**不依靠AS提供的View子类实现Collage**

摘要

本次实验的实验目的是理解UI视图和它们的子视图是如何被画出来的，实现自己的分层次的绘图。

实验设计主要遵循软件设计基本原则，代码样式基本原则。

实验内容主要包括：视图对子视图的增，删，改的管理，布局管理和绘制。

实验结论为：

子类通过继承具备基本的功能，同时，子类又对父类进行具体的实现，各个子类只负责自己实现的那一部分，只包含单一的职责。视图通过这种方式来实现具有各式功能的子视图。父类的方法可以只实现一部分通用功能，剩下的可以交给具体的子类去实现。父类的方法调用子类的这一方法，这种间接的递归调用可以实现多层次的类的完全实现。此实验的doLayout()和draw()正是使用此方法。

**关键词：**collage；view；layout；draw；子视图；

**目录**

**1实验目的和意义**

1.1 实验目的 ……………………………………………………………………… 5

1.2 实验意义 ……………………………………………………………………… 5

**2实验设计**

2.1概述 ………………………………………………………………………… 5

2.2项目架构 ………………………………………………………………………… 6

2.3测试截图 ………………………………………………………………………… 7

**结论** …………………………………………………………………………………… 9

**参考文献 ………………………………………………………………………………** 10

**1实验目的和意义**

**1.1实验目的**

本实验用于理解UI视图和它们的子视图这样的多层次结构是如何被绘制的，并实现自己版本的视图绘制。

