**移动平台应用开发**

**实验报告**

**The Share**

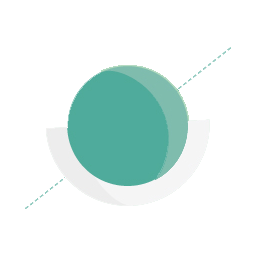
——基于环信3.0 SDK的即时通讯以及画板交互的APP

小组成员：

1511413 陈齐翔 软件学院

1511419 公岩松 软件学院

1. **内容摘要**



(图)The Share

名称：The Share

类型：即时通讯工具

当前版本：v1.0.0

大小：XX MB

更新时间：2017年5月29日星期一

兼容性：需要Android 4.1.0或更高版本，与平板兼容

产品简介：The Share是一款IM即时通讯APP，轻松实现您与好友间的交流，拉近彼此间距离；除此之外，我们还加强了传统IM软件的通讯功能，使其兼具娱乐和教育性，通过它，您能更加高效、完整的去表达您的想法或创意，让距离不再是创意的障碍。

1. **功能介绍**

1．即时通讯，特色功能之一。极简精致的界面，小众的聊天风格，使用户可以不在局限于传统、大众的通讯软件，让手机更加与众不同。

2．画板共享，特色功能之一。在开启语音通话的基础上，用户可以彼此间共享同一块画板，任何操作都将同时显示到两人屏幕上。搭配语音功能可以让用户的想法更加高效、完整的表达出来。

3，在图片上绘制。选择一张图片后便可在画板上进行共享，可以在其上做注释、修改等任何操作。

在了解了这些基础功能之后，我们不难发现这款APP用途之广。

在外工作无法辅导孩子学习? 使用The Share，与孩子轻松建立连接，共享一道题目或一篇美文。

图文并茂都无法清晰表达你的创意？使用The Share，图、文、音同时上阵，不怕你的创意不被理解！

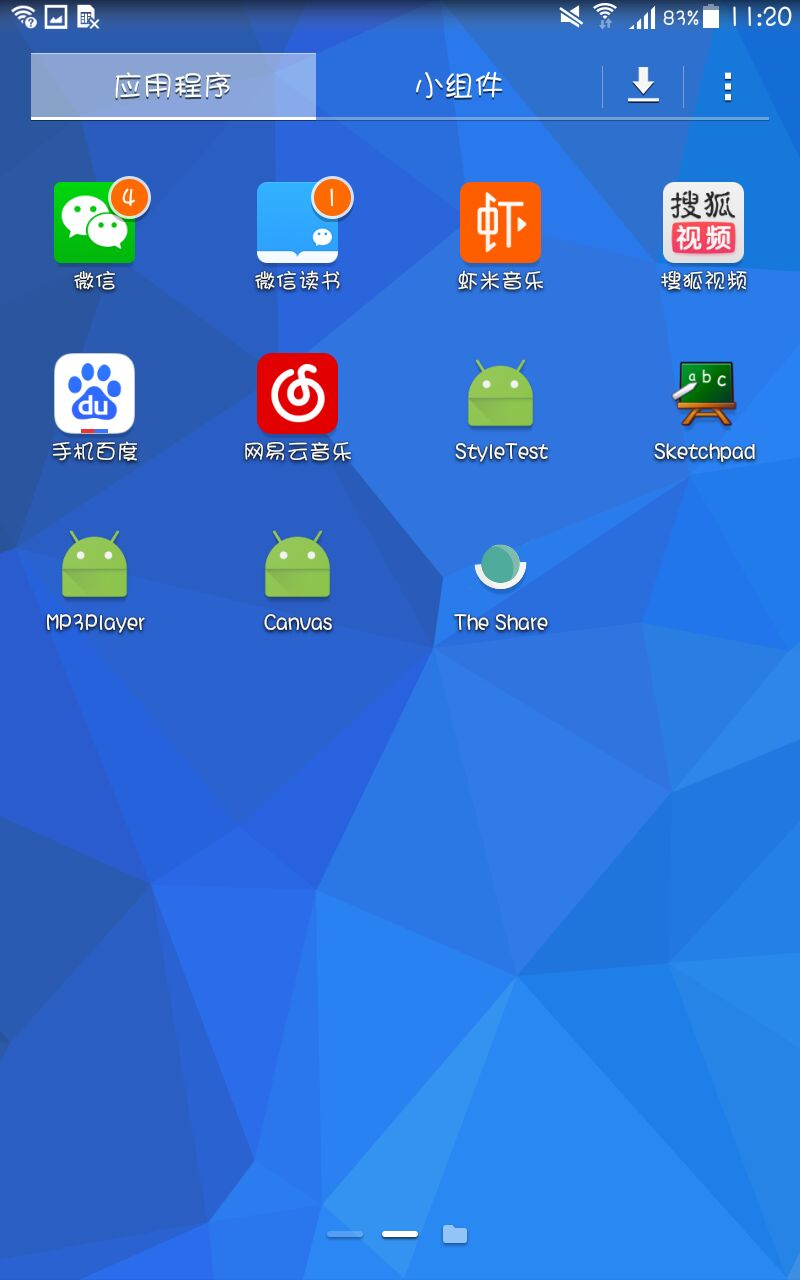
好友们远在天边无法像聚会一样游戏？使用The Share，在画板上任意涂鸦来“你画我猜”（注意别不小心说出你画的是什么）！

