第四讲　　控件的运用

使用说明

【源代码根目录】:　工程目录 WidgetBase

【记号】:

(@编程练习)：表明该实验是需要编将工程文件和实验报告一起提交。

(@团队编程练习):表明该实验是以小组为单位完成的，每个小组完成一份程序和报告即可，报告和工程和其他的实验报告和工程最后要分开交。

本次实验的个人编程部分，请建立在名为 KH4 的 project 中。团队编程练习建立 KHTD4 的工程。

【实验报告的要求】

1、 文件名规范：　学号+姓名+实验名称.doc

2、 内容格式见　实验报告格式.doc

目录

【课堂实验 KT4.1】:UI 中的多线程

【工程模块】: UiThreadSample
【工程描述】: 该程序展示了 UI 线程外更新 UI 的几种基本方法。
【目标要求】:

- 掌握 UI 线程外更新 UI 的几种方法

【实验步骤】:
1、运行程序，点击按钮观察现象并分析。
2、修改如下代码，更换不同的 loadimage 函数运行。

```
bt.setOnClickListener(new View.OnClickListener(){

    public void onClick(View v) {

        // loadImage1();
       // loadImage2();
        // loadImage3();
        // loadImage4();
        loadImage5();
    }
});
```

【知识点注释】

- 关于 UI 线程必须知道的两件事：

1、 直接设置或调用 VIEW 类的属性方法都必须在 UI 主线程中进行。如果想在 UI 主线程外更改 VIEW 对象的可视化属性，则需要按照本例中的方法来做。

2、 在 UI 主线程中不要执行可能会引起阻塞的操作（例如文件读写，网络访问等）

- 本例因为要访问网络因此需声明权限（AndroidManifest.xml）

```
<uses-permission android:name="android.permission.INTERNET" />
```

- 异步任务类是 android 提供的简洁明了的执行 UI 线程外操作的方式，类声明时需要提供泛型参数：

AsyncTask 定义了三种泛型类型 Params，Progress 和 Result。
Params 启动任务执行的输入参数，比如 HTTP 请求的 URL。
Progress 后台任务执行的百分比。用于显示后台执行的进度。
Result 后台执行任务最终返回的结果，比如 String （可以是集合类型）

通常需要实现两个回调函数：

```
protected Bitmap doInBackground(String... urls)
```

该函数在 UI 线程外执行，用于需要后台执行耗时阻塞的操作。

```
protected void onPostExecute(Bitmap result)
```

该函数在 UI 线程中执行，可以操控全部的 UI 控件。

【问题】

- 若将本例中的 loadImage2,和 loadImage5 做如下修改，会有什么现象，请分析原因。

```
private void loadImage2(){
    new Thread(new Runnable() {
        public void run() {
            //final Bitmap bitmap =
loadImageFromRawResource(R.raw.image1);
            iv.post(new Runnable() {
                public void run() {
                final Bitmap bitmap = loadImageFromNetwork();
                    iv.setImageBitmap(bitmap);
                    // iv.setImageResource(R.drawable.image1);
                }
            });
        }
    }).start();
}
```

```
private void loadImage5(){
    new Thread(new Runnable() {
        public void run() {

            MainActivity.this.runOnUiThread(new Runnable() // 工
作线程刷新 UI
            {//这部分代码将在 UI 线程执行，实际上是 runOnUiThread
post Runnable 到 UI 线程执行了
                @Override
                public void run() {
          final Bitmap bitmap = loadImageFromNetwork();
                    iv.setImageBitmap(bitmap);
                }
            });
                            }
```

- （@编程练习）在一个 Activity 中放置一个 textview，一个按钮，按动按钮后，textview 会从屏幕左侧走到右侧并循环滚动（走马灯效果）。再按后停止滚动。请分别用本例中的 3 种线程方法设计。

设置和获取控件的坐标属性可以用：setLeft()、getLeft()，获取屏幕宽度的方法请自行网上查找。

- **（@编程练习）**模仿本例使用 `AsyncTask` 方法，从网络上下载一个网页，并将该网页内容（`html` 文本）显示在 `Textview` 中。

【课堂实验 KT4.2】:自定义 View

【工程模块】: CustomComponentSample
【工程描述】: 该程序展示了完全自定义控件的方法。
【目标要求】:

理解 view 类绘制的基本原理
掌握自定义 view 子类的基本方法
掌握在 xml 中定义自定义控件属性的基本方法
掌握自动刷新 view 的基本方法

【实验步骤】:

