OPEN-LIBRARY

PROJECT REPORT

FOR THE FULLMINT OF Vth SEMESTER

BACHELOR OF COMPUTER APPLICATIONS

(2022-2025)

SUBMITTED BY

NAME: Ashish Kothari

Roll No.: 2221283 (17)

Under the guidance of

DR. RK BISHT

Designation: Professor

SCHOOL OF COMPUTING

GRAPHIC ERA HILL UNIVERSITY

05 / 12 / 2024

CERTIFICATE

This is to certify that the thesis titled **"Tic Tac Toe" submitted** by **Ashish Kothari** to Graphic Era Hill University for the award of the degree of **Bachelor's of Computer Applications**, is a bona fide record of the research work done by her under our supervision.

The contents of this project in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Dehradun
Date: 5-12-2024

DR. RK BISHT
Project Guide
Professor
GEHU, Dehradun

ACKNOWLEDGEMENT

I convey my sincere gratitude for giving the opportunity to prepare the project work in Tic Tac Toe android game. I express my sincere thanks to the members of School of Computing.

I am thankful to Dr. RK Bisht for his guidance during my project work and sparing his valuable time for the same.

Roll No: 2221283

Abstract

This Android application presents a classic Tic Tac Toe game, designed to provide a fun and engaging experience for users of all ages. Developed using Java and XML, the app offers a simple and intuitive interface, allowing players to easily engage in single-player or two-player matches. Key features include:

- **Classic Gameplay:** Adherence to the traditional rules of Tic Tac Toe.
- **User-Friendly Interface:** A clean and intuitive design for seamless gameplay.
- **Two-Player Mode:** Local multiplayer option for competitive fun.

This project demonstrates the effective application of Android development principles to create a timeless game, providing a platform for strategic thinking and friendly competition.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1. Introduction

1.1 Project Overview This document outlines the design and implementation of a Tic Tac Toe Android game. The game allows two players to play against each other on a 3x3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.

1.2 Problem Statement The problem is to create a simple, yet engaging, Android game that can be played by two players on a single device.

1.3 Project Objectives The primary objective is to develop a functional Tic Tac Toe game for Android devices. Specific objectives include:

- Create a user-friendly interface.
- Implement the core game logic.
- Test and debug the application.
- Ensure smooth gameplay and performance.

1.4 Target Audience The target audience for this game includes individuals of all ages who enjoy casual games.

1.5 Scope of the Project The scope of this project is limited to a basic Tic Tac Toe game with two-player mode. Future enhancements may include AI opponents, different board sizes, and additional game modes.

CHAPTER 2

SYSTEM ANALYSIS

2. System Analysis

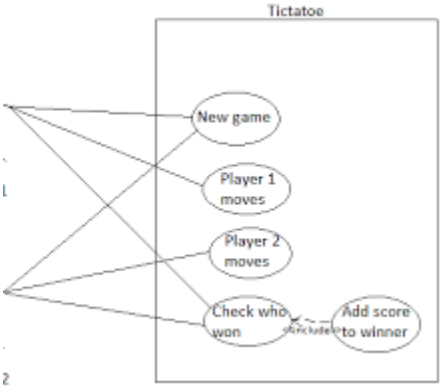
2.1 Functional Requirements

- Display a 3x3 game board.
- Allow two players to take turns.
- Detect winning conditions (horizontal, vertical, diagonal).
- Detect a draw condition.
- Display a message indicating the winner or a draw.
- Reset the game board for a new game.

2.2 Non-functional Requirements

- User-friendly interface.
- Responsive design.
- Fast performance.
- Error handling and robustness.

2.3 Use Case Diagram



CHAPTER 3

SYSTEM DESIGN

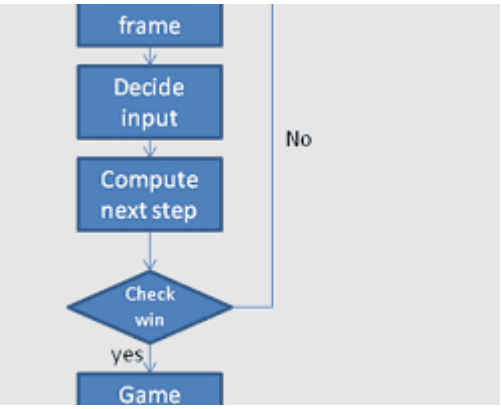
3. System Design

3.1 Architectural Design

The application will follow a simple MVC (Model-View-Controller) architecture.

- Model: Handles the game logic, including the game state and the rules.
- View: Renders the user interface, including the game board and buttons.
- Controller: Handles user input and updates the model and view accordingly.

