

# 移动互联网技术及应用

## 大作业报告

姓名	班级	学号
陈立伟	2017211308	2017211396

2020.6

## 目录

1. 相关技术.....	3
2. 系统功能需求.....	3
3. 系统设计与实现.....	3
4. 系统可能的扩展.....	7
5. 总结体会.....	7

# 1. 相关技术

我负责的部分是视频播放部分及相关扩展功能，技术要点是 **VideoView** 播放视频相关实现（包括 **url** 的获取、单机界面视频暂停/播放等）以及双击响应方法的实现（双击点赞出现爱心特效），另外还包括使用 **intent** 进行页面跳转和传值。

# 2. 系统功能需求

本项目来源于移动互联网课程期末大作业，要求我们实现一个短视频播放 APP，实现视频信息流列表显示（包含封面图）、视频播放功能以及若干扩展功能。最终实现的 APP 应根据视频信息流列表点击获取 **URL** 信息并进行跳转播放。其中我负责视频播放部分的设计与实现。

根据项目描述，用户在使用本 APP 时首先进入的应该是视频信息流显示页面。当用户找到感兴趣的视频时可通过点击封面的方式跳转到视频播放页面。观看视频时，用户可通过点击视频的方式进行暂停/播放切换，去忙临时发生的事情。播放完毕后，用户可点击回退键返回视频信息流页面再次寻找感兴趣的视频。

# 3. 系统设计与实现

本项目主要可分为两大模块——即视频信息流页面和播放页面，其中我负责的是视频播放页面的设计实现。

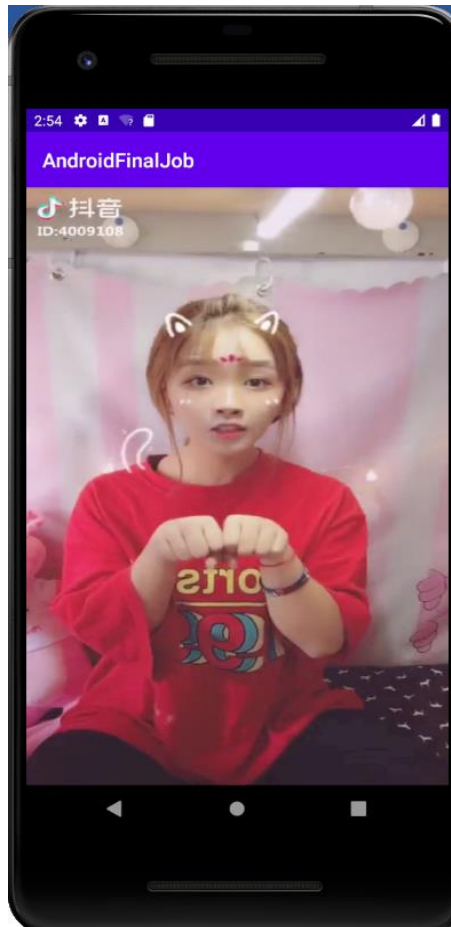
考虑到本项目的受众、功能需求以及开发时间等方面的要求，我决定采用页面简洁、操作便捷、功能健全的原则进行设计实现。

由于该视频播放软件受众是短视频用户，其特点为观看的视频持续时间短，视频切换频率高，故我采用的设计是放弃进度条模块，跳转后立即播放，播放过程中可以通过单击屏幕进行暂停/播放的切换，使得用户能够不受进度条占据屏幕的困扰，播放页面更加简洁美观。

跳转部分的设计为：单击首页的视频封面即可跳转到相关的播放页面，操作简单明了，任何人都可以立即上手使用。

下面我将就本页面的主要技术手段及实现效果做简要阐述，具体实现参见源码。

关于视频播放部分，由于视频短小，我简单采用 **VideoView** 控件作为前端组件，省去了进度条等遮挡界面又对用户体验没有太大提升的组件。其播放界面如下图所示：



其布局代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <RelativeLayout
        android:id="@+id/r1_like"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <ImageView
        android:id="@+id/iv_easy_like"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:visibility="gone"
        android:layout_centerInParent="true"
        android:background="@drawable/a_like_s"
    />
    </RelativeLayout>
</FrameLayout>
```

控制部分代码详见源码，这里简要说一下实现过程和关键代码段。首先页面跳转在 MyAdapter 中，采用 intent 进行页面跳转，同时将视频所需的 url 信息传值到新的 videoplayer activity 中，从而实现了指定页面跳转并规定了跳转后视频源的选取。跳转与接收代码段如下：

