Progress Report on App Development (Week 04)

-by Ritesh Kumar

An Intern in UpSkill Campus, 12/07/2023

Ritesh Kumar

I am pleased to present you with a comprehensive report on app development (Week 04), which provides an overview of “Tic Tac Game in Android App” with the process, challenges, and best practices for successful app development. This report aims to make understanding of the key aspects of app development and making informed decisions in this domain.

**Progress Report on App Development (Week 04)**

# An Overview on App Development

In the 4th week of “6 Weeks of App Development – Project-Based Learning”. We will be building a Tic Tac Toe Game Project using **Java** and **XML** in Android. The Tic Tac Toe Game is based on a two-player game. Each player chooses between X and O. Player play one move at a time simultaneously. In a move, a player can choose any position from a 3×3 grid. The goal here is to get three consecutive X or O in a horizontal, vertical, or diagonal direction. There will be a single activity in this application. This activity will show a 3×3 grid. The status of the game will be displayed at the bottom

Let’s move toward the project

* A Tic Tac Toe Game in Android App

## Content:

* Create a New Project
* Pre-task like adding of images
* Working with the activity\_main.xml file
* Working with the MainActivity.java file
* Source Code
* Output

### Challenges and Hurdles:

* **Add images in your android app**
* **Change the style to NoActionBar in themes.xml file**
* **Writing some simple logic for Tic Tac Toe Game**
* **Recursive Toast in Android**

#### Lesson Learned:

* **A Tic Tac Toe Game in Android App**

We are going to implement this project using the **Java**language.

**Step By Step Implementation of Project 3:**

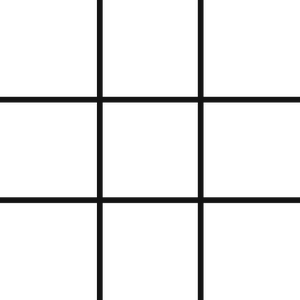
**Step 1: Create a New Project**

To create a new project in Android Studio please refer to How to Create/Start a New Project in Android Studio. Note that select Java as the programming language. Drag and drop the template image to drawable.

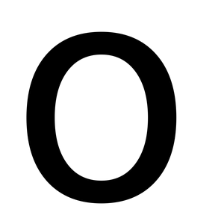
**Step 2: Before going to he coding section first you have to do some pre-task**

**Add Images:**All the images are listed below. Save them in your drawable folder in resources. Go to the **app > res > drawable** and paste the following files:

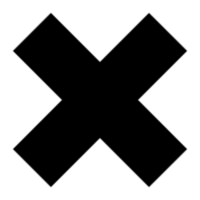
* **Grid**



* **O**



* **X**



**Change the style to NoActionBar in themes.xml file:**

<style name=”AppTheme” parent=”Theme.AppCompat.NoActionBar”>

**Step 3: Working with the activity\_main.xml file**

The XML codes are used to build the structure of the activity as well as its styling part. It contains a **TextView**at the very top of the activity to display the title. Then it contains an **ImageView** of the grid and in each box, there is an ImageView. At the bottom of the activity, there is a TextView to display the status of the game.

Below is the code for the **activity\_main.xml** file.

**Xml**

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

<**androidx.constraintlayout.widget.ConstraintLayout**

    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="@color/green"

    tools:context=".MainActivity">

    <!--title text-->

    <**TextView**

        android:id="@+id/textView"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_marginTop="23dp"

        android:textSize="45sp"

        android:textStyle="bold"

        app:fontFamily="cursive"

        app:layout\_constraintLeft\_toLeftOf="parent"

        app:layout\_constraintRight\_toRightOf="parent"

        app:layout\_constraintTop\_toTopOf="parent" />

    <!--image of the grid-->

    <**ImageView**

        android:id="@+id/imageView"

        android:layout\_width="0dp"

        android:layout\_height="wrap\_content"

        android:contentDescription="Start"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/textView"

        app:srcCompat="@drawable/grid" />

    <**LinearLayout**

        android:id="@+id/linearLayout"

        android:layout\_width="0dp"

        android:layout\_height="420dp"

        android:orientation="vertical"

        app:layout\_constraintBottom\_toBottomOf="@+id/imageView"

        app:layout\_constraintEnd\_toEndOf="@+id/imageView"

        app:layout\_constraintStart\_toStartOf="@+id/imageView"

        app:layout\_constraintTop\_toTopOf="@+id/imageView">

        <**LinearLayout**

            android:layout\_width="match\_parent"

            android:layout\_height="match\_parent"

            android:layout\_weight="1"

            android:orientation="horizontal">

            <!--images of the grid boxes-->

            <**ImageView**

                android:id="@+id/imageView0"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="0" />

