

Tic Tac Toe Game

Individual Project

Name: ZHOU Ziyang

student ID: 55921210

Email: ziyanzhou5-c@my.cityu.edu.hk

Date: 2020/3/10

Here is the report of my “Tic Tac Toe” individual project.

I will introduce my project in modules respectively. The complete codes is attached at the end of report. If there is any problem in report or project source file, please contact me. Thank you!

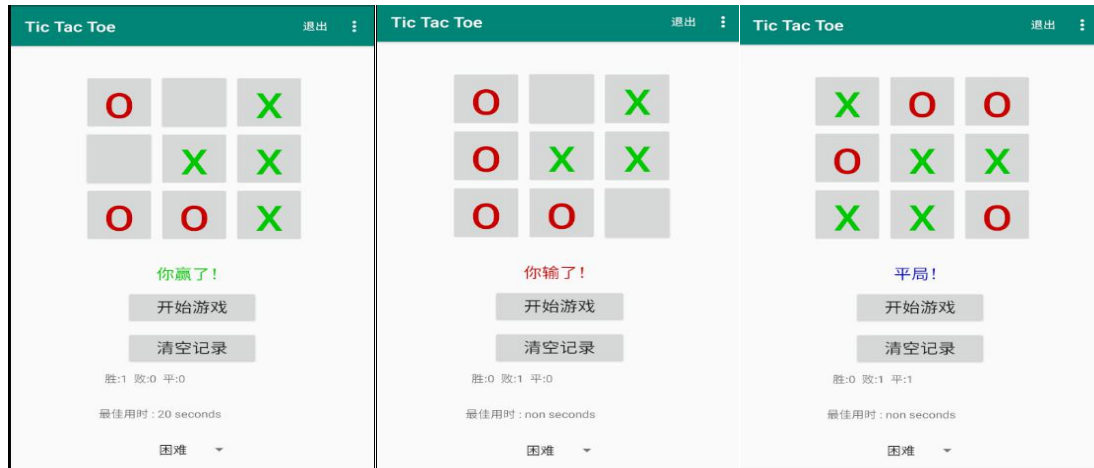
Catalog:

I.	“Tic Tac Toe” game implementation	2
II.	Support multiple language	2
III.	Support portrait and landscape display mode	3
IV.	Android and players take turns to start the game first, and game record	8
V.	Create option menu with difficulty choose and game exit	9
VI.	Adding sound affects to the game	11
VII.	Adding history time record	12
VIII.	Difficulty choose in game view	13
IX.	History record clear	14
X.	User preference	15
	Attachment (full codes of files)	16

I . “Tic Tac Toe” game implementation

This part includes the basic functions of “Tic Tac Toe” game. The code is given so I omit this part here.

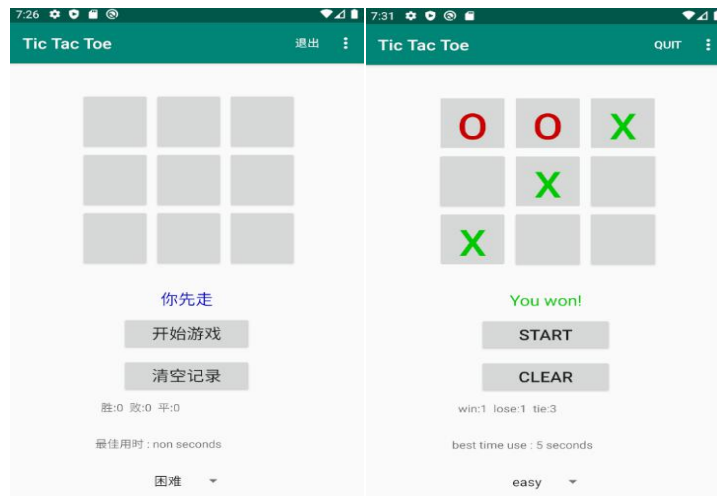
Here are screenshots of “Tic Tac Toe” game.



II . support multiple language

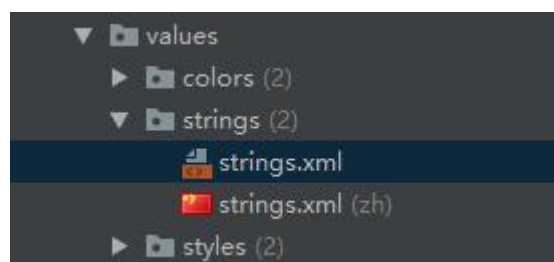
My project support English and Chinese.

Screenshots:



Code and Set up:

In res->values->strings, there are two strings.xml files to support English and Chinese.



In strings.xml :

```
<resources>
  <string name="app_name">Tic Tac Toe</string>
  <string name="your_turn">It is your turn!</string>
```

```

<string name="android turn">It is Android\'s turn!</string>
<string name="tie">It\'s a tie!</string>
<string name="won">You won!</string>
<string name="lose">You lose!</string>
<string name="begin">You go first.</string>
<string name="restart">Start</string>
<string name="a"> win:</string>
<string name="b"> lose:</string>
<string name="c"> tie:</string>
<string name="difficulty">difficulty</string>
<string name="clear">Clear</string>
<string name="quit">quit</string>
<string name="easy">easy</string>
<string name="hard">hard</string>
<string name="expert">expert</string>
<string name="bestTime">best time use :</string>
<string-array name="level">
    <item>easy</item>
    <item>hard</item>
    <item>expert</item>
</string-array>
</resources>

```

ln strings.xml(ch)

