**自定义控件**

View是android UI控件的基类

按类型划分，自定义View的实现方式可分为三种：

**自绘控件**

在View上所展现的内容全部都是绘制出来的

绘制的代码是写在onDraw()方法中的

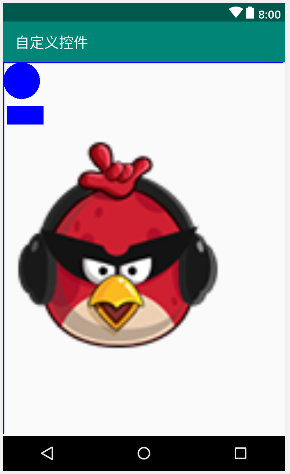
自定义的View在界面上显示，只需要像使用普通的控件一样来使用自定义的View就可以了

activity\_demo.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"**>  
  
 *<!--使用自绘控件在布局中，需要全名（包名.类名）-->*<**com.amaker.hjs.customcontrol.view.MyView  
 android:layout\_width="800dp"  
 android:layout\_height="800dp"** />  
</**LinearLayout**>

MyView.java

**package** com.amaker.hjs.customcontrol.view;  
  
**import** android.content.Context;  
**import** android.graphics.Bitmap;  
**import** android.graphics.BitmapFactory;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**import** com.amaker.hjs.customcontrol.R;  
  
**public class** MyView **extends** View {  
 **private** Paint **mPaint**;*//画笔* **private** Bitmap **mBitmap**;  
  
 */\*\*必须定义  
 \*以代码的方式添加View到容器时会用到这个  
 \** ***@param context*** *\*/* **public** MyView(Context context) {  
 **super**(context);  
 init();  
 }  
  
 */\*\*必须定义  
 \*当以XML布局文件的方式使用时，自动调用  
 \** ***@param context*** *\** ***@param attrs*** *控件的属性  
 \*/* **public** MyView(Context context, AttributeSet attrs) {  
 **super**(context, attrs);  
 init();  
 }  
  
 */\*\*  
 \*  
 \** ***@param context*** *\** ***@param attrs*** *\** ***@param defStyleAttr*** *默认样式  
 \*/* **public** MyView(Context context, AttributeSet attrs, **int** defStyleAttr) {  
 **super**(context, attrs, defStyleAttr);  
 init();  
 }  
  
 */\*\*  
 \*当View需要呈现出来时自动调用  
 \** ***@param canvas*** *画布  
 \*/* @Override  
 **protected void** onDraw(Canvas canvas) {  
 canvas.drawCircle(50,50,50,**mPaint**);  
 **mPaint**.setStyle(Paint.Style.***FILL***);*//实心* canvas.drawRect(10,120,110,170,**mPaint**);  
 canvas.drawBitmap(**mBitmap**,10,200,**null**);  
 **super**.onDraw(canvas);  
 }  
  
 **private void** init(){  
 **mPaint**=**new** Paint();  
 **mPaint**.setColor(Color.***BLUE***);  
 **mPaint**.setStyle(Paint.Style.***STROKE***);*//单线条* **mPaint**.setStrokeWidth(3);*//线条宽度* **mBitmap**=BitmapFactory.*decodeResource*(getResources(), R.drawable.***f***);  
 **mBitmap**=Bitmap.*createScaledBitmap*(**mBitmap**,600,600,**true**);  
  
 }  
  
}



**轮播图的实现：**

MyView2.java

**package** com.amaker.hjs.customcontrol.view;  
  
**import** android.content.Context;  
**import** android.graphics.Bitmap;  
**import** android.graphics.BitmapFactory;  
**import** android.graphics.Canvas;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**import** com.amaker.hjs.customcontrol.R;

//如果创建这个类实例，将会是一个子线程  
**public class** MyView2 **extends** View **implements** Runnable{  
 **private int x**;  
 **private** Bitmap **bmpBackground**;  
 **private boolean flag**;  
 **public** MyView2(Context context) {  
 **super**(context);  
 init();  
 }  
  
 **public** MyView2(Context context, AttributeSet attrs) {  
 **super**(context, attrs);  
 init();  
 }  
  
 **public** MyView2(Context context, AttributeSet attrs, **int** defStyleAttr) {  
 **super**(context, attrs, defStyleAttr);  
 init();  
 }  
  
