

Project Report: Tic Tac Toe Android App

Course: Mobile App Development

Submitted by:

Ahtisham Tanveer

Summary:

This project is a mobile application developed using Java and Android Studio that allows users to play the classic Tic Tac Toe game. It features two modes: Play with a Friend and Play with Computer. The computer uses the Minimax algorithm to make smart, unbeatable moves. The app has a clean, user-friendly interface with a neon-themed design. It includes win/draw detection, reset functionality, and smooth switching between modes.

The project helped the team gain practical experience in Android development, UI design, game logic implementation, and AI techniques.

Tools and Technologies Used:

Tool/Technology	Purpose
Java	Main programming language
Android Studio	IDE for Android development
XML	Layout and UI Design
Minimax Algorithm	AI logic for computer player

XML Code:

```
</> activity_main.xml x MainActivity.java </> btn_neon.xml </> s
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:orientation="vertical"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:gravity="top|center_horizontal"
8     android:padding="16dp"
9     android:background="@color/background_black">
10
11 <!-- Spacer -->
12 <View
13     android:layout_width="match_parent"
14     android:layout_height="20dp" />
15
16 <!-- Title at the top -->
17 <TextView
18     android:id="@+id/titleTextView"
19     android:layout_width="wrap_content"
20     android:layout_height="wrap_content"
21     android:text="Welcome to tic_tac_toe game"
22     android:textSize="26sp"
23     android:textStyle="bold"
24     android:textColor="@color/button_text"
25     android:layout_marginBottom="16dp"
26     android:shadowColor="#00FFFF"
27     android:shadowDx="0"
28     android:shadowDy="0"
29     android:shadowRadius="8" />
30
31 <!-- Game Mode Selection -->
32 <RadioGroup
33     android:id="@+id/modeSelector"
34     android:layout_width="wrap_content"
35     android:layout_height="wrap_content"
36     android:orientation="horizontal"
37     android:layout_marginBottom="20dp">
38
39
```

```

39     <RadioButton
40         android:id="@+id/playWithComputer"
41         android:layout_width="wrap_content"
42         android:layout_height="wrap_content"
43         android:text="Play with Computer"
44         android:textColor="@color/button_text"
45         android:buttonTint="@color/button_text"
46         android:checked="true" />
47
48     <RadioButton
49         android:id="@+id/playWithFriend"
50         android:layout_width="wrap_content"
51         android:layout_height="wrap_content"
52         android:text="Play with Friend"
53         android:textColor="@color/button_text"
54         android:buttonTint="@color/button_text"
55         android:layout_marginStart="16dp" />
56 </RadioGroup>

```

```

57
58 <!-- spacer added here to push the grid lower -->

```

```

59 <View
60     android:layout_width="match_parent"
61     android:layout_height="40dp" />
62

```

```

63 <!-- Game Grid -->

```

```

64 <GridLayout
65     android:id="@+id/gridLayout"
66     android:layout_width="wrap_content"
67     android:layout_height="wrap_content"
68     android:columnCount="3"
69     android:rowCount="3"
70     android:layout_marginBottom="20dp">
71

```

```

<!-- Winner text -->

```

```

<TextView
    android:id="@+id/winnerTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textColor="@color/button_text"
    android:textSize="24sp"
    android:layout_marginBottom="16dp"
    android:shadowColor="#00FFFF"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="8" />

```

```

<!-- Reset Button -->

```

```

<Button
    android:id="@+id/resetButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Reset Game"
    android:backgroundTint="@color/navy_blue"
    android:textColor="@color/button_text"
    android:textSize="18sp"
    android:paddingLeft="24dp"
    android:paddingRight="24dp"
    android:layout_marginTop="8dp" />