3.2 Data Flow Diagram



Class Diagram for Tic Tac Toe

3.3 Class Diagram

3.4 User Interface Design

3.4.1 Main Menu

- Title of the game
- Play button
- Exit button
- 3.4.2 Game Board
- 3x3 grid of cells

- Player turn indicator
- 3.5 Win/Loss Screen
- Winner/loser message
- Play again button

CHAPTER 4

IMPLEMENTATION

4. Implementation

4.1 Development Environment

- Android Studio
- Java
- XML

4.2 Code Structure

- Activity Layout: Defines the user interface using XML layouts.
- Game Logic: Implements the core game logic, including turn management, win/loss detection, and board updates.

APPENDIX (CODE)

activity_splash.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/splash_bg_color"
    android:orientation="vertical">

   <ImageView
      android:layout_width="120dp"
      android:layout_height="120dp"
      android:contentDescription="@string/image"
      android:src="@drawable/tic_tac_toe"
      android:layout_gravity="center_horizontal"
      android:layout_marginTop="70dp"
      />

   <TextView
      android:id="@+id/splash_screen"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/tic_tac_toe"
      android:textSize="60sp"
      android:fontFamily="@font/splash_font"
      android:textAppearance="@font/splash_font"
      android:layout_gravity="center_horizontal"
      android:layout_marginVertical="150dp"/>

</LinearLayout>
```

SplashActivity.java

```
package com.example.tictactoe;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class SplashActivity extends AppCompatActivity {
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_splash);
      Intent game = new Intent(SplashActivity.this, MainActivity.class);
      new Handler().postDelayed(new Runnable() {
         @Override
         public void run() {
            startActivity(game);
            finish();
         }
      }, 500);

   }
}
```

6.1 Splash Activity

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical"
    >

    <ImageView
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:contentDescription="@string/image"
        android:src="@drawable/tic_tac_toe"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="50dp"
/>

    <GridLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:columnCount="3"
        tools:ignore="UselessParent"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="160dp">

        <Button android:id="@+id/btn1"
            android:background="@drawable/square_button"
            android:onClick="check"
            android:layout_marginBottom="5dp"
            android:backgroundTint="@color/button_color"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:textSize="30sp"/>
        <Button android:id="@+id/btn2"
            android:background="@drawable/square_button"
            android:layout_marginStart="4dp"
            android:onClick="check"
            android:layout_marginBottom="5dp"
            android:backgroundTint="@color/button_color"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:textSize="30sp"/>
        <Button android:id="@+id/btn3"
            android:background="@drawable/square_button"
            android:layout_marginStart="5dp"
            android:layout_marginBottom="5dp"
            android:onClick="check"
            android:backgroundTint="@color/button_color"
            android:layout_width="80dp"
```

```
        android:layout_height="80dp"
        android:textSize="30sp"/>
    <Button android:id="@+id/btn4"
        android:background="@drawable/square_button"
        android:layout_marginBottom="5dp"
        android:onClick="check"
        android:backgroundTint="@color/button_color"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:textSize="30sp"/>
    <Button android:id="@+id/btn5"
        android:background="@drawable/square_button"
        android:layout_marginStart="5dp"
        android:layout_marginBottom="5dp"
        android:onClick="check"
        android:backgroundTint="@color/button_color"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:textSize="30sp"/>
    <Button android:id="@+id/btn6"
        android:background="@drawable/square_button"
        android:layout_marginStart="5dp"
        android:layout_marginBottom="5dp"
        android:onClick="check"
        android:backgroundTint="@color/button_color"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:textSize="30sp"/>
    <Button android:id="@+id/btn7"
        android:background="@drawable/square_button"
        android:onClick="check"
        android:backgroundTint="@color/button_color"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:textSize="30sp"/>
    <Button android:id="@+id/btn8"
        android:background="@drawable/square_button"
        android:layout_marginStart="5dp"
        android:onClick="check"
        android:backgroundTint="@color/button_color"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:textSize="30sp"/>
    <Button android:id="@+id/btn9"
        android:background="@drawable/square_button"
        android:layout_marginStart="5dp"
        android:onClick="check"
        android:backgroundTint="@color/button_color"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:textSize="30sp"
        />
</GridLayout>

<Button
    android:id="@+id/replay_button"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/replay"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="100dp"
        android:backgroundTint="@color/button_restart"
        android:padding="15dp"
        />

</LinearLayout>
```

