编写聊天界面

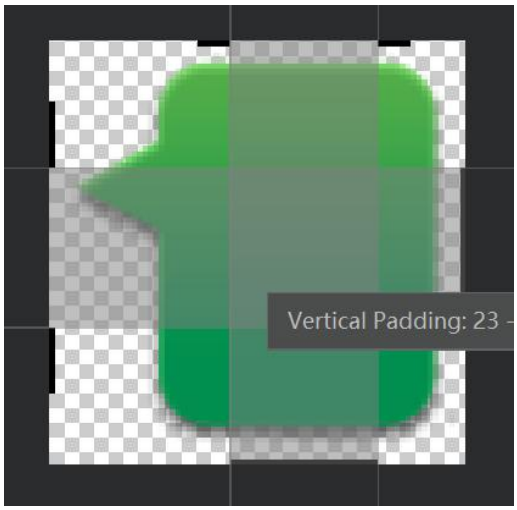
1、创建新项目

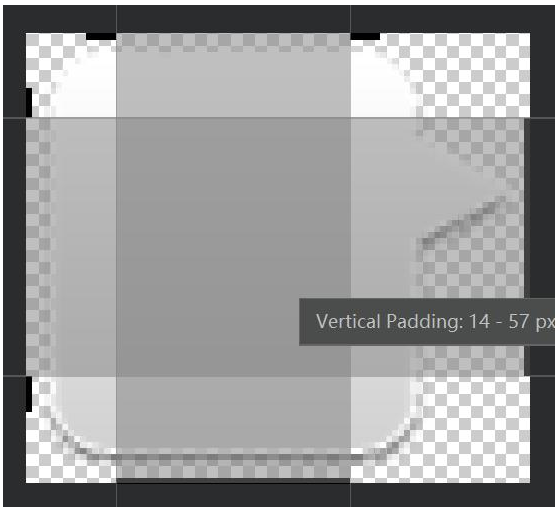
本实验为仿微信聊天界面实现。

2、资源

（1）实验过程中用到的图片，请右键另存下来。


（2）请依据上次课的案例，将这两张图片制作成 9-patch 图片





3、添加依赖

```
activity_main.xml × MainActivity.kt × build.gradle (:app) ×
You can use the Project Structure dialog to view and edit your project configuration
                minifyEnabled false
                proguardFiles getDefaultProguardFile('proguard-android-optimize.
            }
        }
    }

    dependencies {
        implementation fileTree(dir: "libs", include: ["*.jar"])
        implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
        implementation 'androidx.core:core-ktx:1.3.2'
        implementation 'androidx.appcompat:appcompat:1.2.0'
        implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
        implementation 'androidx.recyclerview:recyclerview:1.1.0'
        testImplementation 'junit:junit:4.12'
        androidTestImplementation 'androidx.test.ext:junit:1.1.2'
        androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'

    }
```

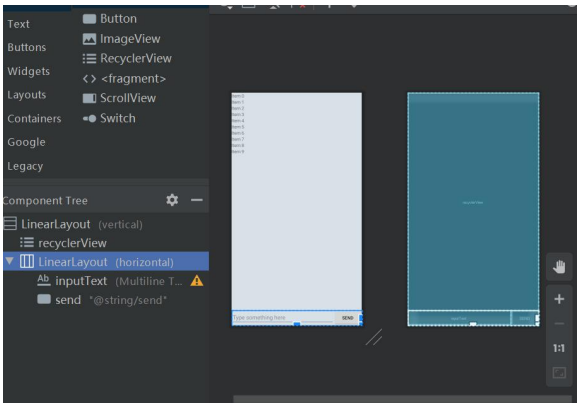
4、编写主界面

修改 activity_main.xml 中的代码

```
MsgAdapter.kt × msg_right_item.xml × activity_main.xml ×

android:background="#d8e0e8" >

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/inputText"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:hint="@string/type"
        android:maxLines="2" />
    <Button
        android:id="@+id/send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/send" />
</LinearLayout>
```



