第 1 单元
UI 设计案例

1.1 ProgressBar（进度条）

本节引言

本节给用户介绍的是 Android 基本 UI 控件中的 ProgressBar（进度条），ProgressBar 的应用场景很多，比如用户登录时，向后台发送请求，以及等待服务器返回信息，这个时候会用到进度条；或进行一些比较耗时的操作需要等待一段较长的时间，这个时候如果没有提示，用户可能会以为程序 Carsh 或者手机死机了，这样会大大降低用户体验的舒适感。因此在需进行耗时操作的地方，添加上进度条，让用户知道当前程序正在执行中，也可以直观地告诉用户当前任务的执行进度等。

1. 常用属性讲解

（1）类关系图。

从官方文档，可以看到如下的类关系图：

ProgressBar 继承自 View 类，直接子类有 AbsSeekBar 和 ContentLoadingProgressBar，其中 AbsSeekBar 的子类有 SeekBar 和 RatingBar，可见这两者也是基于 ProgressBar 实现的。

（2）常用属性详解。

android:max：进度条的最大值；

android:progress：进度条已完成进度值；

android:progressDrawable：设置轨道对应的 Drawable 对象；

android:indeterminate：如果设置成 true，则进度条不精确显示进度；

android:indeterminateDrawable：设置不显示进度的进度条的 Drawable 对象；

android:indeterminateDuration：设置不精确显示进度的持续时间；

android:secondaryProgress：二级进度条，类似于视频播放的一条是当前播放进度，一条是缓冲进度，前者通过 progress 属性进行设置。

（3）对应在 Java 中用户可调用方法。

getMax()：返回这个进度条的范围的上限；

getProgress()：返回进度；

getSecondaryProgress()：返回次要进度；

incrementProgressBy（int diff）：指定增加的进度；

isIndeterminate()：指示进度条是否在不确定模式下；

setIndeterminate（boolean indeterminate）：设置不确定模式下。

2. 系统默认进度条使用实例

（1）新建一个 module，命名为 ProgressBarDemo。

（2）布局文件 activity_main.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"
    tools:context=".MainActivity">

<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:gravity="center_horizontal"
        android:text="系统默认进度条"
        android:textSize="20sp" />
<ProgressBar
        android:layout_marginTop="30dp"
        android:id="@+id/progressBar1"
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

<ProgressBar
        android:layout_marginTop="30dp"
        android:id="@+id/progressBar2"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:max="1000"
        android:progress="120"/>
<Button
        android:id="@+id/bt1"
        android:layout_marginTop="30dp"
        android:layout_width="match_parent"
```

```
            android:layout_height="wrap_content"
            android:text="新增"
            android:textSize="20sp"
            android:onClick="addValue" />
    <ProgressBar
            android:layout_marginTop="30dp"
            android:id="@+id/progressBar3"
            style="?android:attr/progressBarStyleSmall"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>
```

（3）MainActivity.java。

```
public class MainActivity extends AppCompatActivity {
    ProgressBar progressBar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        progressBar = findViewById(R.id.progressBar2);
    }
    public void addValue(View view) {
        int progress = progressBar.getProgress(); //获得当前进度值
        progress += 5; //每点击一次按钮，进度值加 5
        if(progress>1000){
            progress=1000;
        }
        progressBar.setProgress(progress);
    }
}
```

（4）运行效果，见图 1.1。

图 1.1

1.2 SeekBar（拖动条）

本节引言

本节学习 Android 基本 UI 控件中的拖动条——SeekBar，相信大多数用户对它并不陌生，最常见的地方就是音乐播放器或视频播放器，音量控制或者播放进度控制，都用到了这个 SeekBar。SeekBar 的类结构如下：