1、 运行程序，观察现象。
2、 理解分析代码

【知识点注释】

- **自定义控件的基本过程**
  - 自定义控件首先需要生成一个继承自 View 的子类：

```
public class GameView extends View
```

  - 重载 onDraw 方法

```
public void onDraw(Canvas canvas)
```

onDraw 方法决定了 View 对象的外观，每当系统开始显示 View 对象时就会回调该对象的 onDraw 函数。开发者可以通过参数 canvas 来绘制自定义控件的外观（仅限于 2D 图形图像）。本例中绘制了一个矩形:

```
canvas.drawRect(mX, mY, mX+80, mY+40, mPaint);
```

mX,mY 是矩形左上顶点坐标，mX+80, mY+40 是右下顶点坐标，mPaint 是一个画笔对象，可以决定绘制的颜色以及是否实心或者空心等特性。

- **如何在在 xml 中使用自定义控件**
  - 若要在 xml 中使用自定义控件，则必须实现如下形式的构造函数：

```
public GameView(Context context, AttributeSet as)
```

  - 在 xml 中可以自定义属性资源，本例中定义在 res/string.xml 中（也可以定义在其他 xml 中，比如 attrs.xml 或 style.xml，只要用<resource>标签包围即可）：

```
<resources>
    <string name="app_name">CustomComponentsSample</string>
    <declare-styleable name="GV_atts">
        <attr name="gx" format="integer" />
        <attr name="gy" format="integer" />
        <attr name="gcolor1" format="color" />
        <attr name="gcolor2" format="color" />
        <attr name="gcolor3" format="color" />
        <attr name="gcolor4" format="color" />
    </declare-styleable>
</resources>
```

这是一个属性集资源，其名称为"GV_atts"。其中有若干个属性，比如属性名为"gx"的属性，其数值类型为"integer"。

- 在 xml 中插入自定义控件：

```
<com.example.me.customcomponentssample.GameView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/gv"
    tools:gx="100"
    tools:gy="240"
    tools:gcolor1="#ff0000ff"
    tools:gcolor2="#ff00ff00"
    tools:gcolor3="#ffff0000"
    tools:gcolor4="#fffffff0"
    />
```

注意：1、自定义控件必须带有完整的包名，比如：

<com.example.me.customcomponentssample.GameView>。而不能是<GameView>

2、如果自定义控件类被定义成 Activity 的内部类，则引用的语法是：

<com.example.me.customcomponentssample.MainActivity$GameView>

3、 属性前的名字空间（namespace）应是 tools 而不是常见的 android，这是工程向导为我们添加好的。

- 在程序中获取 xml 中定义的属性值

```
public GameView(Context context, AttributeSet as)
{
   super(context,as);
   TypedArray a = context.obtainStyledAttributes(as,
             R.styleable.GV_atts);
          mX = a.getInteger(R.styleable.GV_atts_gx, 22);
          mY = a.getInteger(R.styleable.GV_atts_gy, (320-80)/2);
          mcolor1=a.getColor(R.styleable.GV_atts_gcolor1, Color.BLUE);
          mcolor2=a.getColor(R.styleable.GV_atts_gcolor2, Color.GREEN);
          mcolor3=a.getColor(R.styleable.GV_atts_gcolor3, Color.RED);
          mcolor4=a.getColor(R.styleable.GV_atts_gcolor4, Color.YELLOW);
          //
          //mcolor4 =
as.getAttributeResourceValue("cn.edu.shu.cs.android.experiment06.game_myview", "gcolor4",
Color.YELLOW);
}
```

注意:属性集对应的名称是:R.styleable.GV_atts，而属性对应的名称是属性集名+下横线+属性名，例如：gx 属性对应的引用常量是 R.styeable.GV_atts_gx。

- **如何让自定义控件自动刷新**

让控件刷新，就是要让 UI 线程回调该控件的 onDraw 函数，方式是调用该控件的 invalidate()方法。为了让自定义控件不断地自动刷新，可以创建一个线程不断地发出 invalidate 的请求。`mGameView.invalidate();`

```
// 开启线程
```

```
new Thread(new GameThread()).start();

class GameThread implements Runnable
{
public void run()
{
while (!Thread.currentThread().isInterrupted())
{
try
{
Thread.sleep(100);
}
catch (InterruptedException e)
{
Thread.currentThread().interrupt();
}
//使用 postInvalidate 可以直接在线程中更新界面
mGameView.postInvalidate();
}
}
}
```

注意：Thread.sleep(100);的睡眠单位是 ms，100ms=1/10s。也就是每隔 100ms 发出一次刷新 view 的请求。
mGameView.postInvalidate();这里没用 mGameView.Invalidate();是因为该函数只能用在 UI 线程内。

