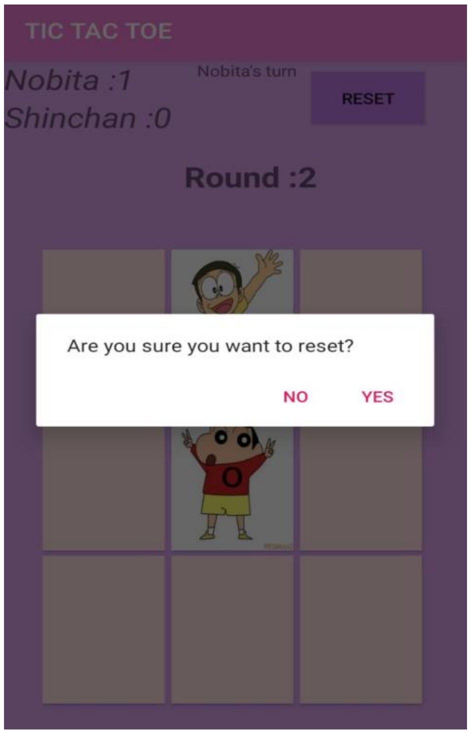


Notification for the next player's move**:**





Reset:


TIC TAC TOE

Nobita :1

Shinchan :0

Nobita's turn

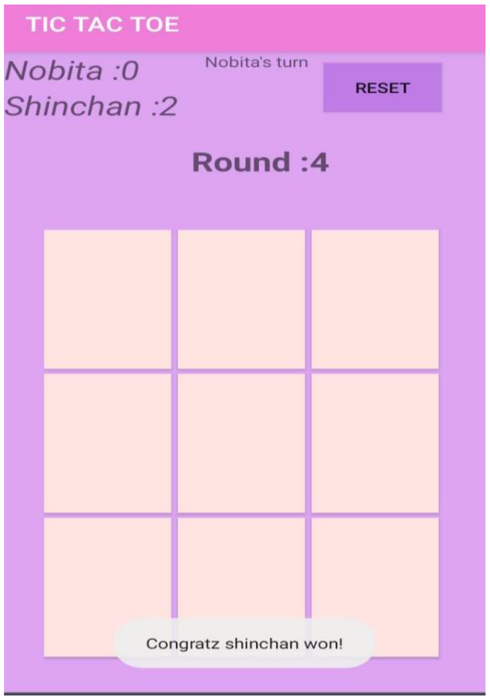
RESET

Round :2

Are you sure you want to reset?

