

Polliwog v-1.0

A tool of android demo rapid generation for college student.

设计者: 高翔宇

开发者: 汤胜中, 祝一帆

协助者: 莫良言, 何梦娇

(Email:cwing.gao@gmail.com / 302787376@qq.com)

Introduce

小蚂蚁是针对大学生或其他具有一定计算机基础的非专业人群提供的一套快速开发 Android App Demo 的实现框架和辅助软件。该设计框架与 Android Studio(From Google)软件结合, 使用我们的辅助软件(Polliwog v-1.0)可以实现了从可视化的界面开发到 Java 代码实现的自动生成, 自动生成部分主要是形式固定的代码部分, 对于面向用户的应用型 App 的 Demo 开发最为合适, 用户可以根据需求自主调整自动生成的内容和格式, 修改我们的框架以适合自身的开发需求。

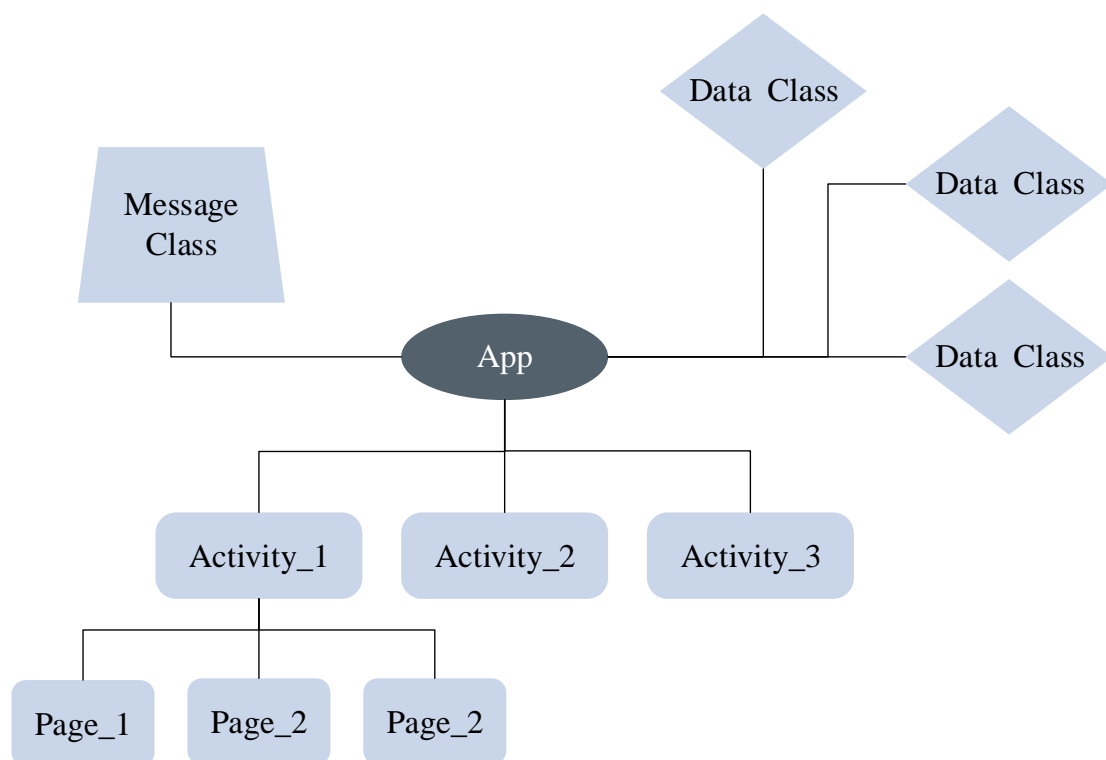
目前只有一套开发框架设计, 在后续版本更新迭代过程中, 我们将提供更多的 App 设计框架以及更个性化的辅助软件, 满足更多 App 开发需求。

Detailed introduction

Framework introduction

该框架主要是围绕 Activity 做的设计，在其层次之下设计了 Page 类，用以将每个布局文件进行实现，而后使用界面时创建 Page 对象来完成界面控件的实际功能。