【问题】

1、通常为编码方便，我们也不另外定义 Runnable 类，而是直接让 GameView 实现 Runnable

接口，请改写本例并调试。

2、（@编程练习）

创建一个电子表自定义控件，该控件获取当前的系统时间（小时：分钟：秒）并不断地刷新显示。要求：

为该控件定义属性，并在 xml 中设置其属性（比如：24 小时还是 12 小时格式或者字体颜色等）。

java 代码：

```
import java.text.SimpleDateFormat;

SimpleDateFormat formatter = new SimpleDateFormat ("yyyy 年 MM 月 dd 日  HH:mm:ss ");
Date curDate = new Date(System.currentTimeMillis());//获取当前时间
String str = formatter.format(curDate);
```

以上可以获取当前的年月时分,也可以分开写（如下）:

java 代码：

```
SimpleDateFormat sDateFormat = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
String date = sDateFormat.format(new java.util.Date());
```

如果想获取当前的年月,则可以这样写(只获取时间或秒种一样):

java 代码：

```
SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM");
String date=sdf.format(new java.util.Date());
```

当然还有就是可以指定时区的时间(待):

```
  df=DateFormat.getDateTimeInstance(DateFormat.FULL,DateFormat.FULL,Locale.CHINA);
System.out.println(df.format(new Date()));
```

取得系统日期:

```
Calendar c = Calendar.getInstance();
year = c.get(Calendar.YEAR)
month = c.grt(Calendar.MONTH)
day = c.get(Calendar.DAY_OF_MONTH)
```

取得系统时间：

```
Calendar c = Calendar.getInstance();
hour = c.get(Calendar.HOUR_OF_DAY);
minute = c.get(Calendar.MINUTE)
```

利用 Time 获取:

```
Time t=new Time(); // or Time t=new Time("GMT+8");  加上 Time Zone 资料。
  t.setToNow(); //  取得系统时间。

   int year = t.year;

  int month = t.month;
  int date = t.monthDay;
  int hour = t.hour; // 0-23
  int minute = t.minute;
  int second = t.second;
```

【课堂实验 KT4.3】:2D 绘图

【工程模块】：Canvas2DSample
【工程描述】：该程序展示了使用 canvas ,paint 对象进行 2d 绘图的基本方法。
【目标要求】：
掌握基本的 2d 绘图方法
掌握旋转 canvas 绘图的方法

【实验步骤】:
1、运行程序，观察现象。
2、分析理解代码
【知识点注释】:
使用旋转画布的方法，可以绘制按照角度（xxf）旋转的效果,函数原型如下：
canvas.rotate(float degree);（按照默认原点旋转 degree 度）
Canvas.rotate(float degree,float px,float py); //px,py 是旋转原点（不动点）的坐标

```
/* 线锁定画布 */
        canvas.save();
/* 旋转画布 */
        canvas.rotate(45.0f);
        mPaint.setColor(Color.RED);
/* 绘制矩形 */
        canvas.drawRect(50, 5, 90, 25, mPaint);
/* 解除画布的锁定 */
        canvas.restore();
```

【问题】

1、canvas.rotate(float degree);（按照默认原点旋转 degree 度）

观察并修改代码，根据现象分析这个默认原点在哪里。

2、（@编程练习）

利用旋转画布的方法设计一个会走动的挂钟，该控件获取当前的系统时间（小时：分钟：秒）并不断地刷新显示。要求：有时，分，秒三根指针，每隔 1 秒刷新一次。

【课堂实验 KT3.4】:触屏事件-手指画

【工程模块】: FingerPaintSample
【工程描述】: 展示了一个随手涂鸦的绘图程序。

【目标要求】:
掌握触屏事件的编程方法。

【实验步骤】:
1、 运行程序观察并分析代码。
【知识点注释】
触屏事件如下，注意该触屏事件是定义在 MyView 中的，view 和 activity 均可接受触屏事件，只是响应范围不同。

```
@Override
    public boolean onTouchEvent(MotionEvent event) {
        float x = event.getX();
        float y = event.getY();

        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                touch_start(x, y);
                invalidate();
                break;
            case MotionEvent.ACTION_MOVE:
                touch_move(x, y);
                invalidate();
                //mX = x;
                //mY = y;
                break;
            case MotionEvent.ACTION_UP:
                touch_up();
                invalidate();
                break;
        }
        return true;
    }
}
```

【问题】

（@团队编程练习）

综合利用本章所学知识，设计开发一个互动小游戏。
比如：推箱子，扫雷，汉诺塔，射击，乒乓球等