MainActivity.java

```
package com.example.tictactoe;

import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    int flag = 0;
    int count = 0;
    String b1, b2, b3, b4, b5, b6, b7, b8, b9;
    Handler handler = new Handler(Looper.getMainLooper());

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button replay = findViewById(R.id.replay_button);
        replay.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                restart();
            }
        });
    }

    public void check(View view)
    {
        Button btn1 = findViewById(R.id.btn1);
        Button btn2 = findViewById(R.id.btn2);
        Button btn3 = findViewById(R.id.btn3);
        Button btn4 = findViewById(R.id.btn4);
        Button btn5 = findViewById(R.id.btn5);
        Button btn6 = findViewById(R.id.btn6);
        Button btn7 = findViewById(R.id.btn7);
        Button btn8 = findViewById(R.id.btn8);
        Button btn9 = findViewById(R.id.btn9);
        Button currentBtn = (Button) view;
        if (currentBtn.getText().toString().isEmpty()) {
            if (flag == 0) {
                currentBtn.setText("X");
                flag = flag + 1;
            } else {
                currentBtn.setText("O");
                flag = flag - 1;
```

```
}
count++;
if (count >= 5) {
    b1 = btn1.getText().toString();
    b2 = btn2.getText().toString();
    b3 = btn3.getText().toString();
    b4 = btn4.getText().toString();
    b5 = btn5.getText().toString();
    b6 = btn6.getText().toString();
    b7 = btn7.getText().toString();
    b8 = btn8.getText().toString();
    b9 = btn9.getText().toString();

    if (!b1.isEmpty() && b1.equals(b2) && b2.equals(b3)) {
        Toast.makeText(this, b1 + " is winner", Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                restart();
            }
        }, 1000);

    } else if (!b4.isEmpty() && b4.equals(b5) && b5.equals(b6)) {
        Toast.makeText(this, b4 + " is winner" , Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                restart();
            }
        }, 1000);
    } else if (!b7.isEmpty() && b7.equals(b8) && b8.equals(b9)) {
        Toast.makeText(this, b7 + " is winner", Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                restart();
            }
        }, 1000);
    } else if (!b1.isEmpty() && b1.equals(b4) && b4.equals(b7)) {
        Toast.makeText(this, b1 + " is winner", Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                restart();
            }
        }, 1000);
    } else if (!b2.isEmpty() && b2.equals(b5) && b5.equals(b8)) {
        Toast.makeText(this, b2 + " is winner", Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
```

```
                restart();
            }
        }, 1000);
    } else if (!b3.isEmpty() && b3.equals(b6) && b6.equals(b9)) {
        Toast.makeText(this, b3 + " is winner", Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                restart();
            }
        }, 1000);
    } else if (!b1.isEmpty() && b1.equals(b5) && b5.equals(b9)) {
        Toast.makeText(this, b1 + " is winner", Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                restart();
            }
        }, 1000);
    } else if (!b3.isEmpty() && b3.equals(b5) && b5.equals(b7)) {
        Toast.makeText(this, b3 + " is winner", Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                restart();
            }
        }, 1000);
    } else if (!b1.isEmpty() && !b2.isEmpty() && !b3.isEmpty() && !b4.isEmpty() &&
!b5.isEmpty() && !b6.isEmpty() && !b7.isEmpty() && !b8.isEmpty() && !b9.isEmpty()){
        Toast.makeText(this, "Match Draw !", Toast.LENGTH_SHORT).show();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                restart();
            }
        }, 1000);
    }
  }
 }
}
public void restart(){
    Button btn1 = findViewById(R.id.btn1);
    Button btn2 = findViewById(R.id.btn2);
    Button btn3 = findViewById(R.id.btn3);
    Button btn4 = findViewById(R.id.btn4);
    Button btn5 = findViewById(R.id.btn5);
    Button btn6 = findViewById(R.id.btn6);
    Button btn7 = findViewById(R.id.btn7);
    Button btn8 = findViewById(R.id.btn8);
    Button btn9 = findViewById(R.id.btn9);
    count = 0;
```

```
        flag = 0;
        btn1.setText("");
        btn2.setText("");
        btn3.setText("");
        btn4.setText("");
        btn5.setText("");
        btn6.setText("");
        btn7.setText("");
        btn8.setText("");
        btn9.setText("");
    }
}
```

