ABOUT SRS TIC TAC TOE APK

TABLE OF CONTENTS
- Application information
- Applications Description
- How to install SRS Tic Tac Toe APK for Android?
- How to install SRS Tic Tac Toe APK for PC (Windows 7/8/10 or MAC)?

APPLICATION INFORMATION:
- Version: "1.0"
- Updated on: 04.12.2021
- Released on: 04.12.2021
- Download size: 4 mb
- Application requirements:
    1. Control vibration
    2. Prevent phone from sleeping
    3. Run at start-up

APPLICATION FEATURES:
- You can see the player's turn after each move.
- Offline feature
- You get alerts if anyone wins or if the game is a draw.

DESCRIPTION:

- You can play the tic tac toe game and relax.

HOW TO INSTALL SRS TIC TAC TOE APK FOR AN ANDROID

- Download SRS Tic Tac Toe APK file from SameAPK.com, then follow these steps:

UPDATE PHONE SETTINGS
- Go to your phone Settings page
- Tap Security or Applications (varies with device)
- Check the Unknown Sources box
- Confirm with OK

GO TO DOWNLOADS
- Open Downloads on your device by going to My Files or Files

- Tap Install when prompted, the APK file you downloaded will be installed on your device.

HOW TO INSTALL TIC TAC TOE APK ON WINDOWS 7/8/9/10 OR MAC PC?

- Download Tic Tac Toe APK file from SameAPK.com to your PC (ex: /Users/xxx/Downloads/), then follow these steps:

USING EMULATOR

- Download And Install one Emulator Softwares (Ex: Bluestacks, GenyMoti on, NoxPlayer)

SOURCE CODE:

activity_main.xml:

```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:text=" Tic Tac Toe "
        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"
```

```xml
    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"
```

```xml
        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"
```

```xml
                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>
```

MainActivity.java:

```java
package com.example.tictactoe;

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
```

```java
// 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);
```

```java
            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;
```

```java
      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);
   }
}
```
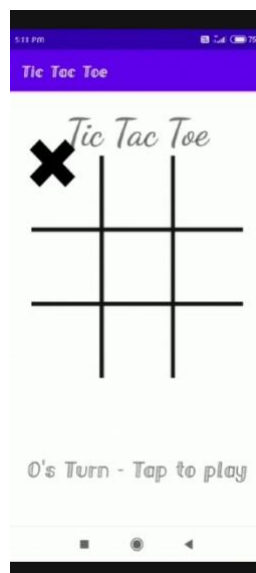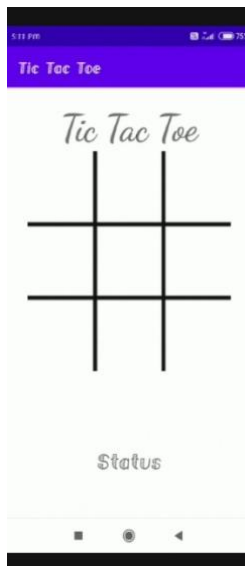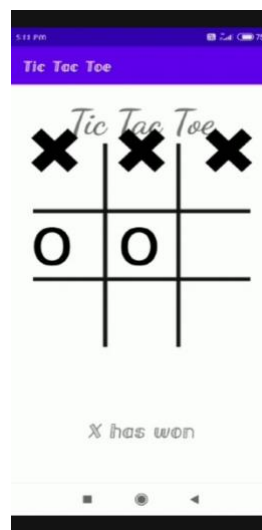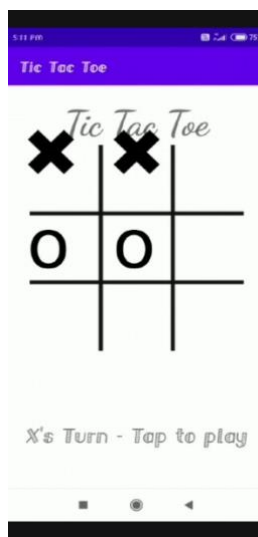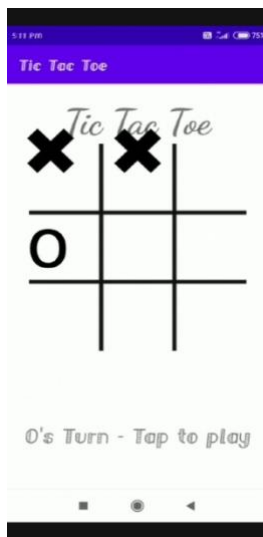
OUTPUT:
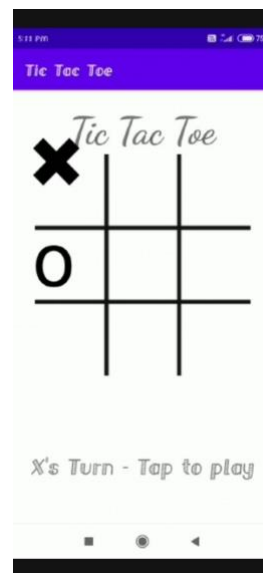
RESULT:

Thus the project for creating tic tac toe was executed successfully.