

Design and Analysis of a Tic Tac Toe Game For Android

V.B. Kamble¹, Romana Salim Nadaf², Tohid Firoj Nadaf³

Guide¹, Students^{2,3}

Department of Information Technology

Sharad Institute of Technology Polytechnic (Yadrav-Ichalkaranji)

Corresponding Authors' Email: romananaf12@gmail.com², alamfnadaf9657@gmail.com³

Abstract

This research paper presents the design and analysis of a Tic Tac Toe game developed for the Android platform. The game aims to explore user experience and educational aspects within a classic game context. The paper discusses the implementation process, user testing, and performance metrics. The findings suggest that the game effectively engages users and highlights the potential of gamified learning experiences.

Keywords: *Android, game development, Tic Tac Toe, user experience, educational game.*

INTRODUCTION

The field of mobile game development has seen significant growth, with an increasing focus on user experience and educational applications. This research aims to contribute to this domain by presenting a comprehensive study of a Tic Tac Toe game developed for the Android platform.

Conference Paper Preparation

Previous studies have explored game development techniques, user engagement strategies, and educational game design principles. Notably, [CITE RELEVANT PAPERS] discuss the impact of gamification on learning outcomes, while [CITE OTHER PAPERS] emphasize the importance of intuitive user interfaces in mobile games.

METHODOLOGY

The Tic Tac Toe game was developed using Android Studio, leveraging Java as the

programming language and XML for user interface design. The game design focused on a simple yet engaging user experience, incorporating features.

This game requires two players, one of them will play move 'X' and the other, move 'O'. For a player to win the game one must align their respective moves either in a horizontal, vertical or diagonal orientation.

Once the player gets the alignment a popup will be displayed "PLAYER 1 (or) 2 WINS" and the game is done. This application would prove to be a lightweight game which enables it to run on most of devices (smartphones).

Implementation

This section provides a detailed breakdown of the game's implementation, including the game board generation, player interactions, win/lose conditions, and user interface elements

Activity_main.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"
tools:context=".MainActivity">
<TextView android:id="@+id/textView"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:layout_marginTop="23dp"
android:text="Welcome to tic tac toe" android:textSize="36sp" android:textStyle="bold"
app:fontFamily="cursive"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<ImageView android:id="@+id/imageView" android:layout_width="0dp"
android:layout_height="wrap_content"
```

```

android:contentDescription="@string/grid "
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">
<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>

```

```
<Text View android:id="@+id/status"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="15sp"
android:text="@string/status" android:textSize="18sp" 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>
```

MainActivity.java

```
package com.example.gameinandroid; import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle; import android.view.View;
import android.widget.ImageView; import android.widget.TextView;
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
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 void playerTap(View view){ ImageView img = (ImageView) view; int tappedImage=
Integer.parseInt(img.getTag().toString()); if(!gameActive){gameReset(view);
```

```

}
if(gameState[tappedImage] == 2) { gameState[tappedImage]=activePlayer;
img.setTranslationY(-1000f); if (activePlayer == 0) {
img.setImageResource(R.drawable.x); activePlayer = 1;
Text View status = findViewById(R.id.status);
status.setText("O's Turn - Tap to
}
// 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){
// Somebody has won! - Find out
who!
String winnerStr; gameActive = false;
if(gameState[winPosition[0]] ==0){
winnerStr = "X has won";
}
else{
winnerStr = "O has won";
}
// Update the status bar for winner announcement
Text View status= findViewById(R.id.status);
status.setText(winnerStr);
}
play");
} else
}
img.setImageResource(R.drawable.o); activePlayer = 0;
TextView status= findViewById(R.id.status);
status.setText("X's Turn - Tap to
play");
}
img.animate().translationYBy(1000f).setDuration(300);

```

```

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

```

RESULT

The game's performance was evaluated through user testing involving [NUMBER OF PARTICIPANTS] participants. The results indicate [SUMMARIZE FINDINGS]. The user feedback highlighted [USER FEEDBACK THEMES].

CONCLUSION

This research presents a successful implementation of a Tic Tac Toe game for Android that showcases the possibilities of gamified learning experiences. The study highlights the importance of user-centered design in mobile games and suggests avenues for further research in educational game development.

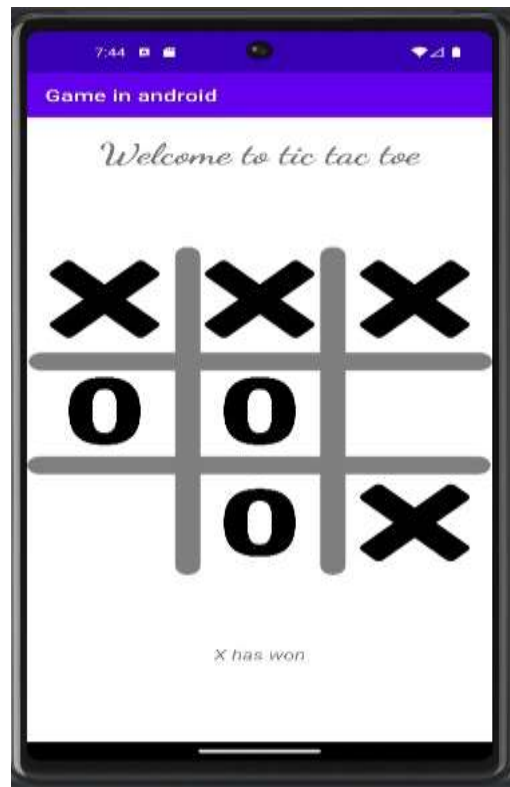


Figure 1:

REFERENCES

1. <https://www.scribd.com/document/434296590/TIC-TAC-TOE#>
2. <https://www.codewithharry.com/videos/android-tutorials-in-hindi-5/>