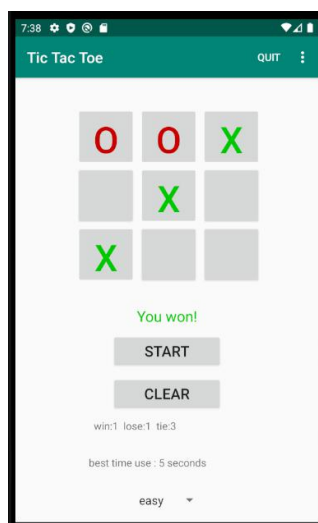
```

<resources>
    <string name="app name">Tic Tac Toe</string>
    <string name="your turn">轮到你了</string>
    <string name="android turn">轮到安卓了</string>
    <string name="tie">平局! </string>
    <string name="won">你赢了! </string>
    <string name="lose">你输了! </string>
    <string name="begin">你先走</string>
    <string name="restart">开始游戏</string>
    <string name="a"> 胜:</string>
    <string name="b"> 败:</string>
    <string name="c"> 平:</string>
    <string name="difficulty">难度</string>
    <string name="clear">清空记录</string>
    <string name="quit">退出</string>
    <string name="easy">简单</string>
    <string name="hard">困难</string>
    <string name="expert">专家</string>
    <string name="bestTime">最佳用时 :</string>
    <string-array name="level">
        <item>简单</item>
        <item>困难</item>
        <item>专家</item>
    </string-array>
</resources>

```

III . support portrait and landscape display mode

Screenshots:



Layout xml code:

In portrait xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:padding="10dp" >
    <TableLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:layout_gravity="center_horizontal" >
        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/button0"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:gravity="center"
                android:text="1"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button1"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="2"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button2"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_gravity="center_vertical"
                android:layout_marginLeft="5dp"
                android:text="3"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
        </TableRow>
        <TableRow
            android:id="@+id/tableRow2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/button3"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="4"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button4"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="5"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button5"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="6"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
        </TableRow>
        <TableRow
            android:id="@+id/tableRow3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/button6"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:onClick="onButtonClicked"
                android:text="7"
                android:textSize="50sp" />
        </TableRow>
    </TableLayout>
</LinearLayout>
```

```

        android:textSize="50sp" />

        <Button
            android:id="@+id/button7"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:layout_marginLeft="5dp"
            android:text="8"
            android:onClick="onButtonClicked"
            android:textSize="50sp" />

        <Button
            android:id="@+id/button8"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:layout_marginLeft="5dp"
            android:text="9"
            android:onClick="onButtonClicked"
            android:textSize="50sp" />

    </TableRow>
</TableLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:orientation="vertical"
    android:padding="10dp">

    <TextView
        android:id="@+id/information"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:gravity="center_horizontal"
        android:text="info"
        android:textSize="20sp"></TextView>

    <Button
        android:id="@+id/button_restart"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="10dp"
        android:onClick="newGame"
        android:text="@string/restart"
        android:textSize="20sp"></Button>

    <Button
        android:id="@+id/button_clear"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="10dp"
        android:text="@string/clear"
        android:textSize="20sp"
        android:onClick="Clear"></Button>

    <TextView
        android:id="@+id/show_results"
        android:layout_width="200dp"
        android:layout_height="40dp"
        android:layout_marginLeft="80dp"
        android:layout_marginTop="10dp"
    ></TextView>

    <TextView
        android:id="@+id/show_bestTime"
        android:layout_width="200dp"
        android:layout_height="40dp"
        android:layout_marginLeft="80dp"
        android:layout_marginTop="10dp"
    ></TextView>

    <Spinner
        android:id="@+id/Level"
        android:entries="@array/level"
        android:tag="@string/difficulty"
        android:layout_width="100dp"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="140dp"
        android:layout_height="wrap_content" />

</LinearLayout>
</LinearLayout>

```

In landscape xml file:

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TableLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:layout_marginLeft="100dp"
        android:layout_gravity="center_horizontal" >
        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="200dp"
            android:layout_height="200dp" >
            <Button
                android:id="@+id/button0"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:gravity="center"
                android:text="1"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button1"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="2"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button2"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_gravity="center_vertical"
                android:layout_marginLeft="5dp"
                android:text="3"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
        </TableRow>
        <TableRow
            android:id="@+id/tableRow2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/button3"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="4"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button4"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="5"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button5"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="6"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
        </TableRow>
        <TableRow
            android:id="@+id/tableRow3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/button6"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="7"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
```

```

        android:id="@+id/button7"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:layout_marginLeft="5dp"
        android:text="8"
        android:onClick="onButtonClicked"
        android:textSize="50sp" />
    <Button
        android:id="@+id/button8"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:layout_marginLeft="5dp"
        android:text="9"
        android:onClick="onButtonClicked"
        android:textSize="50sp" />
    </TableRow>
</TableLayout>
<LinearLayout
    android:layout_width="200dp"
    android:layout_height="340dp"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="20dp"
    android:orientation="vertical"
    android:layout_marginLeft="400dp"
    android:padding="10dp" >
    <TextView
        android:id="@+id/information"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:gravity="center_horizontal"
        android:text="info"
        android:textSize="20sp" >
    </TextView>
    <Button
        android:id="@+id/button_restart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:text="@string/restart"
        android:textSize="20sp"
        android:onClick="newGame" >
    </Button>
    <Button
        android:id="@+id/button_clear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:text="@string/clear"
        android:textSize="20sp"
        android:onClick="Clear"></Button>

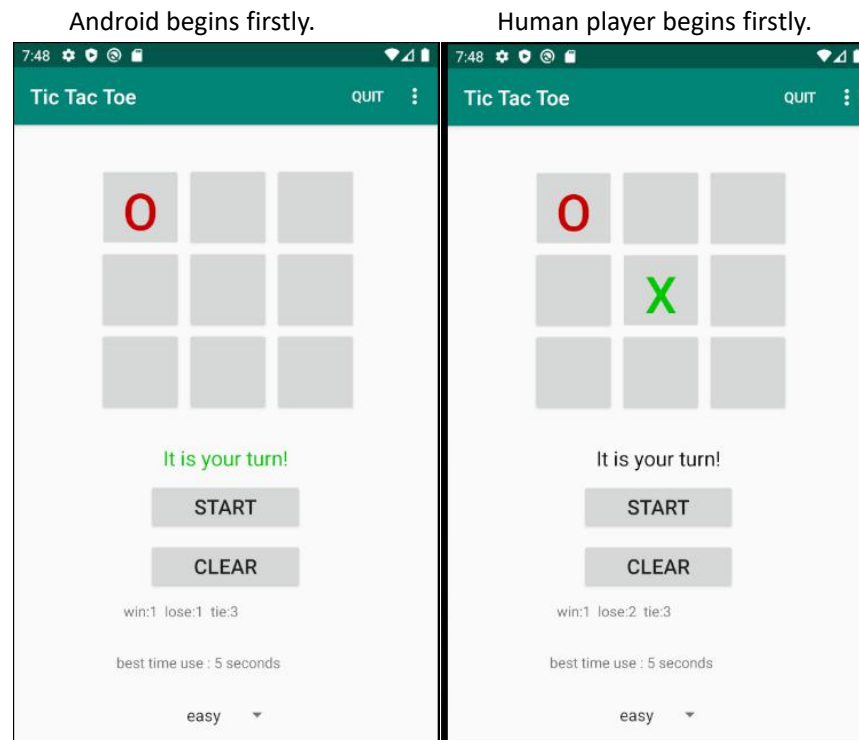
    <TextView
        android:id="@+id/show_results"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
    ></TextView>
    <TextView
        android:id="@+id/show_bestTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
    ></TextView>

    <Spinner
        android:id="@+id/Level"
        android:entries="@array/level"
        android:tag="@string/difficulty"
        android:layout_width="100dp"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="10dp"
        android:layout_height="wrap_content" />
</LinearLayout>
</RelativeLayout>

