

```

/**
 *
 * Creative Commons (CC) 2019 Marcos Vinícius da Silva Santos and Marcos Antonio dos Santos
 *
 * Licensed under the Creative Commons, Version 4.0; you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * https://creativecommons.org/licenses/by-nc-sa/4.0/
 * https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode
 * Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)
 *
 * Licenciado pela Creative Commons, Versão 4.0; você não pode usar este arquivo, exceto em conformidade com a Licença.
 * Você pode obter uma cópia da Licença em
 * https://creativecommons.org/licenses/by-nc-sa/4.0/deed.pt\_BR
 * https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.pt
 * Atribuição-NãoComercial-CompartilhaIgual 4.0 Internacional (CC BY-NC-SA 4.0)
 *
 * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and limitations under the License.
 */

```

## APLICATIVO JOGO DA PALAVRA

### Código res\values\styles.xml

```

<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>

```

### Código res\values\colors.xml

```

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

```

```

<resources>

    <color name="colorPrimary">#008577</color>
    <color name="colorPrimaryDark">#474946</color>
    <color name="colorAccent">#D81B60</color>

    <color name="colorPurple">#AB47BC</color>
    <color name="colorDarkPurple">#641570</color>
    <color name="colorThinMagenta">#F196BA</color>
    <color name="colorRed">#FF0000</color>
    <color name="colorDarkRed">#7E0A0A</color>

    <color name="colorGradientStart">#474946</color>
    <color name="colorGradientCenter">#AB47BC</color>
    <color name="colorGradientEnd">#181818</color>

</resources>

```

### Código res\anim\smalltobig.xml

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:startOffset="300"
    >

    <scale
        android:fromYScale="0.0"
        android:fromXScale="0.0"
        android:toYScale="1.0"
        android:toXScale="1.0"
        android:duration="400"

```

```
android:pivotY="50%"
android:pivotX="50%"
/>
```

```
</set>
```

### Arquivo de imagem res\drawable\marcos.png



### Código res\drawable\toast\background\_image.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">

    <item>
        <shape xmlns:android="http://schemas.android.com/apk/res/android"
            android:shape="rectangle"
            >
            <corners android:bottomRightRadius="10dp"
                android:radius="10dp"
            />
            <gradient android:angle="45"
                android:endColor="@color/colorDarkRed"
                android:startColor="@color/colorRed"
                android:gradientRadius="270"
```

```
    android:type="linear"
```

```
  />
```

```
    <size android:width="123dp"
```

```
        android:height="41dp"/>
```

```
    <stroke android:dashGap="2dp"
```

```
        android:dashWidth="2dp"
```

```
        android:width="2dp"
```

```
        android:color="#FFFFFF"/>
```

```
  </shape>
```

```
</item>
```

```
<item android:drawable="@drawable/marcos"
```

```
    android:width="32dp"
```

```
    android:height="32dp"
```

```
    android:left="2dp"
```

```
    android:gravity="center_vertical">
```

```
</item>
```

```
</layer-list>
```

### **Código arquivo res\drawable\back.xml**

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

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:shape="rectangle">
```

```

<solid android:color="@android:color/white"/>

<stroke android:width="2dp"
    android:color="@color/colorThinMagenta"/>

<corners android:radius="8dp"/>

<size android:width="61dp"
    android:height="68dp"/>

</shape>

```

#### **Código archivo res\drawable\bgapp.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">

    <solid android:color="@android:color/white"/>

    <stroke android:width="2dp"
        android:color="@color/colorThinMagenta"/>

    <corners android:radius="8dp"/>

    <size android:width="61dp"
        android:height="68dp"/>

</shape>

```

**Código arquivo res\layout\activity\_main.xml**

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

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:id="@+id/coordinatorLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgapp">

    <ScrollView

        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout

            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical" >

            <TextView

                android:id="@+id/textViewScreen"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_marginTop="30dp"
                android:text="Vamos Começar!"
                android:textColor="#FFF"
                android:textSize="22sp" />
```

```
<LinearLayout
    android:layout_width="290dp"
    android:layout_height="150dp"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"
    android:background="@drawable/bgquestion"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textViewQuestion"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="24dp"
        android:lineSpacingExtra="2dp"
        android:text="Palavra representa uma fruta em português"
        android:textAlignment="center"
        android:textColor="#AB47BC"
        android:textSize="22sp" />

</LinearLayout>

<EditText
    android:id="@+id/editTextInput"
    android:layout_width="290dp"
    android:layout_height="80dp"
    android:layout_gravity="center"
    android:layout_marginTop="30dp"
    android:layout_marginBottom="10dp"
```

```
android:background="@drawable/bgpurple"  
android:clickable="false"  
android:focusable="false"  
android:gravity="center"  
android:padding="10dp"  
android:textColor="#AB47BC"  
android:textSize="32sp" />
```

<TextView