**1.2实验意义**

理解交互层的构建，管理父亲和孩子视图的树，当子视图被增加，删除，修改的时候，更新界面

实现多层次视图的布局管理和绘制。

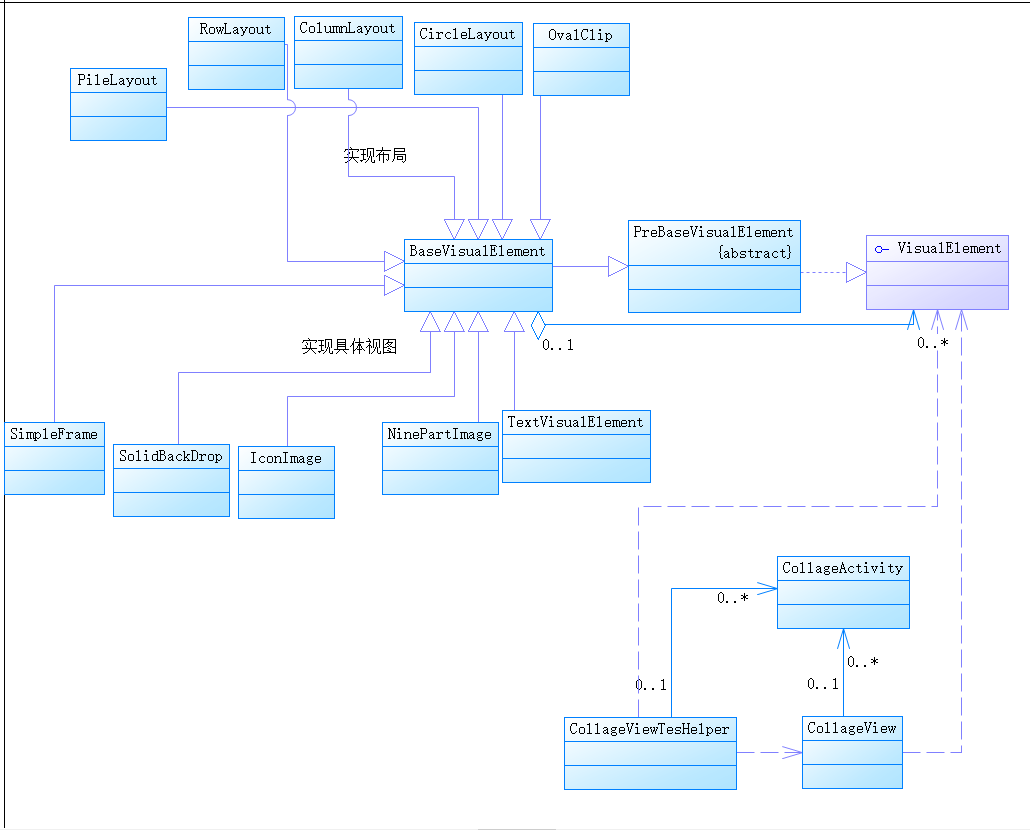
学会使用Canvas的translate和clip方法

**2实验设计**

**2.1概述**

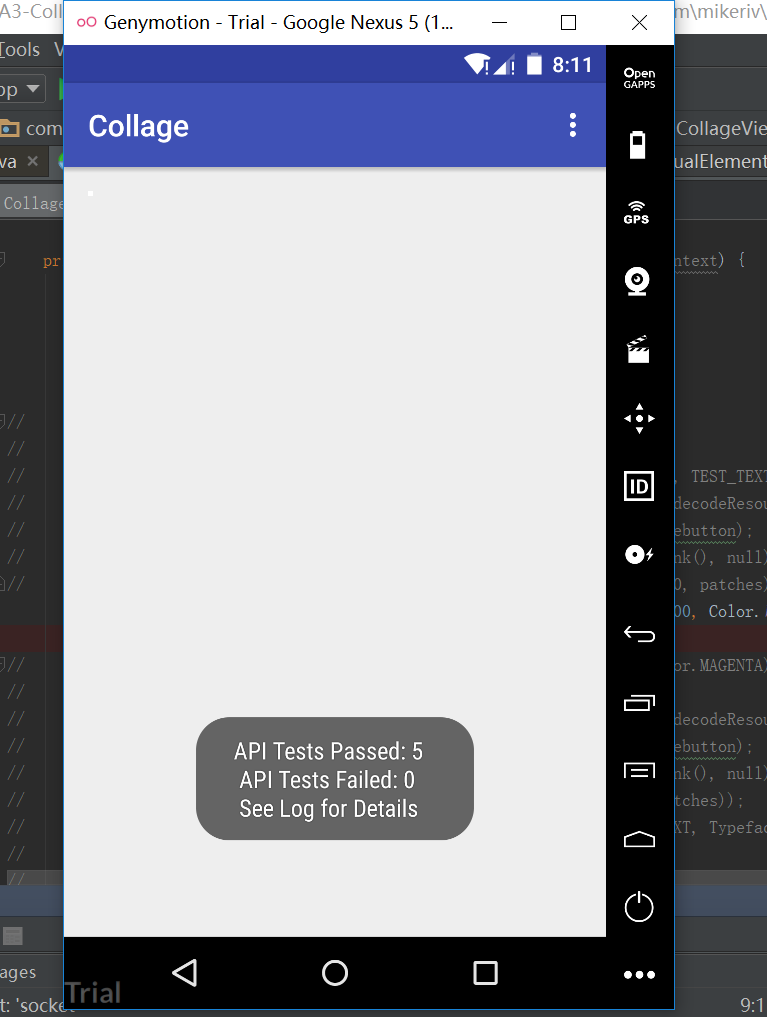
提供VisualElement, PrebaseVisualElement, CollageView, CollageActivity, CollageViewTesHelper接口或类，写一个BaseVisualElement继承PrebaseVisualElement,实现VisualElement的所有方法，写5个具体子视图实现具体的绘制，（SimpleFrame， SolidBackDrop， IconImage，NinePartImage, TextVisualElement）,写4个子类实现特定的布局（PileLayout， RowLayout， ColumnLayout, CircleLayout），写一个OvalClip剪切它的子视图来适应它的边界。写完以后取消CollageViewTesHelper的runTest的注释，来测试。

**2.2项目架构**



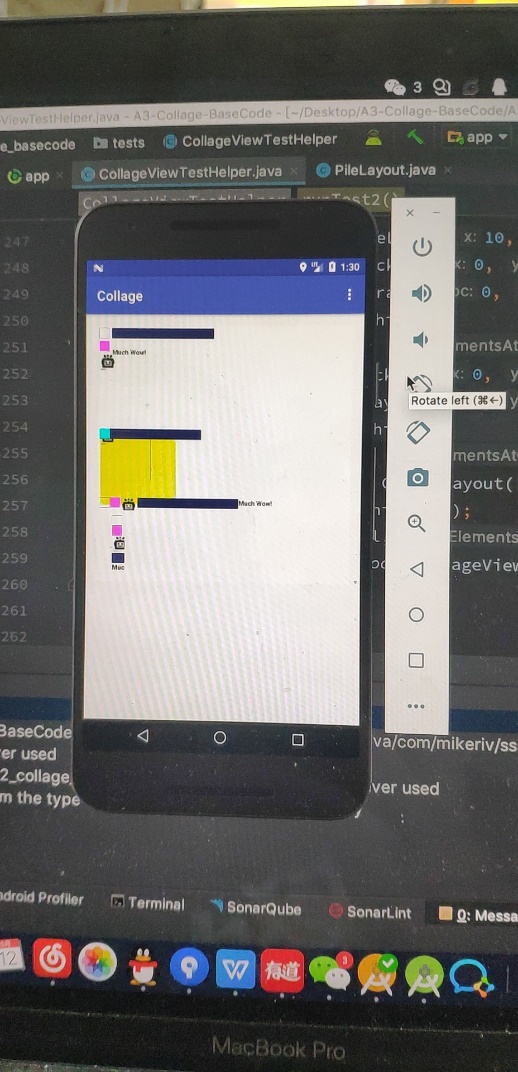
**2.3测试截图**

Test1:

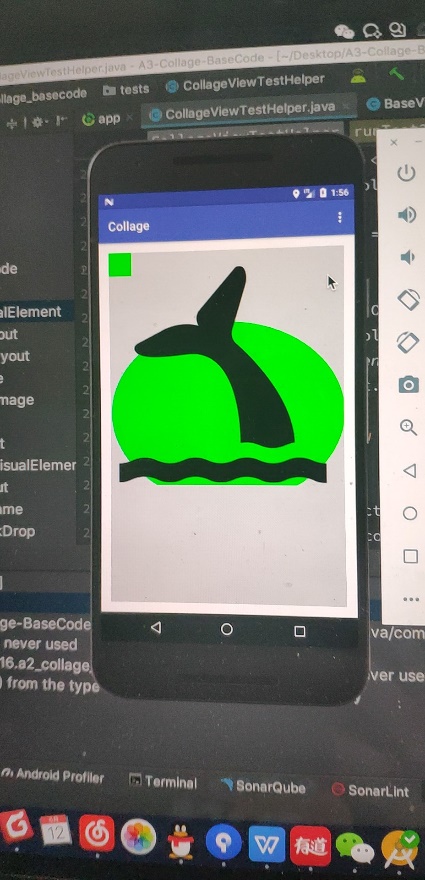


Test2和test3在我的电脑上测试会报错，但是我反复检查也找不出问题出在哪里，我把项目发给我一个懂android studio的亲戚，请他帮我看一下问题在哪里，结果在他的电脑上能运行，在这里使用的是在他的电脑上的运行图片

Test2:



Test3:



**结论**

关于视图的MVC架构使代码的组织变得简洁易懂，层次分明。Drawing包下VisualElement接口到PrebaseVisualElement抽象类到BaseVisualElement实体类，再到BaseVisualElement的各个子类，抽象层次逐渐降低，具体类依赖抽象，使得具体类可以单一的，完整的负责自己的职责，减小类的粒度。BaseVisualElement中有一个对VisualElement的引用list<VisualElement>，来建立视图的树,在doLayout中对每个此list中的元素调用doLayout,来实现对一棵树中所有的视图的布局，在draw中对每个此list中的元素调用draw来实现对一棵树中所有视图的绘制。界面的绘制是一个很好的关于具有多层次结构的项目的模型，做这个实验有利于之后的类似项目的开发。

**参考文献**

[1] 李望江. Android Canvas绘图描述Android Canvas 方法总结[J/OL]. <https://blog.csdn.net/qq_35427437/article/details/80045074>，2018.04.23

[2] Amy\_LuLu\_. Android之Drawable-NinePatchDrawable[J/OL]. <https://www.jianshu.com/p/9fa8d4b2e180>，2018.02.06

[3]L有只大大熊. NinePatchDrawable 简单代码应用[J/OL]. <https://blog.csdn.net/ten11/article/details/52791055>，2016.10.11

[4]zhaizu. 安卓中计算 TextView 的宽高、打点[J/OL]. https://blog.csdn.net/zhaizu/article/details/51723244，2016.06.20

[5]drinkingcode. Android中关于字体宽高的获取[J/OL]. <https://blog.csdn.net/u010661782/article/details/52805939>, 2016.10.13

[6]bendan50. 关于Android中字体宽度的计算（附高度）[J/OL]. <https://blog.csdn.net/bendan50/article/details/86082995>, 2019.01.08

[7]qq\_26626431. Canvas DrawText详解[J/OL]. <https://blog.csdn.net/qq_26626431/article/details/80963296>， 2018.07.08

[8]ITzhongzi. Android编程之绘图canvas基本用法示例[J/OL]. <https://www.jb51.net/article/123132.htm>, 2017.09.07

[9] 哆啦miss梦. Android canvas绘图基础之运动的时钟[J/OL]. <https://blog.csdn.net/qq_30806949/article/details/51054663>, 2016.04.04

[10]打火机维修员. 画布的移动和旋转,旋转效果图如下[J/OL]. <https://blog.csdn.net/u014644594/article/details/81187047>， 2018.07.24

[11]chuanli5157. Android canvas clipPath 问题[J/OL]. <https://blog.csdn.net/chuanli5157/article/details/53955468>, 2016.12.13

[12]陈守印. Andrdoid自定义View之canvas.clipPath(path);[J/OL]. <https://www.jianshu.com/p/57ed51627948>， 2017.08.09

[13]netwp. 如何使用List<父类>引用List<子类>[J/OL]. <https://bbs.csdn.net/topics/391814413?list=lz>, 2015.08.21

[14]chenxibobo. Android中为什么主线程不会因为Looper.loop()方法造成阻塞[J/OL]. <https://www.cnblogs.com/chenxibobo/p/9640472.html>, 2018.09.13