ANDROID APPLICATION

TIC-TAC-TOE

BY

Kilari Venkata Sai Sanjana

(192011124, CSE)

Harini. A

(192011344, CSE)

Java code:

```
package com.dataflair.ticgame;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {


    private int grid_size;
    TableLayout gameBoard;
    TextView txt_turn;
    char [][] my_board;
    char turn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        grid_size = Integer.parseInt(getString(R.string.size_of_board));
        my_board = new char [grid_size][grid_size];
        gameBoard = (TableLayout) findViewById(R.id.mainBoard);
        txt_turn = (TextView) findViewById(R.id.turn);

        resetBoard();
        txt_turn.setText("Turn: "+turn);

        for(int i = 0; i< gameBoard.getChildCount(); i++){
            TableRow row = (TableRow) gameBoard.getChildAt(i);
            for(int j = 0; j<row.getChildCount(); j++){
                TextView tv = (TextView) row.getChildAt(j);
                tv.setText(R.string.none);
                tv.setOnClickListener(Move(i, j, tv));
            }
        }

        Button reset_btn = (Button) findViewById(R.id.reset);
```

```
        reset_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent current = getIntent();
                finish();
                startActivity(current);
            }
        });
    }

    protected void resetBoard(){
        turn = 'X';
        for(int i = 0; i< grid_size; i++){
            for(int j = 0; j< grid_size; j++){
                my_board[i][j] = ' ';
            }
        }
    }

    protected int gameStatus(){

        //0 Continue
        //1 X Wins
        //2 O Wins
        //-1 Draw

        int rowX = 0, colX = 0, rowO = 0, colO = 0;
        for(int i = 0; i< grid_size; i++){
            if(check_Row_Equality(i,'X'))
                return 1;
            if(check_Column_Equality(i, 'X'))
                return 1;
            if(check_Row_Equality(i,'O'))
                return 2;
            if(check_Column_Equality(i,'O'))
                return 2;
            if(check_Diagonal('X'))
                return 1;
            if(check_Diagonal('O'))
                return 2;
        }

        boolean boardFull = true;
        for(int i = 0; i< grid_size; i++){
            for(int j = 0; j< grid_size; j++){
                if(my_board[i][j]==' ')
                    boardFull = false;
            }
        }
        if(boardFull)
            return -1;
        else return 0;
    }

    protected boolean check_Diagonal(char player){
        int count_Equal1 = 0,count_Equal2 = 0;
        for(int i = 0; i< grid_size; i++)
            if(my_board[i][i]==player)
                count_Equal1++;