```
    android:text="Letras"  
    android:textSize="24dp"  
    android:textColor="#FFF"  
    android:id="@+id/textViewTitle"  
    android:layout_marginRight="28dp"  
    android:layout_marginLeft="28dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="16dp"/>
```

<LinearLayout

```
    android:id="@+id/layoutParentLine1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="24dp"  
    android:layout_marginRight="24dp"  
    android:layout_marginBottom="16dp"  
    android:orientation="horizontal" />
```

<LinearLayout



```
android:id="@+id/layoutParentLine2"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginLeft="24dp"  
android:layout_marginRight="24dp"  
android:layout_marginBottom="16dp"  
android:orientation="horizontal" />
```

```
<LinearLayout
```

```
    android:id="@+id/layoutParentLine3"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="24dp"  
    android:layout_marginRight="24dp"  
    android:layout_marginBottom="16dp"  
    android:orientation="horizontal" />
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

### **Código archivo res\layout\activity\_show\_correct.xml**

```
<?xml version="1.0" encoding="utf-8"?>  
  
<LinearLayout  
    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:orientation="vertical"  
android:background="@drawable/bgapp"  
tools:context=".MainActivity">
```

```
<TextView  
    android:id="@+id/textScreen"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_marginTop="40dp"  
    android:text="Ponto de Verificação"  
    android:textColor="#FFF"  
    android:textSize="22sp" />
```

```
<ImageView  
    android:id="@+id/bigboss"  
    android:layout_width="180dp"  
    android:layout_height="180dp"  
    android:layout_gravity="center"  
    android:layout_marginTop="30dp"  
    android:src="@drawable/marcos" />
```

```
<TextView  
    android:id="@+id/textQuestion"  
    android:layout_width="230dp"
```

```
android:layout_height="wrap_content"
android:layout_gravity="center"
android:layout_marginTop="20dp"
android:text="Aí sim! Parabéns! Vamos para a próxima?"
android:textAlignment="center"
android:textColor="#FFF"
android:textSize="28sp" />
```

<Button

```
android:id="@+id/textTitle"
android:layout_width="280dp"
android:layout_height="70dp"
android:layout_gravity="center"
android:layout_marginTop="40dp"
android:background="@drawable/bgbtn"
android:paddingBottom="10dp"
android:text="Continuar"
android:textAllCaps="false"
android:textColor="@color/colorPurple"
android:textSize="28sp" />
```

<TextView

```
android:id="@+id/textBtn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:layout_marginTop="30dp"
android:text="Voltar ao Início"
android:textColor="#AD87FF"
android:textSize="22sp" />
```

```
</LinearLayout>
```

### **Código arquivo MainActivity.java**

```
package com.example.jogodapalavra;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import org.w3c.dom.Text;

import java.util.ArrayList;
import java.util.Random;

public class MainActivity extends AppCompatActivity {

    private LinearLayout mLinearLayoutLine1, mLinearLayoutLine2, mLinearLayoutLine3;
    private EditText mEditTextInput;
    private TextView mTextViewScreen, mTextViewQuestion, mTextViewTitle;

    private Random rnd = new Random();
```

```
private int pressCounter = 0;
private int maxPresCounter = 0;
private int lengthKeysEmpty = 0;

private String[] keys;
ArrayList<String> mStringArrayList = new ArrayList<String>();

private String textAnswer = "PITANGATUBA"; // uva laranja

private void adicionarLetraExtra(){
    String[] listaLetras = {"X", "Y", "Z", "W", "K", "B", "G", "E" };
    int posicao = rnd.nextInt( listaLetras.length + 1 );
    mStringArrayList.add( listaLetras[posicao] );
}

private void adicionarEspacosVazios(String textAnswer){
    lengthKeysEmpty=0;
    if( textAnswer.length() < 5 ){
        lengthKeysEmpty = 5 - textAnswer.length() -1;
    } else if( textAnswer.length() < 10 ){
        lengthKeysEmpty = 10 - textAnswer.length() -1;
    } else if ( textAnswer.length() < 15 ){
        lengthKeysEmpty = 15 - textAnswer.length() -1;
    }

    for(int j = 1; j <= lengthKeysEmpty ; j++){
        mStringArrayList.add("");
    }
}
```

```

}

private void obterMinhasLetras(String textAnswer){
    for(int i = 0; i < textAnswer.length(); i++){
        mStringArrayList.add(String.valueOf(textAnswer.charAt(i)));
    }
    adicionarLetraExtra();
    adicionarEspacosVazios(textAnswer);
}