```

IV . Android and players take turns to start the game first, and game record

Screenshots:



Game history record is behind



Code in MainActivity :

```
// game record
private Button clearRecord;
private int a,b,c,first;
private String A,B,C;
```

In Oncreate() method: decide who first

```
//---Human goes first
if (first==0)
{
    first=1;
    mInfoTextView.setText(R.string.begin);
}
else // android goes first
{
    first=0;
    int location=mGame.getComputerMove(ch);
    mInfoTextView.setText(R.string.your_turn);
    setMove(TicTacToeGame.COMPUTER_PLAYER, location);
}
```


Outside the Oncreate method: show game history record

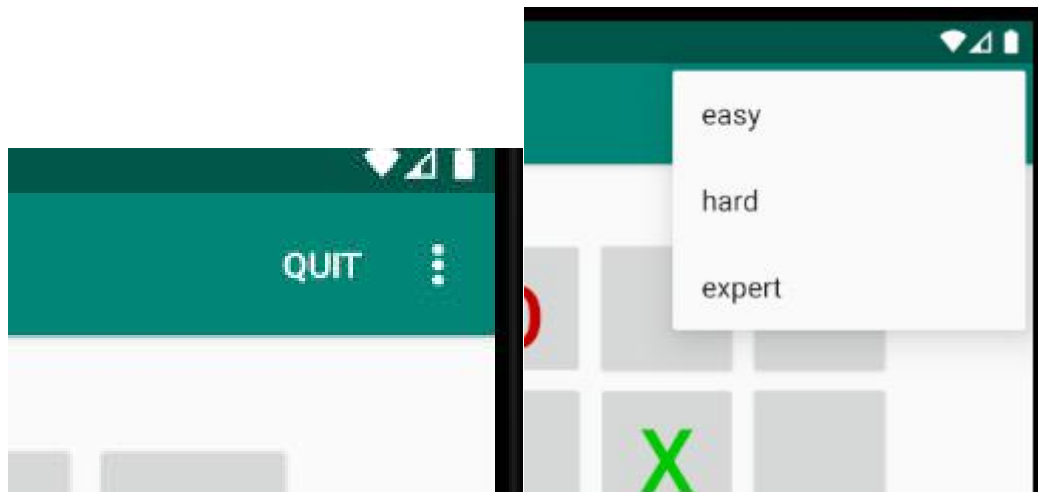
```
private void makeRecord()
{
    String x;
    if (bestTime==-1)
    {
        x=getResources().getString(R.string.bestTime)+" non seconds";
    }
    else
    {
        x=getResources().getString(R.string.bestTime)+" "+bestTime+" seconds";
    }
    bestRecord.setText(x);

    A=getResources().getString(R.string.a);
    B=getResources().getString(R.string.b);
    C=getResources().getString(R.string.c);

    String S=" "+A+a+" "+B+b+" "+C+c;
    showResults.setText(S);
    savePreferences();
}
```

V . Create option memu with difficulty choose and game exit

Screenshots:



Codes of memu:

Codes in res->memu->quit.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu 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"
    tools:context=".MainActivity">
    <item
        android:id="@+id/menu easy"
        android:title="@string/easy" />
    <item
        android:id="@+id/menu hard"
        android:title="@string/hard" />
    <item
        android:id="@+id/menu expert"
        android:title="@string/expert" />
    <item
        android:id="@+id/Quit"
        android:title="@string/quit"
        app:showAsAction="ifRoom" />
</menu>
```

Codes in MainActivity:

Outside of Oncreate() method:

```
// difficulty choose and game exit
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present
    getMenuInflater().inflate(R.menu.quit, menu);
    return true; }
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    switch (id) {
        case R.id.Quit:
            finish();
            return true;
        case R.id.menu_easy:
            ch=1;
            return true;
        case R.id.menu_hard:
            ch=2;
            return true;
        case R.id.menu_expert:
            ch=3;
            return true;
    }
    return false; }
```

I modify the original `getComputerMove()` function in `TicTacToeGame.java`. Now it needs a parameter of difficulty level to get next movement. (1: easy 2: difficulty 3: expert)

AI with different difficulties in TicTacToeGame.java :

```
public int getComputerMove(int level)
{
    // 1: easy 2: difficulty 3: expert
    int move;
    if(level==1)
    {
        for (int i = 0; i < BOARD_SIZE; i++) {
            if (mBoard[i] != HUMAN_PLAYER && mBoard[i] != COMPUTER_PLAYER) {
                mBoard[i] = COMPUTER_PLAYER;
                return i;
            }
        }
    }

    // First see if there's a move O can make to win
    for (int i = 0; i < BOARD_SIZE; i++) {
        if (mBoard[i] != HUMAN_PLAYER && mBoard[i] != COMPUTER_PLAYER) {
            char curr = mBoard[i];
            mBoard[i] = COMPUTER_PLAYER;
            if (checkForWinner() == 3) {
                System.out.println("Computer is moving to " + (i + 1));
                return i;
            }
            else
                mBoard[i] = curr;
        }
    }

