## Хранение данных в общих настройках

### Цель работы

Научиться сохранять данные и состояние приложения в файле общих настроек, освоить операции чтения и записи в них, восстанавливать состояние из сохраненного файла

### Задания для выполнения

Создайте приложение, единственная активность которого состоит из формы, в которую пользователь вводит следующую информацию о студенте: имя, группа (выбор одного элемента из нескольких заранее определенных), возраст (в виде ползунка), дата рождения (в виде поля ввода даты).

Решение этого задания объединено с решением первых дополнительных заданий и содержится ниже.

### Контрольные вопросы

1. Что такое общие настройки и зачем они нужны?

Настройки представляют собой хранилище с группой пар ключ-значение. Они чаще всего используются для обеспечения постоянства данных между пользовательскими сессиями и доступа к ним компонентов приложения. Общие настройки нужны для того, чтобы сохранить данные для дальнейшего использования, например данные о пользователе, настройки конфигурации и т. д.

1. Чем отличаются приватные и публичные общие настройки?

Приватные общие настройки отличаются от публичных общих настроек тем, что каждая активность может использовать приватные, и доступ из других активностей к ним будет невозможен.

1. Какие данные не целесообразно хранить в файле общих настроек?

В файлах не целесообразно хранить большие или сложные данные, например, большие текстовые заметки. В файлах настроек однозначно не следует хранить чувствительные данные: пароли, данные банковских карт и т. д., так как настройки хранятся в незашифрованном виде.

### Дополнительные задания

1. Модифицируйте программу таким образом, чтобы все необходимые данные автоматически сохранялись при остановке приложения и восстанавливались при возобновлении работы.
2. (\*) Добавьте возможность в форме делать снимок в камеры телефона. Загруженный снимок должен сохраняться в отдельный файл приложения и так же восстанавливаться при следующем запуске.

**AndroidManifest.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.datastorage">  
  
 <uses-feature android:name="android.hardware.camera"  
 android:required="true" />  
 <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"  
 android:maxSdkVersion="18" />  
  
 <application  
 android:icon="@mipmap/ic\_launcher"  
 android:allowBackup="true"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.DataStorage">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 <provider  
 android:name="androidx.core.content.FileProvider"  
 android:authorities="com.example.datastorage.fileprovider"  
 android:exported="false"  
 android:grantUriPermissions="true">  
 <meta-data  
 android:name="android.support.FILE\_PROVIDER\_PATHS"  
 android:resource="@xml/file\_paths" />  
 </provider>  
  
 </application>  
  
</manifest>

**MainActivity.java**

package com.example.datastorage;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.content.FileProvider;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Environment;  
import android.provider.MediaStore;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.SeekBar;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import java.io.File;  
import java.util.Arrays;  
  
public class MainActivity extends AppCompatActivity {  
  
 private final int AGE\_OFFSET = 17;  
  
 EditText etName, editTextDate;  
 TextView ageView;  
 SeekBar seekBar;  
 Spinner group;  
 ImageView photo;  
  
 SharedPreferences pref;  
 SharedPreferences.Editor ed;  
  
 File file = null;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 etName = findViewById(R.id.etName);  
 ageView = findViewById(R.id.ageView);  
 editTextDate = findViewById(R.id.editTextDate);  
 group = findViewById(R.id.group);  
 seekBar = findViewById(R.id.seekBar);  
 photo = findViewById(R.id.photo);  
  
 pref = getPreferences(MODE\_PRIVATE);  
 ed = pref.edit();  
  
 seekBar.setOnSeekBarChangeListener(  
 new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean b) {  
 ageView.setText(String.valueOf(progress + AGE\_OFFSET));  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
 }  
 );  
  
 load();  
 }  
  
 @Override  
 protected void onStop() {  
 save();  
 super.onStop();  
 }  
  
 public void onClick(View view) {  
 Intent takePictureIntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
 File photoFile = createImageFile();  
 Uri photoURI = FileProvider.*getUriForFile*(this,  
 "com.example.datastorage.fileprovider",  
 photoFile);  
 takePictureIntent.putExtra(MediaStore.*EXTRA\_OUTPUT*, photoURI);  
 startActivityForResult(takePictureIntent, 0);  
 }  
  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode == *RESULT\_OK*) {  
 loadPhoto();  
 }  
 }  
  
