



#R4E

Software Developer

Android

Primeiro Projeto - Layouts



Conteúdo

- Elaboração da Primeira App
- Jetpack Compose
- Layouts

Layouts

- Os elementos da IU são hierárquicos, com elementos contidos em outros. No Compose, você cria uma hierarquia de IU chamando funções combináveis usando outras funções desse tipo.



Adicione vários textos

- Até agora, você criou sua primeira função de composição e uma prévia. Para conhecer mais recursos do Jetpack Compose, vamos criar uma tela de mensagens simples com uma lista de mensagens que pode ser aberta com algumas animações.
- Comece complementando o elemento de composição da mensagem, mostrando o nome do autor e o conteúdo dela. Primeiro, você precisa mudar o parâmetro de composição para aceitar um objeto `Message` em vez de `String` e adicionar outra função de composição `Text` à `MessageCard`. Não se esqueça de também atualizar a prévia.

Adicione vários textos

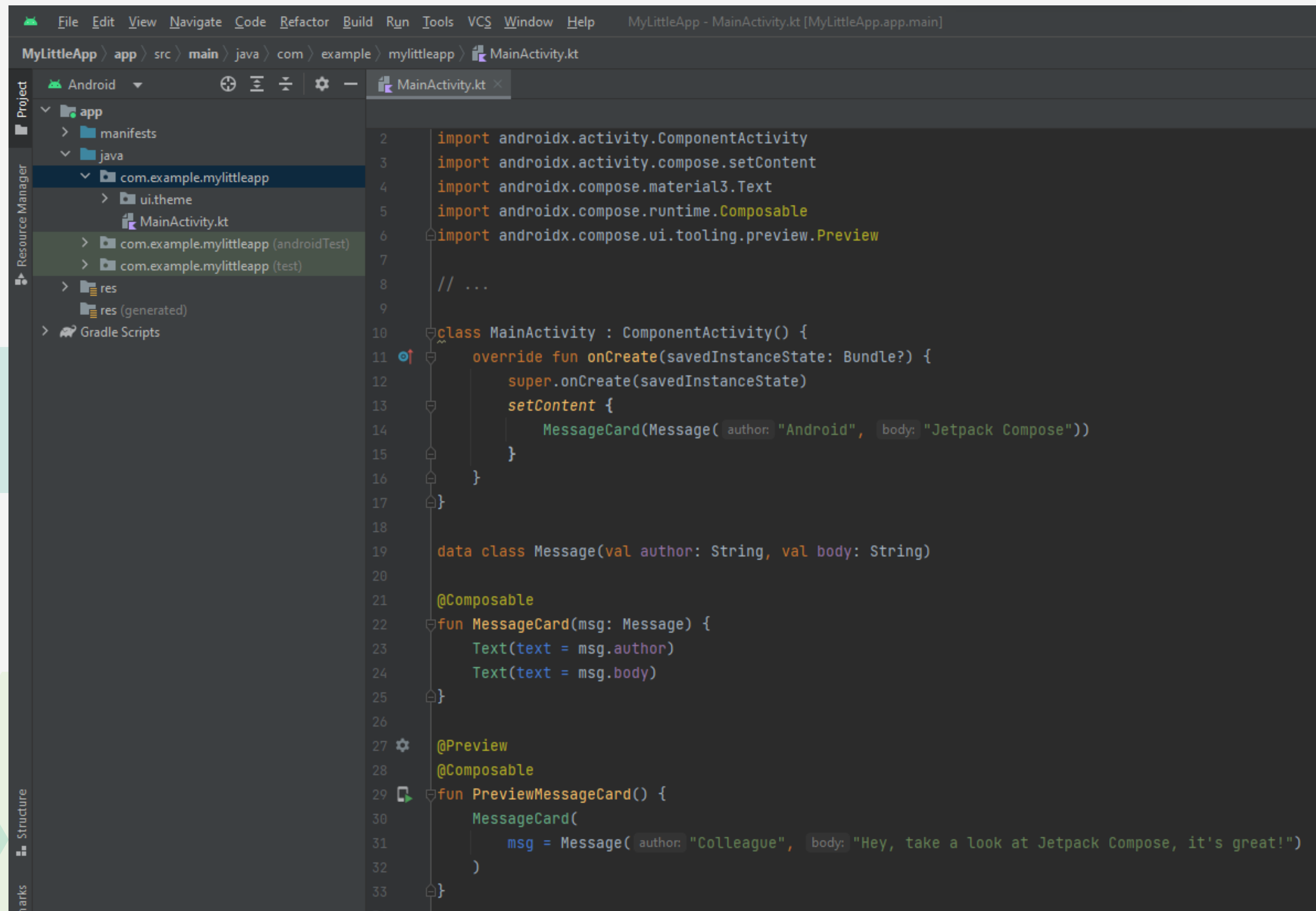
```
// ...
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            MessageCard(Message("Android", "Jetpack Compose"))
        }
    }
}
```

```
data class Message(val author: String, val body: String)
```

```
@Composable
fun MessageCard(msg: Message) {
    Text(text = msg.author)
    Text(text = msg.body)
}
```

```
@Preview
@Composable
fun PreviewMessageCard() {
    MessageCard(
        msg = Message("Colleague", "Hey, take a look at Jetpack Compose, it's great!")
    )
}
```

Adicione vários textos



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help MyLittleApp - MainActivity.kt [MyLittleApp.app.main]

MyLittleApp > app > src > main > java > com > example > mylittleapp > MainActivity.kt

Project
└─ app
   └─ manifests
   └─ java
      └─ com.example.mylittleapp
         └─ ui.theme
            └─ MainActivity.kt
         └─ com.example.mylittleapp (androidTest)
         └─ com.example.mylittleapp (test)
└─ res
   └─ res (generated)
└─ Gradle Scripts