主界面共有三个控件：

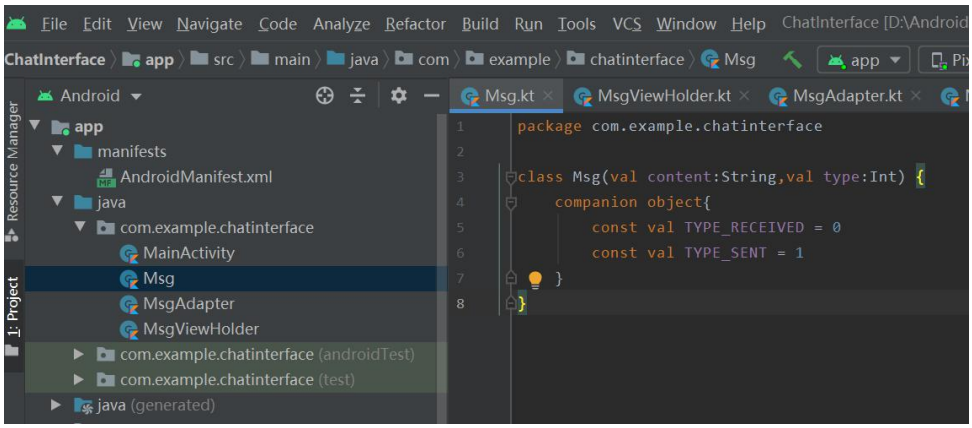
RecyclerView 位于界面上方，占据整个界面（**android:layout_weight="1"**）

下方是一个线性布局容器，因为没有设置 **android:layout_weight**，所以只能依赖于自身内部控件的大小来占据剩余的区域

线性布局容器内部左侧是一个文本输入框，右侧是发送按钮。

5、定义消息实体类

使用面向对象的角度解决问题，既然是发送消息，那么必须得存在消息的数据载体，在面向对象的世界里面，这个载体就意味着存在一个类（实体类）。



属性 content：消息的内容，

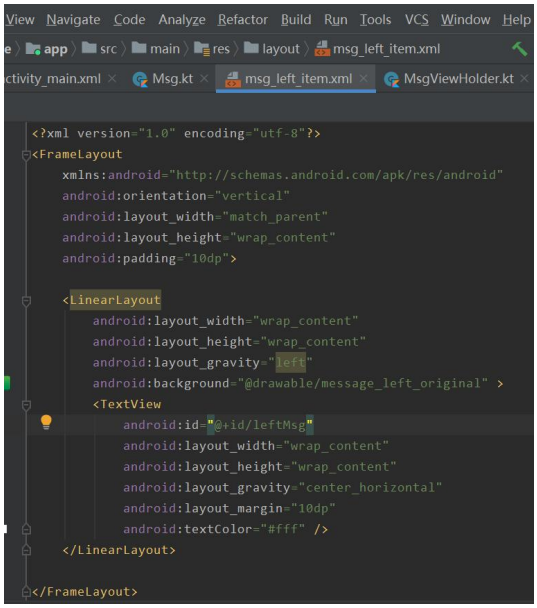
属性 type：消息的类型，

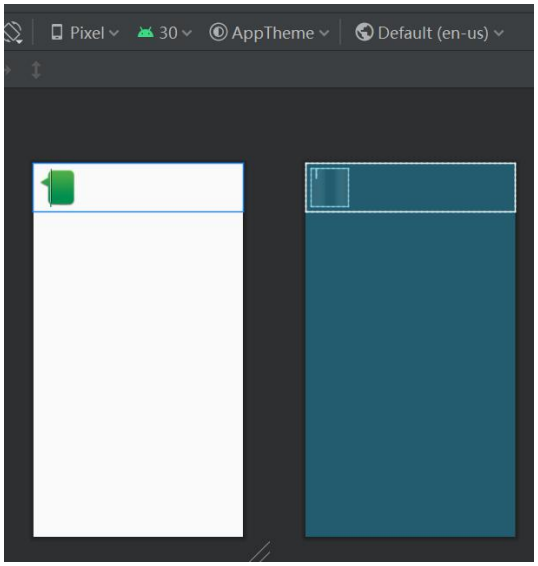
常量 TYPE_RECEIVED 与 TYPE_SENT 分别对应于属性 type 的两个值，分别代表该消息是发出的消息还是接受的消息。