 private void save() {  
 saveName();  
 saveAge();  
 saveGroup();  
 saveDate();  
 savePhotoPath();  
 ed.commit();  
 }  
  
 private void saveName() {  
 String name = etName.getText().toString();  
 ed.putString("name", name);  
 }  
  
 private void saveAge() {  
 int age = seekBar.getProgress() + AGE\_OFFSET;  
 ed.putInt("age", age);  
 }  
  
 private void saveGroup() {  
 String group = this.group.getSelectedItem().toString();  
 ed.putString("group", group);  
 }  
  
 private void saveDate() {  
 String date = editTextDate.getText().toString();  
 ed.putString("date", date);  
 }  
  
 private void savePhotoPath() {  
 if (file != null) {  
 String path = file.getAbsolutePath();  
 ed.putString("path", path);  
 }  
 }  
  
 private void load() {  
 loadName();  
 loadAge();  
 loadGroup();  
 loadDate();  
 loadPhoto();  
 }  
  
 private void loadName() {  
 String name = pref.getString("name", "");  
 etName.setText(name);  
 }  
  
 private void loadAge() {  
 int ageProgress = pref.getInt("age", 18) - AGE\_OFFSET;  
 seekBar.setProgress(ageProgress);  
 }  
  
 private void loadGroup() {  
 String[] groups = getResources().getStringArray(R.array.groups);  
 String defaultGroup = groups[0];  
 String group = pref.getString("group", defaultGroup);  
 int group\_index = Arrays.asList(groups).indexOf(group);  
 this.group.setSelection(group\_index);  
 }  
  
 private void loadDate() {  
 String date = pref.getString("date", "");  
 editTextDate.setText(date);  
 }  
  
 private void loadPhoto() {  
 String path = pref.getString("path", "");  
 if (!path.equals("")) {  
 file = new File(path);  
 BitmapFactory.Options bmOptions = new BitmapFactory.Options();  
 Bitmap imageBitmap = BitmapFactory.decodeFile(file.getAbsolutePath(), bmOptions);  
 photo.setImageBitmap(imageBitmap);  
 }  
 }  
  
 private File createImageFile() {  
 File storageDir = getExternalFilesDir(Environment.DIRECTORY\_PICTURES);  
 file = new File(storageDir.getPath() + "/" + "photo\_"  
 + System.currentTimeMillis() + ".jpg");  
 return file;  
 }  
}

**activity\_main.xml**

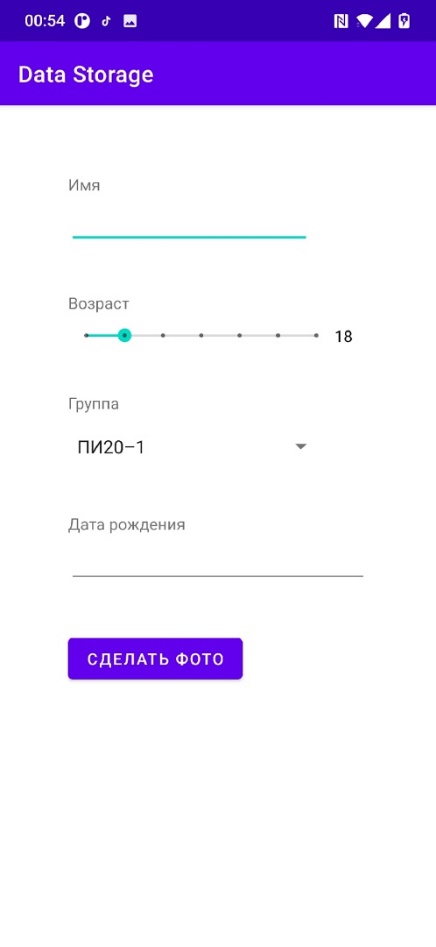
*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:layout\_margin="60dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginBottom="40dp">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Имя" />  
  