框架结构如下图：



这要求开发人员在开发过程中，布局文件的设计需具有整体性，一个完成的页面由一个***.xml 文件完成或最终将各种零散控件在一个***.xml 文件内引用。

除 Page 类外，还有 Adapter 类、Msg 类等。其中 Adapter 类

是为 Listview 等控件设计实现的，以方便将 Item 布局通过适配器的方式加入到 Listview 控件中。Msg 类是一个消息声明类，其内声明 Page 与 Page 间、Page 与 Activity 间的信息编码和命令编码。当然，你可以通过修改我们的代码生成新的类生成器，以适应你自己设计得框架。

Page 自身通信和 Page 与 Activity 间的通信利用 Handler 实现。

Introduction of auxiliary software

XML 文件编写

关于布局文件，其开发流程与原开发流程没有区别，主要使用 Android Studio 内部的开发工具以可视化的拖拉开发与代码编写的结合来完成。

其中，为方便辅助代码生成工具的使用，每个 Page 布局要有一个整体性的布局文件来组合完成，并且你文件命名在前缀或者后缀出应统一，Adapter 的 Item 界面也是如此。

Java 代码生成

生成 XML 代码之后，就可以用辅助软件按照固定格式生成 Java 代码，开发者只需要填写控件的响应逻辑部分就可以完成一个简单 Demo 的开发。

辅助软件使用说明

辅助软件可以将多个布局文件的组成的文件夹统一识别转换

成 Java 代码，也可以单独单个文件转换。在整体工程生成时，需要在设置内先设置好识别字符，以让软件区分你的布局文件类型，比如如果所有的 Page 布局文件均有前缀（或者后缀）Page，你便可以在设置内设置，设置内容将以文件形式保存在本地，以供继续使用，单个文件生成则不需要。

每个控件要生成怎样的实现函数是存储在 widget_collection.csv 文件内的，转换时软件首先识别 XML 文件然后读取 widget_collection.csv 内的设定生成 Java 代码，所以使用者可在不修改程序源码的基础上自行扩展控件类型和控件的专属函数。

Appendix

Java 代码生成范例

PageClass

```
package cwing; //工程包

//基础包含
import android.os.Handler;
import android.os.Message;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;

//布局控件类包含
import android.widget.Button;
```

```

public class PageName implements View.OnClickListener, {
    AppCompatActivity parent=null; //父活动
    MainActivity.MainHandler mainHandler=null;    //核心句柄处理器
    //检测到的所有控件声明,格式:
    //      privat 控件类 控件类名+名字=null;
    private Button buttonName=null;

    //构造方法
    public PageGet(AppCompatActivity parent, Handler mainHandler){
        this.parent= parent;
        this.mainHandler=( MainActivity.MainHandler)mainHandler;
    }

    public void setUI() {
        parent.setContentView(R.layout.page_name);
        ActionBar actionBar = parent.getSupportActionBar();
        actionBar.hide();
    }

    public void loadControl() {
        buttonName=(Button) parent.findViewById(R.id.butt_back);
    }

    @Override
    public void onClick(View view) {    //控件专有函数

        Message msg=new Message();
        switch (view.getId()){
            case R.id.butt_back:
                break;
        }

    }

    void memoryClear()
    {
    }

    public void onClose() {