  
 *//当View呈现出来时，自动调用这个方法* @Override  
 **protected void** onDraw(Canvas canvas) {  
 **super**.onDraw(canvas);  
 moveBackground(canvas);  
 }  
  
 **private void** init(){  
 **bmpBackground** = BitmapFactory.*decodeResource*(getResources(), R.drawable.***nba***);  
  
 }  
  
 **public void** setFlag(**boolean** flag){  
 **this**.**flag**=flag;  
 }  
  
 **private void** moveBackground(Canvas canvas){  
 **x**-=10;  
 **int** x2=**bmpBackground**.getWidth()-(-**x**);  
 **if** (x2<=0){  
 **x**=0;  
 canvas.drawBitmap(**bmpBackground**,**x**,10,**null**);*//重绘画布在初始位置* }**else**{  
 *//画布左移* canvas.drawBitmap(**bmpBackground**,**x**,10,**null**);  
 canvas.drawBitmap(**bmpBackground**,x2,10,**null**);  
 }  
  
 }  
  
 @Override  
 **public void** run() {  
 **while** (**flag**){  
 *//子线程不可以直接调用onDraw方法来进行修改* postInvalidate();*//在工作子线程中对View进行重新呈现  
  
 /\*Android的invalidate与postInvalidate都是用来刷新界面的  
 \* 在UI主线程中更新View，用invalidate()，本质是调用View的onDraw()绘制  
 \* 在工作子线程中更新View，用postInvalidate()  
 \* \*/  
  
 /\*也可以在主线程创建一个Handler，然后在工作线程中用这个handler来传送信息到主线程的队列中，  
 然后在主线程的Handler中重写handleMessage()方法来处理接收的信息，方法内部用invalidate更新  
 界面\*/* }  
 }  
}

MainActivity.java

**package** com.amaker.hjs.customcontrol;  
  
**import** android.os.Trace;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
  
**import** com.amaker.hjs.customcontrol.view.MyView2;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** MyView2 **view**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 **view** = **new** MyView2(**this**);  
 setContentView(**view**);  
  
 *//在主线程内设置标识为真，同意开始工作线程* **view**.setFlag(**true**);  
  
 *//启动工作线程* **new** Thread(**view**).start();  
  
  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 *//当结束Activity之前，在主线程内设置标识为假，同意开始工作线程* **view**.setFlag(**false**);  
 **super**.onDestroy();  
 }  
}

**组合控件**

并不需要自己去绘制视图上显示的内容，而只是用系统原生的控件

可以将几个系统原生的控件组合到一起，这样创建出的控件就被称为组合控件

布局文件关联形式：

mytitle.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="horizontal"  
 android:background="#FF0"**>  
  
 <**Button  
 android:id="@+id/btnBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="返回"  
 android:textColor="#FFF"**/>  
  
 <**TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:text="这是黄继升创建的Title控件"  
 android:textAppearance="?android:attr/textAppearanceLarge"  
 android:textColor="#FFF"** />  
</**LinearLayout**>

MyTitle.java

**package** com.amaker.hjs.customcontrol.view;  
  
**import** android.content.Context;  
**import** android.util.AttributeSet;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.widget.LinearLayout;  
  
**import** com.amaker.hjs.customcontrol.R;  
  
*/\*以关联布局文件方式创建View\*/***public class** MyTitle **extends** LinearLayout {  
 **public** MyTitle(Context context) {  
 **super**(context);  
 *//解析XML* View view = inflater(context);  
 **this**.addView(view);*//把这个view添加到自身上* }  
  
  
*/\*以代码方式创建View\*/* **public** MyTitle(Context context, AttributeSet attrs) {  
 **super**(context);  
 *//解析XML* View view = inflater(context);  
 **this**.addView(view);*//把这个view添加到自身上* }  
  
 **public** View inflater(Context context){  
 LayoutInflater inflater = LayoutInflater.*from*(context);  
 View view = inflater.inflate(R.layout.***mytitle***,**null**);  
 **return** view;  
 }  
}