    // See if there's a move O can make to block X from winning
    for (int i = 0; i < BOARD_SIZE; i++) {
        if (mBoard[i] != HUMAN_PLAYER && mBoard[i] != COMPUTER_PLAYER) {
            char curr = mBoard[i]; // Save the current number
            mBoard[i] = HUMAN_PLAYER;
            if (checkForWinner() == 2) {
                mBoard[i] = COMPUTER_PLAYER;
                System.out.println("Computer is moving to " + (i + 1));
                return i;
            }
            else
                mBoard[i] = curr;
        }
    }

    // Generate random move
    if(level == 2)
    {
        do
        {
            move = mRand.nextInt(BOARD_SIZE);
        }
    }
}
```

```

    } while (mBoard[move] == HUMAN_PLAYER || mBoard[move] == COMPUTER_PLAYER);
    mBoard[move] = COMPUTER_PLAYER;
    return move;
}
else
{
    for (int i=0;i<9;i++)
    {
        if (mBoard[X[i]] != HUMAN_PLAYER && mBoard[X[i]] != COMPUTER_PLAYER) {
            mBoard[X[i]] = COMPUTER_PLAYER;
            return X[i];
        }
    }
    return 0;
}
}

```

VI . adding sound affects to the game

In my project, there are three kinds of sound to play when you win , lose and tie respectively. Select and download three feasible music files and put them into **res->raw folder**. Then we can use them to play in app.

Screenshots: non

Codes in MainActivity:

```

// button music
private SoundPool soundPool;
private int soundIDwin, soundIDtie, soundIDlose;
private MediaPlayer music;

private void PlayMusic(int MusicId) {
    music = MediaPlayer.create(this, MusicId);
    music.start();
}

@SuppressWarnings("NewApi")
private void initSound() {
    soundPool = new SoundPool.Builder().build();
    soundIDwin = soundPool.load(this, R.raw.win, 1);
    soundIDtie = soundPool.load(this, R.raw.tie, 1);
    soundIDlose = soundPool.load(this, R.raw.lose, 1);
}

private void playSoundWin() {
    soundPool.play(soundIDwin, 1f, 1f, 0, 0, 1);
}

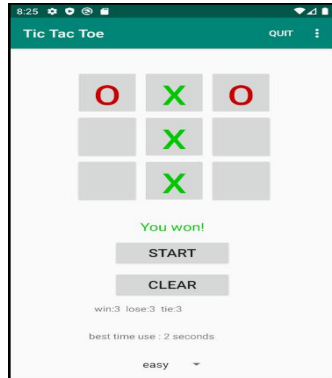
private void playSoundTie() {
    soundPool.play(soundIDtie, 1f, 1f, 0, 0, 1);
}

private void playSoundLose() {
    soundPool.play(soundIDlose, 1f, 1f, 0, 0, 1);
}

```

VII . adding history time record

When player win, there will be a time he spend. If his time is less than the best record, the best record will be replaced and updated in TextView.



Codes in MainActivity:

```
// time record
Calendar calendars;
private int bestTime, startTime, endTime;
private TextView bestRecord;

private int getTime() {
    calendars = Calendar.getInstance();
    calendars.setTimeZone(TimeZone.getTimeZone("GMT+8:00"));
    int min = Integer.parseInt(String.valueOf(calendars.get(Calendar.MINUTE)));
    int second = Integer.parseInt(String.valueOf(calendars.get(Calendar.SECOND)));
    return min*60+second;
}

private void updateTimeRecord()
{
    endTime=getTime();
    if (startTime>endTime) endTime+=3600;
    if (endTime-startTime<bestTime||bestTime==--1)
    {
        bestTime=endTime-startTime;
        makeRecord();
    }
}

private void makeRecord()
{
    String x;
    if (bestTime==--1)
    {
        x=getResources().getString(R.string.bestTime)+" non seconds";
    }
    else
    {
        x=getResources().getString(R.string.bestTime)+" "+bestTime+" seconds";
    }
    bestRecord.setText(x);

    A=getResources().getString(R.string.a);
    B=getResources().getString(R.string.b);
    C=getResources().getString(R.string.c);

    String S=" "+A+a+" "+B+b+" "+C+c;
    showResults.setText(S);
    savePreferences();
}
```

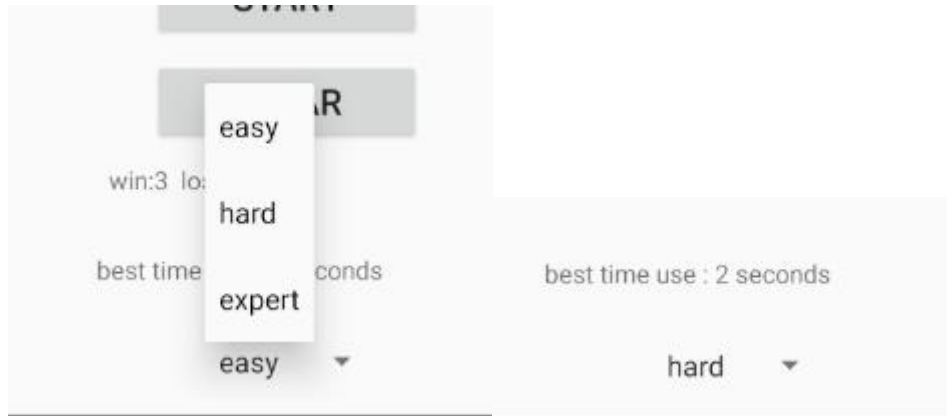
In newGame() method:

```
startTime=getTime();
```

VIII . difficulty choose in game view

I use Spinner in game view to choose game difficulty.

Screenshots:



Codes in layout:

```
<Spinner
    android:id="@+id/Level"
    android:entries="@array/level"
    android:tag="@string/difficulty"
    android:layout_width="100dp"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="140dp"
    android:layout_height="wrap_content" />
```

Codes in string.xml

```
<string-array name="level">
    <item>easy</item>
    <item>hard</item>
    <item>expert</item>
</string-array>
```

Codes in MainActivity.java

In Oncreate() method:

```
choose = (Spinner) findViewById(R.id.Level);
choose.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        //获取到 Spinner 下列选项值
        String val = choose.getItemAtPosition(position).toString();
        //Toast.makeText(MainActivity.this, val, Toast.LENGTH_LONG).show();

        if (val.equals("简单") || val.equals("easy")) ch=1;
        else if (val.equals("困难") || val.equals("hard")) ch=2;
        else if (val.equals("专家") || val.equals("expert")) ch=3;
        else
        {
            Toast.makeText(MainActivity.this, "wrong!", Toast.LENGTH_LONG).show();
        }
    }

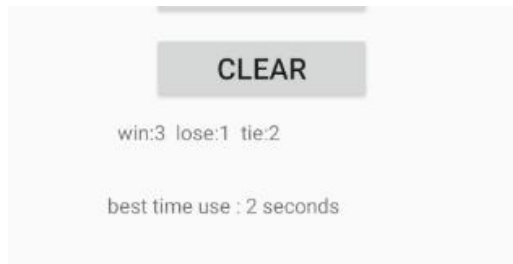
    @Override
    public void onNothingSelected(AdapterView<?> parent) {
        ch=1;
    }
});
```

IX . history record clear

I set a button to clear all history records, including win-lose-tie data and best spending time record.

Screenshots:

Before click "CLEAR"



after click "CLEAR"



Codes in MainActivity.java

```
// game record  
private Button clearRecord;  
private int a,b,c first;  
private String A,B,C;
```

```
// clear record  
public void Clear(View v)  
{  
    a=0;  
    b=0;  
    c=0;  
    bestTime=-1;  
    makeRecord();  
}
```

Codes in Layout:

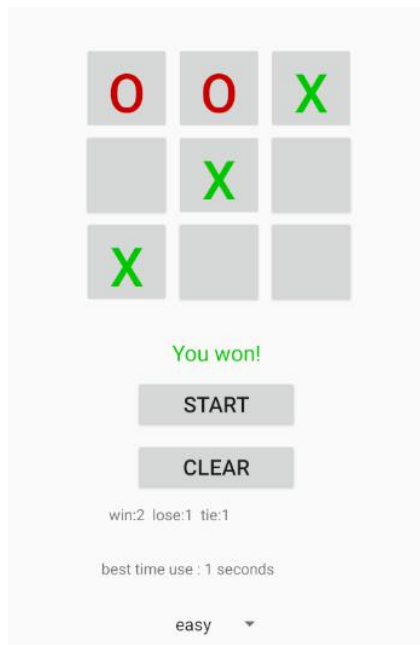
```
<Button  
    android:id="@+id/button_clear"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal"  
    android:layout_marginTop="20dp"  
    android:text="@string/clear"  
    android:textSize="20sp"  
    android:onClick="Clear"></Button>
```

X . user preference

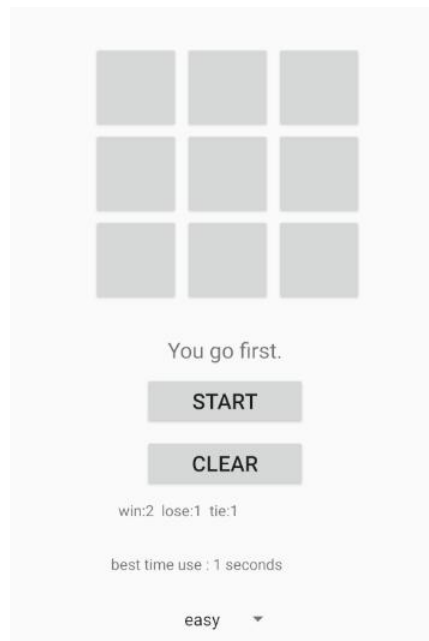
I set the user preference to show last data record includes win-lose-tie record and best-time record.

Screenshot :

Before I close the game



close the game and open it again



Codes in MainActivity.java

```
public void savePreferences() {
    SharedPreferences pref = getSharedPreferences("TicTac", MODE_PRIVATE);
    String a=String.valueOf(a);
    String b=String.valueOf(b);
    String c=String.valueOf(c);
    String t=String.valueOf(bestTime);
    pref.edit().putString("a", a).apply();
    pref.edit().putString("b", b).apply();
    pref.edit().putString("c", c).apply();
    pref.edit().putString("t", t).apply();
}

public void loadPreferences() {
    SharedPreferences pref = getSharedPreferences("TicTac", MODE_PRIVATE);
    a=Integer.parseInt(pref.getString("a", "0"));
    b=Integer.parseInt(pref.getString("b", "0"));
    c=Integer.parseInt(pref.getString("c", "0"));
    bestTime=Integer.parseInt(pref.getString("t", "-1"));
}

@Override
protected void onStart() {
    super.onStart();
    loadPreferences();
}
```

Codes in Oncreate() method:

```
loadPreferences();
```

Codes in makeRecord()

```
savePreferences();
```

Attachment

Complete codes in files

MainActivity.java

```
package com.example.tictactoe;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.media.MediaPlayer;
import android.media.SoundPool;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Calendar;
import java.util.TimeZone;

public class MainActivity extends AppCompatActivity {
    // Represents the internal state of the game

    private TicTacToeGame mGame;
    private Button mBoardButtons[];
    // Various text displayed
    private TextView mInfoTextView;
    private TextView showResults;

    // game record
    private Button clearRecord;
    private int a,b,c,first;
    private String A,B,C;

    // Restart Button
    private Button startButton;
    Boolean mGameOver;

    // difficulty choose
    private Spinner choose;
    private int ch;

    // button music
    private SoundPool soundPool;
    private int soundIDwin,soundIDtie,soundIDlose;
    private MediaPlayer music;

    // time record
    Calendar calendars;
    private int bestTime,startTime,endTime;
    private TextView bestRecord;

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

        initSound();

        startTime=getTime();
        loadPreferences();
        first=0;
        bestRecord=(TextView)findViewById(R.id.show_bestTime);
        showResults=(TextView) findViewById(R.id.show_results);

        makeRecord();

        choose = (Spinner) findViewById(R.id.Level);
        choose.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
```