        for(int i = 0; i< grid_size; i++)
            if(my_board[i][grid_size -1-i]==player)
```

```
                count_Equal2++;
        if(count_Equal1== grid_size || count_Equal2== grid_size)
            return true;
        else return false;
    }

    protected boolean check_Row_Equality(int r, char player){
        int count_Equal=0;
        for(int i = 0; i< grid_size; i++){
            if(my_board[r][i]==player)
                count_Equal++;
        }

        if(count_Equal== grid_size)
            return true;
        else
            return false;
    }

    protected boolean check_Column_Equality(int c, char player){
        int count_Equal=0;
        for(int i = 0; i< grid_size; i++){
            if(my_board[i][c]==player)
                count_Equal++;
        }

        if(count_Equal== grid_size)
            return true;
        else
            return false;
    }

    protected boolean Cell_Set(int r, int c){
        return !(my_board[r][c]==' ');
    }

    protected void stopMatch(){
        for(int i = 0; i< gameBoard.getChildCount(); i++){
            TableRow row = (TableRow) gameBoard.getChildAt(i);
            for(int j = 0; j<row.getChildCount(); j++){
                TextView tv = (TextView) row.getChildAt(j);
                tv.setOnClickListener(null);
            }
        }
    }

    View.OnClickListener Move(final int r, final int c, final TextView tv){

        return new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if(!Cell_Set(r,c)) {
                    my_board[r][c] = turn;
                    if (turn == 'X') {
                        tv.setText(R.string.X);
                        turn = 'O';
                    } else if (turn == 'O') {
                        tv.setText(R.string.O);
                        turn = 'X';
                    }
```

```
                    if (gameStatus() == 0) {
                        txt_turn.setText("Turn: Player " + turn);
                    }
                    else if(gameStatus() == -1){
                        txt_turn.setText("This is a Draw match");
                        stopMatch();
                    }
                    else{
                        txt_turn.setText(turn+" Loses!");
                        stopMatch();
                    }
                }
                else{
                    txt_turn.setText(txt_turn.getText()+" Choose an Empty
Call");
                }
            }
        };
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.menu_board, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

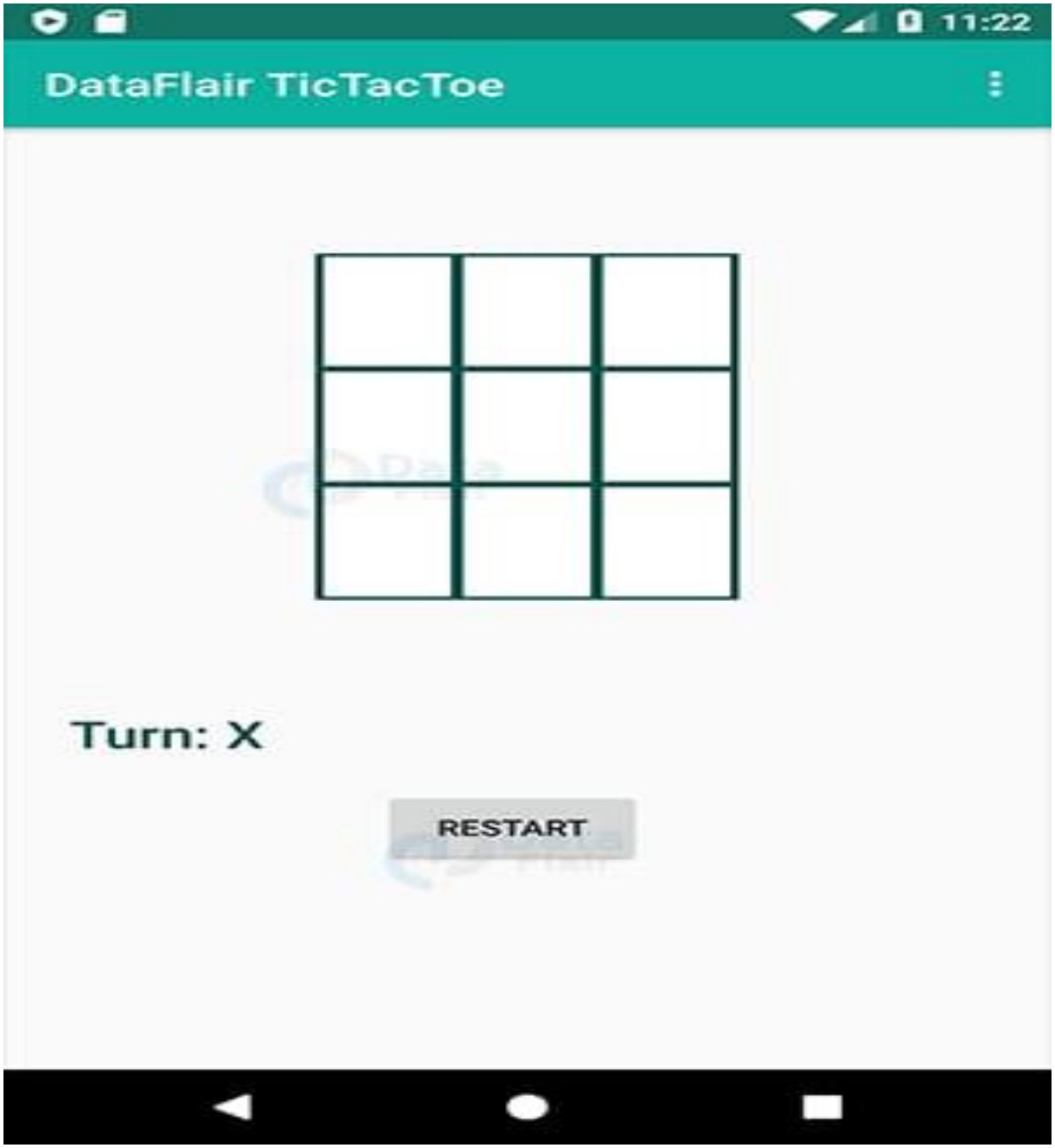
XML Code:

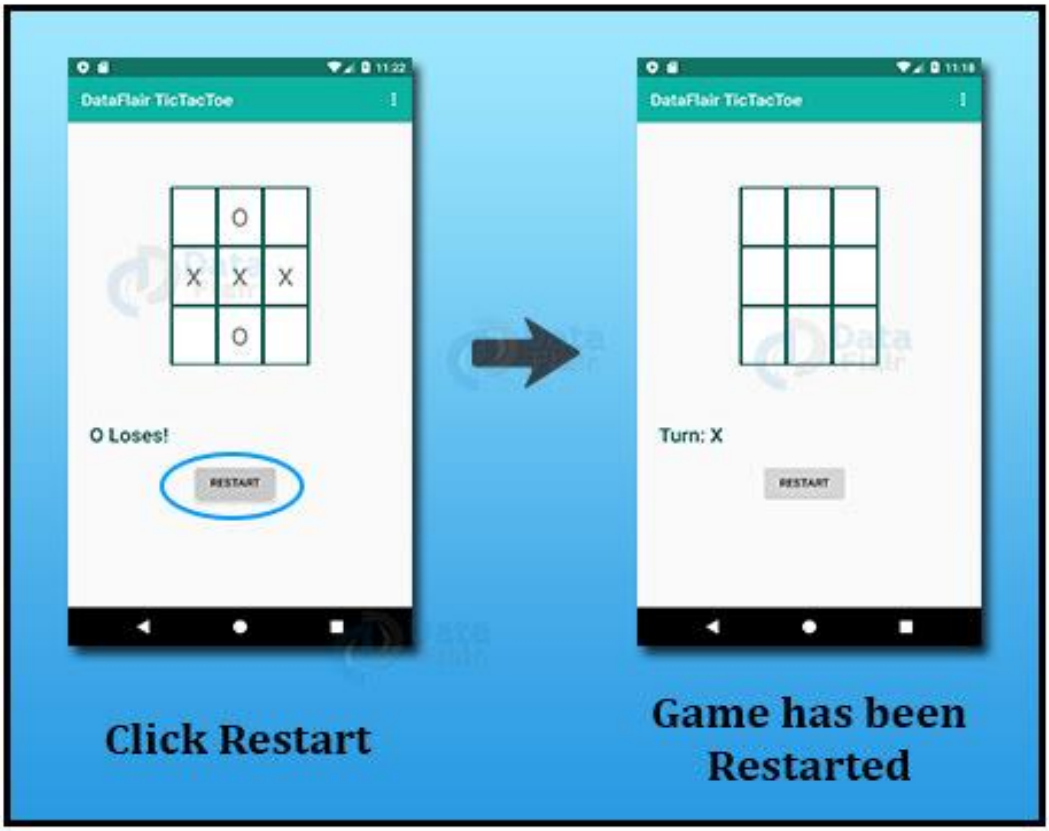
```
<?xml version="1.0"?>