            <**ImageView**

                android:id="@+id/imageView1"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="1" />

            <**ImageView**

                android:id="@+id/imageView2"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="2" />

        </**LinearLayout**>

        <**LinearLayout**

            android:layout\_width="match\_parent"

            android:layout\_height="match\_parent"

            android:layout\_weight="1"

            android:orientation="horizontal">

            <**ImageView**

                android:id="@+id/imageView3"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="3" />

            <**ImageView**

                android:id="@+id/imageView4"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="4" />

            <**ImageView**

                android:id="@+id/imageView5"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="5" />

        </**LinearLayout**>

        <**LinearLayout**

            android:layout\_width="match\_parent"

            android:layout\_height="match\_parent"

            android:layout\_weight="1"

            android:orientation="horizontal">

            <**ImageView**

                android:id="@+id/imageView6"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="6" />

            <**ImageView**

                android:id="@+id/imageView7"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="7" />

            <**ImageView**

                android:id="@+id/imageView8"

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:layout\_weight="1"

                android:onClick="playerTap"

                android:padding="20sp"

                android:tag="8" />

        </**LinearLayout**>

    </**LinearLayout**>

    <!--game status text display-->

    <**TextView**

        android:id="@+id/status"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_marginBottom="15sp"

        android:text="Status"

        android:textSize="28sp"

        android:textStyle="italic"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/linearLayout" />

</**androidx.constraintlayout.widget.ConstraintLayout**>

**Step 4: Working with the MainActivity.java file**

We will create a two-dimensional array that will store all the winning positions. We will create a function that will run when a box inside the grid is clicked. Inside this function, we will first check if the box selected is empty or not. After that, we will set the image of X if the last move was of O or we will set the image of O if the last move was of X. Then we will check if the move has reached the move position and then reset the game.

Below is the code for the **MainActivity.java** file. Comments are added inside the code to understand the code in more detail.

**Java**

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.ImageView;

**import** android.widget.TextView;

**import** androidx.appcompat.app.AppCompatActivity;

**public** **class** MainActivity **extends** AppCompatActivity {

**boolean** gameActive = **true**;

    // Player representation

    // 0 - X

    // 1 - O

**int** activePlayer = 0;

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

    // State meanings:

    //    0 - X

    //    1 - O

    //    2 - Null

    // put all win positions in a 2D array

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

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

            {0, 4, 8}, {2, 4, 6}};

**public** **static** **int** counter = 0;

    // this function will be called every time a

    // players tap in an empty box of the grid

**public** **void** playerTap(View view) {

        ImageView img = (ImageView) view;

**int** tappedImage = Integer.parseInt(img.getTag().toString());

        // game reset function will be called

        // if someone wins or the boxes are full

**if** (!gameActive) {

            gameReset(view);

        }

        // if the tapped image is empty

**if** (gameState[tappedImage] == 2) {

            // increase the counter

            // after every tap

            counter++;

            // check if its the last box

**if** (counter == 9) {

                // reset the game

                gameActive = **false**;

            }

            // mark this position

            gameState[tappedImage] = activePlayer;

            // this will give a motion

            // effect to the image

            img.setTranslationY(-1000f);

            // change the active player

            // from 0 to 1 or 1 to 0

**if** (activePlayer == 0) {

                // set the image of x

                img.setImageResource(R.drawable.x);

                activePlayer = 1;

                TextView status = findViewById(R.id.status);

                // change the status

                status.setText("O's Turn - Tap to play");

            } **else** {

                // set the image of o

                img.setImageResource(R.drawable.o);

                activePlayer = 0;

                TextView status = findViewById(R.id.status);

                // change the status

                status.setText("X's Turn - Tap to play");

            }

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

        }

**int** flag = 0;

        // Check if any player has won

**for** (**int**[] winPosition : winPositions) {

**if** (gameState[winPosition[0]] == gameState[winPosition[1]] &&

                    gameState[winPosition[1]] == gameState[winPosition[2]] &&

                    gameState[winPosition[0]] != 2) {

                flag = 1;

                // Somebody has won! - Find out who!