 <EditText  
 android:id="@+id/etName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:text="" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginBottom="40dp" >  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:text="Возраст" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <SeekBar  
 android:id="@+id/seekBar"  
 style="@style/Widget.AppCompat.SeekBar.Discrete"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="8"  
 android:max="6"  
 android:progress="1" />  
  
 <TextView  
 android:id="@+id/ageView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="18"  
 android:textColor="#000000" />  
  
 </LinearLayout>  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginBottom="40dp" >  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:text="Группа" />  
  
 <Spinner  
 android:id="@+id/group"  
 android:layout\_width="229dp"  
 android:layout\_height="37dp"  
 android:entries="@array/groups" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginBottom="40dp" >  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Дата рождения" />  
  
 <EditText  
 android:id="@+id/editTextDate"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="date" />  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/takePhotoBtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Сделать фото"  
 android:onClick="onClick" />  
  
 <ImageView  
 android:id="@+id/photo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
</LinearLayout>

**groups.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<resources>  
 <string-array name="groups">  
 <item>ПИ20–1</item>  
 <item>ПИ20–2</item>  
 <item>ПИ20–3</item>  
 <item>ПИ20–4</item>  
 <item>ПИ20–5</item>  
 <item>ПИ20–6</item>  
 </string-array>  
</resources>

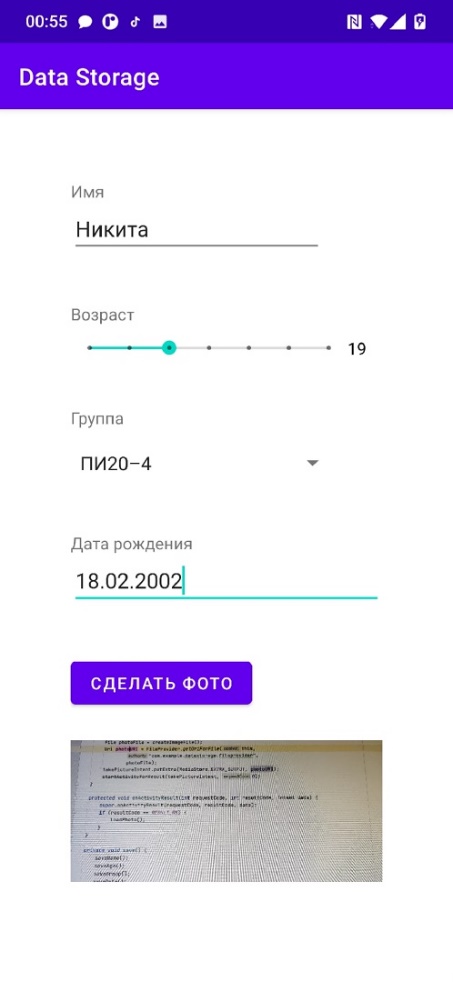
**file\_paths.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<paths>  
 <external-files-path name="my\_images" path="Pictures" />  
</paths>



Изображение выглядит как текст

Автоматически созданное описание



1. (\*) Создайте мобильную игру в крестики-нолики используя специально подготовленные графические элементы. Модифицируйте программу таким образом, чтобы прогресс игрока сохранялся между запусками программы. Придумайте схему хранения состояния игры.

**MainActivity.java**

package com.example.tictactoe;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.os.Debug;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
import java.util.Arrays;  
  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView textView;  
 private int[] buttonsIds;  
  
 private boolean firstPlayerIsActive;  
 private enum FieldState {  
 *EMPTY*,  
 *FIRST\_PLAYER*,  
 *SECOND\_PLAYER* }  
 private FieldState[] gameState;  
  
 private final int[][] WINNING\_POSITIONS = {  
 {0, 1, 2}, {3, 4, 5}, {6, 7, 8},  
 {0, 3, 6}, {1, 4, 7}, {2, 5, 8},  
 {0, 4, 8}, {2, 4, 6}  
 };  
 private final String FIRST\_PLAYER\_SIGN = "X";  
 private final String SECOND\_PLAYER\_SIGN = "O";  
 private final String FIRST\_PLAYER\_NAME = "Игрок 1";  
 private final String SECOND\_PLAYER\_NAME = "Игрок 2";  
  