您还可以用它实现远程产品展示、作品设计等功能。请尽情发挥您的想象力和创造力，The Share远不止于此……

1. **使用说明**

**1.有关界面**

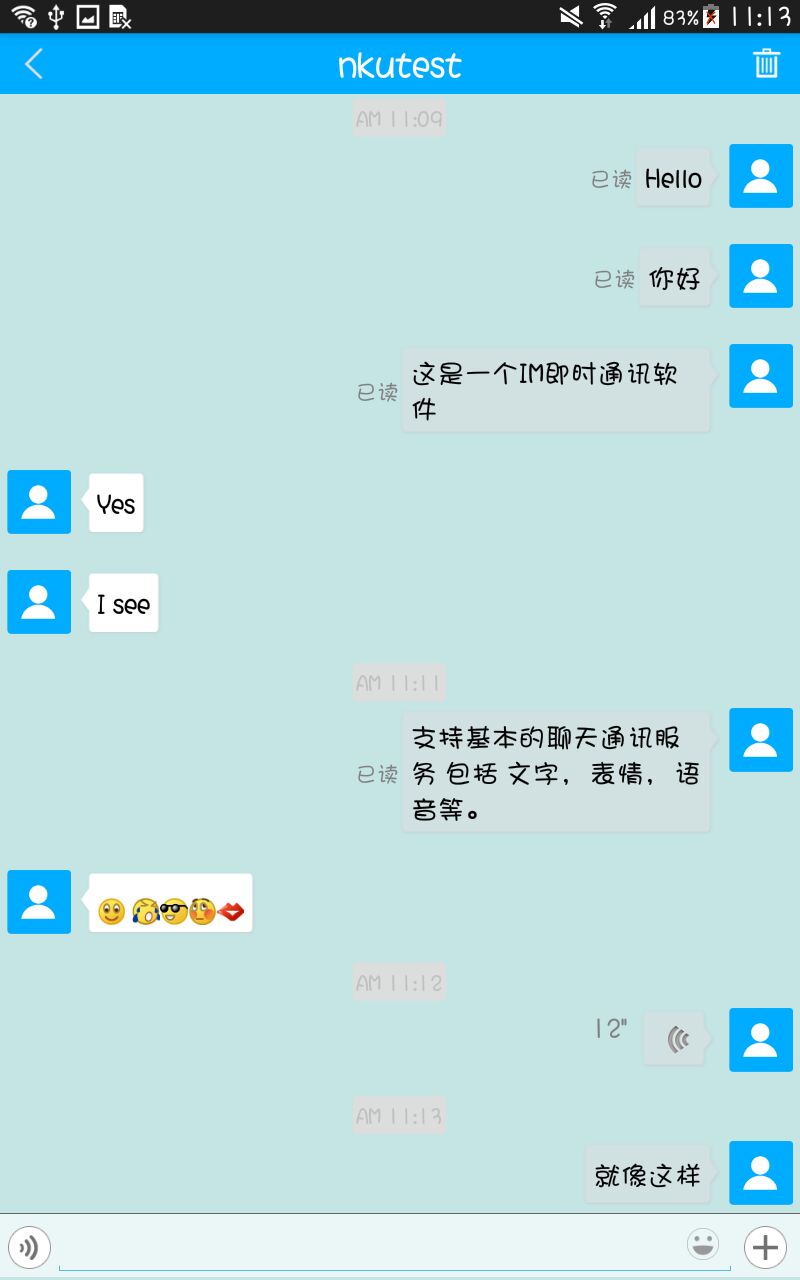
APP图标以及界面样式如下列图片：





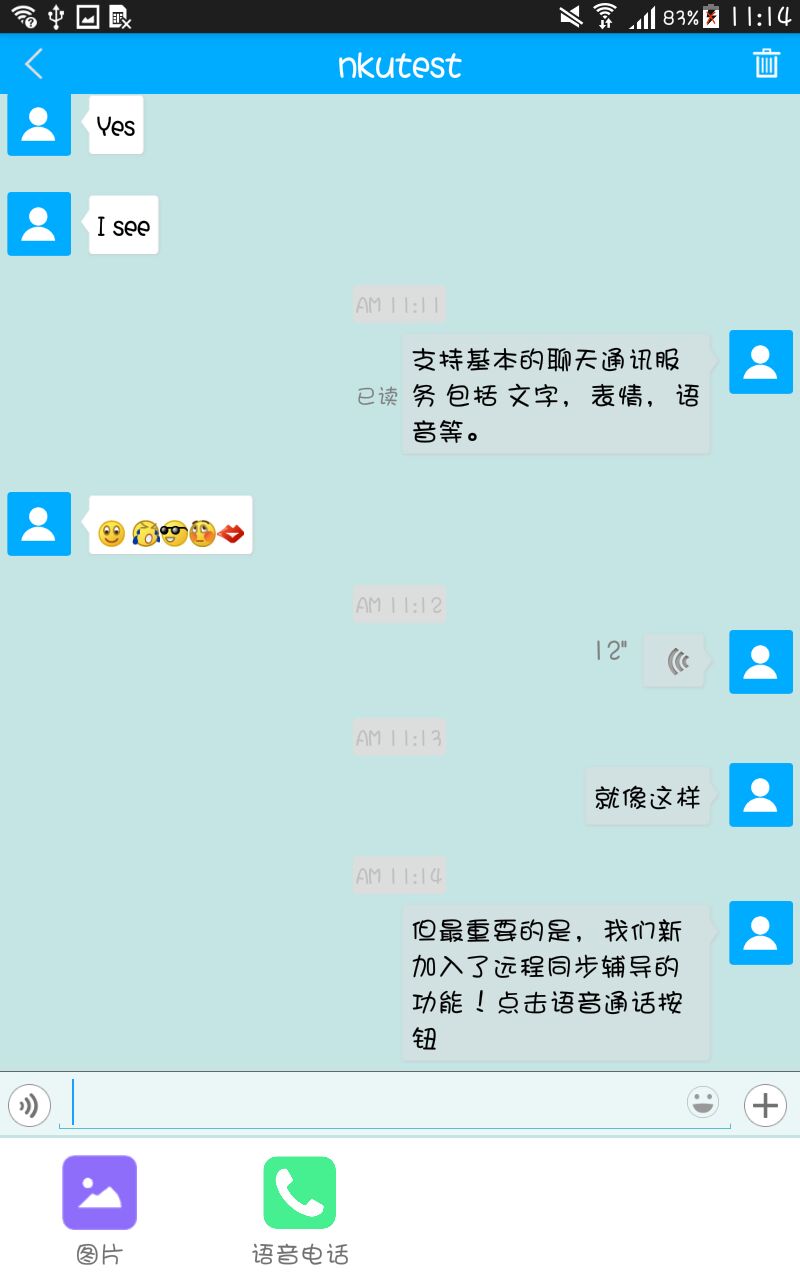
