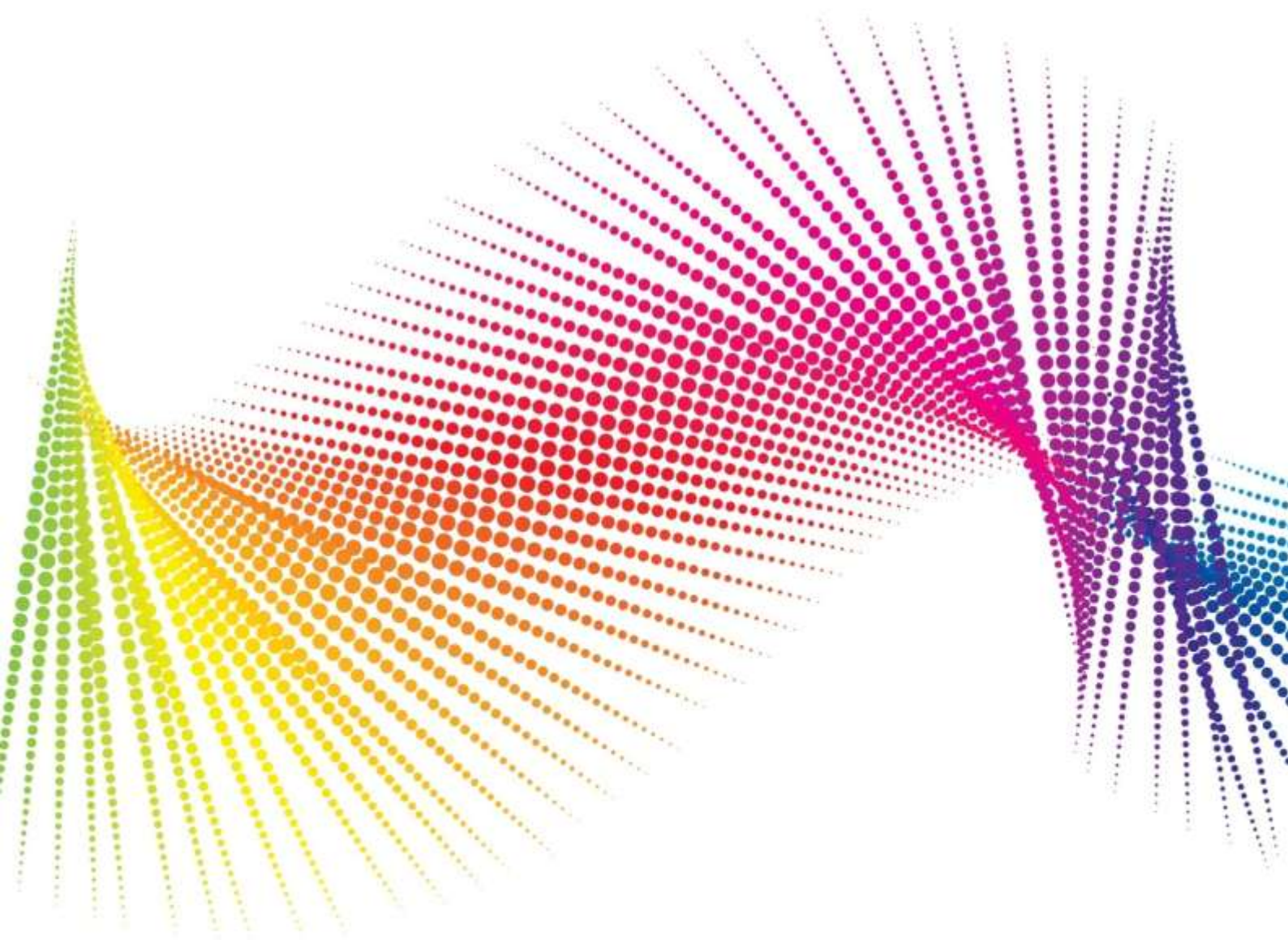


Computação Móvel

Aula 10



Este material é parte integrante da disciplina oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