```

```

        return false;
    }
}

```

Adapter

```

public class AdapterHomeList extends ArrayAdapter<ContentCourseItem> {

    private int resourceId;
    private AppCompatActivity parent=null;
    private List<ContentCourseItem> list;
    Main.MainHandler mainHandler=null;

    public AdapterHomeList(@NonNull Context context, @LayoutRes int
resource, List<ContentCourseItem> objects, Main.MainHandler handler) {
        super(context, resource, objects);
        this.parent=(AppCompatActivity) context;
        Log.d("resourceId", resource + "");
        this.resourceId = resource;
        this.list=objects;
        this.mainHandler=handler;
    }

    @Override
    public View getView(int position, @Nullable final View convertView,
@NonNull ViewGroup parent) {
        Log.d("resourceId", resourceId + "");
        View view = LayoutInflater.from(getContext()).inflate(resourceId,
parent, false);

        ContentCourseItem content=(ContentCourseItem) getItem(position);

        ImageView image1=(ImageView) view.findViewById(R.id.image_1);
        TextView textName1=(TextView)
view.findViewById(R.id.text_name_1);
        TextView textTeacher1=(TextView)
view.findViewById(R.id.text_teacher_1);
        TextView textType1=(TextView) view.findViewById(R.id.text_type_1);

        image1.setImageBitmap(content.image_1);
    }
}

```

```

        textName1.setText(content.name_1);
        textTeacher1.setText(content.teacher_1);
        textType1.setText(content.type_1);

        ImageView image2=(ImageView) view.findViewById(R.id.image_2);
        TextView textName2=(TextView)
view.findViewById(R.id.text_name_2);
        TextView textTeacher2=(TextView)
view.findViewById(R.id.text_teacher_2);
        TextView textType2=(TextView) view.findViewById(R.id.text_type_2);

        image2.setImageBitmap(content.image_2);
        textName2.setText(content.name_2);
        textTeacher2.setText(content.teacher_2);
        textType2.setText(content.type_2);

        LinearLayout linearLayout1=(LinearLayout)
view.findViewById(R.id.linear_1);
        LinearLayout linearLayout2=(LinearLayout)
view.findViewById(R.id.linear_2);

        linearLayout1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Message msg=new Message();
                msg.what= Msg.ACTION_INTRO_COURSE;
                mainHandler.sendMessage(msg);
            }
        });

        linearLayout2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Message msg=new Message();
                msg.what= Msg.ACTION_INTRO_COURSE;
                mainHandler.sendMessage(msg);
            }
        });

        return view;
    }
}

```

控件识别附表（可自行扩展）

名称	功能描述	所属包	使用及函数
Button	按钮	android.widget.Button	onClick(View v)
ToggleButton	按钮	android.widget.ToggleButton	
Textview	文本	android.widget.TextView	——
checkbox	复选框	android.widget.CheckBox	
RadioButton	单选按钮	android.widget.RadioButton	
checkedTextView	文本	android.widget.CheckedTextView	
spinner	下拉框	android.widget.Spinner	
progressbar	进度条	android.widget.ProgressBar	
seekbar	进度条	android.widget.SeekBar	
QuickcontactBadge	联系人快捷 标示	android.widget.QuickContactBad ge	
ratingbar	触摸/拖动 或使用键来 设置评分	android.widget.RatingBar	
switch	开关	android.widget.switch	
Space	开关	android.widget.Space	
grid layout	网格布局	——	

framelayout	帧布局	android.widget.FrameLayout	
linearlayout	线性布局	android.widget.LinearLayout	
relativelayout	相对布局	android.widget.RelativeLayout	
tablelayout	表格布局	android.widget.TableLayout	
tablerow	表格布局	android.widget.TableRow	
radiogroup	单选框组	android.widget.RadioGroup	
listview	列表视图	android.widget.ListView	
expandablelistview	可展开的列表组件	android.widget.ExpandableListView	
scrollview	滚动视图	android.widget.ScrollView	
horizontalscrollview	水平滚动控件	android.widget.HorizontalScrollView	
tabhost	导航栏布局	android.widget.TabHost	
webview	Web 视图	android.webkit.WebView	
searchview	搜索框	android.widget.SearchView	
imagebutton	图像按钮	android.widget.ImageButton	
imageview	图片视图	android.widget.ImageView	
videoview	影像视图	android.widget.VideoView	
datepicker	日期选择器	android.widget.DatePicker	
timepicker	时间选择器	android.widget.TimePicker	
calenderview	日历对话框	——	
textclock	文字时钟	android.widget.TextClock	