Resource Manager

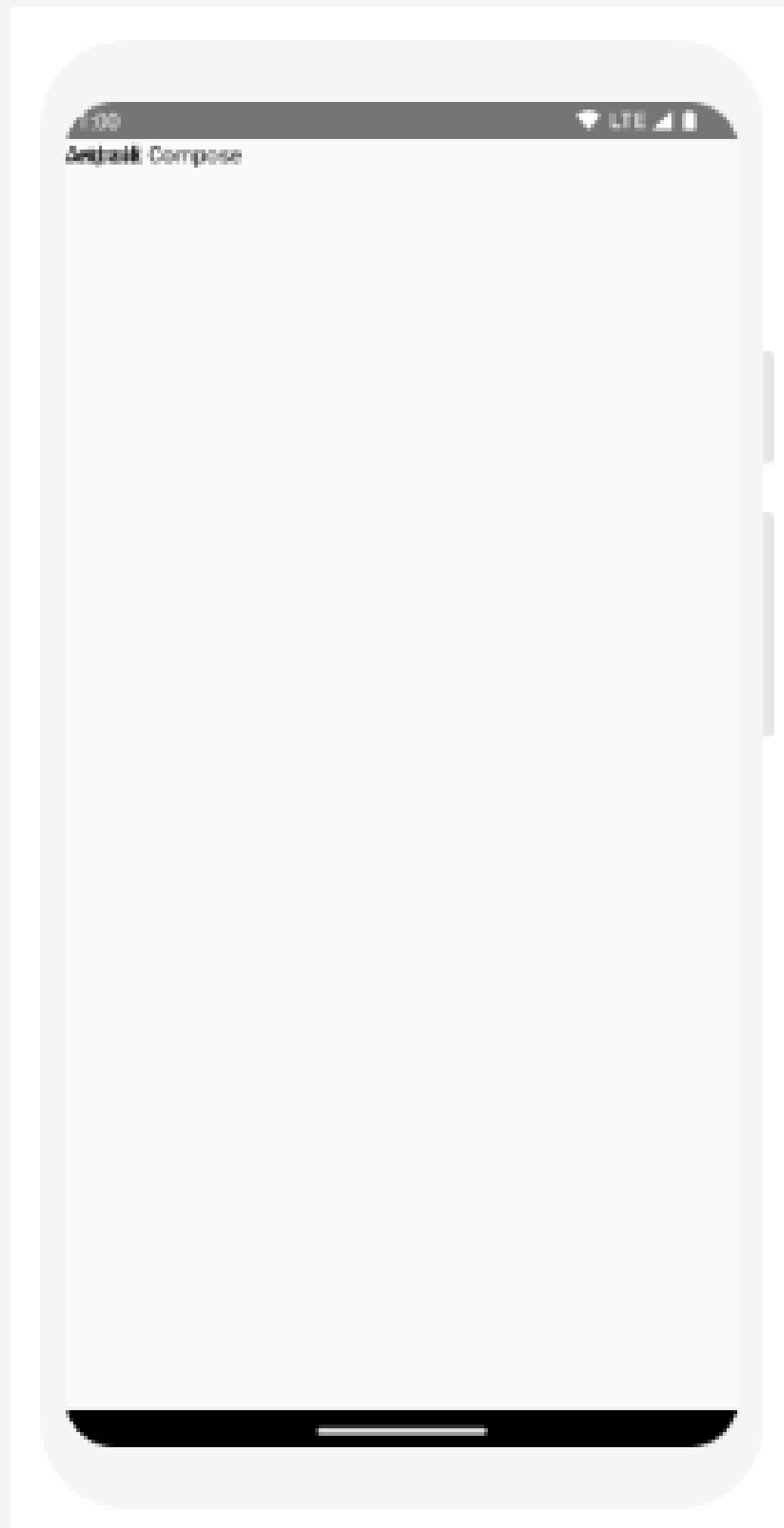
Structure

marks

2 import androidx.activity.ComponentActivity
3 import androidx.activity.compose.setContent
4 import androidx.compose.material3.Text
5 import androidx.compose.runtime.Composable
6 import androidx.compose.ui.tooling.preview.Preview
7
8 // ...
9
10 class MainActivity : ComponentActivity() {
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContent {
14             MessageCard(Message( author: "Android", body: "Jetpack Compose"))
15         }
16     }
17 }
18
19 data class Message(val author: String, val body: String)
20
21 @Composable
22 fun MessageCard(msg: Message) {
23     Text(text = msg.author)
24     Text(text = msg.body)
25 }
26
27 @Preview
28 @Composable