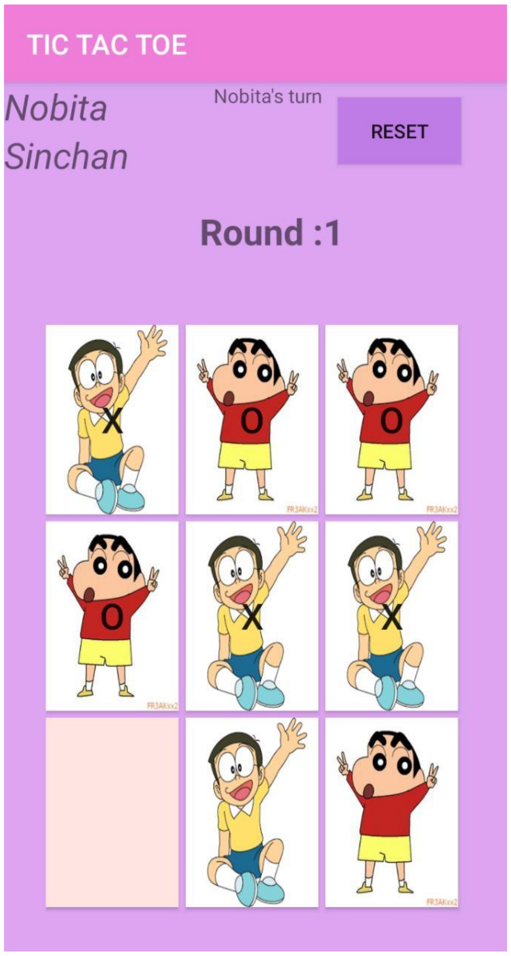
NO YES

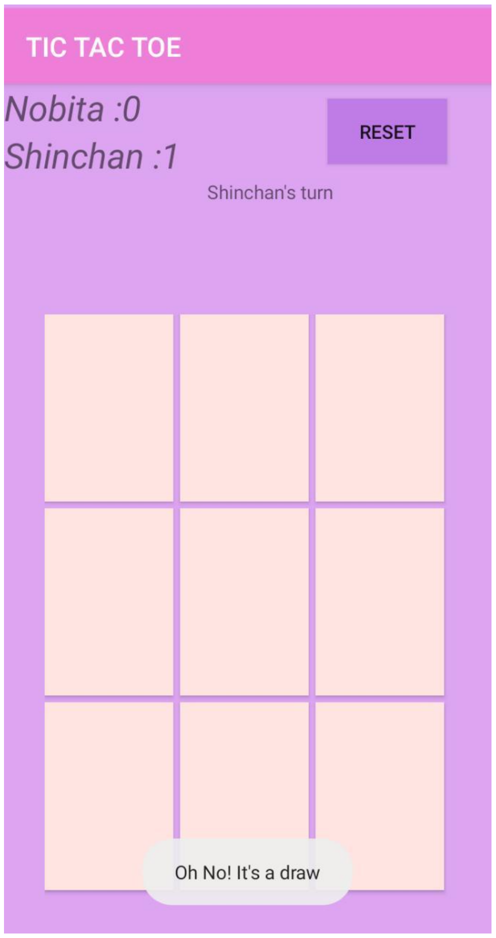
Displaying who won:



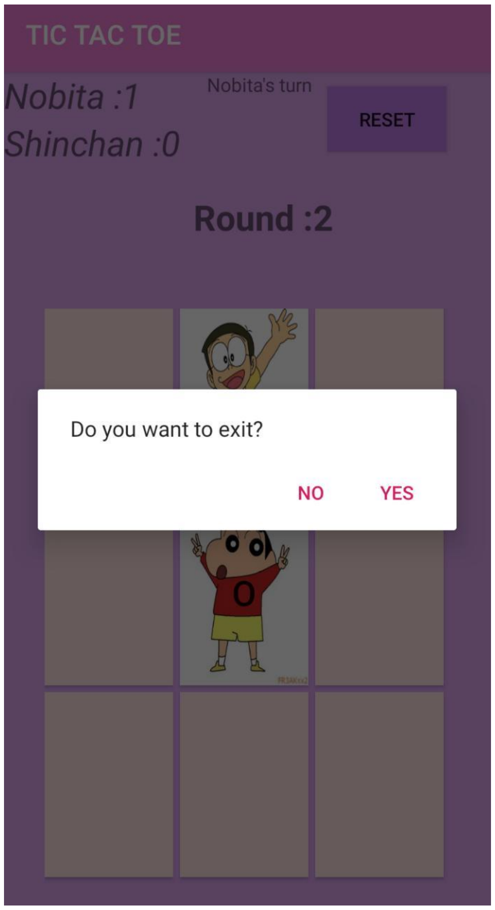
Nobita :0
Shinchan :2
Nobita's turn
Round :4
Congratz shinchan won!

Draw:


TIC TAC TOE
Nobita
Sinchan
Nobita's turn
RESET
Round :1
X
O
O
O
X
X


TIC TAC TOE
Nobita :0
Shinchan :1
RESET
Shinchan's turn
Oh No! It's a draw

Exit:



Do you want to exit?

