Mobile Application Development Laboratory
Lab 2

Tic Tac Toe

Roll: ***106118103*** *Name:* ***V. Aananth***

Aim:

To make an android application for a two player tic tac toe game.

Description of App:

Tic-tac-toe, or noughts and crosses, 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 contiguous diagonal, horizontal, or vertical line is the winner.

Following features are implemented in the application:

- A 3x3 Tic Tac Toe grid.
- Allow players to make moves by clicking on the grids.
- Check and notify (using a toast) if the game is over, and if any player has won the game.
- Button to reset the board.

Device Specifications:

Model: Poco F1
Android Version: 9 (API Level 28)
Resolution: 2160 x 1080 pixels

Technical Concepts Learnt:

- To create a Grid Layout and add child views.
- To programmatically create and insert Buttons.
- To programmatically change the visibility of views.
- To programmatically change the text and background color of views.
- To create and display toasts.
- To create games in an Object Oriented Manner.

Source Code:

(i) TicTacToe.java - The game logic class which is used by the Activity to create and manage the game.

```
package com.example.tictactoe;

public class TicTacToe {
    int grid[] = {-1, -1, -1, -1, -1, -1, -1, -1, -1}; // 1 for P1
and 2 for P2
    int winner; // 1 - P1, 2 - P2, 3 - Draw
    int numPlays = 0;
    int currentPlayer = -1;
    public static final int PLAYER1 = 1;
    public static final int PLAYER2 = 2;
    public static final int DRAW = 3;

    // Constructor
    TicTacToe() {
        winner = -1;
        currentPlayer = PLAYER1;
    }

    // Check if game is over
    public void checkIfGameIsOver () {
        if((grid[0] == 1 && grid[3] == 1 && grid[6] == 1) ||
                (grid[0] == 1 && grid[1] == 1 && grid[2] == 1) ||
                (grid[0] == 1 && grid[4] == 1 && grid[8] == 1) ||
                (grid[1] == 1 && grid[4] == 1 && grid[7] == 1) ||
                (grid[2] == 1 && grid[5] == 1 && grid[8] == 1) ||
                (grid[3] == 1 && grid[4] == 1 && grid[5] == 1) ||
                (grid[6] == 1 && grid[7] == 1 && grid[8] == 1) ||
                (grid[2] == 1 && grid[4] == 1 && grid[6] == 1)
        ) {
            winner = PLAYER1;
        } else if((grid[0] == 2 && grid[3] == 2 && grid[6] == 2) ||
                (grid[0] == 2 && grid[1] == 2 && grid[2] == 2) ||
                (grid[0] == 2 && grid[4] == 2 && grid[8] == 2) ||
                (grid[1] == 2 && grid[4] == 2 && grid[7] == 2) ||
```

```
                (grid[2] == 2 && grid[5] == 2 && grid[8] == 2) ||
                (grid[3] == 2 && grid[4] == 2 && grid[5] == 2) ||
                (grid[6] == 2 && grid[7] == 2 && grid[8] == 2) ||
                (grid[2] == 2 && grid[4] == 2 && grid[6] == 2)
        ) {
            winner = PLAYER2;
        } else if (numPlays == grid.length) {
            winner = DRAW;
        }
    }

    // Function to make a move
    public boolean makeMove(int i) {
        if(winner != -1 || i >= grid.length || grid[i] != -1) {
            return false;
        }

        grid[i] = currentPlayer;
        currentPlayer = currentPlayer == PLAYER1 ? PLAYER2 : PLAYER1;

        numPlays++;

        checkIfGameIsOver();

        return true;
    }

    public void reset() {
        winner = -1;
        numPlays = 0;
        for (int i = 0; i < 9; i++) grid[i] = -1;
        currentPlayer = PLAYER1;
    }
}
```

(ii) MainActivity.java - Used to render the game and related views.