    <LinearLayout tools:context=".MainActivity"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:android="http://schemas.android.com/apk/res/android">
```

```
<TableLayout
    android:id="@+id/mainBoard"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="top|center"
    android:clickable="true"
    android:gravity="center"
    android:layout_marginTop="55dp"
    android:nestedScrollingEnabled="false"
    android:padding="10dp">
    <TableRow
        android:id="@+id/row0"
        style="@style/TableRow"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

     <TextView
         style="@style/LeftCell"
         android:width="50dp"
         android:layout_column="0"></TextView>

        <TextView
            style="@style/MiddleCell"
            android:width="50dp"
            android:layout_column="1"></TextView>

        <TextView
            style="@style/RightCell"
            android:width="50dp"
            android:layout_column="2"></TextView>

    </TableRow>

    <TableRow
        android:id="@+id/row1"
        style="@style/TableRow"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            style="@style/LeftCell"
            android:width="50dp"
            android:layout_column="0"></TextView>

        <TextView
            style="@style/MiddleCell"
            android:width="50dp"
            android:layout_column="1"></TextView>

        <TextView

            style="@style/RightCell"
            android:width="50dp"
            android:layout_column="2"
            ></TextView>

    </TableRow>

    <TableRow
```

```
                android:id="@+id/row2"
                style="@style/TableRow"
                android:layout_width="match_parent"
                android:layout_height="match_parent">

                <TextView
                    style="@style/LeftCell"
                    android:width="50dp"
                    android:layout_column="0"></TextView>

                <TextView
                    style="@style/MiddleCell"
                    android:width="50dp"
                    android:layout_column="1"></TextView>

                <TextView
                    style="@style/RightCell"
                    android:width="50dp"
                    android:layout_column="2"></TextView>

            </TableRow>

        </TableLayout>

        <TextView android:layout_height="wrap_content"
            android:layout_width="fill_parent"
            android:id="@+id/turn"
            android:padding="10dp"
            android:layout_marginTop="50dp"
            android:gravity="left"
            android:textColor="@color/black"
            android:fontFamily="sans-serif-medium"
            android:textSize="24dp"
            android:text="          Turn:  "/>

    <Button
        android:id="@+id/reset"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="132dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="   Restart   " />

</LinearLayout>
```

Output:




DataFlair TicTacToe
O
X X X
O
O Loses!
RESTART
DataFlair TicTacToe
Turn: X
RESTART
Click Restart
Game has been Restarted