7. PROGRAM CODE

XML File:

```xml
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#dca4f1"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:id="@+id/rel"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/tv1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:freezesText="true"
            android:text="Nobita"
            android:textStyle="italic"
            android:textSize="26sp" />

        <TextView
            android:id="@+id/tv2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tv1"
            android:freezesText="true"
            android:text="Sinchan"
            android:textStyle="italic"
            android:textSize="26sp" />

        <Button
            android:id="@+id/btnreset"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_centerVertical="true"
            android:background="#bf7be6"
            android:layout_marginEnd="33dp"
            android:text="reset" />
        <TextView
            android:id="@+id/display"
            android:layout_width="wrap_content"
            android:layout_height="50dp"
            android:layout_marginLeft="150dp"
            android:text=""
            tools:layout_height="wrap_content"
            tools:layout_width="wrap_content"
            tools:textColor="#ff0" />
```

```
</RelativeLayout>



<TextView
    android:id="@+id/round"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/rel"
    android:textStyle="bold"
    android:textSize="26sp"
    android:text="Round :1"
    android:textAlignment="center"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="140dp"
    tools:layout_marginLeft="140dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"

    android:layout_marginLeft="30dp"
    android:layout_marginTop="50dp"
    android:layout_marginRight="30dp"
    android:layout_weight="1">

    <Button
        android:id="@+id/button_00"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginRight="5dp"

        android:layout_marginBottom="5dp"
        android:layout_weight="1"
        android:background="#ffe4e1"
        android:freezesText="true"
        android:textSize="26sp" />


    <Button
        android:id="@+id/button_01"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginRight="5dp"
        android:layout_marginBottom="5dp"
        android:layout_weight="1"
        android:background="#ffe4e1"
        android:freezesText="true"
        android:textSize="26sp" />

    <Button
        android:id="@+id/button_02"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginRight="5dp"
        android:layout_marginBottom="5dp"
```

```
            android:layout_weight="1"
            android:background="#ffe4e1"
            android:freezesText="true"
            android:textSize="26sp" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"
        android:layout_weight="1">

        <Button
            android:id="@+id/button_10"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_marginRight="5dp"
            android:layout_marginBottom="5dp"
            android:layout_weight="1"
            android:background="#ffe4e1"
            android:freezesText="true"
            android:textSize="26sp" />

        <Button
            android:id="@+id/button_11"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_marginRight="5dp"
            android:layout_marginBottom="5dp"
            android:layout_weight="1"
            android:background="#ffe4e1"
            android:freezesText="true"
            android:textSize="26sp" />

        <Button
            android:id="@+id/button_12"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_marginRight="5dp"
            android:layout_marginBottom="5dp"
            android:layout_weight="1"
            android:background="#ffe4e1"
            android:freezesText="true"
            android:textSize="26sp" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"
        android:layout_marginBottom="30dp"
        android:layout_weight="1">

        <Button
```

```
            android:id="@+id/button_20"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#ffe4e1"
            android:layout_marginBottom="5dp"
            android:layout_marginRight="5dp"
            android:freezesText="true"
            android:textSize="0sp"/>

        <Button
            android:id="@+id/button_21"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#ffe4e1"
            android:freezesText="true"
            android:layout_marginBottom="5dp"
            android:layout_marginRight="5dp"

            android:textSize="0sp"/>

        <Button
            android:id="@+id/button_22"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:layout_marginBottom="5dp"
            android:background="#ffe4e1"
            android:freezesText="true"
            android:layout_marginRight="5dp"
            android:textSize="0sp"/>

    </LinearLayout>

</LinearLayout>
```

