

目 录

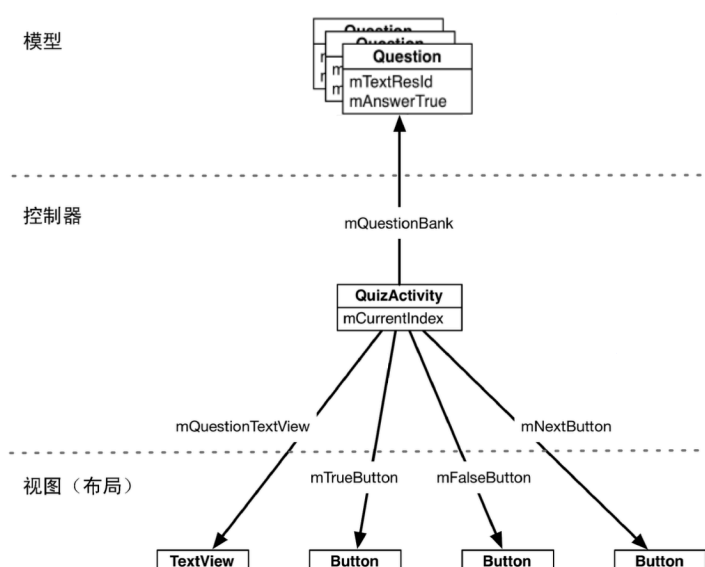
0.1	MVC 设计模式	2
0.1.1	设计模型层	2
0.1.2	设计视图层	3
0.1.3	设计控制层	5

0.1 MVC 设计模式

Android 应用可以基于模型-控制器-视图的架构模式来进行设计，应用的任何对象，归根结底都属于模型对象、视图对象以及控制对象中的一种：

1. 模型对象，用于存储和管理应用数据。
2. 视图对象，用于绘制在屏幕上能看见的对象。
3. 控制对象，是视图对象与模型对象的联系纽带。

接下来，通过一个实例来学习 MVC 设计模式，首先看一下这个应用的 MVC 模型：



0.1.1 设计模型层

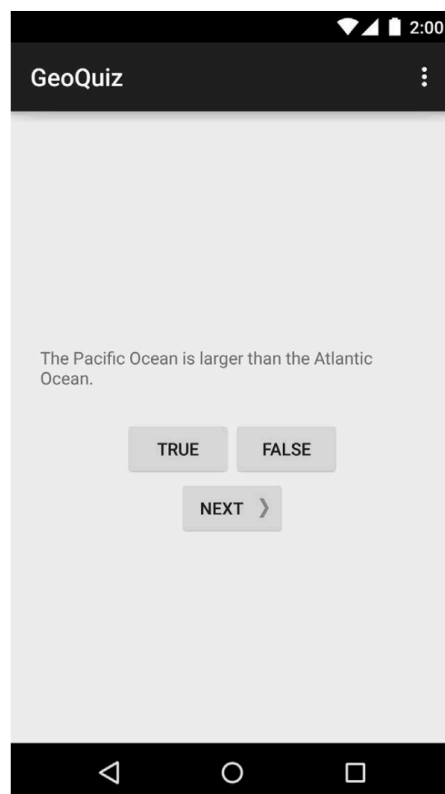
Question 类中有问题文本和问题答案：

```
1 public class Question {
2     private int mTextResId;
3     private boolean mAnswerTrue;
4
5     public Question(int textRestId, boolean answerTrue)
6     {
7         mTextResId = textRestId;
8         mAnswerTrue = answerTrue;
9     }
10
11     public int getTextResId()
```

```
12     {  
13         return mTextResId;  
14     }  
15  
16     public void setTextResId(int textResId)  
17     {  
18         mTextResId = textResId;  
19     }  
20  
21     public boolean isAnswerTrue()  
22     {  
23         return mAnswerTrue;  
24     }  
25  
26     public void setAnswerTrue(boolean answerTrue)  
27     {  
28         mAnswerTrue = answerTrue;  
29     }  
30 }
```