注意这个地方，用了一个伴生类，主要作用是为了实现全局属性，const 只有在单例、伴生类、顶层方法中才能使用。

6、编写 RecyclerView 的子项布局

（1）新建 **msg_left_item.xml**





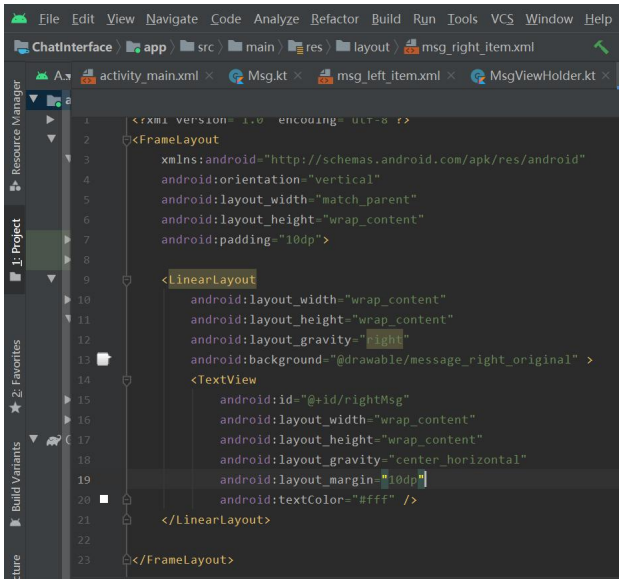
这是接收消息的子项布局，因此可以看到 LinearLayout 的定位是靠左，用了接收消息的图片作为背景。

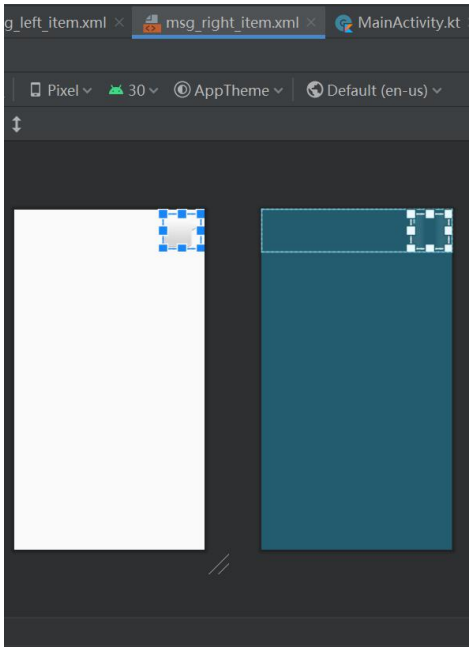
知识点：这里用了一个帧布局（**FrameLayout**），FrameLayout（帧布局）可以说是六大布局中最为简单的一个布局，帧布局的特性如下：

- 这个布局直接在屏幕上开辟出一块空白的区域
- 当我们往里面添加控件的时候，会默认把他们放到这块区域的左上角；
- 这种布局方式却没有任何的定位方式，所以它应用的场景并不多；
- 帧布局的大小由控件中最大的子控件决定，如果控件的大小一样大的话，那么同一时刻就只能看到最上面的那个组件，后续添加的控件会覆盖前一个；
- 虽然默认会将控件放置在左上角,但是我们也可以通过 layout_gravity 属性,指定到其他的位置。