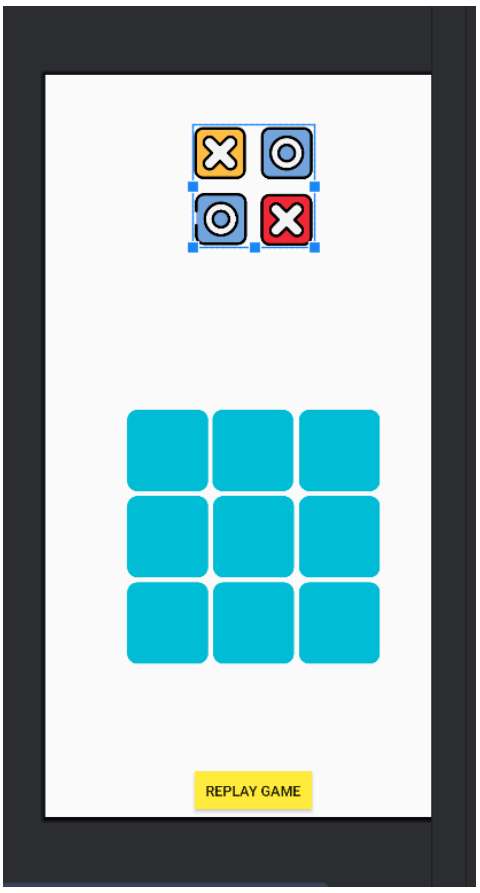


Figure 6.2 Game Activity

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.TicTacToe"
        tools:targetApi="31">
        <activity
            android:name=".SplashActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name=".MainActivity"
            android:exported="false">
        </activity>


    </application>

</manifest>
```

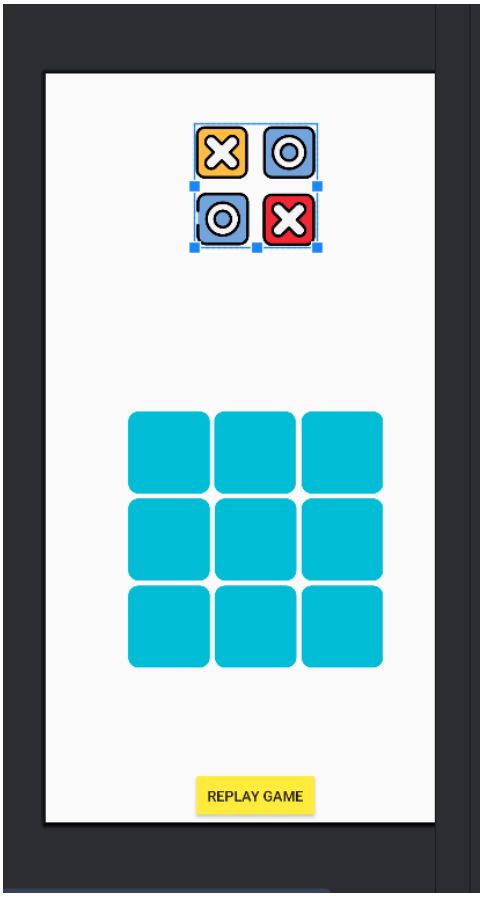


Figure 6.3 Grid Layout
Figure 6.4 Replay Button

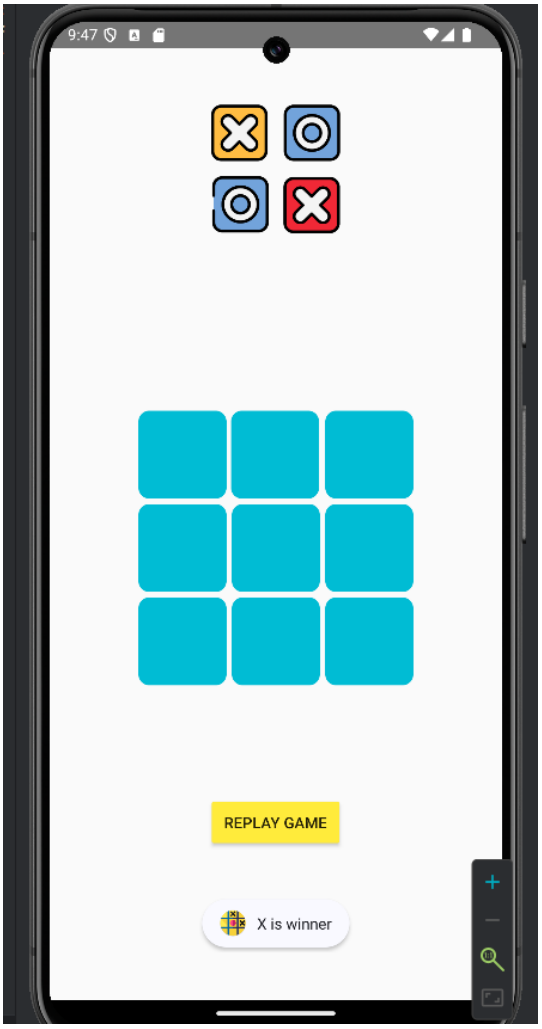


Figure 6.5 Toast Result

REFERENCES

- **6. References**
- Android Developer Documentation
- Relevant tutorials and articles