29 fun PreviewMessageCard() {
30     MessageCard(
31         msg = Message( author: "Colleague", body: "Hey, take a look at Jetpack Compose, it's great!")
32     )
33 }
```

Adicione vários textos

- Esse código cria dois elementos de texto dentro da visualização do conteúdo. No entanto, como você não deu nenhuma informação sobre como organizar os elementos, eles são mostrados uns sobre os outros, o que deixa o texto ilegível.



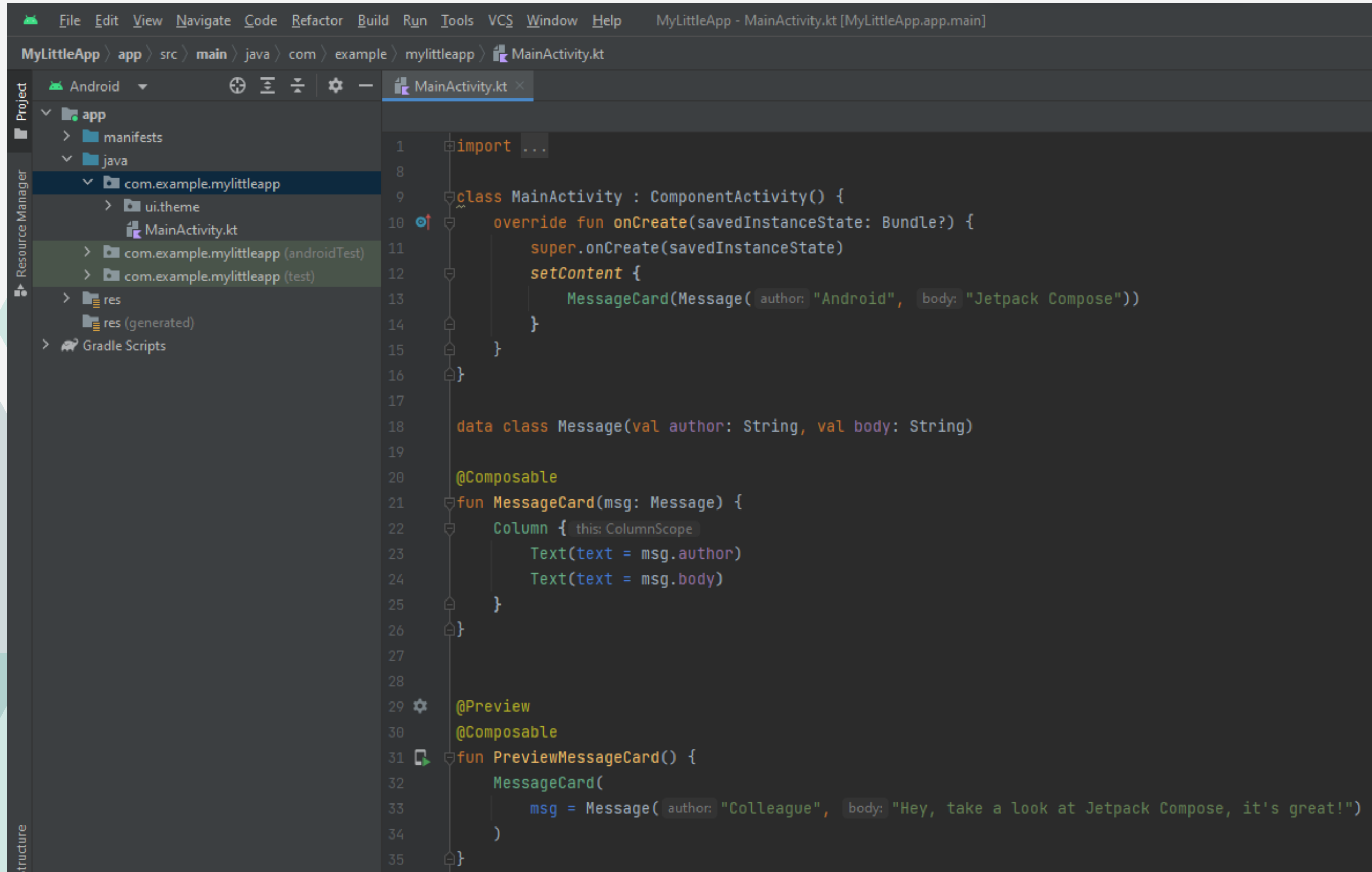
Como usar uma coluna

- A função Column permite organizar os elementos verticalmente. Adicione uma Column à função MessageCard.
- Você pode usar a Row para organizar os itens horizontalmente e a Box para empilhar os elementos.

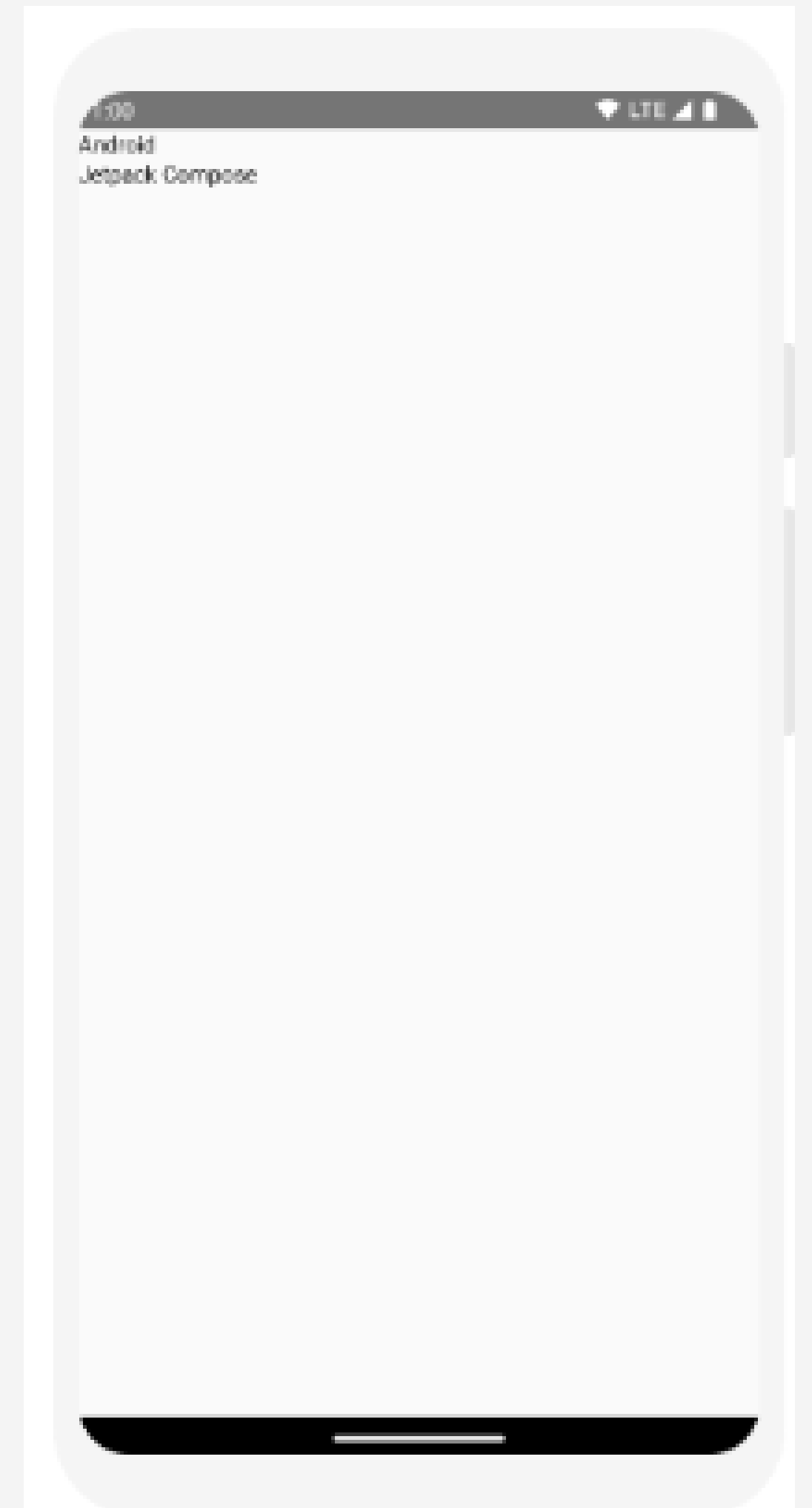
Como usar uma coluna