                String winnerStr;

                // game reset function be called

                gameActive = **false**;

**if** (gameState[winPosition[0]] == 0) {

                    winnerStr = "X has won";

                } **else** {

                    winnerStr = "O has won";

                }

                // Update the status bar for winner announcement

                TextView status = findViewById(R.id.status);

                status.setText(winnerStr);

            }

        }

        // set the status if the match draw

**if** (counter == 9 && flag == 0) {

            TextView status = findViewById(R.id.status);

            status.setText("Match Draw");

        }

    }

    // reset the game

**public** **void** gameReset(View view) {

        gameActive = **true**;

        activePlayer = 0;

**for** (**int** i = 0; i < gameState.length; i++) {

            gameState[i] = 2;

        }

        // remove all the images from the boxes inside the grid

        ((ImageView) findViewById(R.id.imageView0)).setImageResource(0);

        ((ImageView) findViewById(R.id.imageView1)).setImageResource(0);

        ((ImageView) findViewById(R.id.imageView2)).setImageResource(0);

        ((ImageView) findViewById(R.id.imageView3)).setImageResource(0);

        ((ImageView) findViewById(R.id.imageView4)).setImageResource(0);

        ((ImageView) findViewById(R.id.imageView5)).setImageResource(0);

        ((ImageView) findViewById(R.id.imageView6)).setImageResource(0);

        ((ImageView) findViewById(R.id.imageView7)).setImageResource(0);

        ((ImageView) findViewById(R.id.imageView8)).setImageResource(0);

        TextView status = findViewById(R.id.status);

        status.setText("X's Turn - Tap to play");

    }

    @Override

**protected** **void** onCreate(Bundle savedInstanceState) {

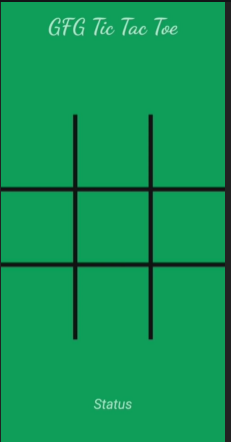
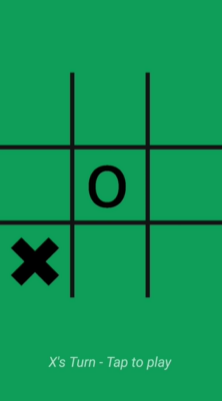
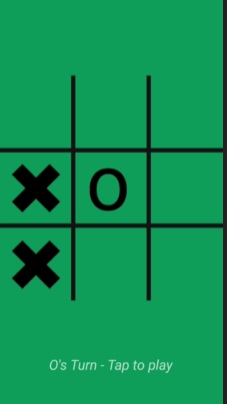
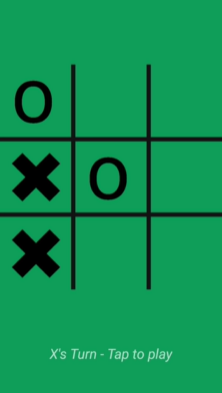
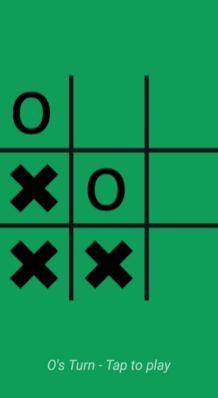
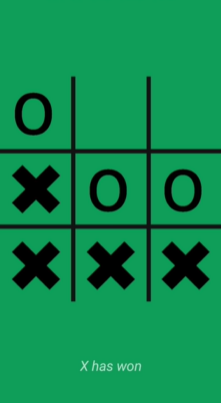
**super**.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

    }

}

**Output:**

**      **

##### **References:**

* <https://developer.android.com/studio>
* https://developer.android.com/
* <https://www.geeksforgeeks.org/>

Please note that this report provides a general overview of app development, and specific recommendations or strategies should be tailored to your organization's unique requirements and goals. Should you require further assistance or have any questions, please do not hesitate to contact me.

Thank you for the opportunity to prepare this report, and I hope it proves valuable in your app development endeavors. I look forward to discussing this topic further and assisting you in your future projects.

Sincerely,

Ritesh Kumar (9330771068)

Email: kumarritesh6769@gmail.com