DemoActivity.java

**package** com.amaker.hjs.customcontrol;  
  
**import** android.app.Activity;  
**import** android.os.Bundle;  
  
**import** com.amaker.hjs.customcontrol.view.MyTitle;  
**import** com.amaker.hjs.customcontrol.view.MyTitle2;  
**import** com.amaker.hjs.customcontrol.view.MyView2;  
  
**public class** DemoActivity **extends** Activity {  
 **private** MyTitle **myTitle**;  
 **private** MyTitle2 **myTitle2**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 **myTitle** = **new** MyTitle(**this**);setContentView(**myTitle2**);  
 }  
}

代码直接实现形式：

MyTitle2.java

**package** com.amaker.hjs.customcontrol.view;  
  
**import** android.content.Context;  
**import** android.content.res.TypedArray;  
**import** android.graphics.Color;  
**import** android.util.AttributeSet;  
**import** android.view.Gravity;  
**import** android.widget.Button;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
  
**import** com.amaker.hjs.customcontrol.R;  
  
**public class** MyTitle2 **extends** LinearLayout {  
 **private** TextView **tvTitle**;  
 **private** Button **btnBack**;  
 **private** Context **context**;  
  
 **private** String **title**=**"This is Title"**;  
 **public** MyTitle2(Context context) {  
 **super**(context);  
 **this**.**context**=context;  
 init();  
 }  
  
 **public** MyTitle2(Context context, AttributeSet attrs) {  
 **super**(context,attrs);  
 **this**.**context**=context;  
 TypedArray ta = context.obtainStyledAttributes(attrs, R.styleable.***myTitleAttrs***);  
 **title**=ta.getString(R.styleable.***myTitleAttrs\_title***);  
 init();  
 }  
  
 **private void** init(){  
 *//由于没有布局文件可以关联后拿到控件，所以需要自己去new出来* **tvTitle**=**new** TextView(**context**);  
 **btnBack**=**new** Button(**context**);  
  
 *//设置排列方式* **this**.setOrientation(LinearLayout.***HORIZONTAL***);  
  
 *//必须是父容器（LinearLayout）定义的参数  
 //参数：宽度、高度、权重  
 //参数：宽度、高度* LayoutParams params = **new** LayoutParams(280,LayoutParams.***WRAP\_CONTENT***,1);  
 **tvTitle**.setLayoutParams(params);  
 **tvTitle**.setText(**title**);  
 **tvTitle**.setTextColor(Color.***WHITE***);  
 **tvTitle**.setGravity(Gravity.***CENTER\_HORIZONTAL***);  
  
 **btnBack**.setText(**"点击我"**);  
 **btnBack**.setTextColor(Color.***WHITE***);  
  
 **this**.setOrientation(LinearLayout.***HORIZONTAL***);  
 **this**.setBackgroundColor(Color.***YELLOW***);  
  
 *//添加控件到容器中* **this**.addView(**btnBack**);  
 **this**.addView(**tvTitle**);  
 }  
  
  
}

res/values/attr.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**resources**>  
 <**declare-styleable name="myTitleAttrs"**>  
 <**attr name="title" format="string"**></**attr**>  
 </**declare-styleable**>  
</**resources**>

activity\_demo2.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:mytitle2="http://schemas.android.com/apk/res/com.amaker.hjs.customcontrol.view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
  
 <**com.amaker.hjs.customcontrol.view.MyTitle2  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
</**LinearLayout**>

DemoActivity.java

**package** com.amaker.hjs.customcontrol;  
  
**import** android.app.Activity;  
**import** android.os.Bundle;  
  
**import** com.amaker.hjs.customcontrol.view.MyTitle;  
**import** com.amaker.hjs.customcontrol.view.MyTitle2;  
**import** com.amaker.hjs.customcontrol.view.MyView2;  
  
**public class** DemoActivity **extends** Activity {  
 **private** MyTitle **myTitle**;  
 **private** MyTitle2 **myTitle2**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);**myTitle2**= **new** MyTitle2(**this**);setContentView(**myTitle2**);  
 *//或  
 //*setContentView(R.layout.***activity\_demo2***);  
 }  
}