**2.聊天功能**

我们在语音通话之前有类似于QQ、微信等APP的聊天功能，支持发表情，文字，语音。下面是聊天的功能以及界面：



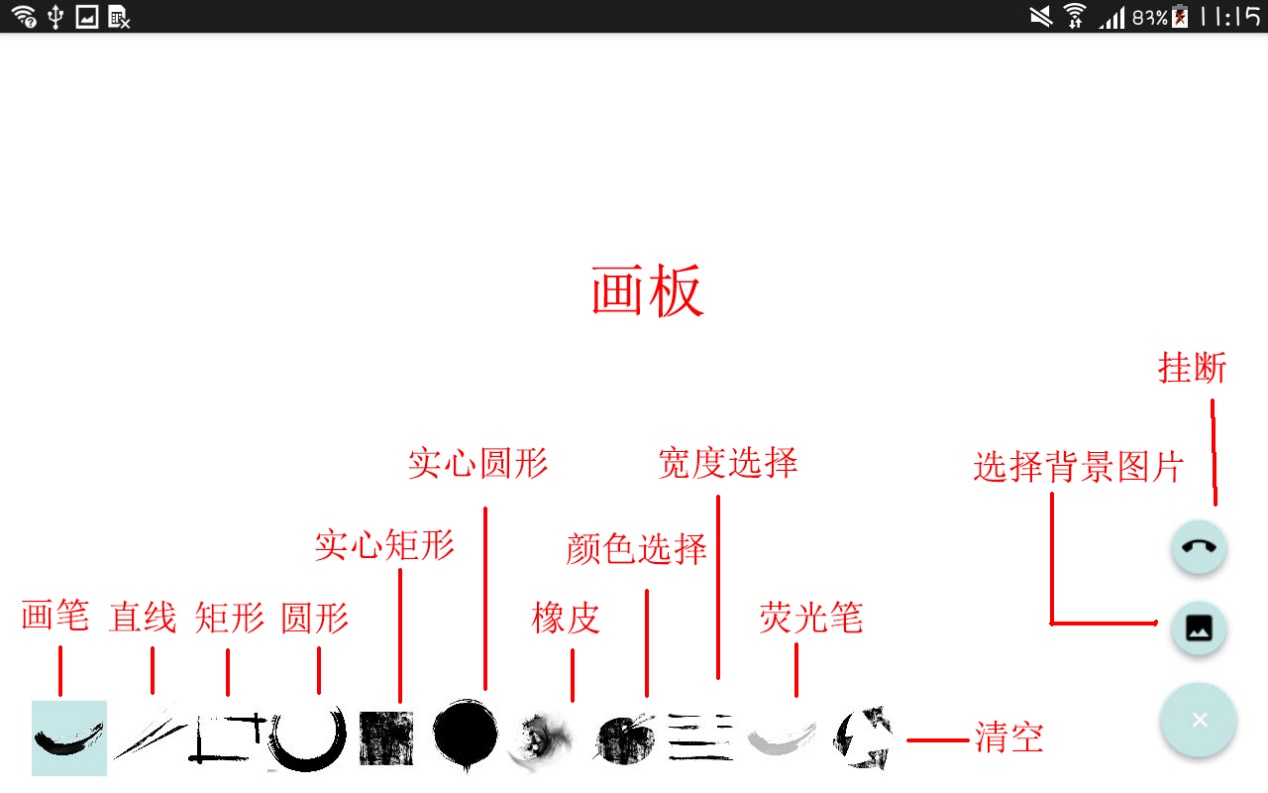
**2.语音通话**

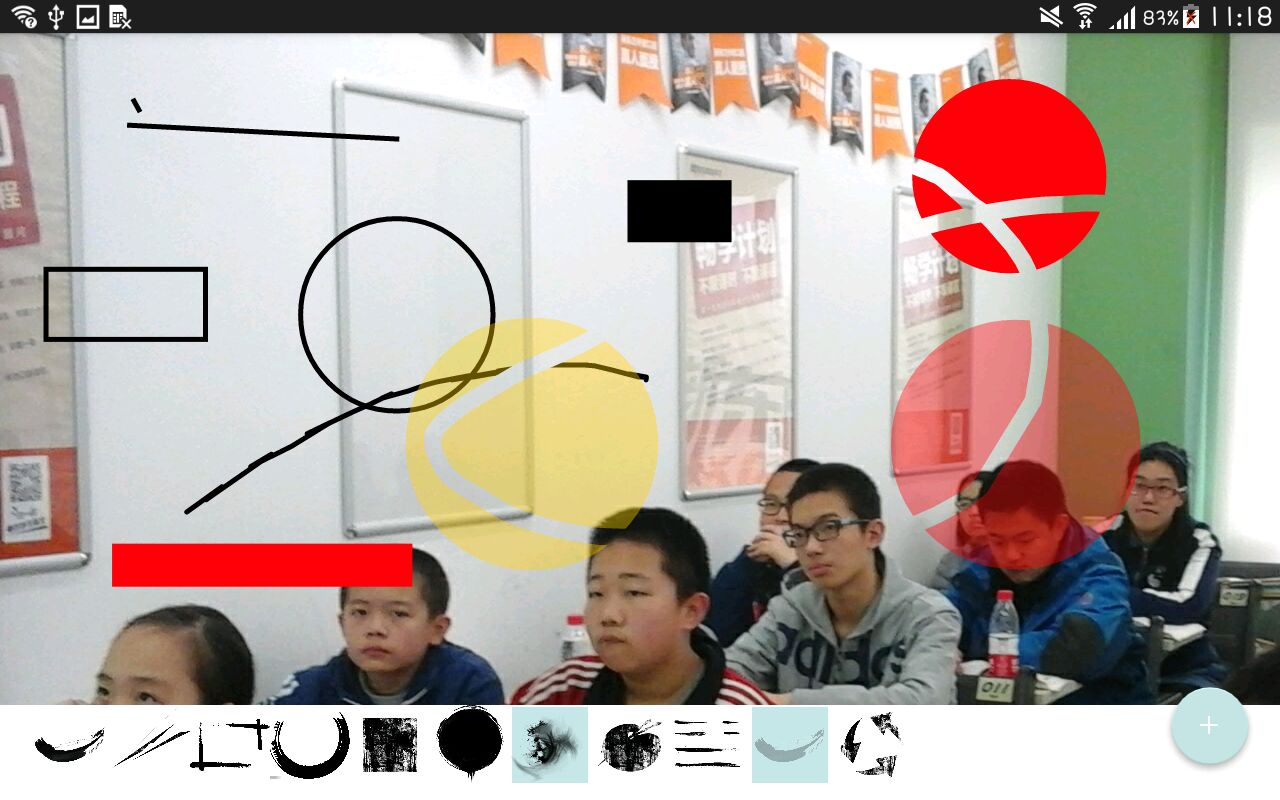
语音通话使用了环信的服务器，通话的即时性可以得到相对保障。通话的过程与平时打电话无异，在这里就不赘述了，下列图片是如何开启通话。（开启语音通话同时会开启画板）