 SharedPreferences pref;  
 SharedPreferences.Editor editor;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 textView = findViewById(R.id.*textView*);  
 buttonsIds = new int[]{  
 R.id.*field0*,  
 R.id.*field1*,  
 R.id.*field2*,  
 R.id.*field3*,  
 R.id.*field4*,  
 R.id.*field5*,  
 R.id.*field6*,  
 R.id.*field7*,  
 R.id.*field8*,  
 };  
  
 pref = getPreferences(*MODE\_PRIVATE*);  
 editor = pref.edit();  
  
 loadGameState();  
 }  
  
 @Override  
 protected void onStop() {  
 saveGameState();  
 super.onStop();  
 }  
  
 public void onFieldButtonClick(View view) {  
 Button button = (Button) view;  
  
 if (button.getText().toString().equals("")) {  
 int gameStatePointer = getGameStatePointer(view);  
 if (firstPlayerIsActive) {  
 gameState[gameStatePointer] = FieldState.*FIRST\_PLAYER*;  
 changeActivePlayer(SECOND\_PLAYER\_NAME);  
 } else {  
 gameState[gameStatePointer] = FieldState.*SECOND\_PLAYER*;  
 changeActivePlayer(FIRST\_PLAYER\_NAME);  
 }  
 updateFields();  
 if (isWin()) {  
 onWin();  
 } else if (isDraw()) {  
 onDraw();  
 }  
 }  
  
 saveGameState();  
 }  
  
 public void onResetButtonClick(View view) {  
 startNewGame();  
 }  
  
 private int getGameStatePointer(View view) {  
 String buttonIdName = view.getResources().getResourceEntryName(view.getId());  
 int buttonIdNameLength = buttonIdName.length();  
 return Integer.*parseInt*(  
 buttonIdName.substring(buttonIdNameLength - 1, buttonIdNameLength)  
 );  
 }  
  
 private void changeActivePlayer(String playerName) {  
 textView.setText("Ходит " + playerName);  
 firstPlayerIsActive = !firstPlayerIsActive;  
 }  
  
 private void updateFields() {  
 for (int i = 0; i < buttonsIds.length; i++) {  
 Button button = findViewById(buttonsIds[i]);  
 switch (gameState[i]) {  
 case *EMPTY*:  
 button.setText("");  
 break;  
 case *FIRST\_PLAYER*:  
 button.setText(FIRST\_PLAYER\_SIGN);  
 break;  
 case *SECOND\_PLAYER*:  
 button.setText(SECOND\_PLAYER\_SIGN);  
 break;  
 }  
 }  
 }  
  
 private boolean isWin() {  
 for (int[] winningPosition : WINNING\_POSITIONS) {  
 if (gameState[winningPosition[0]] == gameState[winningPosition[1]] &&  
 gameState[winningPosition[1]] == gameState[winningPosition[2]] &&  
 gameState[winningPosition[0]] != FieldState.*EMPTY*) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
 private void onWin() {  
 textView.setText("Победа");  
 setEnabledToFields(false);  
 }  
  
 private void setEnabledToFields(boolean value) {  
 for (int buttonId : buttonsIds) {  
 Button button = findViewById(buttonId);  
 button.setEnabled(value);  
 }  
 }  
  
  
 private boolean isDraw() {  
 for (int buttonId : buttonsIds) {  
 Button button = findViewById(buttonId);  
 if (button.getText().toString().equals("")) {  
 return false;  
 }  
 }  
 return true;  
 }  
  
 private void onDraw() {  
 textView.setText("Ничья");  
 setEnabledToFields(false);  
 }  
  
 private void startNewGame() {  
 setInitState();  
 textView.setText("Ходит " + FIRST\_PLAYER\_NAME);  
 firstPlayerIsActive = true;  
 setEnabledToFields(true);  
 }  
  