```
java.lang.Object
  ↳ android.view.View
      ↳ android.widget.ProgressBar
          ↳ android.widget.AbsSeekBar
              ↳ android.widget.SeekBar
```

SeekBar 是 ProgressBar 的子类，也就是 ProgressBar 的属性都可以用。它还有一个自己的属性：android:thumb，允许用户自定义滑块。

1. SeekBar 的基本用法

SeekBar 的基本用法很简单，常用的属性有如下几个，Java 代码里只要 set × × × 即可。

android:max="100" //滑动条的最大值

android:progress="60" //滑动条的当前值

android:secondaryProgress="70" //二级滑动条的进度

android:thumb = "@mipmap/sb_icon" //滑块的 drawable

SeekBar 的事件监听器是 SeekBar.OnSeekBarChangeListener，用户只需重写三个对应的即可，方法为：

onProgressChanged：进度发生改变时会触发

onStartTrackingTouch：按住 SeekBar 时会触发

onStopTrackingTouch：放开 SeekBar 时触发

2. SeekBar 案例实现

（1）新建一个 module，命名为 SeekBarDemo。

（2）布局文件 activity_main.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"
    tools:context=".MainActivity">

<SeekBar
    android: 用_marginTop="20dp"
    android:id="@+id/seekBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="100"
    />
<ProgressBar
        android:layout_marginTop="20dp"
        android:id="@+id/progressBar"
        style="@android:style/Widget.ProgressBar.Horizontal"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:max="100"
        />
</LinearLayout>
```

（3）MainActivity.java。

```
public class MainActivity extends AppCompatActivity {
    SeekBar seekBar;
    ProgressBar progressBar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        seekBar = findViewById(R.id.seekBar);
        progressBar = findViewById(R.id.progressBar);
        seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                //fromUser--是否是用户主动拖拽进度条发生的改变
                if(fromUser){
                    progressBar.setProgress(progress);
                }
            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
                Toast.makeText(MainActivity.this,"开始拖拽",
                Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
                Toast.makeText(MainActivity.this,"放开拖拽",
                Toast.LENGTH_SHORT).show();
            }
        });
```

```
        }
    }
```

（4）运行效果，见图 1.2。

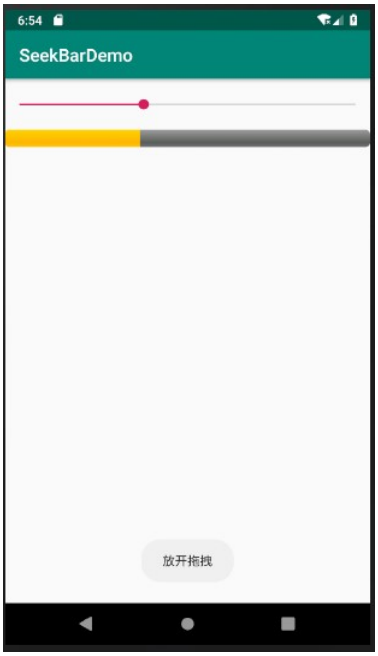

图 1.2

1.3 RatingBar（星级评分条）

本节引言

本节学的 RatingBar（星级评分条）比较简单，在网上购过物的用户对这个应该不会感到陌生。买家对卖家评分的时候就会用到这个星级评分条。从官方文档可以看出，RatingBar 和 SeekBar 的类结构是一样的，也是 ProgressBar 的子类。

也就是说 RatingBar 同样拥有 ProgressBar 的相关属性，接下来我们来探究 RatingBar 特有的属性。

1. RatingBar 基本使用

5.0 的 RatingBar 原生模样，见图 1.3。

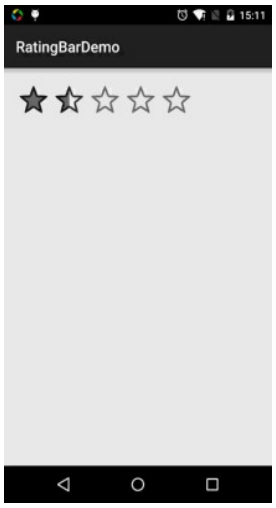

图 1.3

——相关属性：

android:isIndicator：是否用作指示，用户无法更改，默认 false；

android:numStars：显示多少个星星，必须为整数；

android:rating：默认评分值，必须为浮点数；

android:stepSize：评分每次增加的值，必须为浮点数。

除了上面这些，还有两种样式供用户选择，分别是：

style="?android:attr/ratingBarStyleSmall"

style="?android:attr/ratingBarStyleIndicator"