0.1.2 设计视图层

我们想要的图形界面如下所示：



设计视图层，也就是写 XML 文件和 string 资源文件，以下是视图界面文件：

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:gravity="center"
6      android:orientation="vertical">
7
8      <TextView
9          android:id="@+id/question_text_view"
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:padding="24dp"
13     />
14
15     <LinearLayout
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:orientation="horizontal"
19     >
20
21         <Button
22             android:id="@+id/true_button"
23             android:layout_width="wrap_content"
24             android:layout_height="wrap_content"
25             android:text="@string/true_button"
26         />
27
28         <Button
29             android:id="@+id/false_button"
30             android:layout_width="wrap_content"
31             android:layout_height="wrap_content"
32             android:text="@string/false_button"
33         />
34     </LinearLayout>
35
36     <Button
37         android:id="@+id/next_button"
38         android:layout_width="wrap_content"
39         android:layout_height="wrap_content"
40         android:text="@string/next_button"
41     />
42
43 </LinearLayout>
```

还有相应的 string 资源文件：

```
1  <resources>
2      <string name="app_name">GeoQuiz</string>
3      <string name="question_text">
4          Constantionople is the largest city in Turkey.
5      </string>
6      <string name="true_button">TRUE</string>
7      <string name="false_button">FALSE</string>
```

```

8      <string name="next_button">NEXT</string>
9      <string name="correct_button">Correct!</string>
10     <string name="incorrect_button">Incorrect!</string>
11
12     <string name="question_oceans">The Pacific Ocean is larger than the
13         Atlantic Ocean.</string>
14     <string name="question_mideast">The Suez Canal connects the Red Sea and the
15         Indian Ocean.</string>
16     <string name="question_africa">The source of the Nile River is in Egypt.</
17         string>
18     <string name="question_americas">The Amazon River is the longest river in
19         the Americas.</string>
20     <string name="question_asia">Lake Baikal is the world\'s oldest and deepest
21         freshwater lake.</string>
22 </resources>

```

0.1.3 设计控制层

在控制层部分，我们根据资源 ID 获得组件对象，然后就可以对这个组件对象自由发挥。在控制层，也会创建模型层定义的对象，与视图层的组件对象进行交互：

```

1      package geoquiz.android.bignerdranch.com.geoquiz;
2
3      import android.support.v7.app.AppCompatActivity;
4      import android.os.Bundle;
5      import android.view.View;
6      import android.widget.Button;
7      import android.widget.TextView;
8      import android.widget.Toast;
9
10     public class QuizActivity extends AppCompatActivity {
11         private Button mTrueButton;
12         private Button mFalseButton;
13         private Button mNextButton;
14         private TextView mQuestionTextView;
15
16         private Question[] mQuestionBank = new Question[]{
17             new Question(R.string.question_oceans, true),
18             new Question(R.string.question_mideast, false),
19             new Question(R.string.question_africa, true),
20             new Question(R.string.question_americas, true),
21             new Question(R.string.question_asia, true)
22         };
23
24         private int mCurrentIndex = 0;
25
26         private void updateQuestion()
27         {
28             int question_text = mQuestionBank[mCurrentIndex].getTextResId();
29             mQuestionTextView.setText(question_text);
30         }
31
32         private void checkAnswer(boolean answer)

```

```
33     {
34         boolean rightAnswer = mQuestionBank[mCurrentIndex].isAnswerTrue();
35
36         if(rightAnswer == answer)
37             Toast.makeText(QuizActivity.this, R.string.correct_button, Toast.
38                 LENGTH_SHORT).show();
39         else
40             Toast.makeText(QuizActivity.this, R.string.incorrect_button, Toast.
41                 LENGTH_SHORT).show();
42     }
43
44     @Override
45     protected void onCreate(Bundle savedInstanceState) {
46         super.onCreate(savedInstanceState);
47         setContentView(R.layout.activity_quiz);
48
49         mTrueButton = (Button)findViewById(R.id.true_button);
50         mFalseButton = (Button)findViewById(R.id.false_button);
51         mNextButton = (Button)findViewById(R.id.next_button);
52
53         mQuestionTextView = (TextView)findViewById(R.id.question_text_view);
54
55         updateQuestion();
56
57         mTrueButton.setOnClickListener(new View.OnClickListener() {
58             @Override
59             public void onClick(View v)
60             {
61                 checkAnswer(true);
62             }
63         });
64
65         mFalseButton.setOnClickListener(new View.OnClickListener() {
66             @Override
67             public void onClick(View v)
68             {
69                 checkAnswer(false);
70             }
71         });
72
73         mNextButton.setOnClickListener(new View.OnClickListener() {
74             @Override
75             public void onClick(View v)
76             {
77                 mCurrentIndex = (mCurrentIndex + 1) % mQuestionBank.length;
78                 updateQuestion();
79             }
80         });
81     }
82 }
```