```

@Override
public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
    //获取到 Spinner 下列选项值
    String val = choose.getItemAtPosition(position).toString();
    //Toast.makeText(MainActivity.this, val, Toast.LENGTH_LONG).show();

    if (val.equals("简单") || val.equals("easy")) ch=1;
    else if (val.equals("困难") || val.equals("hard")) ch=2;
    else if (val.equals("专家") || val.equals("expert")) ch=3;
    else
    {
        Toast.makeText(MainActivity.this, "wrong!", Toast.LENGTH_LONG).show();
    }
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
    ch=1;
}
});

// Buttons making up the board

mGame = new TicTacToeGame();

mBoardButtons = new Button[mGame.BOARD_SIZE];
mBoardButtons[0] = (Button) findViewById(R.id.button0);
mBoardButtons[1] = (Button) findViewById(R.id.button1);
mBoardButtons[2] = (Button) findViewById(R.id.button2);
mBoardButtons[3] = (Button) findViewById(R.id.button3);
mBoardButtons[4] = (Button) findViewById(R.id.button4);
mBoardButtons[5] = (Button) findViewById(R.id.button5);
mBoardButtons[6] = (Button) findViewById(R.id.button6);
mBoardButtons[7] = (Button) findViewById(R.id.button7);
mBoardButtons[8] = (Button) findViewById(R.id.button8);
mInfoTextView = (TextView) findViewById(R.id.information);
mGame = new TicTacToeGame();

startNewGame();
}

//--- Set up the game board.
private void startNewGame() {
    mGameOver = false;
    mGame.clearBoard();
    //---Reset all buttons
    for (int i = 0; i < mBoardButtons.length; i++) {
        mBoardButtons[i].setText("");
        mBoardButtons[i].setEnabled(true);
        mBoardButtons[i].setOnClickListener(new ButtonClickListener(i));
    }
    //---Human goes first
    if (first==0)
    {
        first=1;
        mInfoTextView.setText(R.string.begin);
    }
    else // android goes first
    {
        first=0;
        int location=mGame.getComputerMove(ch);
        mInfoTextView.setText(R.string.your_turn);
        setMove(TicTacToeGame.COMPUTER_PLAYER, location);
    }
}

private class ButtonClickListener implements View.OnClickListener {
    int location;

    public ButtonClickListener(int location) {
        this.location = location;
    }

    @Override
    public void onClick(View v) {
        if (mGameOver == false) {
            if (mBoardButtons[location].isEnabled()) {
                setMove(TicTacToeGame.HUMAN_PLAYER, location);
                //--- If no winner yet, let the computer make a move
                int winner = mGame.checkForWinner();
                if (winner == 0) {
                    mInfoTextView.setText(R.string.android_turn);
                    int move = mGame.getComputerMove(ch);
                    setMove(TicTacToeGame.COMPUTER_PLAYER, move);
                    winner = mGame.checkForWinner();
                }
            }
        }
    }
}

```

```

        }
        if (winner == 0) {
            mInfoTextView.setTextColor(Color.rgb(0, 0, 0));
            mInfoTextView.setText(R.string.your_turn);
        } else if (winner == 1) {
            playSoundLie();
            c+=1;
            mInfoTextView.setTextColor(Color.rgb(0, 0, 200));
            mInfoTextView.setText(R.string.tie);
            makeRecord();
            mGameOver = true;
        } else if (winner == 2) {
            updateTimeRecord();
            playSoundWin();
            a+=1;
            mInfoTextView.setTextColor(Color.rgb(0, 200, 0));
            mInfoTextView.setText(R.string.won);
            makeRecord();
            mGameOver = true;
        } else {
            playSoundLose();
            b+=1;
            mInfoTextView.setTextColor(Color.rgb(200, 0, 0));
            mInfoTextView.setText(R.string.lose);
            makeRecord();
            mGameOver = true;
        }
    }
}

private void setMove(char player, int location) {
    mGame.setMove(player, location);
    mBoardButtons[location].setEnabled(false);
    mBoardButtons[location].setText(String.valueOf(player));
    if (player == TicTacToeGame.HUMAN_PLAYER)
        mBoardButtons[location].setTextColor(Color.rgb(0, 200, 0));
    else
        mBoardButtons[location].setTextColor(Color.rgb(200, 0, 0));
}

//--- OnClickListener for Restart a New Game Button
public void newGame(View v) {
    startTime=getTime();
    startNewGame();
}

// clear record
public void Clear(View v)
{
    a=0;
    b=0;
    c=0;
    bestTime=-1;
    makeRecord();
}

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

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    switch (id) {
        case R.id.Quit:
            finish();
            return true;
        case R.id.menu_easy:
            ch=1;
            return true;
        case R.id.menu_hard:
            ch=2;
            return true;
        case R.id.menu_expert:
            ch=3;
            return true;
    }
    return false; }

private void PlayMusic(int MusicId) {
    music = MediaPlayer.create(this, MusicId);
    music.start();
}

```

```

@SuppressLint("NewApi")
private void initSound() {
    soundPool = new SoundPool.Builder().build();
    soundIDwin = soundPool.load(this, R.raw.win, 1);
    soundIDtie = soundPool.load(this, R.raw.tie, 1);
    soundIDlose = soundPool.load(this, R.raw.lose, 1);
}

private void playSoundWin() {
    soundPool.play(soundIDwin, 1f, 1f, 0, 0, 1);
}
private void playSoundTie() {
    soundPool.play(soundIDtie, 1f, 1f, 0, 0, 1);
}
private void playSoundLose() {
    soundPool.play(soundIDlose, 1f, 1f, 0, 0, 1);
}

private int getTime() {
    calendars = Calendar.getInstance();
    calendars.setTimeZone(TimeZone.getTimeZone("GMT+8:00"));
    int min = Integer.parseInt(String.valueOf(calendars.get(Calendar.MINUTE)));
    int second = Integer.parseInt(String.valueOf(calendars.get(Calendar.SECOND)));
    return min*60+second;
}

private void updateTimeRecord()
{
    endTime=getTime();
    if (startTime>endTime) endTime+=3600;
    if (endTime-startTime<bestTime||bestTime==-1)
    {
        bestTime=endTime-startTime;
        makeRecord();
    }
}

private void makeRecord()
{
    String x;
    if (bestTime==-1)
    {
        x=getResources().getString(R.string.bestTime)+" non seconds";
    }
    else
    {
        x=getResources().getString(R.string.bestTime)+" "+bestTime+" seconds";
    }
    bestRecord.setText(x);

