



Gujarat Technological University

Chandkheda, Ahmedabad
Affiliated



V.V.P. Engineering College, Rajkot

A
Project Report
on

Tic Tac Toe

Under the course of
Mobile Application Development (3170726)
B.E. Semester-VII

(Computer Engineering)

No.	Name of Student	Enrolment No.
1	Darshan Upadhyay	180470107058
2	Dishen Makwana	180470107035
3	Dhruv Vachhani	180470107059
4	Rashesh Padia	180470107041
5	Sharwil koradia	180470107029

Nivid Limasiya
(Assistant Prof.)

Dr. Tejas Patalia
(HOD)

Academic Year
(2021-22)



V.V.P. ENGINEERING COLLEGE

Department Of Computer Engineering

Vision of the Institute

- To be an exemplary institute, transforming students into competent professionals with human values.

Mission of the Institute

- To provide a conducive academic environment for strengthening technical capabilities of the students.
- To strengthen linkage with industries, alumni and professional bodies.
- To organize various co-curricular and extra-curricular activities for overall development of the students.
- To practice good governance and conduct value- based activities for making students responsible citizens.

Vision of the Department

- Transforming students into globally efficient professionals with moral values.

Mission of the Department

- To provide a strong foundation of computer engineering through effective teaching learning process.
- To enhance industry linkage & alumni network for better placement and real-world exposure.
- To provide various opportunities & platforms for all round development of students & encourage them for value-based practices.



V.V.P. ENGINEERING COLLEGE

RAJKOT

Certificate

This is to certify that

Mr. DISHEN MAKWANA, Enrolment No: 180470107035, Branch: Computer Engineering, Semester: 7 has satisfactorily completed the course in the subject: **Mobile Application Development (3170726)** within the four walls of V.V.P. Engineering College, Rajkot.

Date of Submission:

Prof. Nivid Limasiya,
Staff In-Charge

Head of Department,
Department of Computer Engineering,
V.V.P. Engineering College



V.V.P. Engineering College
Department of Computer Engineering
Final Assignment
AY. 2021-22 (ODD)

Semester: 7

Subject: Mobile Application Development

Subject Code: 3170726

Course Outcome	Student will be able to
CO1	Explain fundamentals of android operating system.
CO2	Design user interface and activity for android applications.
CO3	Use database and content providers in android applications.
CO4	Use multimedia, camera, and location-based services in android applications.
CO5	Develop an Android application and upload it on Google Play store.

Index

Sr No.	Context	Page No.
1	Introduction	6
2	What is Android	7
3	What is Android Studio	7
4	Structural Layout of Android Studio	8
5	Tic Tac Toe Game	9
6	MainActivity.java	9
7	Main_activity.xml	14
8	Project Photos	15

Introduction

Objective

- Tic-tac-toe, noughts and crosses, or Xs and Os/“X’y O’sies”, 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 diagonal, horizontal, or vertical row is the winner.
- To win the game, a player must place three of their marks in a horizontal, vertical, or diagonal row.
- Players soon discover that the best play from both parties leads to a draw. Hence, tic-tac-toe is most often played by young children, who often have not yet discovered the optimal strategy.
- [Incidence structure](#) for tic-tac-toe. Because of the simplicity of tic-tac-toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship and the branch of AI that deals with the searching of [game trees](#).
- It is straightforward to write a computer program to play tic-tac-toe perfectly or to enumerate the 765 essentially different positions (the [state space complexity](#)) or the 26,830 possible games up to rotations and reflections (the [game tree complexity](#)) on this space.
- If played optimally by both players, the game always ends in a draw, making tic-tac-toe a [futile game](#).

Project Context

- Tic-Tac-Toe is one type of basic game which any one can easily play. The rules of games are very simple and here to play a game we require fix two player. The put something one by one and if in diagonal, row or column we get same result then that player win that game.
- Here we use 0 and X for putting in 3x3 grid or plan.