——事件处理：只需为 RatingBar 设置 OnRatingBarChangeListener 事件，然后重写下 onRatingChanged()方法即可。

2. RatingBar 案例实现

（1）新建一个 module，命名为 RatingBarDemo。

（2）布局文件 activity_main.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"
    tools:context=".MainActivity">

<RatingBar
        android:layout_marginTop="20dp"
        android:id="@+id/ratingBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:numStars="5"
        android:stepSize="0.5"
        />
<TextView
        android:layout_marginTop="20dp"
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="用户评分：0 颗星"
        android:textSize="30sp"
        />
</LinearLayout>
```

（3）MainActivity.java。

```
public class MainActivity extends AppCompatActivity {
    RatingBar ratingBar;
    TextView textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ratingBar = findViewById(R.id.ratingBar);
        textView = findViewById(R.id.textView);
        ratingBar.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {
            @Override
            public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
                if(fromUser){
                    textView.setText("用户评分：+rating+"颗星");
                }
            }
        });
    }
}
```

（4）运行效果，见图 1.4。

图 1.4

1.4 ScrollView（滚动条）

本节引言

本节讲解 Android 基本 UI 控件中的 ScrollView（滚动条），也叫竖直滚动条，对应于另外一个水平方向上的滚动条：HorizontalScrollView。从官方文档可以看到类的结构如下：

原来它是一个 FrameLayout 的容器，只是在它的基础上添加了滚动，允许可显示比实际多的内容。

另外，可在里面放置一个子元素，可以是单一的组件，又或者一个布局包裹着的复杂的层次结构。

如遇到可能显示不完的情况，用户可以直接在布局的外层套上一个 ScrollView 或者 HorizontalScrollView。

1. 实际开发中的常见需求

实际开发中常见的一些需求如下：

（1）滚动到底部/顶部

用户可以直接利用 ScrollView 所提供的 fullScroll()方法：

scrollView.fullScroll（ScrollView.FOCUS_DOWN）；滚动到底部

scrollView.fullScroll（ScrollView.FOCUS_UP）；滚动到顶部

用户在使用时要注意异步的问题，即在 addView 后，有可能还没有显示完，如果此时直接调用，可能会无效，则需要用户写 handler 予以更新。

（2）滚动显示 View 的全部内容

用户可以用 ScrollView 包含一个布局，同时在布局中可以有多个组件。

2. ScrollView 案例实现

（1）新建一个 module，命名为 ScrollViewDemo。

（2）布局文件 activity_main.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"
    tools:context=".MainActivity">
    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <SeekBar
                android:layout_marginTop="20dp"
                android:id="@+id/seekBar"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:max="100"/>
            <ProgressBar
                android:layout_marginTop="20dp"
                android:id="@+id/progressBar"
                style="@android:style/Widget.ProgressBar.Horizontal"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
```

```
            android:max="100"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="2003 年 10 月，Andy Rubin 等人创建 Android 公司，并组建
Android 团队。
```

2005 年 8 月 17 日，Google 低调收购了成立仅 22 个月的高科技企业 Android 及其团队。Andy Rubin 成为 Google 公司工程部副总，继续负责 Android 项目。

2007 年 11 月 5 日，谷歌公司正式向外界展示了这款名为 Android 的操作系统，并且在这天谷歌宣布建立一个全球性的联盟组织，该组织由 34 家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成，并与 84 家硬件制造商、软件开发商及电信营运商组成开放手持设备联盟（Open Handset Alliance）来共同研发改良 Android 系统，这一联盟将支持谷歌发布的手机操作系统以及应用软件，Google 以 Apache 免费开源许可证的授权方式，发布了 Android 的源代码。