    A=getResources().getString(R.string.a);
    B=getResources().getString(R.string.b);
    C=getResources().getString(R.string.c);

    String S=" "+A+a+" "+B+b+" "+C+c;
    showResults.setText(S);
    savePreferences();
}

public void savePreferences() {
    SharedPreferences pref = getSharedPreferences("TicTac", MODE_PRIVATE);
    String a=String.valueOf(a);
    String b=String.valueOf(b);
    String c=String.valueOf(c);
    String t=String.valueOf(bestTime);
    pref.edit().putString("a", a).apply();
    pref.edit().putString("b", b).apply();
    pref.edit().putString("c", c).apply();
    pref.edit().putString("t", t).apply();
}

public void loadPreferences() {
    SharedPreferences pref = getSharedPreferences("TicTac", MODE_PRIVATE);
    a=Integer.parseInt(pref.getString("a", "0"));
    b=Integer.parseInt(pref.getString("b", "0"));
    c=Integer.parseInt(pref.getString("c", "0"));
    bestTime=Integer.parseInt(pref.getString("t", "-1"));
}

@Override
protected void onStart() {
    super.onStart();
    loadPreferences();
}

```

TicTacToeGame.java

```
package com.example.tictactoe; /* TicTacToeConsole.java
 * By Frank McCown (Harding University)
 *
 * This is a tic-tac-toe game that runs in the console window. The human
 * is X and the computer is O.
 */

import android.annotation.SuppressLint;
import android.media.MediaPlayer;
import android.media.SoundPool;
import android.widget.Toast;

import java.util.InputMismatchException;
import java.util.Random;
import java.util.Scanner;

public class TicTacToeGame {

    // Characters used to represent the human, computer, and open spots
    public static final char HUMAN_PLAYER = 'X';
    public static final char COMPUTER_PLAYER = 'O';
    public static final char OPEN_SPOT = ' ';
    public static final int BOARD_SIZE = 9;
    private char mBoard[] = { '1', '2', '3', '4', '5', '6', '7', '8', '9' };
    private int X[] = {4, 0, 2, 6, 8, 1, 3, 5, 7};
    private Random mRand;

    public TicTacToeGame() {
        // Seed the random number generator
        mRand = new Random();
        char turn = HUMAN_PLAYER; // Human starts first
        int win = 0; // Set to 1, 2, or 3 when game is over
    }

    public void clearBoard() {
        for (int i = 0; i < BOARD_SIZE; i++) {
            mBoard[i] = OPEN_SPOT;
        }
    }

    /** Set the given player at the given location on the game board */
    public void setMove(char player, int location) {
        mBoard[location] = player;
    }

    // Check for a winner. Return
    // 0 if no winner or tie yet
    // 1 if it's a tie
    // 2 if X won
    // 3 if O won
    public int checkForWinner() {

        // Check horizontal wins
        for (int i = 0; i <= 6; i += 3) {
            if (mBoard[i] == HUMAN_PLAYER &&
                mBoard[i+1] == HUMAN_PLAYER &&
                mBoard[i+2] == HUMAN_PLAYER)
                return 2;
            if (mBoard[i] == COMPUTER_PLAYER &&
                mBoard[i+1] == COMPUTER_PLAYER &&
                mBoard[i+2] == COMPUTER_PLAYER)
                return 3;
        }

        // Check vertical wins
        for (int i = 0; i <= 2; i++) {
            if (mBoard[i] == HUMAN_PLAYER &&
                mBoard[i+3] == HUMAN_PLAYER &&
                mBoard[i+6] == HUMAN_PLAYER)
                return 2;
            if (mBoard[i] == COMPUTER_PLAYER &&
                mBoard[i+3] == COMPUTER_PLAYER &&
                mBoard[i+6] == COMPUTER_PLAYER)
                return 3;
        }

        // Check for diagonal wins
        if ((mBoard[0] == HUMAN_PLAYER &&
            mBoard[4] == HUMAN_PLAYER &&
            mBoard[8] == HUMAN_PLAYER) ||
            (mBoard[2] == HUMAN_PLAYER &&
            mBoard[4] == HUMAN_PLAYER &&
            mBoard[6] == HUMAN_PLAYER))
            return 2;
        if ((mBoard[0] == COMPUTER_PLAYER &&
```

```

        mBoard[4] == COMPUTER_PLAYER &&
        mBoard[8] == COMPUTER_PLAYER ||
        (mBoard[2] == COMPUTER_PLAYER &&
         mBoard[4] == COMPUTER_PLAYER &&
         mBoard[6] == COMPUTER_PLAYER))
    return 3;

    // Check for tie
    for (int i = 0; i < BOARD_SIZE; i++) {
        // If we find a number, then no one has won yet
        if (mBoard[i] != HUMAN_PLAYER && mBoard[i] != COMPUTER_PLAYER)
            return 0;
    }

    // If we make it through the previous loop, all places are taken, so it's a tie
    return 1;
}

public int getComputerMove(int level)
{
    int move;
    if(level==1)
    {
        for (int i = 0; i < BOARD_SIZE; i++) {
            if (mBoard[i] != HUMAN_PLAYER && mBoard[i] != COMPUTER_PLAYER) {
                mBoard[i] = COMPUTER_PLAYER;
                return i;
            }
        }
    }

    // First see if there's a move 0 can make to win
    for (int i = 0; i < BOARD_SIZE; i++) {
        if (mBoard[i] != HUMAN_PLAYER && mBoard[i] != COMPUTER_PLAYER) {
            char curr = mBoard[i];
            mBoard[i] = COMPUTER_PLAYER;
            if (checkForWinner() == 3) {
                System.out.println("Computer is moving to " + (i + 1));
                return i;
            }
            else
                mBoard[i] = curr;
        }
    }

    // See if there's a move 0 can make to block X from winning
    for (int i = 0; i < BOARD_SIZE; i++) {
        if (mBoard[i] != HUMAN_PLAYER && mBoard[i] != COMPUTER_PLAYER) {
            char curr = mBoard[i]; // Save the current number
            mBoard[i] = HUMAN_PLAYER;
            if (checkForWinner() == 2) {
                mBoard[i] = COMPUTER_PLAYER;
                System.out.println("Computer is moving to " + (i + 1));
                return i;
            }
            else
                mBoard[i] = curr;
        }
    }