```
// ...  
import androidx.compose.foundation.layout.Column  
  
@Composable  
fun MessageCard(msg: Message) {  
    Column {  
        Text(text = msg.author)  
        Text(text = msg.body)  
    }  
}
```

Como usar uma coluna



Como usar uma coluna



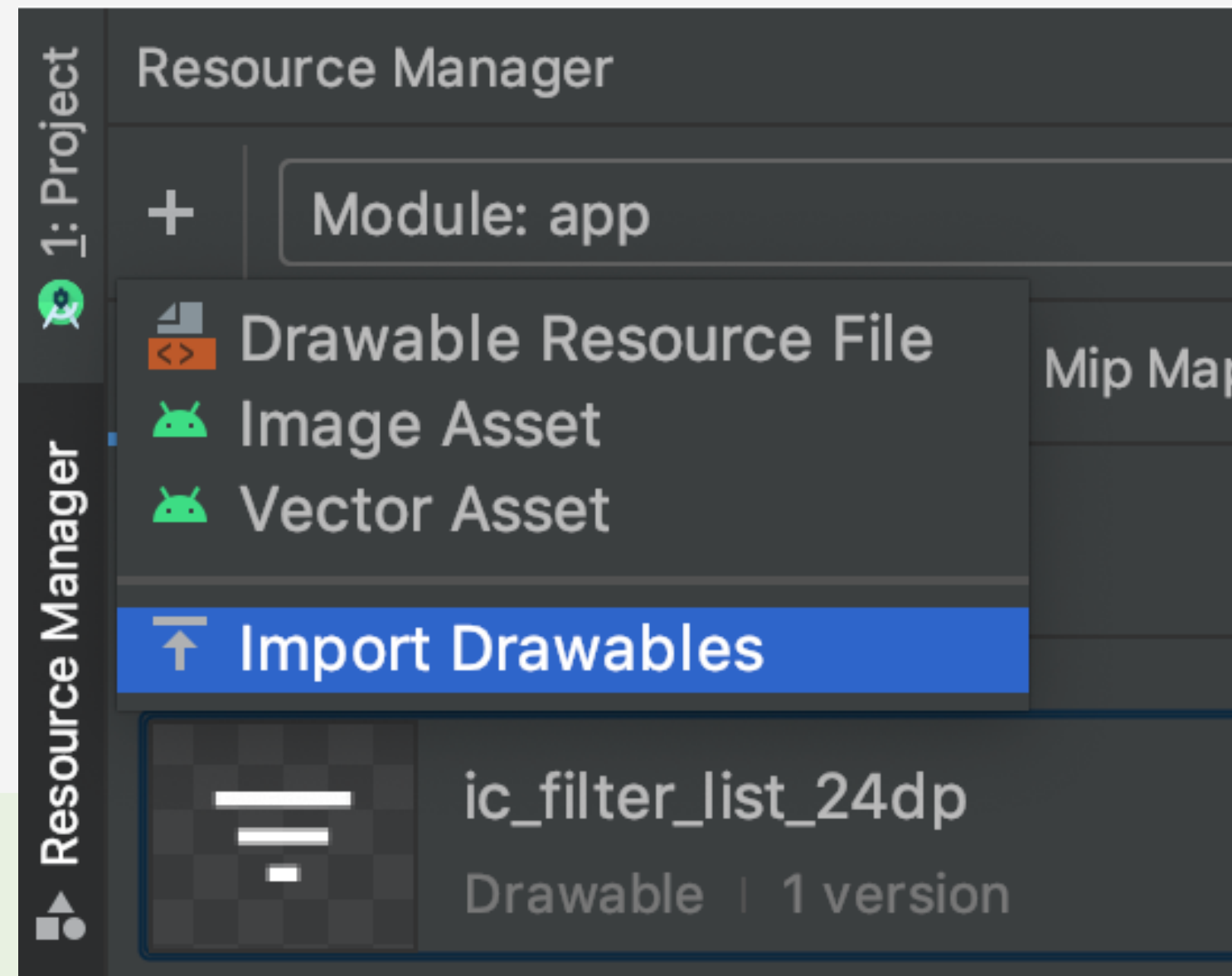
Adicionar um elemento de imagem

- Complemente seu card de mensagens adicionando uma foto do perfil do remetente. Use o Resource Manager para importar uma imagem da sua biblioteca de fotos. Adicione uma função de composição Row para ter um design bem estruturado e uma Image dentro dela:

Importar drawables para seu projeto

- Use o Resource Manager para importar recursos de imagem para seu projeto.
- Para importar recursos de imagem para seu projeto, faça o seguinte:
 - a. Arraste as imagens diretamente para a janela Resource Manager no Android Studio.
 - i. Como alternativa, você pode seguir estas etapas:
 1. Clique no ícone de adição (+).
 2. Escolha Import Drawables, conforme mostrado na Figura 3.
 3. Selecione os arquivos e as pastas que você quer importar.

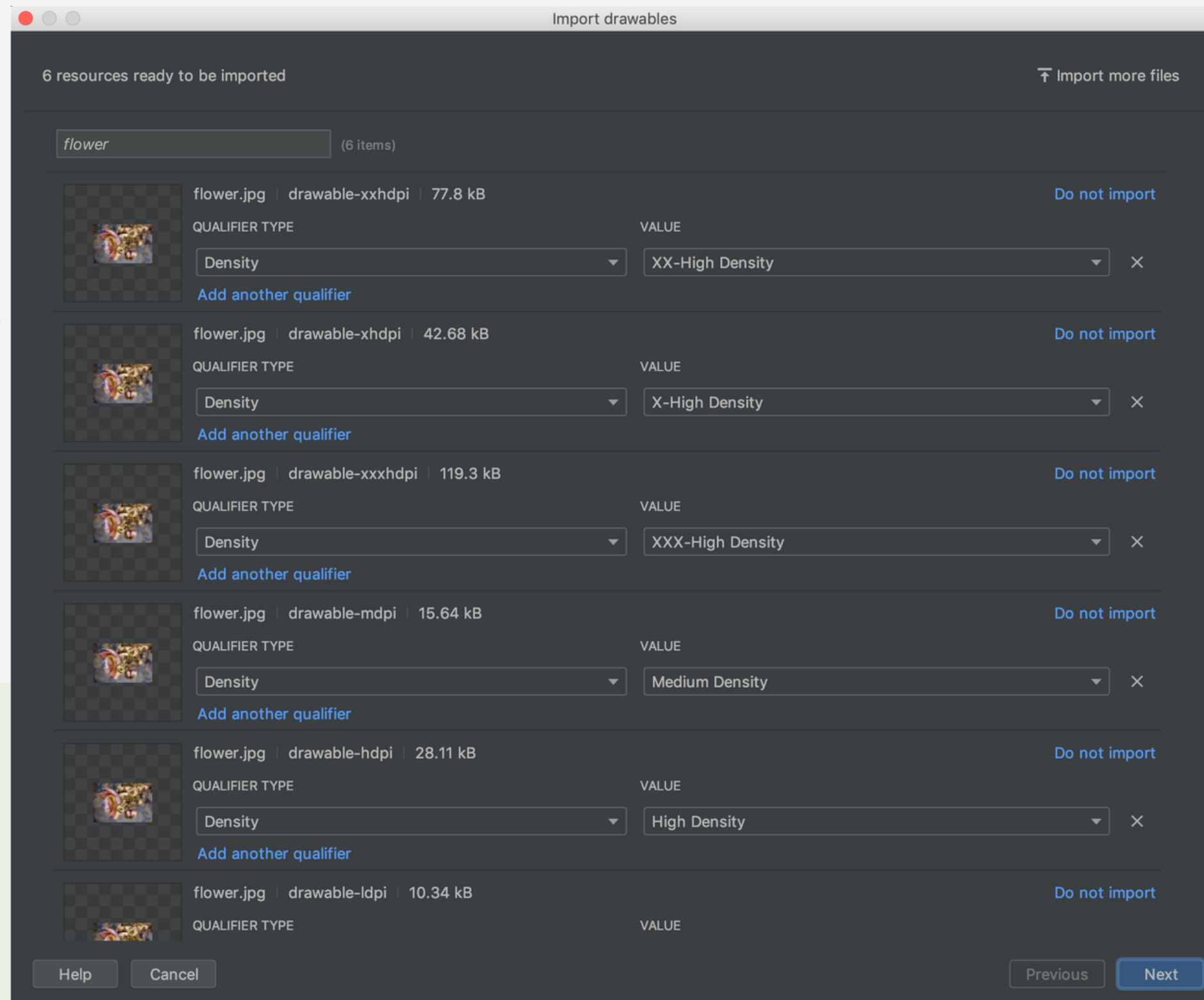
Importar drawables para seu projeto



Importar drawables para seu projeto

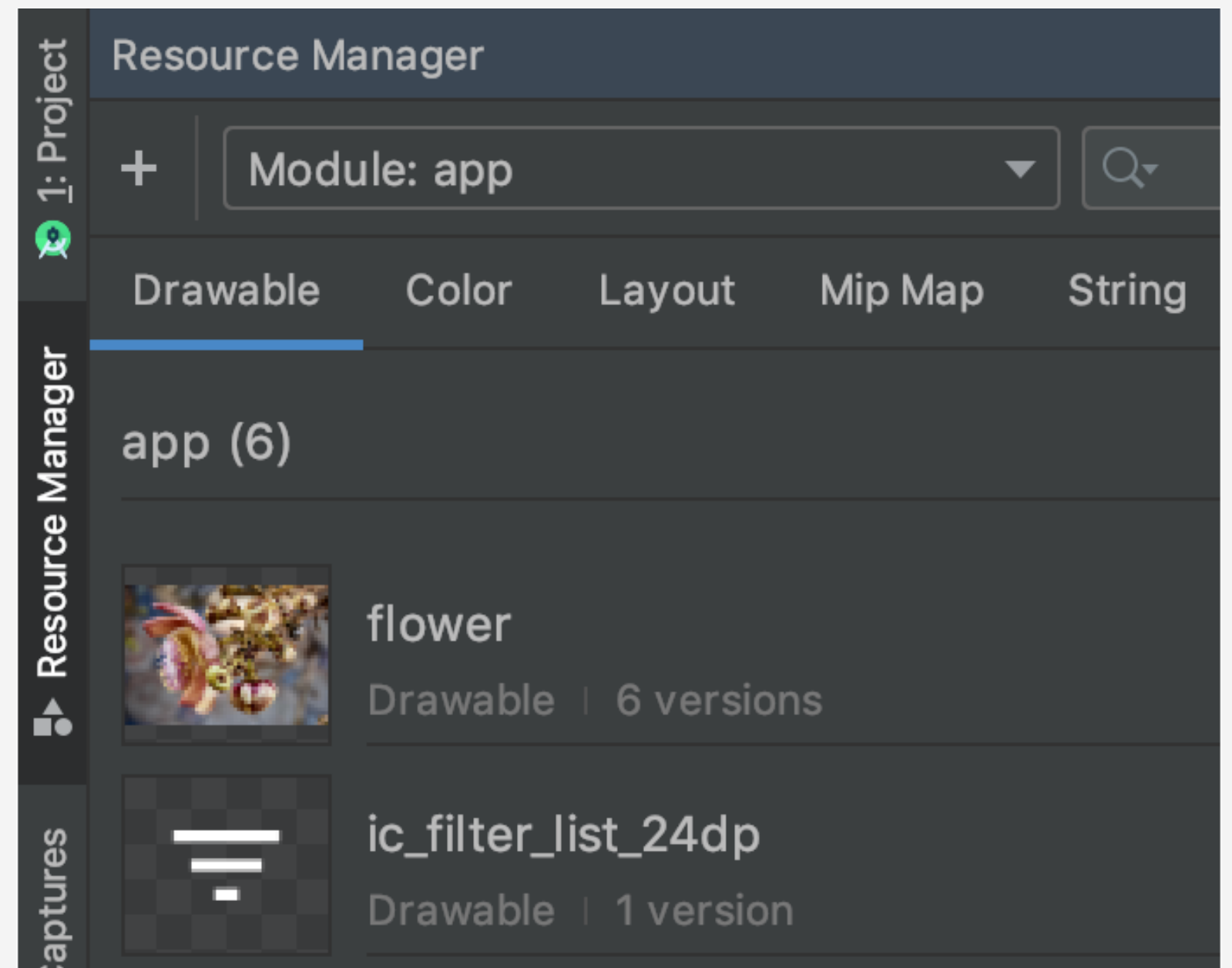
- A caixa de diálogo Import drawables vai aparecer, conforme mostrado na Figura 4. Ela mostra uma lista dos recursos que você está importando. É possível renomear recursos clicando na caixa de texto acima da visualização de um recurso.
- Se você estiver fornecendo várias versões do mesmo recurso, adicione qualificadores de configuração de dispositivos que descrevem a configuração específica a que cada recurso oferece suporte, conforme veremos na seção a seguir.