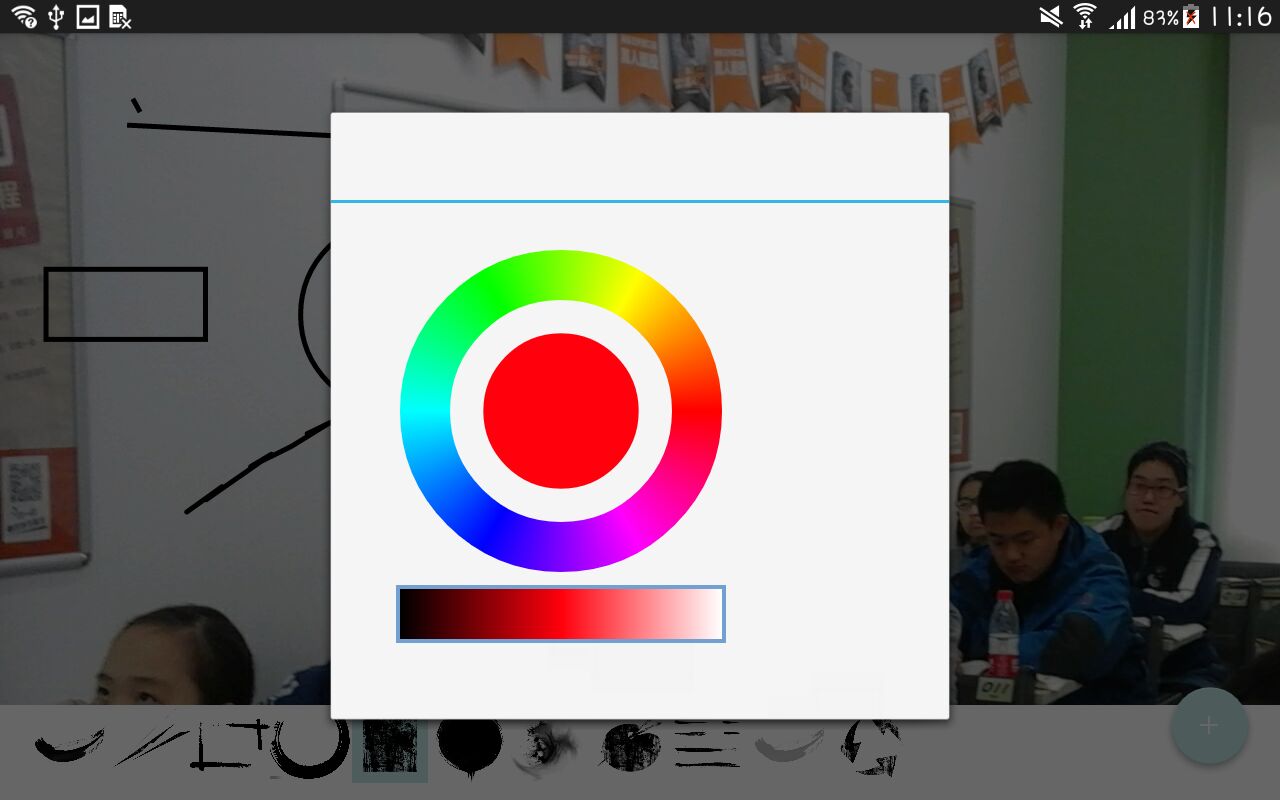


**3.有关画板**

画板功能丰富，可以使用画笔，直线，矩形，圆，实心矩形，实心圆，橡皮，荧光笔模式，颜色选择框，宽度选择框，选择背景图片，清空等。具体效果展示如下图：



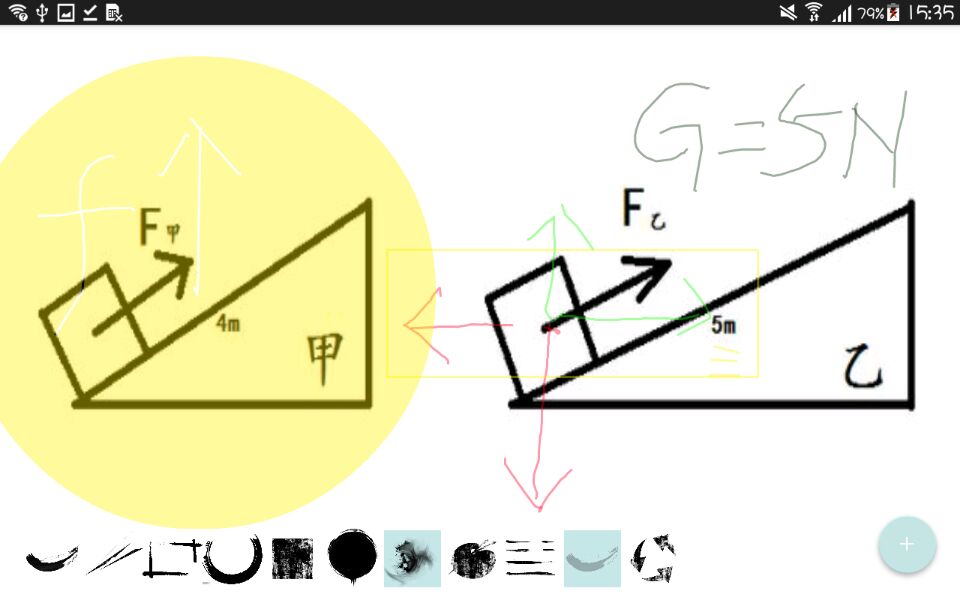


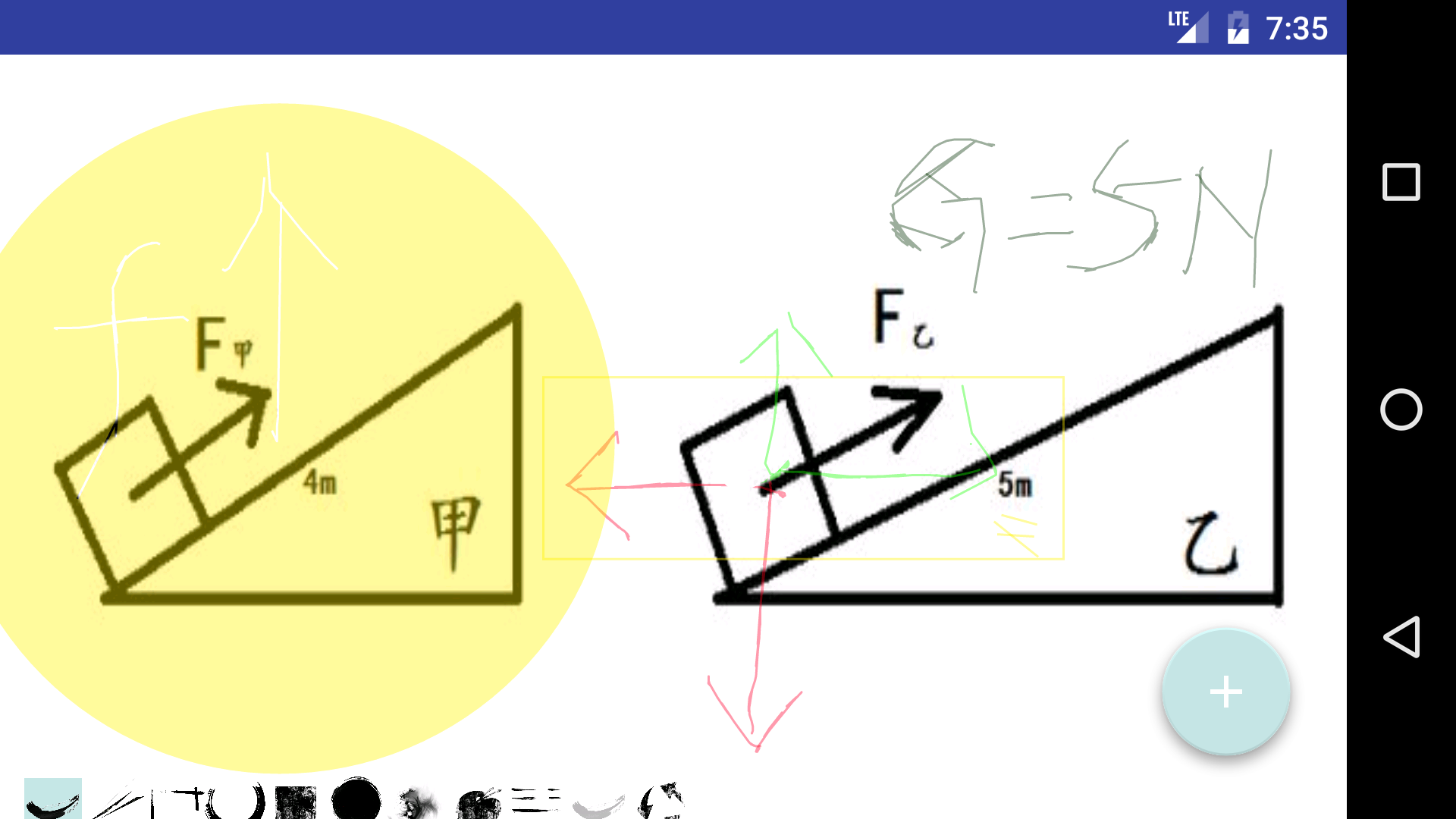


值得注意的是，橡皮擦并不会将背景图片抹去，只会抹去图片上面的绘画内容。

**4.画板的实时交互**

我们使用了一个安卓设备与一个模拟器进行画板的实时交互，效果如下图：



  
我们在选择背景图片时，会将此图片发送到对方的设备中，对方的设备也会显示该背景图片以保证双方得到的信息均保持大致相同。

1. **实现思路**

**1.有关通讯**

环信即时通讯云V3.0是一个Paas平台，其SDK封装了基本的通讯协议、消息收发等功能，方便开发者快速搭建IM应用，并且在日活量低于3万时为免费的，因此本次课题选择了环信的SDK为基础进行开发。