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

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.GridLayout;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    TicTacToe game;
    GridLayout mGridLayout;
    ArrayList<Button> mButtons;
    TextView mCurrentMove, mWinner, mDraw;
    LinearLayout mCurrentMoveContainer, mWinnerContainer;
    Button mResetButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        game = new TicTacToe();
        mGridLayout = (GridLayout) findViewById(R.id.grid);
        mButtons = new ArrayList<Button>();
        mCurrentMove = (TextView) findViewById(R.id.current_move);
        mWinner = (TextView) findViewById(R.id.winner);
        mDraw = (TextView) findViewById(R.id.draw);
        mCurrentMoveContainer = (LinearLayout)
findViewById(R.id.current_move_container);
        mWinnerContainer = (LinearLayout)
findViewById(R.id.winner_container);
        mResetButton = (Button) findViewById(R.id.reset_button);
```

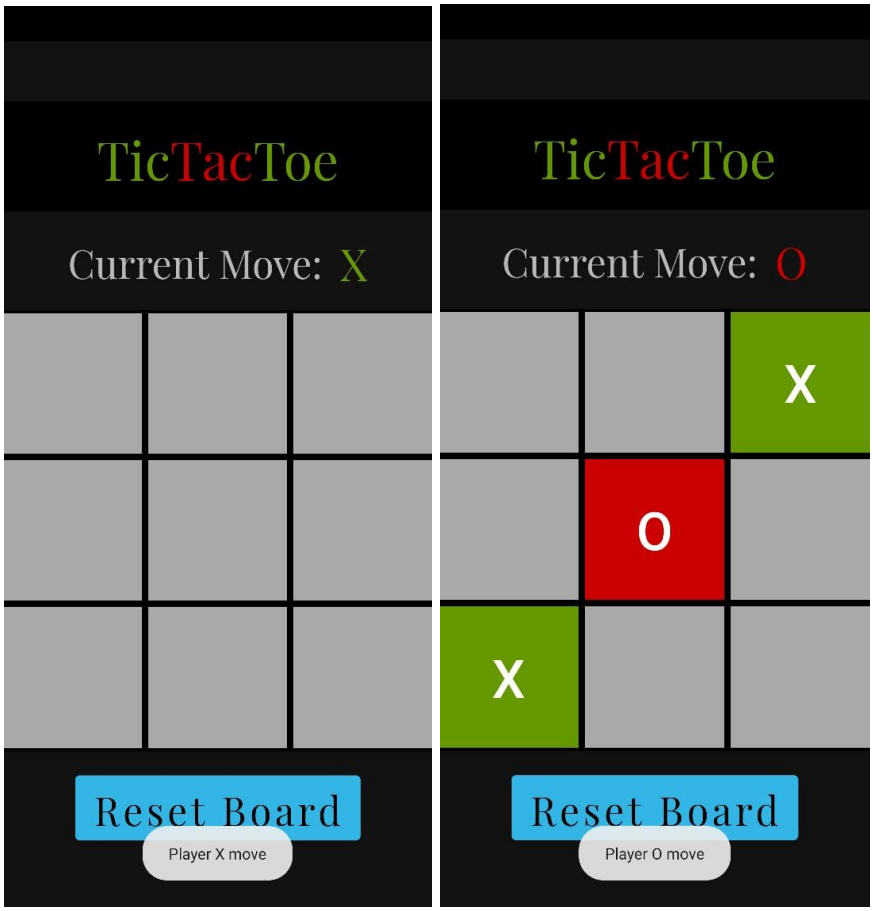
```
        for (int i = 0; i < 9; i++) {
            Button b = new Button(this);

            GridLayout.LayoutParams lp = new
GridLayout.LayoutParams();

            lp.columnSpec = GridLayout.spec(GridLayout.UNDEFINED,
1f);
            lp.rowSpec = GridLayout.spec(GridLayout.UNDEFINED, 1f);

            int margin = 8;
            lp.topMargin = margin;
            lp.bottomMargin = margin;
            lp.leftMargin = margin;
            lp.rightMargin = margin;

            b.setLayoutParams(lp);

            b.setTextSize(48);

            int finalI = i;
            b.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    makeMove(finalI);
                }
            });

            mGridLayout.addView(b);
            mButtons.add(b);
        }

        mResetButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                reset();
            }
        });
```

```
        reset();
    }

    private String getPlayerSymbol(int id) {
        if (id == game.PLAYER1)
            return "X";
        else if (id == game.PLAYER2)
            return "O";

        return "";
    }

    private int getPlayerColor(int id) {
        if (id == game.PLAYER1)
            return 0xff669900;
        else if (id == game.PLAYER2)
            return 0xffcc0000;

        return 0xffA9A9A9;
    }

    private void updateGrid() {
        for (int i = 0; i < 9; i++) {
            mButtons.get(i).setText(getPlayerSymbol(game.grid[i]));

mButtons.get(i).setBackgroundColor(getPlayerColor(game.grid[i]));
        }
    }

    private void updateMove() {
        mCurrentMove.setText(getPlayerSymbol(game.currentPlayer));

mCurrentMove.setTextColor(getPlayerColor(game.currentPlayer));