- Por exemplo, caso esteja fornecendo várias versões do mesmo recurso para diferentes densidades de tela, você pode adicionar um qualificador Density para cada versão. Se dois ou mais recursos tiverem o mesmo nome e qualificadores, apenas uma versão será importada.

Importar drawables para seu projeto



Importar drawables para seu projeto

- Depois de nomear os recursos e adicionar os qualificadores necessários, clique em Next.
- A próxima tela mostra um resumo dos recursos que você está importando. Quando estiver pronto para importar, clique em Import.
- Na janela do Resource Manager, os recursos agora estarão prontos para serem usados no projeto.



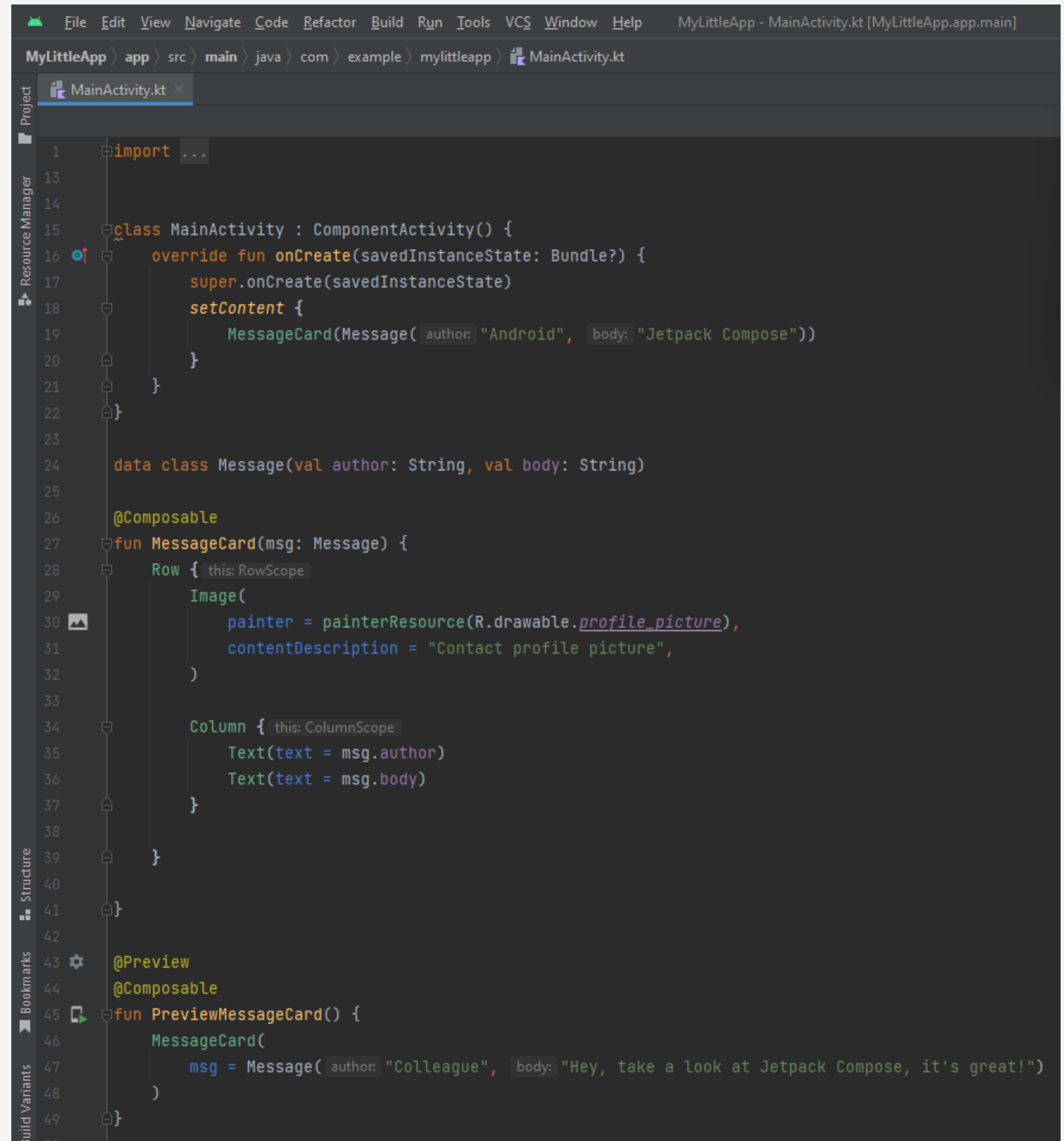
Adicionar um elemento de imagem