2008 年，在 GoogleI/O 大会上，谷歌展示了 AndroidHAL 架构图，在同年 8 月 18 号，Android 获得了美国联邦通信委员会（FCC）的批准，在 2008 年 9 月，谷歌正式发布了 Android 1.0 系统，这也是 Android 系统最早的版本。

2009 年 4 月，谷歌正式推出了 Android 1.5 款手机，从 Android 1.5 版本开始，谷歌开始将 Android 的版本以甜品的名字命名，Android 1.5 命名为 Cupcake（纸杯蛋糕）。该系统与 Android 1.0 相比有了很大的改进。

2009 年 9 月，谷歌发布了 Android 1.6 的正式版，并且推出了搭载 Android 1.6 正式版的手机 HTC Hero（G3），凭借着出色的外观设计以及全新的 Android 1.6 操作系统，HTC Hero（G3）成为当时全球最受欢迎的手机。Android 1.6 也有一个有趣的甜品名称，它被称为 Donut（甜甜圈）。

2010 年 2 月，Linux 内核开发者 Greg Kroah-Hartman 将 Android 的驱动程序从 Linux 内核“状态树”（“staging tree”）上除去，从此，Android 与 Linux 开发主流将分道扬镳。在同年 5 月份，谷歌正式发布了 Android 2.2 操作系统。谷歌将 Android 2.2 操作系统命名为 Froyo，翻译完名为冻酸奶。

2010 年 10 月，谷歌宣布 Android 系统达到了第一个里程碑，即电子市场上获得官方数字认证的 Android 应用数量已经达到了 10 万个，Android 系统的应用增长非常迅速。在 2010 年 12 月，谷歌正式发布了 Android 2.3 操作系统 Gingerbread（姜饼）。

2011 年 1 月，谷歌称每日的 Android 设备新用户数量达到了 30 万部，到 2011 年 7 月，这个数字增长到 55 万部，而 Android 系统设备的用户总数达到了 1.35 亿，Android 系统已经成为智能手机领域占有量最高的系统。

2011 年 8 月 2 日，Android 手机已占据全球智能机市场 48%的份额，并在亚太地区市场占据统治地位，终结了 Symbian（塞班系统）的霸主地位，跃居全球第一。

2011 年 9 月，Android 系统的用户已经达到了 48 万，而在智能手机市场，Android 系统的占有率也达到了 43%。继续在排在移动操作系统首位。谷歌将会发布全新的 Android 4.0 操作系统，这款系统被谷歌命名为 Ice Cream Sandwich（冰激凌三明治）。

2012 年 1 月 6 日，谷歌 Android Market 已有 10 万开发者推出超过 40 万活跃的应用，大多数的应用程序免费。Android Market 应用程序商店目录在新年首周周末突破 40 万，距离突破 30 万应用仅

4 个月。在 2011 年早些时候，Android Market 从 20 万增加到 30 万应用也仅用了四个月。

2013 年 11 月 1 日，Android4.4 正式发布，从具体功能上讲，Android4.4 提供了各种实用小功能，新的 Android 系统更智能，添加更多的 Emoji 表情图案，UI 的改进也更为现代化，如全新的 HelloiOS7 半透明效果。

2015 年，网络安全公司 Zimperium 研究人员警告，安卓(Android)存在“致命”安全漏洞,黑客发送一封彩信便能在用户毫不知情的情况下完全控制手机。

2018 年 10 月，谷歌表示，计划于 2018 年 12 月 6 日停止 Android 系统中的 Nearby Notifications（附近通知）服务，因为 Android 用户收到太多的附近商家推销信息的垃圾邮件。

2020 年 3 月，谷歌的 Android 安全公告中提到，新更新已经提供了 CVE-2020-0069 补丁来解决针对联发科芯片的一个严重安全漏洞。

"/>

```
            </LinearLayout>
        </ScrollView>
    </LinearLayout>
```