 private void setInitState() {  
 gameState = new FieldState[]{  
 FieldState.*EMPTY*,  
 FieldState.*EMPTY*,  
 FieldState.*EMPTY*,  
 FieldState.*EMPTY*,  
 FieldState.*EMPTY*,  
 FieldState.*EMPTY*,  
 FieldState.*EMPTY*,  
 FieldState.*EMPTY*,  
 FieldState.*EMPTY* };  
 updateFields();  
 }  
  
 private void saveGameState() {  
 editor.putString("gameState", Arrays.*toString*(gameState));  
 editor.putBoolean("firstPlayerIsActive", firstPlayerIsActive);  
 editor.commit();  
 }  
  
 private void loadGameState() {  
 String gameStateString = pref.getString("gameState", "");  
 if (gameStateString.equals("")) {  
 startNewGame();  
 } else {  
 firstPlayerIsActive = pref.getBoolean("firstPlayerIsActive", true);  
 String[] gameSatesStrings = gameStateString.substring(  
 1, gameStateString.length() - 1  
 ).split(", ");  
 gameState = new FieldState[9];  
 for (int i = 0; i < gameState.length; i++) {  
 switch (gameSatesStrings[i]) {  
 case "EMPTY":  
 gameState[i] = FieldState.*EMPTY*;  
 break;  
 case "FIRST\_PLAYER":  
 gameState[i] = FieldState.*FIRST\_PLAYER*;  
 break;  
 case "SECOND\_PLAYER":  
 gameState[i] = FieldState.*SECOND\_PLAYER*;  
 break;  
 }  
 }  
 updateFields();  
 if (isWin()) {  
 onWin();  
 } else if (isDraw()) {  
 onDraw();  
 }  
 }  
 }  
}

**activity\_main.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout  
 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="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="30dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginEnd="30dp"  
 android:gravity="center"  
 android:lines="2"  
 android:text="Ходит Игрок 1"  
 android:textColor="@color/black"  
 android:textSize="40dp" />  
  
  
 <Button  
 android:id="@+id/newGameButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/textView"  
 android:layout\_margin="30dp"  
 android:onClick="onResetButtonClick"  
 android:text="Новая игра" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@id/newGameButton"  
 android:layout\_marginHorizontal="30dp"  
 android:layout\_marginTop="130dp"  
 android:layout\_marginBottom="30dp"  
 android:orientation="vertical">  
  
 <LinearLayout style="@style/field\_row">  
  
 <Button  
 android:id="@+id/field0"  
 style="@style/field" />  
  
 <Button  
 android:id="@+id/field1"  
 style="@style/field" />  
  
 <Button  
 android:id="@+id/field2"  
 style="@style/field" />  
  
 </LinearLayout>  
  
 <LinearLayout style="@style/field\_row">  
  
 <Button  
 android:id="@+id/field3"  
 style="@style/field" />  
  
 <Button  
 android:id="@+id/field4"  
 style="@style/field" />  
  
 <Button  
 android:id="@+id/field5"  
 style="@style/field" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/last\_linear"  
 style="@style/field\_row">  
  
 <Button  
 android:id="@+id/field6"  
 style="@style/field" />  
  
 <Button  
 android:id="@+id/field7"  
 style="@style/field" />  
  
 <Button  
 android:id="@+id/field8"  
 style="@style/field" />  
  
 </LinearLayout>  
  
  
 </LinearLayout>  
  
</RelativeLayout>

**styles.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<resources>  
  
 <style name="field">  
 <item name="android:layout\_width">0dp</item>  
 <item name="android:layout\_height">match\_parent</item>  
 <item name="android:layout\_weight">1</item>  
 <item name="android:textSize">40sp</item>  
 <item name="android:layout\_marginLeft">4dp</item>  
 <item name="android:layout\_marginRight">4dp</item>  
 <item name="android:onClick">onFieldButtonClick</item>  
 </style>  
  
 <style name="field\_row">  
 <item name="android:layout\_width">match\_parent</item>  
 <item name="android:layout\_height">0dp</item>  
 <item name="android:layout\_weight">1</item>  
 </style>  
  
</resources>