主要使用的功能有：

**1），用户注册、登录**

因为环信服务器提供了相应接口，因此仅需要将用户密码提交至服务器后台即可实现，通过回调函数判断登录情况。

**2），发起聊天、聊天界面**

联系人列表是一个RecycleList，可显示用户的好友（目前只提供少数测试好友），点击即可进入聊天界面。

聊天界面分为基础功能和扩展功能。基础功能使用的是官方提供的Demo，包括文本、语音、表情消息的显示等；扩展功能为本应用的核心功能：“远程辅导”。

**基础功能--显示、收发消息：**

消息列表为SDK自定义的EMMessageList控件，继承自RecycleList并实现了消息的显示方式和事件响应方式。

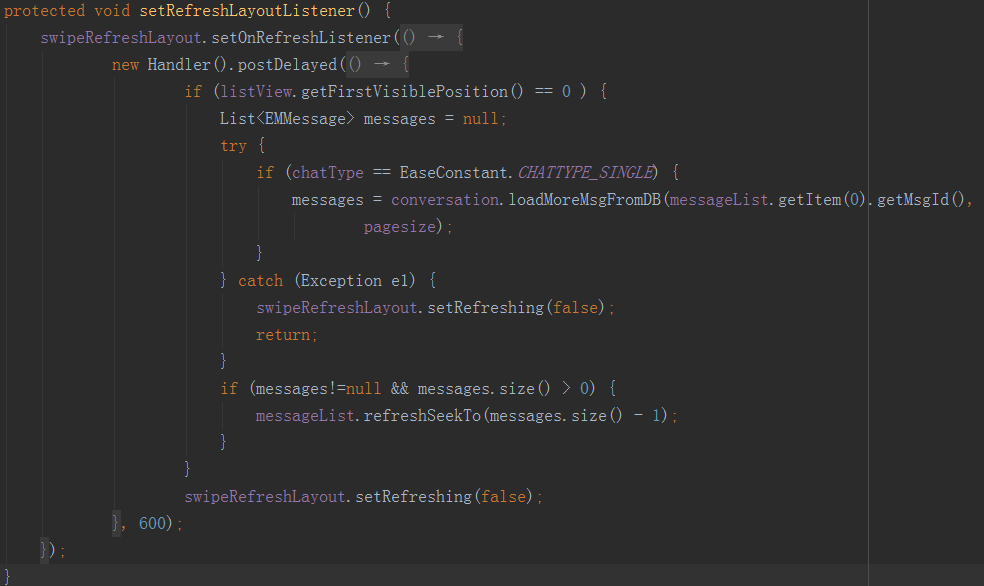
收发消息需注册监听器并实现对应接口。当监听器收到消息时，匹配其发送者和指定接收者，匹配成功时调用SDK方法提取消息内容最后显示到列表控件中。





**基础功能--加载历史对话：**

在聊天页面中下拉消息列表即可实现该功能，注册了对应手势的监听器后仅需向服务器提交申请便可获得更多历史消息，并显示到列表中。

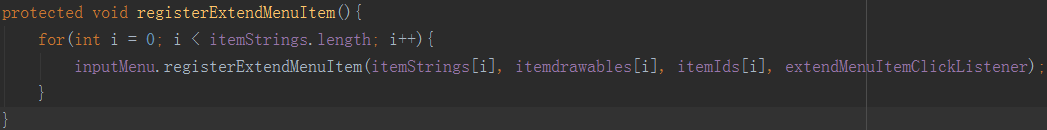


**基础功能--顶部工具栏：**

顶部的TitleBar主要用来显示当前会话对象及提供回退按钮，简单设置TextView及注册回退按钮的监听器便可实现。

**基础功能--扩展工具栏：**

扩展工具栏为点击右下角加号后显示的工具栏，目前包括选择图片及进入远程辅导。设置Item的名字、图标（用选择器实现普通和按下两种状态）、资源ID和消息响应函数，最后调用SDK提供的方法注册后便可实现该效果。

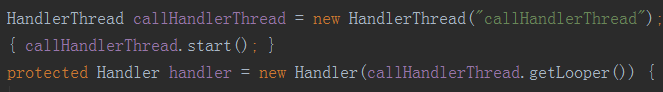


**扩展功能--远程交流：**

远程交流的核心可看作：语音通话+消息收发+画板。

**a)语音通话：**

语音通话核心部分是对后台消息的分发和处理，如发起通话、接受通话、挂断通话等等，可通过一个HandlerThread实现：



再在消息循环体中加入对各类消息的处理方法。

IMG_256 ——建立通话

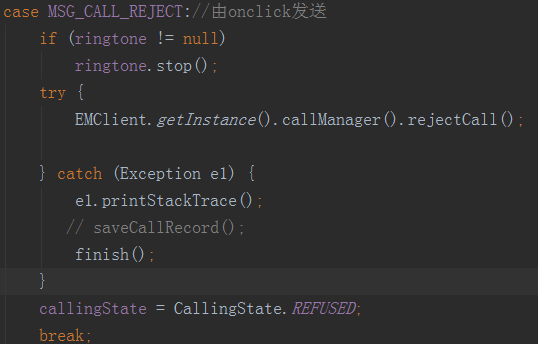
IMG_256 ——接受通话

IMG_256 ——拒绝通话

IMG_256 ——通话结束

IMG_256 ——释放资源

对于不同的消息需要调用SDK提供的对应方法，如：

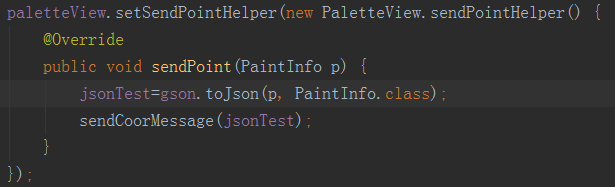


后台初步处理了消息并反馈到Activity后，需切换到远程交流界面（VoiceCallActivity）并根据Intent进行初始化，包括建立请求对话框、初始化画板、注册通话状态监听器等。初始化完成后，语音通讯服务便搭建成功，用户便可以进行基本的语音通话。

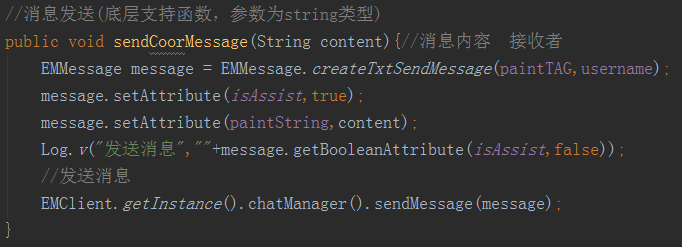
**消息收发（绘画坐标消息和图片消息）：**

这里的消息不同于聊天界面，是绘画过程中产生的绘画消息，包括像素点位置、颜色、透明度等。发送方将位置信息对象（PaintInfo）包装成JSON格式的数据对象后进行发送，接收方解析JSON得到具体信息后便可同步的画到自己的画板上。代码实现：