Code link - <https://github.com/Darshan-upadhyay1110/MyAndroidApps/tree/master/ttt>

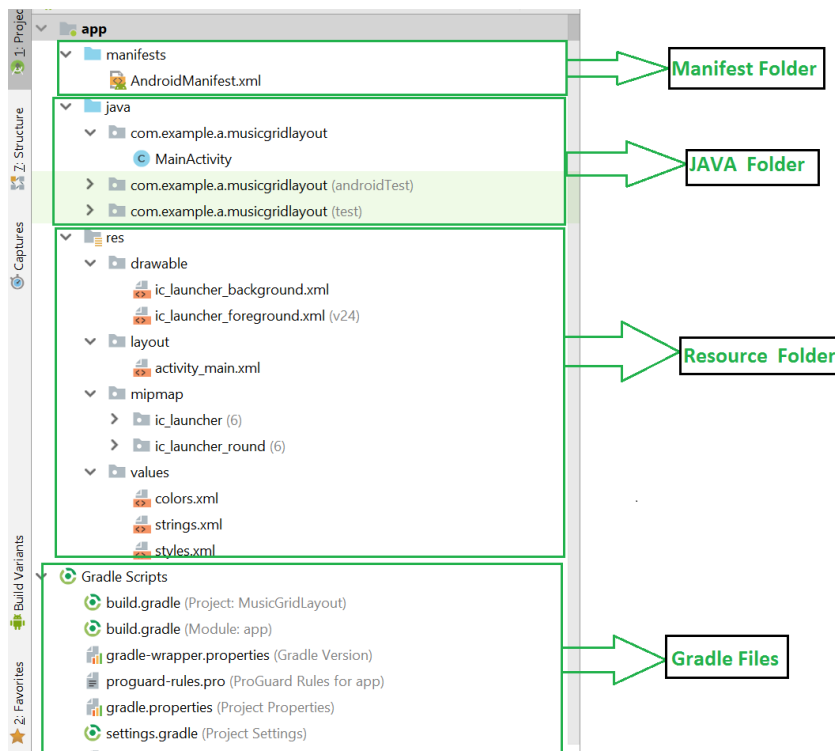
Android

- Android is the best-selling Operating System among various mobile platforms across the globe. Hundreds of millions of mobile devices are powered by Android in more than 190 countries of the world. It conquered around 75% of the global market share by the end of 2020, and this trend is growing bigger every other day.
- Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.
- Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google.
- It was unveiled in November 2007, with the first commercial Android device, the HTC Dream, being launched in September 2008.
- It is free and open-source software; its source code is known as Android Open Source Project (AOSP), which is primarily licensed under the Apache License.
- However most Android devices ship with additional proprietary software pre-installed, most notably Google Mobile Services (GMS), which includes core apps such as Google Chrome, the digital distribution platform Google Play and associated Google Play Services development platform.

Android Studio

- Android is an operating system that is built basically for Mobile phones.
- It is based on the Linux Kernel and other open-source software and is developed by Google.
- It is used for touchscreen mobile devices such as smartphones and tablets. But nowadays these are used in Android Auto cars, TV, watches, camera, etc.
- It has been one of the best-selling OS for smartphones. Android OS was developed by Android Inc. which Google bought in 2005.
- Various applications (apps) like games, music player, camera, etc. are built for these smartphones for running on Android.
- Google Play store features more than 3.3 million apps. The app is developed on an application known as Android Studio.
- These executable apps are installed through a bundle or package called APK(Android Package Kit).

Structural Layout of Android Studio



Manifest Folder: Android Manifest is an XML file that is the root of the project source set. It describes the essential information about the app and the Android build tools, the Android Operating System, and Google Play. It contains the permission that an app might need in order to perform a specific task.

Java Folder: The JAVA folder consists of the java files that are required to perform the background task of the app. It consists of the functionality of the buttons, calculation, storing, variables, toast(small popup message), programming function, etc.

Resource Folder: The res or Resource folder consists of the various resources that are used in the app. This consists of sub-folders like drawable, layout, mipmap, raw, and values. The drawable consists of the images. The layout consists of the XML files that define the user interface layout.

Gradle Files: Gradle is an advanced toolkit, which is used to manage the build process, that allows defining the flexible custom build configurations. Each build configuration can define its own set of code and resources while reusing the parts common to all versions of your app. The Android plugin for Gradle works with the build toolkit to provide processes and configurable settings that are specific to building and testing Android applications.

Tic-Tac-Toe

A Tic-Tac-Toe board is given after some moves are played. Find out if the given board is valid, i.e., is it possible to reach this board position after some moves or not.

Note that every arbitrary filled grid of 9 spaces isn't valid e.g. a grid filled with 3 X and 6 O isn't valid situation because each player needs to take alternate turns.

X	X	O
O	O	X
X	O	X

Valid Board

O	X	X
O	X	X
O	O	X

Invalid Board
(Both X and O cannot win)

Rules of the Game

- The game is to be played between two people (in this program between HUMAN and COMPUTER).
- One of the player chooses 'O' and the other 'X' to mark their respective cells.
- The game starts with one of the players and the game ends when one of the players has one whole row/ column/ diagonal filled with his/her respective character ('O' or 'X').
- If no one wins, then the game is said to be draw.