```

```

private String[] embaralharListaLetras(String[] param){
    int posicao = 0;
    String s = "";
    for(int i = param.length-1 ; i > 0; i--) { //TORRE DE HANOI
        posicao = rnd.nextInt( i + 1);
        s = param[posicao];
        param[posicao] = param[i];
        param[i] = s;
        if(s==""){
            lengthKeysEmpty++;
        }
    }

    return param;
}

```

```

private void fazerValidacaoInput(){
    if( mEditTextInput.getText().toString().equals(textAnswer) ){
        Intent intent = new Intent(this, ShowCorrectActivity.class);
        startActivity(intent);
    }
}

```

```

        mEditTextInput.setText("");
    } else {
        Toast toastMessage = Toast.makeText(getApplicationContext(), "Errada", Toast.LENGTH_SHORT);
        View toastView = toastMessage.getView();
        toastView.setBackgroundResource(R.drawable.toast_background_image);
        toastMessage.show();
        pressCounter=0;
        mEditTextInput.setText("");
        keys = embaralharListaLetras(keys);
        mLinearLayoutLine1.removeAllViews();
        mLinearLayoutLine2.removeAllViews();
        mLinearLayoutLine3.removeAllViews();
        criarVisualizacaoLetras(keys);
    }
}

private void criarVisualizacaoLetras(String[] keys){

    int qteLetras = 0;
    for( String key : keys ){
        if( qteLetras < 5 ){
            adicionarView( (LinearLayout)findViewById(R.id.layoutParentLine1) , key,
            (EditText)findViewById(R.id.editTextInput) ); //line 1
        } else if ( qteLetras < 10) {
            adicionarView( (LinearLayout)findViewById(R.id.layoutParentLine2) , key,
            (EditText)findViewById(R.id.editTextInput) ); //line 2
        } else if( qteLetras < 15) {
            adicionarView( (LinearLayout)findViewById(R.id.layoutParentLine3) , key,
            (EditText)findViewById(R.id.editTextInput) ); //line 3
        } else{
            Toast.makeText(this, "Palavra muito grande", Toast.LENGTH_SHORT).show();

```

```

    }

    qteLetras++;
}

}

private void adicionarView(LinearLayout viewParent, final String text, final EditText editText){

    LinearLayout.LayoutParams linearLayoutParams = new
    LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1.0f);

    linearLayoutParams.rightMargin=20;

    final TextView textView = new TextView(this);

    textView.setLayoutParams(linearLayoutParams);

    textView.setBackground(this.getResources().getDrawable(R.drawable.back));

    textView.setTextColor(this.getResources().getColor(R.color.colorPurple));

    textView.setGravity(Gravity.CENTER);

    textView.setText(text);

    textView.setClickable(true);

    textView.setFocusable(true);

    textView.setTextSize(32);

    textView.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            if(pressCounter < maxPresCounter){

                if(pressCounter == 0){

                    editText.setText("");

                }

                editText.setText(editText.getText().toString() + text);

                textView.animate().alpha(0).setDuration(300);

```



```

        if(text != ""){
            pressCounter++;
        }
        if(pressCounter == maxPresCounter){
            fazerValidacaoInput();
        }
        textView.setOnClickListener(null); //corrige bug pressionamento no vácuo
    }
}
});

viewParent.addView(textView);

}

```

@Override

```

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

    mEditTextInput = findViewById(R.id.editTextInput);

    mLinearLayoutLine1 = findViewById(R.id.layoutParentLine1);
    mLinearLayoutLine2 = findViewById(R.id.layoutParentLine2);
    mLinearLayoutLine3 = findViewById(R.id.layoutParentLine3);

    mTextViewQuestion = findViewById(R.id.textViewQuestion);
    mTextViewScreen = findViewById(R.id.textViewScreen);
    mTextViewTitle = findViewById(R.id.textViewTitle);

```

```
maxPresCounter = textAnswer.length();

obterMinhasLetras(textAnswer);

keys = mStringArrayList.toArray(new String[mStringArrayList.size()]);
keys = embaralharListaLetras(keys);

criarVisualizacaoLetras(keys);

}
}
```

**Código arquivo ShowCorrectActivty.java**

```
package com.example.jogodapalavra;

import android.os.Bundle;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

public class ShowCorrectActivity extends AppCompatActivity {

    private ImageView mBigImage;
    private Animation mSmallToBig;

    @Override
```

```

protected void onCreate(Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_show_correct);

    mSmallToBig = AnimationUtils.loadAnimation(this, R.anim.smalltobig);
    mBigImage = findViewById(R.id.bigboss);
    mBigImage.startAnimation(mSmallToBig);

}

}

```

### Código arquivo AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.jogodapalavra">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

```
</intent-filter>
```

```
</activity>
```

```
<activity android:name=".ShowCorrectActivity"
```

```
    android:parentActivityName=".MainActivity"></activity>
```

```
</application>
```

```
</manifest>
```