其实这里使用帧布局并没有什么特殊意义，但是这个帧布局常常会用在动画场景，因为它的最大特点就是没有布局，适合用于绘画。

（2）新建 msg_right_item.xml





7、编写代码实现功能

思考：

- 模拟数据如何构建？可以使用 List，但是记住，构建时应有发送的消息以及接受的消息。

List.add(msg)，msg 中有两个参数，一个 content 表示文本内容，另一个 Type 表示发送/接收消息的类型

- adapter 面临着两个子项布局的选择，何时采用发送消息的子项布局，何时采用接受消息的子项布局？

将左右两个布局的 ViewHolder 封装在一个同父类 ViewHolder 中，之后分别调用

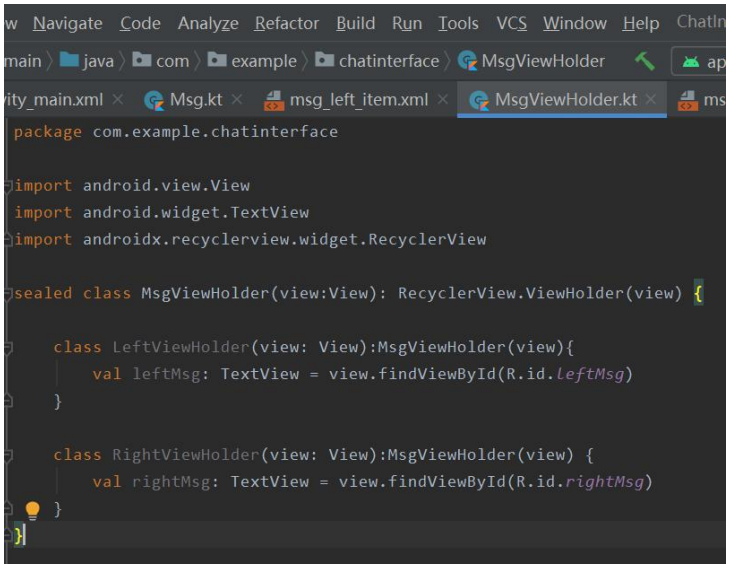
- 发送按钮不是摆设，得起作用，也就是说点击按钮后，会发生什么事情，这需要思考。

点击按钮时，将文本传进 Msg，设置消息类型为发送类型，并加入到 List 中。设置好添加位置在 item 末尾，并滚动视图显示最新消息。

（1）定义 MsgViewHolder

这是个密封类（你也可以尝试设置为 open，看看会不会影响代码的实现），密封类用来表示受限的类继承结构：当一个值为有限几种的类型，而不能有任何其他类型时。它可以不在此文件外被继承，有效保护代码。密封类可以有子类，但是所有的子类都必须要内嵌。

设置为 open，并不影响代码的实现



这里定义了一个父类 MsgViewHolder，目的是在 adapter 中将 LeftViewHolder 与 RightViewHolder 都当成同一对象看待处理，这样在渲染时就无需在代码级进行区分。

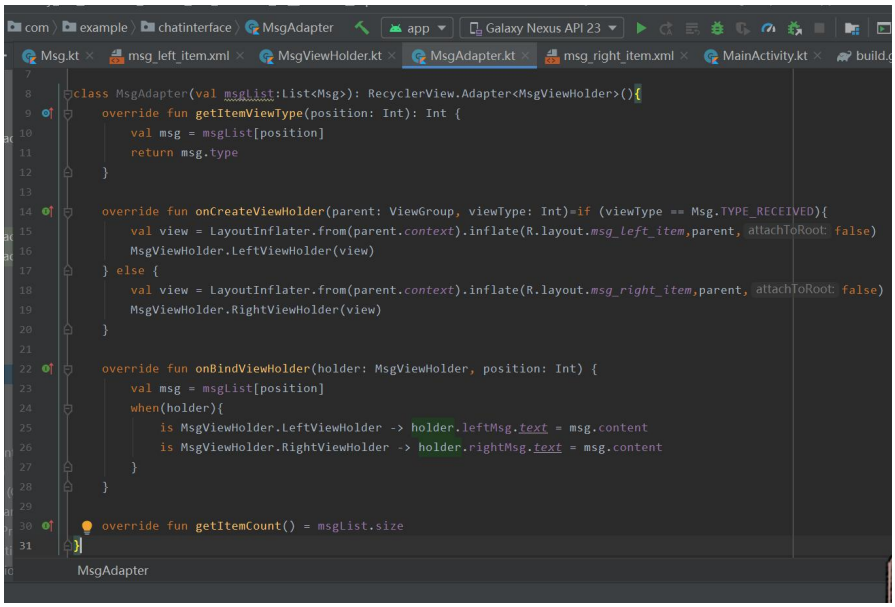
定义子类 LeftViewHolder，从 MsgViewHolder 继承，为什么这么做呢？因为这两个 viewHolder 的东西不同，在具体实现上需要针对具体的布局进行设置，但是在渲染层面又希望 android 统一处理。