Java file:

```
package com.example.yashjhamnani.tictactoe;




import android.annotation.TargetApi;
import android.app.Dialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Build;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.support.annotation.RequiresApi;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import android.support.v7.app.AppCompatDialog;


public class MainActivity extends AppCompatActivity {

    private Button[][] buttons = new Button[3][3];

    private boolean player1Turn = true;

    private int roundCount;
TextView tv3;
int counter=1;
    private int player1Points;
    private int player2Points;

    private TextView textViewPlayer1;
    private TextView textViewPlayer2;
    MediaPlayer ring1;

    int c;
    TextView tv4;
    //EditText user1;
     //EditText user2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {


        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
  //   alert a=new alert();
    // Dialog s=a.onCreateDialog();

       // user1=(EditText)findViewById(R.id.user1);
       c=1;
        //user2= (EditText)findViewById(R.id.user2);
        textViewPlayer1 = findViewById(R.id.tv1);
        textViewPlayer2 = findViewById(R.id.tv2);
        ring1= MediaPlayer.create(MainActivity.this,R.raw.dor);
        //ring2= MediaPlayer.create(MainActivity.this,R.raw.beep);
        tv3=(TextView)findViewById(R.id.display);
        tv3.setText("Nobita's turn");
        tv4=(TextView)findViewById(R.id.round);
        ring1.start();



        for(int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                String buttonID = "button_" + i + j;
                final int resID = getResources().getIdentifier(buttonID, "id",
getPackageName());
                buttons[i][j] = findViewById(resID);
                buttons[i][j].setOnClickListener(new View.OnClickListener() {
                    @TargetApi(Build.VERSION_CODES.JELLY_BEAN)
                    @RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
                    @Override
                    public void onClick(View v) {
                        if (!((Button) v).getText().toString().equals(""))
                        {
                            return;
                        }




                        if (player1Turn) {
                            ((Button) v).setBackgroundResource(R.drawable.nobita);
                            ((Button) v).setText("X");
                            //((Button) v).setTextColor(R.drawable.yello);

                            tv3.setText("Shinchan's turn");


                        } else {
                            ((Button) v).setBackgroundResource(R.drawable.sin);
                            ((Button )v).setText("O");
                            //((Button) v).setTextColor(R.drawable.red);

                            tv3.setText("Nobita's turn");
                        }

                        roundCount++;

                        if (checkForWin()) {
                            if (player1Turn) {
                                player1Wins();
                            } else {
                                player2Wins();
                            }
                        } else if (roundCount == 9) {
                            draw();
                        } else {
                            player1Turn = !player1Turn;
                        }

                    }
                });
            }
        }


        Button buttonReset = findViewById(R.id.btnreset);


        buttonReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this);
```

```
                builder.setMessage("Are you sure you want to reset?")
                        .setCancelable(false)
                        .setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog, int id) {
                                resetGame();
                            }
                        })
                        .setNegativeButton("No", new DialogInterface.OnClickListener()
{
                            public void onClick(DialogInterface dialog, int id) {
                                dialog.cancel();
                            }
                        });
                AlertDialog alert = builder.create();
                alert.show();

            }
        });
    }

    private boolean checkForWin() {
        String[][] field = new String[3][3];

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                field[i][j] = buttons[i][j].getText().toString();
            }
        }

        for (int i = 0; i < 3; i++) {
            if (field[i][0].equals(field[i][1])
                    && field[i][0].equals(field[i][2])
                    && !field[i][0].equals("")) {
                return true;
            }
        }

        for (int i = 0; i < 3; i++) {
            if (field[0][i].equals(field[1][i])
                    && field[0][i].equals(field[2][i])
                    && !field[0][i].equals("")) {
                return true;
            }
        }

        if (field[0][0].equals(field[1][1])
                && field[0][0].equals(field[2][2])
                && !field[0][0].equals("")) {
            return true;
        }

        if (field[0][2].equals(field[1][1])
                && field[0][2].equals(field[2][0])