        Toast toast = Toast.makeText(getApplicationContext(),
                "Player " + getPlayerSymbol(game.currentPlayer) + "
move",
                Toast.LENGTH_SHORT);
```

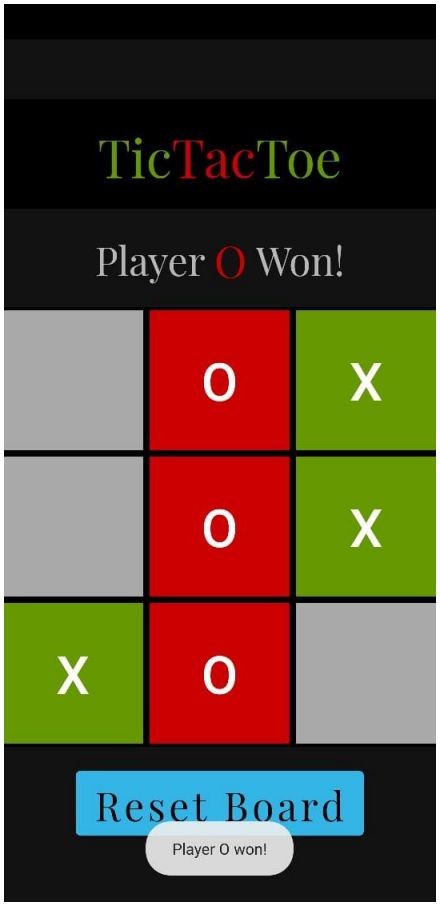
```
        toast.show();
    }

    private void updateWinner() {
        Log.d("winner", game.winner + "");

        if (game.winner == -1)
            return;

        mCurrentMoveContainer.setVisibility(View.GONE);

        if (game.winner == game.DRAW) {
            mDraw.setVisibility(View.VISIBLE);

            Toast toast = Toast.makeText(getApplicationContext(),
                    "Game was Drawn!",
                    Toast.LENGTH_SHORT);

            toast.show();

            return;
        }

        mWinnerContainer.setVisibility(View.VISIBLE);
        mWinner.setText(getPlayerSymbol(game.winner));
        mWinner.setTextColor(getPlayerColor(game.winner));

        Toast toast = Toast.makeText(getApplicationContext(),
                "Player " + getPlayerSymbol(game.winner) + " won!",
                Toast.LENGTH_SHORT);

        toast.show();
    }

    private void makeMove(int i) {
        if (game.makeMove(i)) {
            updateGrid();
            updateMove();
```

```
            updateWinner();
        }
    }

    private void reset() {
        game.reset();
        updateGrid();
        updateMove();

        mCurrentMoveContainer.setVisibility(View.VISIBLE);
        mDraw.setVisibility(View.GONE);
        mWinnerContainer.setVisibility(View.GONE);
    }
}
```

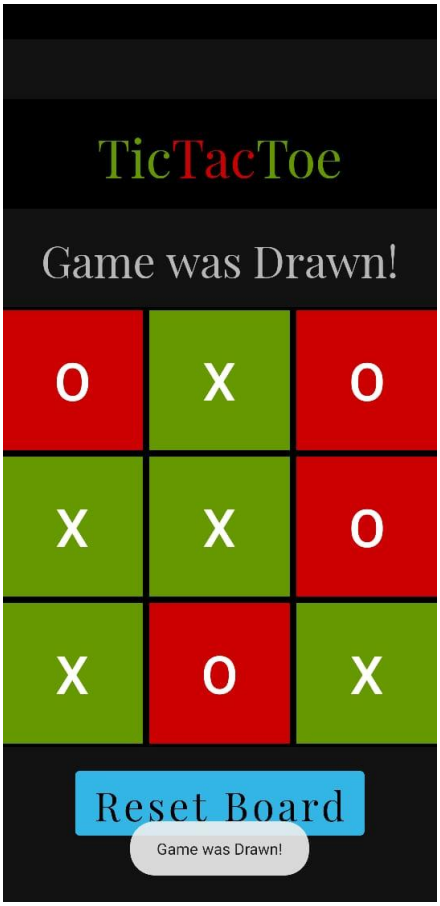
Screenshots:



(i) Initial State (ii) Game in Progress



(iii) Player 'O' Won



(iv) Game Drawn

Video Demo:
https://drive.google.com/file/d/14fA8eS7BUW4Nv_WdIoQFxjKpV6VkEN6l/view?usp=sharing

APK:
https://drive.google.com/file/d/1VRCmpPDfrSp4zJMCiDdue2WUDN7ByVry/view?usp=sharing

Outcomes:

An android application was developed for a two player Tic Tac Toe game. Various concepts in Android App Development were explored including:

- Creating grid layouts and adding child views.
- Programmatically creating and insert Buttons.
- Programmatically changing the visibility of views.
- Programmatically changing the text and background color of views.
- Creating and displaying toasts.