Implementation

In our program the moves taken by the computer and the human are chosen randomly. We use rand() function 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).

The above changes are left as an exercise to the readers.

MainActivity.java

```
package com.example.ttt;  
  
import androidx.appcompat.app.AppCompatActivity;
```

```

import android.os.Bundle;

import android.view.View;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;


public class MainActivity extends AppCompatActivity {

    int player=0;

    int countP1=0;

    int countP2=0;

    int[] gameState={2,2,2,2,2,2,2,2,2};

    int[][] winpos={ {0,1,2}, {3,4,5}, {6,7,8},

        {0,3,6}, {1,4,7}, {2,5,8},

        {0,4,8}, {2,4,6} };


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

    }

    public void OnTap(View view) {

        ImageView img=(ImageView) view;

        int tagnum=Integer.parseInt(img.getTag().toString());

        if(gameState[tagnum]==2)

        {

            gameState[tagnum]=player;

            img.setTranslationY(-1000f);

            if(player==0)

            {

                img.setImageResource(R.drawable.o);

                player=1;

            }

            else

            {

```

```

        img.setImageResource(R.drawable.x);

        player=0;
    }

    img.animate().translationYBy(1000f).setDuration(300);
}

for(int[] a:winpos)
{
    if(gameState[a[0]]==gameState[a[1]]&&gameState[a[1]]==gameState[a[2]]&&gameState[a[0]]!=2)
    {
        if(gameState[a[0]]==0)
        {
            Toast.makeText(this,"Player 1 Win",Toast.LENGTH_SHORT).show();

            countP1++;

            TextView p1=(TextView)findViewById(R.id.scoreOfP1);

            String setp1="" +countP1;

            p1.setText(setp1);

            reset(view);
        }
        else
        {
            Toast.makeText(this,"Player 2 Win",Toast.LENGTH_SHORT).show();

            countP2++;

            TextView p2=(TextView)findViewById(R.id.scoreOfP2);

            String setP2="" +countP2;

            p2.setText(setP2);

            reset(view);
        }
    }
}

}

public void reset(View view) {
    for(int i=0;i<gameState.length;i++)
    {
        gameState[i]=2;
    }
}

```

```

    }

    ((ImageView)findViewById(R.id.image0)).setImageResource(0);
    ((ImageView)findViewById(R.id.image1)).setImageResource(0);
    ((ImageView)findViewById(R.id.image2)).setImageResource(0);
    ((ImageView)findViewById(R.id.image3)).setImageResource(0);
    ((ImageView)findViewById(R.id.image4)).setImageResource(0);
    ((ImageView)findViewById(R.id.image5)).setImageResource(0);
    ((ImageView)findViewById(R.id.image6)).setImageResource(0);
    ((ImageView)findViewById(R.id.image7)).setImageResource(0);
    ((ImageView)findViewById(R.id.image8)).setImageResource(0);

}

public void resetPlayrerscore(View view) {
    countP1=0;
    countP2=0;
    String p1="" + countP1;
    String p2="" + countP2;
    TextView rp1=findViewById(R.id.scoreOfP1);
    rp1.setText(p1);
    TextView rp2=findViewById(R.id.scoreOfP2);
    rp2.setText(p2);
}
}

```

```

1 package com.example.ttt;
2
3
4 import ...
5
6
7
8
9
10
11
12 public class MainActivity extends AppCompatActivity {
13     int player=0;
14     int countP1=0;
15     int countP2=0;
16     int[] gameState={2,2,2,2,2,2,2,2,2};
17     int[][] winpos={{0,1,2}, {3,4,5}, {6,7,8},
18                     {0,3,6}, {1,4,7}, {2,5,8},
19                     {0,4,8}, {2,4,6}};
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25     }
26
27     public void onTap(View view) {
28         ImageView img=(ImageView) view;
29         int tagnum=Integer.parseInt(img.getTag().toString());
30         if(gameState[tagnum]==2)
31         {
32             gameState[tagnum]=player;
33             img.setTranslationY(-1000f);
34             if(player==0)
35             {
36                 img.setImageResource(R.drawable.o);
37                 player=1;
38             }
39             else