```

```

</LinearLayout>

```

JAVA Code:

```
1 package com.example.tic_tac_toe_game;
2
3 import android.os.Bundle;
4 import android.widget.Button;
5 import android.widget.RadioButton;
6 import android.widget.RadioGroup;
7 import android.widget.TextView;
8 import androidx.appcompat.app.AppCompatActivity;
9
10 public class MainActivity extends AppCompatActivity {
11
12     11 usages
13     private Button[] buttons = new Button[9];
14     6 usages
15     private boolean isXTurn = true;
16     13 usages
17     private String[] board = new String[9];
18     2 usages
19     private Button resetButton;
20     6 usages
21     private TextView winnerTextView;
22     2 usages
23     private RadioGroup modeSelector;
24     1 usage
25     private RadioButton playWithComputer, playWithFriend;
26     2 usages
27     private boolean isComputerMode = true;
28
29     @Override
30     @f
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_main);
34
35         for (int i = 0; i < 9; i++) {
36             String buttonID = "button" + i;
37             int resID = getResources().getIdentifier(buttonID, "id", getPackageName());
38             buttons[i] = findViewById(resID);
39             int finalI = i;
40             buttons[i].setOnClickListener(new View.OnClickListener() {
41                 @Override
42                 public void onClick(View v) {
43                     handleClick(finalI);
44                 }
45             });
46         }
47     }
48 }
```

```

31         buttons[i].setOnClickListener( View v -> handleClick(finalI));
32     }
33
34     resetButton = findViewById(R.id.resetButton);
35     winnerTextView = findViewById(R.id.winnerTextView);
36
37     modeSelector = findViewById(R.id.modeSelector);
38     playWithComputer = findViewById(R.id.playWithComputer);
39     playWithFriend = findViewById(R.id.playWithFriend);
40
41     modeSelector.setOnCheckedChangeListener(( RadioGroup group, int checkedId) -> {
42         isComputerMode = (checkedId == R.id.playWithComputer);
43         resetGame();
44     });
45
46     resetButton.setOnClickListener( View v -> resetGame());
47
48     resetGame();
49 }

```

1 usage

```

51 private void handleClick(int index) {
52     if (!board[index].equals("")) return;
53
54     if (isXTurn) {
55         board[index] = "X";
56         buttons[index].setText("X");
57         buttons[index].setTextColor(getResources().getColor(R.color.black));
58     } else {
59         board[index] = "O";
60         buttons[index].setText("O");
61         buttons[index].setTextColor(getResources().getColor(R.color.button_text));
62     }
63
64     if (checkWin(board[index])) {
65         winnerTextView.setText(board[index] + " is Winner!");
66         disableButtons();
67         return;

```

```
public class minimaxivity extends AppCompactivity {
```

```
    private int minimax(String[] currentBoard, int depth, boolean isMaximizing) {  
        currentBoard[0][0] = "O";  
        int score = minimax(currentBoard, depth: depth + 1, isMaximizing: false);  
        currentBoard[0][0] = "";  
        bestScore = Math.max(score, bestScore);  
    }  
    }  
    return bestScore;  
} else {  
    int bestScore = Integer.MAX_VALUE;  
    for (int i = 0; i < 9; i++) {  
        if (currentBoard[i].equals("")) {  
            currentBoard[i] = "X";  
            int score = minimax(currentBoard, depth: depth + 1, isMaximizing: true);  
            currentBoard[i] = "";  
            bestScore = Math.min(score, bestScore);  
        }  
    }  
    return bestScore;  
}  
}
```

2 usages

```
private boolean checkWin(String player) {  
    return checkWin(player, board);  
}
```

3 usages

```
private boolean checkWin(String player, String[] boardToCheck) {  
    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}  
    };  
}
```

```
}
```

```
2 usages
```

```
private boolean isDraw() {  
    for (String cell : board) {  
        if (cell.equals("")) return false;  
    }  
    return true;  
}
```

```
1 usage
```

```
@ private boolean isDraw(String[] boardToCheck) {  
    for (String cell : boardToCheck) {  
        if (cell.equals("")) return false;  
    }  
    return true;  
}
```

```
3 usages
```

```
private void resetGame() {  
    for (int i = 0; i < 9; i++) {  
        board[i] = "";  
        buttons[i].setText("");  
        buttons[i].setEnabled(true);  
    }  
    isXTurn = true;  
    winnerTextView.setText("");  
}
```

```
2 usages
```

```
private void disableButtons() {  
    for (Button btn : buttons) {  
        btn.setEnabled(false);  
    }  
}
```

```
}
```