（2）定义 MsgAdapter

代码与课上的类似，但有个细节，因为我们要考虑到，消息的 **type** 有两种，当它为 0 时，是发送的消息，为 1 时是接受的消息，不同类型的消息对应的子项布局是不同的，所以我们为什么使用了两种 viewHolder；

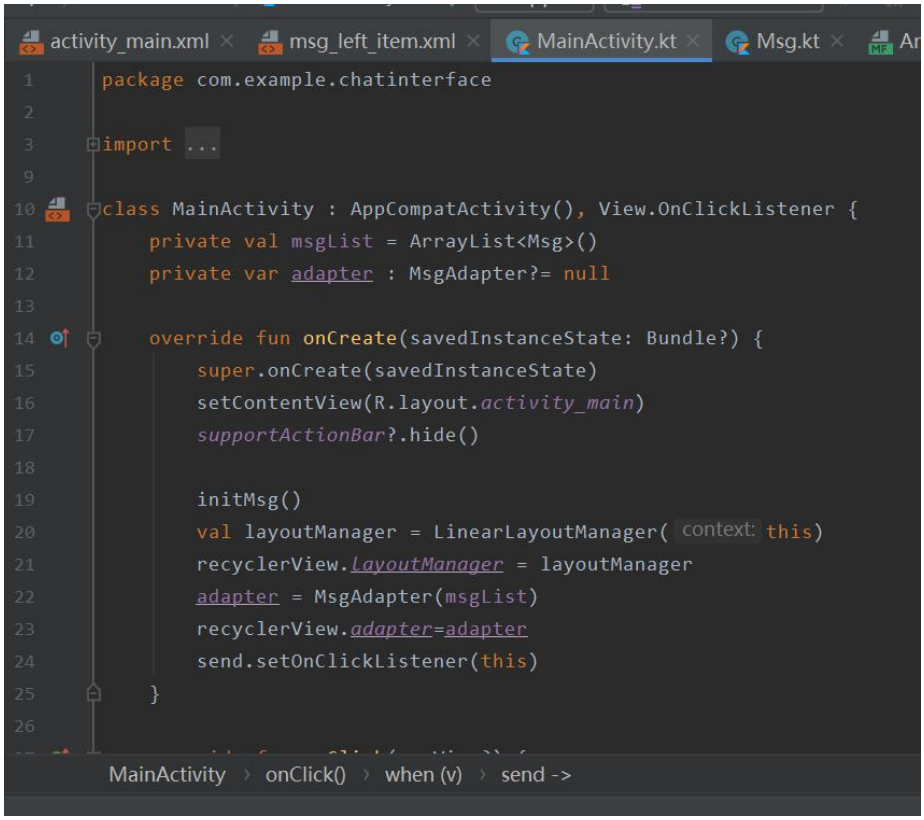
因此我们需要在 onCreateViewHolder 方法中判断 ViewHolder 的类型，如果是 LeftViewHolder 就加载 R.layout.*msg_left_item* 布局，如果是 RightViewHolder 就加载 R.layout.*msg_right_item* 布局

