

# 中山大学移动信息工程学院本科生实验报告

## ( 2017 年秋季学期 )

课程名称：移动应用开发

任课教师：刘宁

年级	大三	专业 ( 方向 )	电政
学号	15331439	姓名	朱理锋
电话	15119665373	Email	<u>820917549@qq.co</u> <u>m</u>
开始日期	2017.11.19	完成日期	2017.11.26

### 一、 实验题目

Android 期中项目

### 二、 实现内容

三国辞典的详情页面和百科页面

### 三、 课堂实验结果

( 1 ) 实验截图

2:32

<

...



# 卫青

归属势力:其他

籍贯:兖州泰山郡平阳 (山东泰安市新泰)

生卒:??

## 人物简介

卫青 男 史实人物 字:仲卿 生卒 ( ? - 前106 ) 籍贯:兖州泰山郡平阳 (山东泰安市新泰)

## 历史记载



## (2) 实验步骤以及关键代码

详情页主要是布局和将数据显示。其中比较重要的代码属于网络头像的加载。由于安卓不像 web 和 uwp 直接将 url 插入就可以加载显示这么方便，所以得另外开一个线程去获取图片，获取到后，再给 imageView 设置 source，更新 ui，代码如下：

```
public static class LoadImagesTask extends AsyncTask<String, Void, Bitmap> {
    private ImageView imageView;
    public LoadImagesTask(ImageView imageView) { this.imageView = imageView; }
    @Override
    protected Bitmap doInBackground(String... params) {
        URL imageUrl = null;
        Bitmap bitmap = null;
        InputStream inputStream = null;
        try {
            imageUrl = new URL(params[0]);
            HttpURLConnection conn = (HttpURLConnection) imageUrl.openConnection();
            conn.setDoInput(true);
            conn.connect();
            inputStream = conn.getInputStream();
            bitmap = BitmapFactory.decodeStream(inputStream);
            inputStream.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return bitmap;
    }
    @Override
    protected void onPostExecute(Bitmap bitmap) {
        imageView.setImageBitmap(bitmap);
    }
}

// 头像
new Tools.LoadImagesTask(avatar).execute(data.avatar);
```

其次就是百科页面和主页的搜索，我们小组是用 **SQLite** 来存三国人物的信息的。搜索为了好做点，是搜索三国人物的名字，为了获取数据，我们得写 **sql** 语句，得到数据后，更新数据，**adapter** 通知更新。其中做了一下优化，如果搜索的 **key** 是空的，返回全部人物。但是又要搜索一遍数据库。所以我们做了一个缓存。搜索空时，直接返回。下一次进入到百科页面时，数据会被更新，不用担心数据不一致的问题。

```
} public List<Character> getSearchCharacters(String key) {
    Cursor cursor = db.query(DbHelper.TABLE_CHARACTER, columns: null, selection: "name like '%" + key + "%'", null, null);
    List<Character> res = cursor.getCount() != 0 ? getAllCharactersByCursor(cursor) : null;
    cursor.close();
    return res;
}

public List<Character> getSearchOwnCharacters(String key) {
    String select = "SELECT * " +
        "FROM " + DbHelper.TABLE_OWN + " AS o " +
        "JOIN " + DbHelper.TABLE_CHARACTER + " AS c " +
        "ON c._id = o.character_id " +
        "WHERE c.name LIKE '%" + key + "%'";
    Cursor cursor = db.rawQuery(select, selectionArgs: null);
    List<Character> res = cursor.getCount() != 0 ? getAllCharactersByCursor(cursor) : null;
    cursor.close();
    return res;
}
```

百科页是没有解锁的人物背景是有未解锁的浅透明度背景，且不能点击。解锁的是没有这个背景和可以点击进入详情页的。而我们的数据库有两张表，一张是解锁的表，一张是全部的表。解锁的表有个 **character\_id**，是全部的表的外键。要知道全部表中哪个人物的解锁状态。我们还是得写一个 **sql** 语句来查询一下。然后根据返回的结果做对应的更新 **ui** 和交互的操作。

```
} public boolean checkIfOwned(Character ch) {
    String select = "SELECT * " +
        "FROM " + DbHelper.TABLE_OWN + " AS o " +
        "WHERE o.character_id = " + String.valueOf(ch._id);
    Cursor cursor = db.rawQuery(select, selectionArgs: null);
    boolean res = cursor.getCount() != 0;
    cursor.close();
    return res;
}
```

```

public void convert(ViewHolder holder, Character data) {
    boolean unlocked = DbReader.getInstance(EncyclopediaActivity.this).checkIfOwned(data);
    TextView name = holder.getView(R.id.name);
    name.setText(data.name);
    TextView belong = holder.getView(R.id.belong);
    belong.setText(data.belong);
    TextView description = holder.getView(R.id.abstract_description);
    description.setText("\t\t\t\t" + data.abstractDescription);
    ConstraintLayout layout = holder.getView(R.id.list_item_wrapper);

    holder.getView(R.id.Locked_mask).setVisibility(unlocked ? View.INVISIBLE : View.VISIBLE);
    holder.getView(R.id.Locked_mask_text).setVisibility(unlocked ? View.INVISIBLE : View.VISIBLE);
    layout.setClickable(unlocked);
}

```

### (3) 实验遇到困难以及解决思路

不懂的问谷歌。安卓发展太久，解决方法层出不穷。

## 四、 课后实验结果

## 五、 实验思考及感想

使用 GitHub 作为代码托管和协作的工具。规范好协作流程。每个人分好模块，并行开发。高效，无误。你值得拥有。