    // Generate random move
    if(level == 2)
    {
        do
        {
            move = mRand.nextInt(BOARD_SIZE);
        } while (mBoard[move] == HUMAN_PLAYER || mBoard[move] == COMPUTER_PLAYER);
        mBoard[move] = COMPUTER_PLAYER;
        return move;
    }
    else
    {
        for (int i=0;i<9;i++)
        {
            if (mBoard[X[i]] != HUMAN_PLAYER && mBoard[X[i]] != COMPUTER_PLAYER) {
                mBoard[X[i]] = COMPUTER_PLAYER;
                return X[i];
            }
        }
        return 0;
    }
}
}

```

Activity_main.xml (here I omit the layout in landscape display mode)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:padding="10dp" >
    <TableLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal" >
        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/button0"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:gravity="center"
                android:text="1"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button1"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="2"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button2"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_gravity="center_vertical"
                android:layout_marginLeft="5dp"
                android:text="3"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
        </TableRow>
        <TableRow
            android:id="@+id/tableRow2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/button3"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="4"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button4"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="5"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
                android:id="@+id/button5"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="6"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
        </TableRow>
        <TableRow
            android:id="@+id/tableRow3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/button6"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="5dp"
                android:text="7"
                android:onClick="onButtonClicked"
                android:textSize="50sp" />
            <Button
```

```

        android:id="@+id/button7"
        android:layout width="80dp"
        android:layout height="80dp"
        android:layout marginLeft="5dp"
        android:text="8"
        android:onClick="onButtonClicked"
        android:textSize="50sp" />
    <Button
        android:id="@+id/button8"
        android:layout width="80dp"
        android:layout height="80dp"
        android:layout marginLeft="5dp"
        android:text="9"
        android:onClick="onButtonClicked"
        android:textSize="50sp" />
    </TableRow>
</TableRow>
</TableLayout>
<LinearLayout
    android:layout width="match parent"
    android:layout height="wrap content"
    android:layout gravity="center_horizontal"
    android:orientation="vertical"
    android:padding="10dp" >
    <TextView
        android:id="@+id/information"
        android:layout width="fill parent"
        android:layout height="wrap content"
        android:layout marginTop="20dp"
        android:gravity="center_horizontal"
        android:text="info"
        android:textSize="20sp" >
    </TextView>
    <Button
        android:id="@+id/button_restart"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:layout gravity="center_horizontal"
        android:layout marginTop="20dp"
        android:text="Start a New Game"
        android:textSize="20sp"
        android:onClick="newGame" >
    </Button>
    <Button
        android:id="@+id/button_clear"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:layout gravity="center_horizontal"
        android:layout marginTop="20dp"
        android:text="clear data"
        android:textSize="20sp"
        android:onClick="Clear">
    </Button>
    <TextView
        android:id="@+id/show_results"
        android:layout width="200dp"
        android:layout height="40dp"
        android:layout marginLeft="80dp"
        android:layout marginTop="20dp"
    ></TextView>
    <TextView
        android:id="@+id/show_bestTime"
        android:layout width="200dp"
        android:layout height="40dp"
        android:layout marginLeft="80dp"
        android:layout marginTop="20dp"
    ></TextView>
    <Spinner
        android:id="@+id/Level"
        android:entries="@array/level"
        android:tag="@string/difficulty"
        android:layout width="100dp"
        android:layout marginTop="40dp"
        android:layout marginLeft="150dp"
        android:layout_height="wrap_content" />

</LinearLayout>
</LinearLayout>

```

Res->Menu->quit.xml (menu layout)

```
<?xml version="1.0" encoding="utf-8"?>
<menu 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"
    tools:context=".MainActivity">
    <item
        android:id="@+id/menu_easy"
        android:title="@string/easy" />
    <item
        android:id="@+id/menu_hard"
        android:title="@string/hard" />
    <item
        android:id="@+id/menu_expert"
        android:title="@string/expert" />
    <item
        android:id="@+id/Quit"
        android:title="@string/quit"
        app:showAsAction="ifRoom" />
</menu>
```

Res->values->strings->strings.xml (English supporting Edition)

```
<resources>
    <string name="app_name">Tic Tac Toe</string>
    <string name="your_turn">It is your turn!</string>
    <string name="android_turn">It is Android's turn!</string>
    <string name="tie">It's a tie!</string>
    <string name="won">You won!</string>
    <string name="lose">You lose!</string>
    <string name="begin">You go first.</string>
    <string name="restart">Start</string>
    <string name="a"> win:</string>
    <string name="b"> lose:</string>
    <string name="c"> tie:</string>
    <string name="difficulty">difficulty</string>
    <string name="clear">Clear</string>
    <string name="quit">quit</string>
    <string name="easy">easy</string>
    <string name="hard">hard</string>
    <string name="expert">expert</string>
    <string name="bestTime">best time use :</string>
    <string-array name="level">
        <item>easy</item>
        <item>hard</item>
        <item>expert</item>
    </string-array>
</resources>
```

Res->values->strings->strings.xml (Chinese supporting Edition)

```
<resources>
    <string name="app_name">Tic Tac Toe</string>
    <string name="your_turn">轮到你了</string>
    <string name="android_turn">轮到安卓了</string>
    <string name="tie">平局! </string>
    <string name="won">你赢了! </string>
    <string name="lose">你输了! </string>
    <string name="begin">你先走</string>
    <string name="restart">开始游戏</string>
    <string name="a"> 胜:</string>
    <string name="b"> 败:</string>
    <string name="c"> 平:</string>
    <string name="difficulty">难度</string>
    <string name="clear">清空记录</string>
    <string name="quit">退出</string>
    <string name="easy">简单</string>
    <string name="hard">困难</string>
    <string name="expert">专家</string>
    <string name="bestTime">最佳用时 :</string>
    <string-array name="level">
        <item>简单</item>
        <item>困难</item>
        <item>专家</item>
    </string-array>
</resources>
```