**继承控件**

并不需要自己重头去实现一个控件，只需要去继承一个现有的控件，然后在这个控件上去增加一些新的功能，就可以形成一个自定义的控件。

继承控件的特点就是不仅能够按照我们的需求加入相应的功能，还可以保留原生控件的所有功能

MyButton.java

**package** com.amaker.hjs.customcontrol.view;  
  
**import** android.annotation.SuppressLint;  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.util.AttributeSet;  
**import** android.view.MotionEvent;  
**import** android.widget.Button;  
  
@SuppressLint(**"AppCompatCustomView"**)  
**public class** MyButton **extends** Button {  
 **private int cx**,**cy**;  
 **private int radius**;  
 **private** Paint **paint**;  
 **private int paintColor**;  
 **public** MyButton(Context context) {  
 **super**(context);  
 init();  
 }  
  
 **public** MyButton(Context context, AttributeSet attrs) {  
 **super**(context, attrs);  
 init();  
 }  
  
 **public** MyButton(Context context, AttributeSet attrs, **int** defStyleAttr) {  
 **super**(context, attrs, defStyleAttr);  
 init();  
 }  
  
 **private void** init(){  
 **paint**=**new** Paint();  
 **paint**.setColor(Color.***RED***);  
 **this**.setText(**""**);  
 **this**.setBackgroundColor(Color.***TRANSPARENT***);  
 }  
  
 **public void** setPaintColor(**int** color){  
 **paintColor**=color;  
 **paint**.setColor(**paintColor**);*//画笔也要改颜色  
  
 //在UI主线程进行界面刷新更改颜色。在主线程中，必须用invalidate()方法；* invalidate();  
 *//如若是在实现了Runnable的实例中，即工作线程中，就用postInvalidate()方法* }  
  
 @Override  
 **protected void** onDraw(Canvas canvas) {  
 **int** width = getWidth();  
 **int** heignt = getHeight();  
  
 **cx**= width/2;  
 **cy**=heignt/2;  
  
 **radius**=**cx**<**cy**?**cx**-10:**cy**-10;  
 canvas.drawCircle(**cx**,**cy**,**radius**,**paint**);  
 **super**.onDraw(canvas);  
 }  
  
 @Override  
 **public boolean** onTouchEvent(MotionEvent event) {  
 **switch** (event.getAction()){  
 **case** MotionEvent.***ACTION\_DOWN***:  
 setPaintColor(Color.***YELLOW***);  
 **return true**;  
 **case** MotionEvent.***ACTION\_UP***:  
 setPaintColor(Color.***BLUE***);  
 **return true**;  
 }  
 **return super**.onTouchEvent(event);  
 }  
}

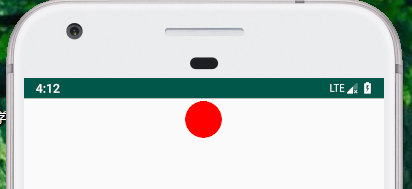
activity\_demo3.xml

*<?***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"**>  
  
 <**com.amaker.hjs.customcontrol.view.MyButton  
 android:id="@+id/myButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="com.amaker.hjs.customcontrol.view.MyButton"** />  
</**LinearLayout**>

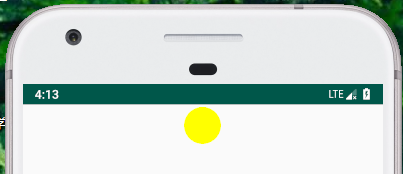
DemoActivity.java

**package** com.amaker.hjs.customcontrol;  
  
**import** android.app.Activity;  
**import** android.os.Bundle;  
  
**import** com.amaker.hjs.customcontrol.view.MyTitle;  
**import** com.amaker.hjs.customcontrol.view.MyTitle2;  
**import** com.amaker.hjs.customcontrol.view.MyView2;  
  
**public class** DemoActivity **extends** Activity {  
 **private** MyTitle **myTitle**;  
 **private** MyTitle2 **myTitle2**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 *//myTitle = new MyTitle(this);  
 //myTitle2= new MyTitle2(this);  
 //setContentView(myTitle);  
 //setContentView(myTitle2);* setContentView(R.layout.***activity\_demo3***);  
 }  
}

初始化：



按下按钮：



松开按钮：