*包装（实现画板的发送接口）：*

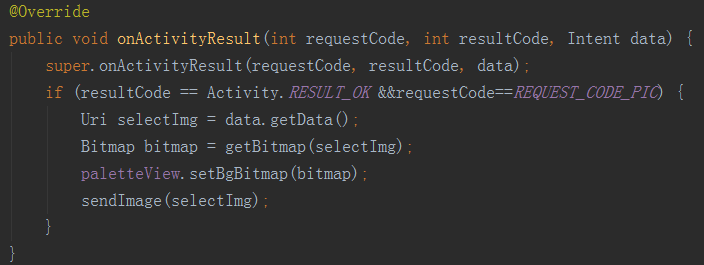


*发送（坐标消息）：*



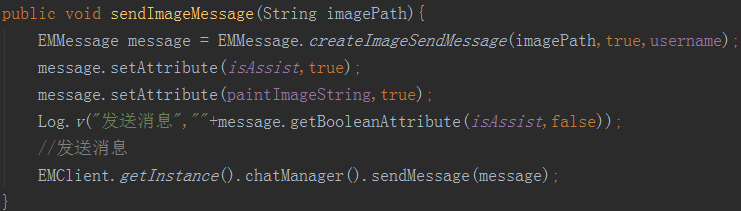
*发送（图片消息）：*

需要先调用Android系统的Action进入相册选择图片，获取到图片Uri后再得到Bitmap。

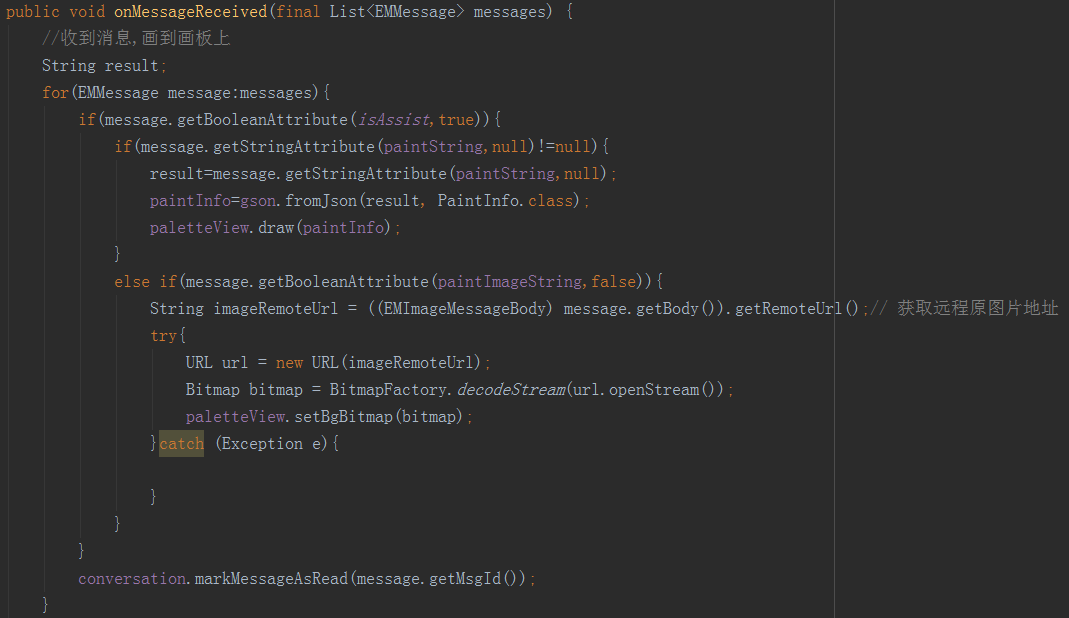


（跳过搜索图片部分）

得到图片后，获取绝对路径，由环信后台转换为网络资源路径发送：



*接收（坐标消息和图片消息）：*



坐标消息由JSON转为对应格式后便可交给画板(paletteView)完成绘画

图片消息为一条URL，转换为Bitmap后便可交给画板完成共享

**2.有关画板**

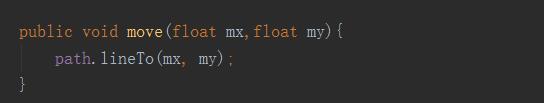
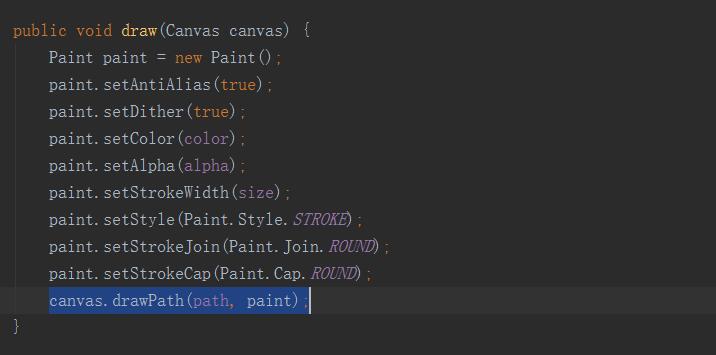
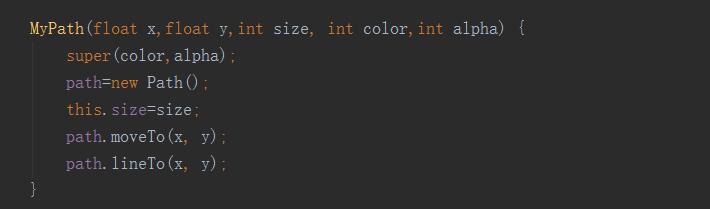
1），画板

由于android中Cavas,Paint类已经相当成熟，制作一个画板非常的容易。我在这里简要地梳理一下思路。首先实现画板我用到了两个类，一个是PaletteView，另一个是Action。这两个类都是我们从一个开源项目中找到的，前者我们几乎全部修改了一遍。后者我们增加了一些必要的属性。

对于Action类，它是一个父类。圆类，矩形类等有关图形的类均是他的子类，这些类的对象保存了一个图形的位置，大小，颜色，透明度等信息。每一个子类都需要重载两个函数，一个是draw()这个函数是用来把对这个对象画在画布上的。另外一个是move()这个函数是用来，当按住屏幕时，需要把沿途手指经过的位置都暂时记录一下，用于即时绘画。

实现画板上的绘画只需要设置好成员变量Canvas和Paint这两个类的对象就能完成大部分的工作，由于太简单这里不加详述。

下面的截图是Action其中一个子类的实现方法，其余类似：



对于PaletteView，有一个成员变量，用ArrayList储存Action类对象，每次绘画完成后会增加一个Action类对象。每一次重绘时会遍历数组，画到画布中。当然每一次重绘也会将当前正在绘制的图形画到画布上。这些均调用Action类对象的draw()函数即可。