                && !field[0][2].equals("")) {
            return true;
        }
        return false;
    }

    private void player1Wins() {
        player1Points++;
        Toast.makeText(this, "Congratz Nobita won!", Toast.LENGTH_SHORT).show();
        ring1.pause();

        CountDownTimer cntr_aCounter = new CountDownTimer(2000, 2000) {
            MediaPlayer ring = MediaPlayer.create(MainActivity.this, R.raw.osin);


            @Override
            public void onTick(long millisUntilFinished) {
                ring.start();
            }

            public void onFinish() {
                //code fire after finish
                ring.stop();
            }

        };cntr_aCounter.start();

        updatePointsText();
        resetBoard();

    }




    private void player2Wins() {
        player2Points++;
        Toast.makeText(this, "Congratz shinchan won!", Toast.LENGTH_SHORT).show();
       ring1.pause();

        CountDownTimer cntr_aCounter = new CountDownTimer(2000, 2000) {
            MediaPlayer ring = MediaPlayer.create(MainActivity.this, R.raw.osin);


            @Override
            public void onTick(long millisUntilFinished) {
                ring.start();
            }

            public void onFinish() {
                //code fire after finish
                ring.stop();
            }

        };cntr_aCounter.start();

        updatePointsText();
        resetBoard();

    }
```

```java
private void draw() {
    Toast.makeText(this, "Oh No! It's a draw", Toast.LENGTH_SHORT).show();
    resetBoard();
}

private void updatePointsText() {
    textViewPlayer1.setText("Nobita :" + player1Points);
    textViewPlayer2.setText("Shinchan :" + player2Points);
}

private void resetBoard() {
    c++;
    counter++;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            buttons[i][j].setText("");
            buttons[i][j].setBackgroundResource(R.drawable.ffe4e1);
        }
    }
    tv4.setText("Round :"+counter);
    if(c%2==0)
    {
        ring1.stop();
        ring1= MediaPlayer.create(MainActivity.this,R.raw.sintune);
        ring1.seekTo(0);
        ring1.start();
    }
    else
    {
        ring1.stop();
        ring1= MediaPlayer.create(MainActivity.this,R.raw.dor);
        ring1.seekTo(0);
        ring1.start();
    }


    roundCount = 0;
    player1Turn = true;
    tv3.setText("Nobita's turn");

}

private void resetGame() {
    player1Points = 0;
    player2Points = 0;
    tv3.setText("Nobita's turn");
    tv4.setText("Round :1");
    counter=0;
    updatePointsText();
    resetBoard();
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    outState.putInt("roundCount", roundCount);
    outState.putInt("player1Points", player1Points);
```

```
        outState.putInt("player2Points", player2Points);
        outState.putBoolean("player1Turn", player1Turn);
    }

    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
        super.onRestoreInstanceState(savedInstanceState);

        roundCount = savedInstanceState.getInt("roundCount");
        player1Points = savedInstanceState.getInt("player1Points");
        player2Points = savedInstanceState.getInt("player2Points");
        player1Turn = savedInstanceState.getBoolean("player1Turn");

    }
    public void onBackPressed() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setMessage("Do you want to exit?")
                .setCancelable(false)
                .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        ring1.stop();
                        MainActivity.this.finish();
                    }
                })
                .setNegativeButton("No", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        dialog.cancel();
                    }
                });
        AlertDialog alert = builder.create();
        alert.show();

    }
}
```

8. TESTING

Testing is a process, which reveals errors in the program. Once the source code has been generated, the software must be tested to uncover as many errors as possible before delivery to the customer. Software testing can be stated as the process of validating and verifying that a computer program/application/product:

(a) meets the requirements that guided its design and development,(b) works as expected, (c)can be implemented with the same characteristics, (d) satisfies the needs of stakeholders.The developed system was tested on the following platforms-

1. MAC OS- The app failed as it is developed in android.

2. Android OS- The app successfully runs on the latest android devices i.e. android, pie, Oreo, etc.

9. CONCLUSION AND FUTURE WORK

Tic-Tac-Toe helps children apply their logic and develop strategy at an early age. It prepares children for more complex games because they have to think of multiple things at one time. Tic-tac-toe helps develop coordination, fine motor skills and visual skills. It is an easy and popular game which is played all over the world.

In future we are planning to add features like database storage in which names of different players and their previous score records will be stored. Introducing AI in the system will enable single player or multiple player modes.

REFERENCES

https://www.stackoverflow.com

https://arxiv.org/ftp/arxiv/papers/1406/1406.5177.pdf

https://www.researchgate.net/publication/316534681_Tic-Tac-Toe