那么如何获取到 ViewType 呢？可以覆盖 getItemViewType 方法，根据数据进行类型的返回。



3）修改 MainActivity

为 RecyclerView 初始化一些数据，并给发送按钮加入事件响应




```kotlin
package com.example.chatinterface

import ...

class MainActivity : AppCompatActivity(), View.OnClickListener {
    private val msgList = ArrayList<Msg>()
    private var adapter : MsgAdapter?= null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        supportActionBar?.hide()

        initMsg()
        val layoutManager = LinearLayoutManager( context: this)
        recyclerView.layoutManager = layoutManager
        adapter = MsgAdapter(msgList)
        recyclerView.adapter=adapter
        send.setOnClickListener(this)
    }
```

```kotlin
    override fun onClick(v: View?) {
        when(v) {
            send -> {
                val content = inputText.text.toString()
                if(content.isNotEmpty()){
                    val msg = Msg(content,Msg.TYPE_SENT)
                    msgList.add(msg)
                    adapter?.notifyItemInserted( position: msgList.size-1)
                    recyclerView.scrollToPosition( position: msgList.size-1)
                    inputText.setText("")

                    val msg2=Msg( content: "TD",Msg.TYPE_RECEIVED)
                    msgList.add(msg2)
                }
            }
        }
    }
```

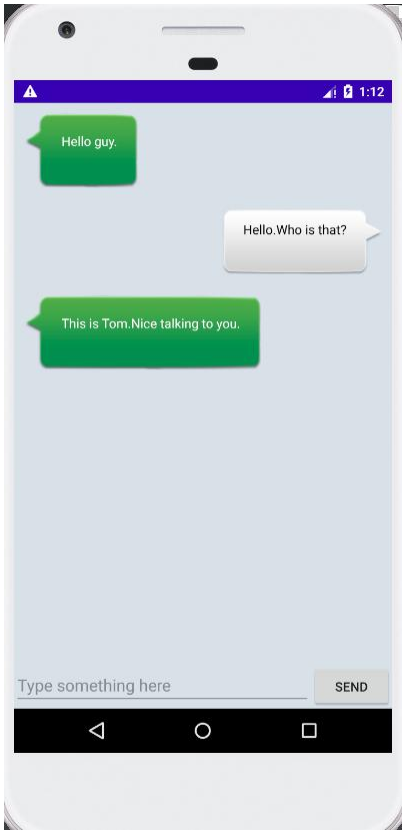
```kotlin
    private fun initMsg(){
        val msg1 = Msg( content: "Hello guy.",Msg.TYPE_RECEIVED)
        msgList.add(msg1)
        val msg2=Msg( content: "Hello.Who is that?",Msg.TYPE_SENT)
        msgList.add(msg2)
        val msg3 = Msg( content: "This is Tom.Nice talking to you.", Msg.TYPE_RECEIVED)
        msgList.add(msg3)
    }
}
```

这里主要注意一下发送按钮点击后的事件，这里首先是根据消息内容创建了一个消息（Msg）对象；其次将此对象放到了数据（msgList）的最后；接着通知（notifyItemInserted）了 adapter 数据发生了变化，这里的参数是一个整型，代表通知

刷新的子项位置是第几个，类似的还有 notifyItemRemoved，删除了子项，也需要刷新；最后将将RecyclerView 定位到最后一行。

8、运行调试

1. **软件启动后的截图：**



2. **发送及发送超过三句话的截图：**

9、遇到问题：

只设置 maxLines=”2”并不能有效控制最多出入两行。

需要加入 inputType=”textMultiLine”属性配合 maxLines 使用。

```
<EditText
    android:id="@+id/inputText"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:hint="@string/type"
    android:inputType="textMultiLine"
    android:maxLines="2" />
```

效果：

无法发送超过 2 行的消息


2:40
Hello guy.
Hello.Who is that?
This is Tom.Nice talking to you.
2 yuan per message.
Reply to TD unsubscribe.
Wishing you a happy life :)
Type something here
SEND