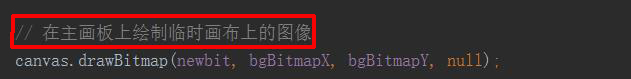
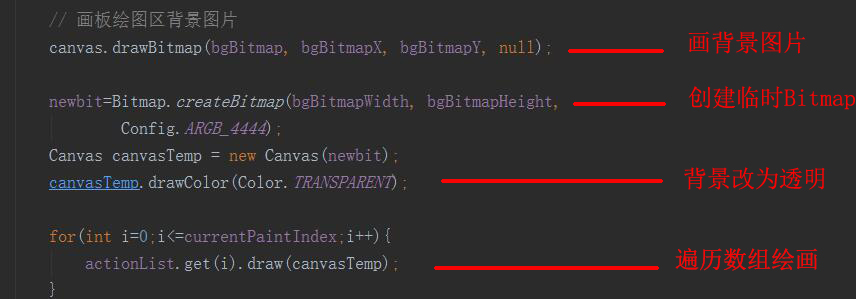
另外为了适应各种屏幕分辨率，我们画布的大小均是根据设备的屏幕进行比例微调的。

2），背景图片与画布分离

在实现画板时有一个重大难关，便是如果加载图片到画布之中，那么当擦除时也会将背景图片一并擦除。在面对这个问题，我们采用了PhotoShop中图层的思想，将PaletteView中设置两个Canvas的成员变量（canvas,canvasTemp），一个用于绘制上层用于用户绘画(canvasTemp)，一个用于绘制底层背景图片和工具栏的(canvas)。

当每次进行重绘时，canvas会先把背景图片画到屏幕上，之后canvasTemp会将前文中所提到的ArrayList<Action>中的对象全都绘制一遍。保存在一个背景为透明的Bitmap中，最后canvas会把刚才生成的Bitmap画到屏幕上。每一次重绘都执行这个过程，那么每次擦除时，实际上只是调用了画笔，把画笔的颜色调为透明，覆盖到原来的画中，这样就使得后面的背景显示出来而且没有被擦除掉。如果不使用临时canvas就会使背景也变成透明。

下面的代码截图就是上述步骤的执行过程：



3），绘制工具栏

工具栏并没有使用Button等成熟的类，都是直接绘制在画布之中的。判断点击与否，只需要判断手指触摸画板的坐标是否在相应的按钮位置之内。每一个按钮的长宽都相同。当按下时会改变背景色。值得注意的是当按下荧光笔模式时，可能与其他功能一同使用，所以专门设置了私有变量看是否处于荧光笔状态，如果在的话，就一直处于按下状态。当显示对话框时，会产生对话框不会立即显示的BUG，在经过添加按钮开关私有变量后判断是否处于打开状态后解决。每个按钮上绘制的图标都在drawable文件夹中。

4）绘制事件的同步

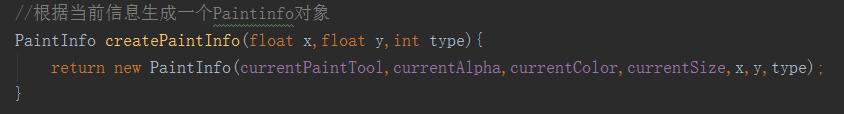
我们专门设置了一个类PaintInfo用于储存当前绘制的信息，并发送到对方的设备中，对方收到了这个类，就会根据类中携带的位置大小种类状态颜色透明度等信息在自己的画布上绘制相同的信息。

信息何时会发送？每当绘画时的点击，移动，松开屏幕时都会发送不同种类的信息，对方也会模拟进行电椅，移动，松开的动作。当然如果点击工具栏中的清空，双方也会同时清空。

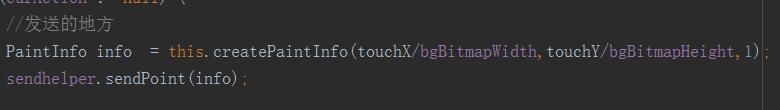
当前活动如何获得发送的消息？我们设置了发送消息时的监听器，每当发送消息时，都会通过监听器获得消息的对象，在经过环信的服务器进行网络传输。

下面给出发送时调用的函数截图：

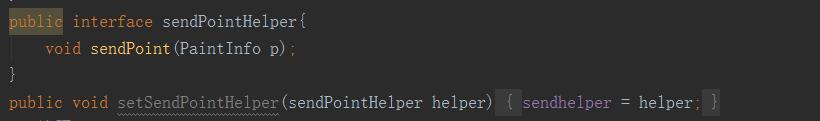
通过当前的信息创建一个信息对象:



发送时调用sendPoint()函数：



主函数重写这个接口，实现监听，并获取对象：



5），对话框生成

我们用到了两种对话框，一种是颜色对话框。由于android没有带自带的颜色选择框，所以我们选择了一个开源项目的颜色对话框，它的大体思路时计算手指点击对话框的坐标生成颜色，通过画布重写监听器获取选中颜色。

另外一个是宽度选择框，实现非常简单，值得一提的也是效仿颜色选择框设置了相应的事件监听改变画布中画笔的宽度。

1. **目前发现的BUG及不足之处**

1. 账号登陆问题，某一设备登陆过一个账号后在登陆新的账号时会显示该账号已登陆，从而无法更换账号，原因不明。

2. 远程辅导时发送的坐标、图片消息等再使用过后无法删除，导致再次进入聊天页面时会将其作为聊天消息的一部分全部加载到消息列表中。

3. 目前不提供添加好友的功能，联系人列表仅是几个测试号。

4. 绘画时由于会发大量消息，导致绘画的显示略慢于手移动的速度。有一定时差，目前可以通过发消息频率降低减少这一问题，但是会导致对方收到的消息曲线平滑度降低，目前依然在寻找更好的平衡点。

*注：目前提供的测试号(可随时注册新账号，语音通话时需要双方均在线)*

*用户名 密码*

*cqxtest 123*

*nkutest 123*

*test\_uesr3 123*

*test\_uesr4 123*