```

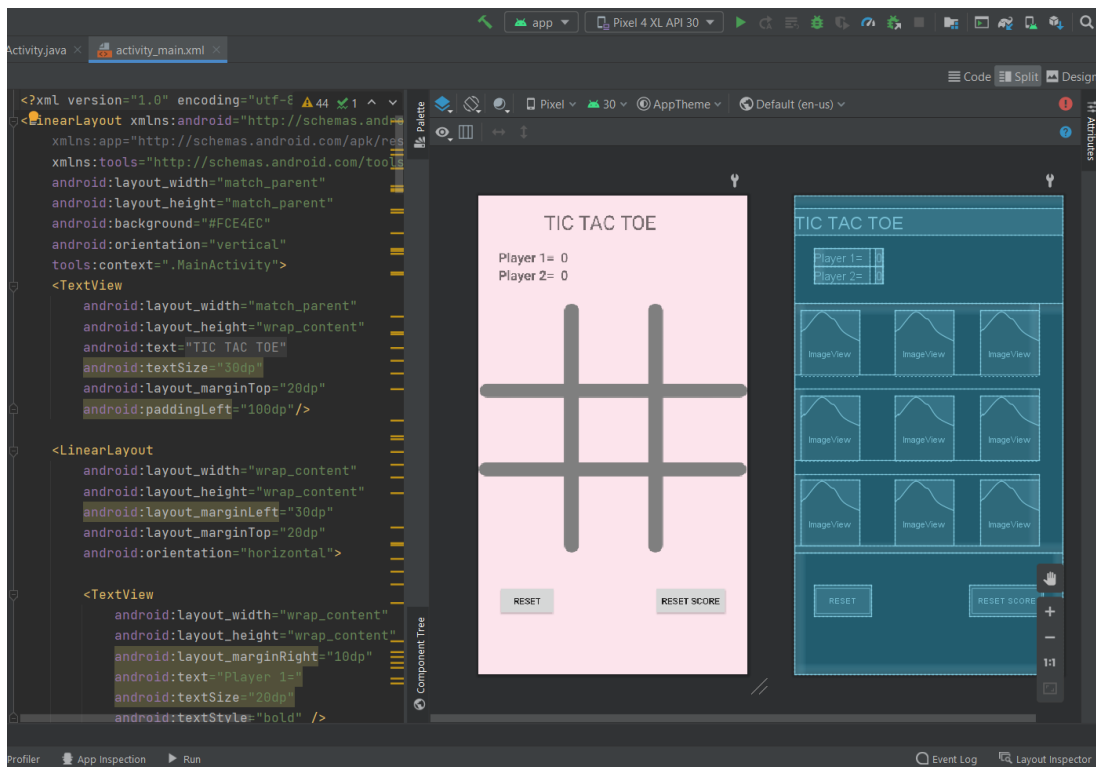
Failed to start monitoring emulator-5554 (40 minutes ago)

```

68     }
69     }
70
71     }
72
73     public void reset(View view) {
74         for(int i=0;i<gameState.length;i++)
75         {
76             gameState[i]=2;
77         }
78         ((ImageView)findViewById(R.id.image0)).setImageResource(0);
79         ((ImageView)findViewById(R.id.image1)).setImageResource(0);
80         ((ImageView)findViewById(R.id.image2)).setImageResource(0);
81         ((ImageView)findViewById(R.id.image3)).setImageResource(0);
82         ((ImageView)findViewById(R.id.image4)).setImageResource(0);
83         ((ImageView)findViewById(R.id.image5)).setImageResource(0);
84         ((ImageView)findViewById(R.id.image6)).setImageResource(0);
85         ((ImageView)findViewById(R.id.image7)).setImageResource(0);
86         ((ImageView)findViewById(R.id.image8)).setImageResource(0);
87     }
88
89
90
91     public void resetPlayerscore(View view) {
92         countP1=0;
93         countP2=0;
94         String p1="" +countP1;
95         String p2="" +countP2;
96         TextView rp1=findViewById(R.id.score0fP1);
97         rp1.setText(p1);
98         TextView rp2=findViewById(R.id.score0fP2);
99         rp2.setText(p2);
100    }

```

Failed to start monitoring emulator-5554 (40 minutes ago)



Main_activity.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:background="#FCE4EC"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TIC TAC TOE"
        android:textSize="30dp"
        android:layout_marginTop="20dp"
        android:paddingLeft="100dp"/>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="30dp"
        android:layout_marginTop="20dp"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginRight="10dp"
            android:text="Player 1="
            android:textSize="20dp"
            android:textStyle="bold" />

```

Main LinerLayout

1. LinerLayout
2. LinerLayout
3. LinerLayout

Grid png | O-X png