```

View.OnClickListener clickListener = new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Toast.makeText(v.getContext(), text: "url:" + position, Toast.LENGTH_SHORT).show();
        Log.d( tag: "URL", mDataset.get(position).feedUrl); //视频地址
        String temp = mDataset.get(position).feedUrl;
        Log.d( tag: "URL", temp); //视频地址

        Intent intent = new Intent(mcontext, videoplayer.class); //选择跳转到的页面
        intent.putExtra( name: "mURL", temp);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        Bundle bundle = new Bundle();
        bundle.putString("mURL", temp);
        intent.putExtras(bundle);
        mcontext.startActivity(intent);
    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.videoplayer);

    //接受Url参数
    String mURL = getIntent().getStringExtra( name: "mURL");

    String videoUrl2 = mURL;
    Uri uri = Uri.parse( videoUrl2 );
    videoView = (VideoView)this.findViewById(R.id.videoView );
    videoView.setVideoURI(uri);
    videoView.requestFocus();
    videoView.start();
}

```

解决了视频源的选取后，只需要设置 `setOnTouchListener` 函数即可实现点击视频暂停与播放功能。关于双击点赞功能，需要设置点击事件。当判断两次点击事件时间间隔小于设定值时，即调用双击事件函数，触发爱心特效。判断双击事件的方法如下：

```

@Override
public boolean onTouch(View v, MotionEvent event) {
    if (MotionEvent.ACTION_DOWN == event.getAction()) {
        count++;
        if (1 == count) {
            firClick = System.currentTimeMillis();
        } else if (2 == count) {
            secClick = System.currentTimeMillis();
            if (secClick - firClick < interval) {
                if (mCallback != null) {
                    mCallback.onDoubleClick();
                } else {
                    Log.e(TAG, msg: "请在构造方法中传入一个双击回调");
                }
                count = 0;
                firClick = 0;
            } else {
                firClick = secClick;
                count = 1;
            }
            secClick = 0;
        }
    }
    return true;
}

```

判断出双击事件后，我们还需要启动爱心图标的显示和动画演示，需要定义运动方法并调用动画方法进行演出。运动方法如下图所示：

```

@SuppressLint("NewApi")
class BezierEvaluator implements TypeEvaluator<PointF> {

    @Override
    public PointF evaluate(float fraction, PointF startValue,
                           PointF endValue) {
        float oneMinusT = 1.0f - fraction;
        PointF point = new PointF();
        PointF point0 = (PointF) startValue;
        PointF point1 = new PointF();
        point1.set(width, 0);

        PointF point2 = new PointF();
        point2.set(0, height);

        PointF point3 = (PointF) endValue;

        point.x = oneMinusT * oneMinusT * oneMinusT * (point0.x) + 4
            * oneMinusT * oneMinusT * fraction * (point0.x) + 2
            * oneMinusT * fraction * fraction * (point0.x) + fraction
            * fraction * fraction * (point0.x);

        point.y = oneMinusT * oneMinusT * oneMinusT * (point0.y)
            + oneMinusT * oneMinusT * fraction * (point0.y) + 2
            * oneMinusT * fraction * fraction * (point2.y) + fraction
            * fraction * fraction * (point3.y);

        return point;
    }
}

```

至此，我们就实现了双击爱心事件的配置，来看一下效果：



若需要观看完整效果可以看演示视频。视频中有提供 demo 完整演示，此处仅简单显示项目效果。

## 4. 系统可能的扩展

若未来将该项目扩展成为视频播放 APP（即引入长视频数据），则只需要将播放页面扩展为播放器页面即可，添加进度条控件保证视频可以任意跳转。此外还可以加入上传视频、收藏视频等功能，集成短视频分享平台。

## 5. 总结体会

本项目采用 **Android Studio** 进行开发，目标用户为闲暇时间较多且比较零碎，喜欢观看短视频的用户，运行系统系统为 **Android 5.0** 及以上。通过本次项目的开发，我对视频播放的相关控件和技术有了更进一步的认识和理解，通过研究实验学会了 **intent** 在 **adapter** 和 **activity** 之间传值的相关应用并能正确进行页面跳转和传值引用。最终开发出符合规范的能够实际运行的短视频 APP。