```
// ...
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Row
import androidx.compose.ui.res.painterResource

@Composable
fun MessageCard(msg: Message) {
    Row {
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = "Contact profile picture",
        )

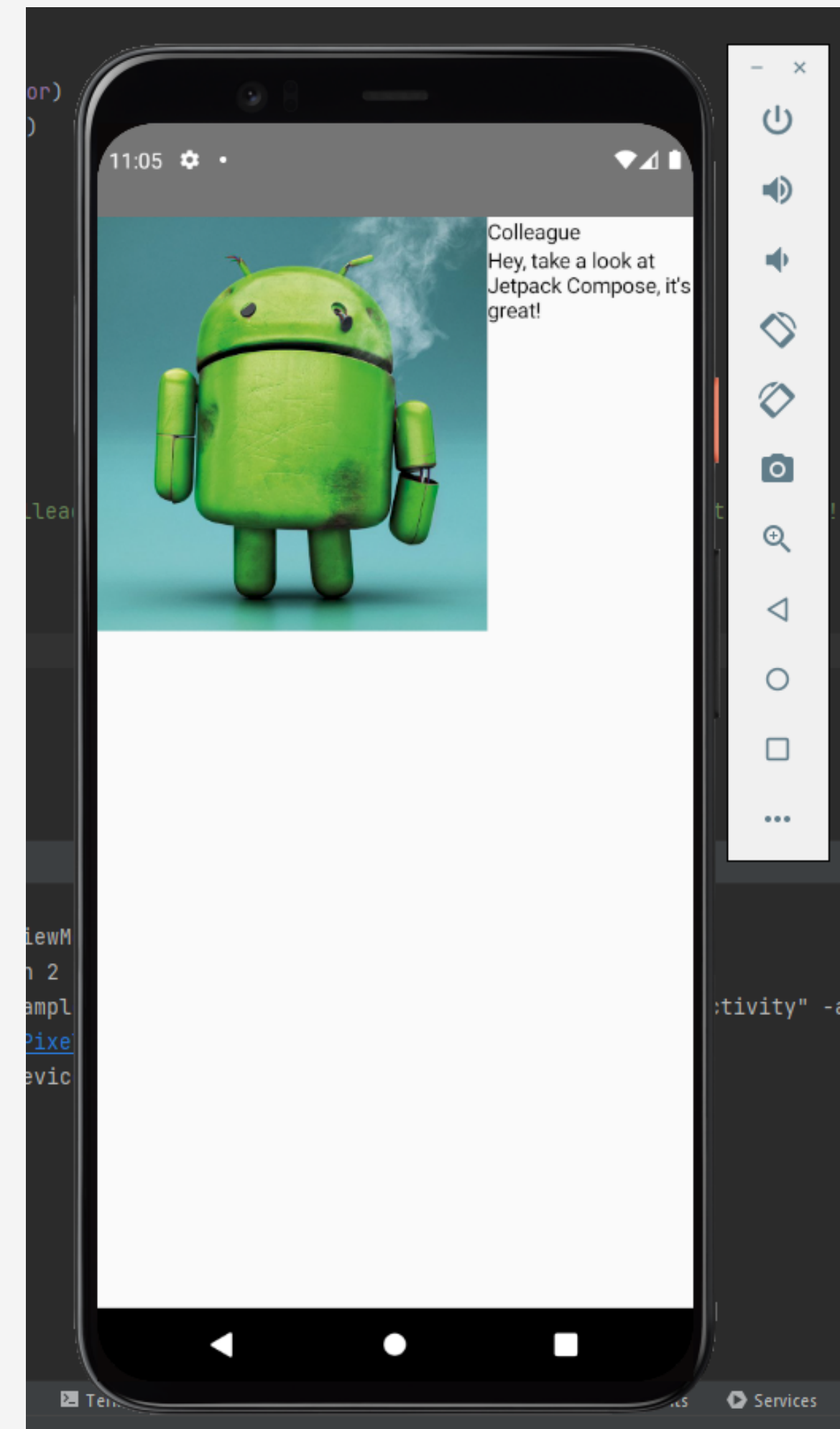
        Column {
            Text(text = msg.author)
            Text(text = msg.body)
        }
    }
}
```

Adicionar um elemento de imagem



```
1  import ...
13
14
15  class MainActivity : AppCompatActivity() {
16      override fun onCreate(savedInstanceState: Bundle?) {
17          super.onCreate(savedInstanceState)
18          setContent {
19              MessageCard(Message(author: "Android", body: "Jetpack Compose"))
20          }
21      }
22  }
23
24  data class Message(val author: String, val body: String)
25
26  @Composable
27  fun MessageCard(msg: Message) {
28      Row { this: RowScope
29          Image(
30              painter = painterResource(R.drawable.profile_picture),
31              contentDescription = "Contact profile picture",
32          )
33
34          Column { this: ColumnScope
35              Text(text = msg.author)
36              Text(text = msg.body)
37          }
38      }
39  }
40
41
42
43  @Preview
44  @Composable
45  fun PreviewMessageCard() {
46      MessageCard(
47          msg = Message(author: "Colleague", body: "Hey, take a look at Jetpack Compose, it's great!")
48      )
49  }
```

Adicionar um elemento de imagem



Configurar o Layout

- O layout da mensagem tem a estrutura certa, mas os elementos não estão bem espaçados e a imagem é muito grande. Para decorar ou configurar uma função de composição, o Compose usa modificadores. Eles permitem alterar o tamanho, o layout e a aparência dos elementos que podem ser compostos ou adicionar interações de alto nível, como tornar um elemento clicável. Eles podem ser encadeados para criar funções combináveis mais detalhadas. Você vai usar alguns deles para melhorar o layout.

Configurar o Layout

```
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.layout.width
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.unit.dp

@Composable
fun MessageCard(msg: Message) {
    // Add padding around our message
    Row(modifier = Modifier.padding(all = 8.dp)) {
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = "Contact profile picture",
            modifier = Modifier
                // Set image size to 40 dp
                .size(40.dp)
                // Clip image to be shaped as a circle
                .clip(CircleShape)
        )
```

```
        // Add a horizontal space between the image and the column
        Spacer(modifier = Modifier.width(8.dp))

        Column {
            Text(text = msg.author)
            // Add a vertical space between the author and message texts
            Spacer(modifier = Modifier.height(4.dp))
            Text(text = msg.body)
        }
    }
}
```

Configurar o Layout



```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            MessageCard(Message( author: "Android", body: "Jetpack Compose"))
        }
    }
}

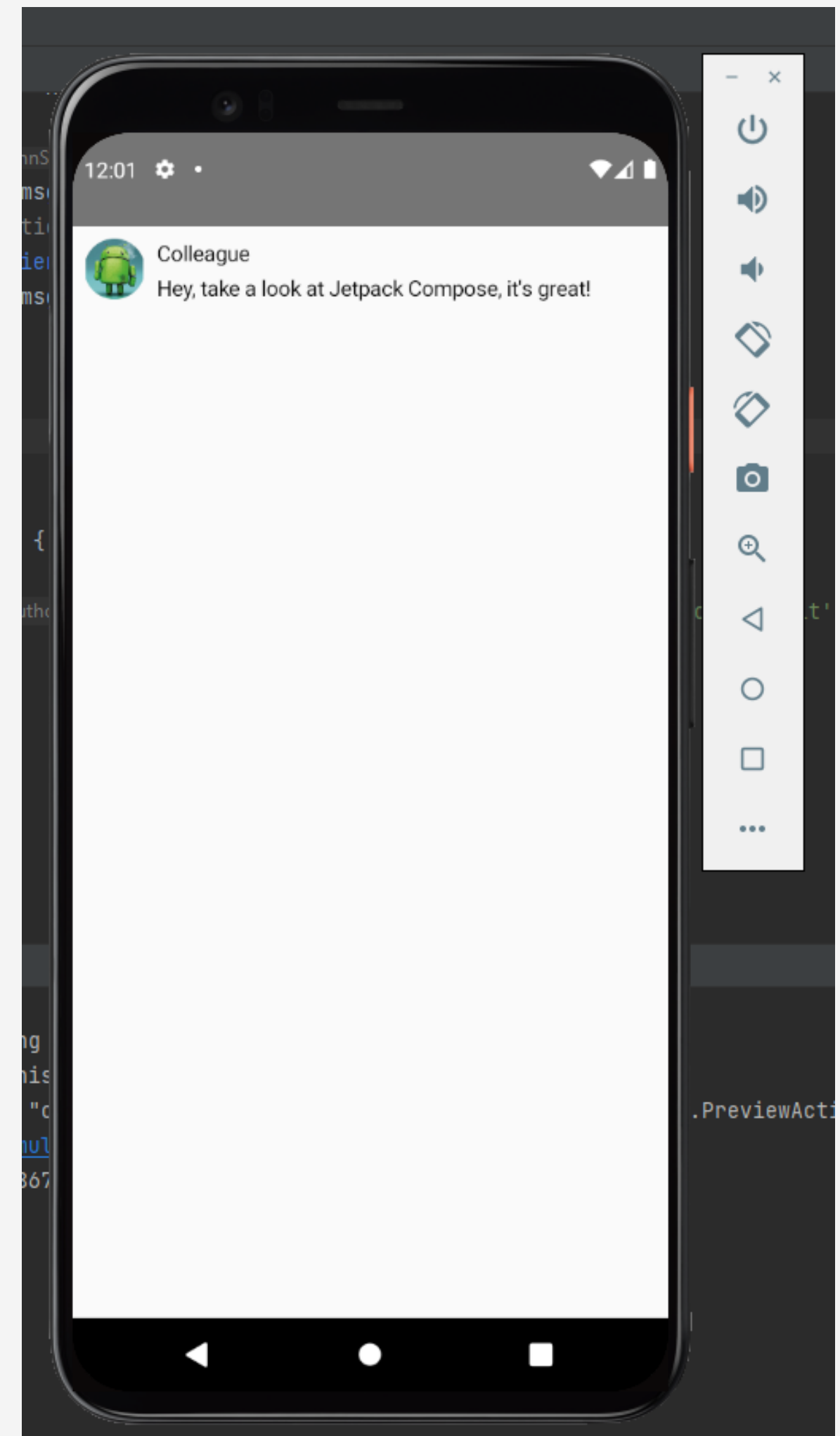
data class Message(val author: String, val body: String)

@Composable
fun MessageCard(msg: Message) {
    // Add padding around our message
    Row(modifier = Modifier.padding(all = 8.dp)) { this: RowScope
        Image(
            painter = painterResource(R.drawable.profile_picture),
            contentDescription = "Contact profile picture",
            modifier = Modifier
                // Set image size to 40 dp
                .size(40.dp)
                // Clip image to be shaped as a circle
                .clip(CircleShape)
        )

        // Add a horizontal space between the image and the column
        Spacer(modifier = Modifier.width(8.dp))

        Column { this: ColumnScope
            Text(text = msg.author)
            // Add a vertical space between the author and message texts
            Spacer(modifier = Modifier.height(4.dp))
            Text(text = msg.body)
        }
    }
}
```

Configurar o Layout





#R4E

Software Developer

Android

Primeiro Projeto - Layouts