Aula 10: Interface com usuário android

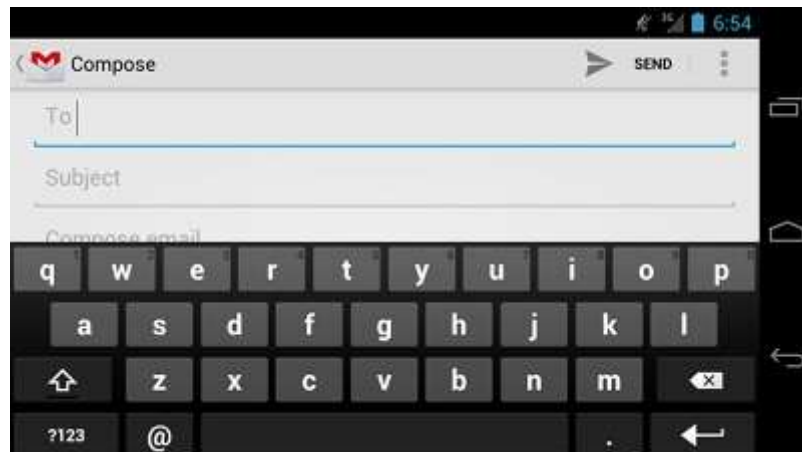
Objetivo: Demonstrar as formas de construção de uma interface para um usuário android, apresentando os objetos fornecidos pelo sistema android que têm por objetivo a construção das referidas interfaces e suas respectivas funções dentro do programa android.

Campo de texto

Um campo de texto permite ao usuário digitar o texto em sua aplicação. Ele pode ser composto de uma ou varias linhas. Quando se pressiona um campo de texto, esse recebe o cursor e exibe automaticamente o teclado. Além de digitação, campos de texto permitem uma variedade de outras atividades, como a seleção de texto (cortar, copiar, colar) e a visualização de dados através de autopreenchimento.

Um campo de texto pode ser adicionado em um layout utilizando-se objeto EditText. Normalmente deve-se fazê-lo em seu layout XML com um elemento <EditText>. Como:

```
<EditText
    android:id="@+id/postal_address"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="@string/postal_address_hint"
    android:inputType="textPostalAddress|
        textCapWords|
        textNoSuggestions" />
```



Campo texto selecionado

Botões

Um botão consiste em um texto ou um ícone que comunica ao sistema que a ação ocorre quando o usuário o pressiona. Em seguida temos alguns exemplos de botões.



Dependendo se queremos um botão com o texto, um ícone, ou ambos pode-se criar o layout do botão de três maneiras:

Somente com texto, usando a classe Button, por exemplo:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
... />
```

Somente com um ícone, usando a classe ImageButton, por exemplo:

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/button_icon"
    ... />
```

Com texto e um ícone, usando a classe Button com o atributo android:drawableLeft, por exemplo:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    android:drawableLeft="@drawable/button_icon"
    ... />
```

Respondendo ao evento click

Quando o usuário pressiona um botão, o objeto Button recebe um evento on-click. Para definir o manipulador de eventos de click de um botão, adicione o atributo android:onClick para o elemento <Button> em seu layout XML. O valor para este atributo deve ser o nome do método que se deseja chamar, em resposta a um evento de click. Por exemplo, aqui está um layout com um botão usando android:onClick.

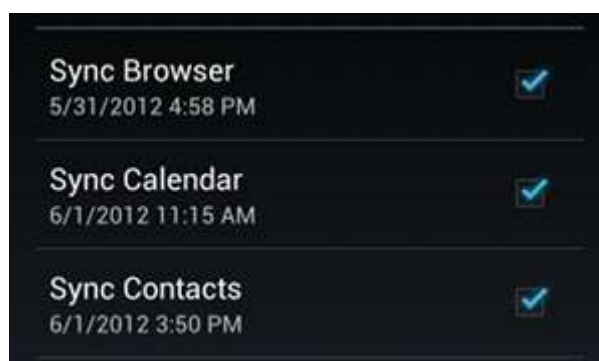
```
<?xml version="1.0" encoding="utf-8"?>
<Button xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send"
    android:onClick="sendMessage" />
```

Dentro da atividade que hospeda esse layout, o método a seguir manipula o evento click

```
public void sendMessage(View view) {  
    // Do something in response to button click  
}
```

Caixas de verificação

Caixas de verificação permitem que o usuário selecione uma ou mais opções de um conjunto. Normalmente, deve-se apresentar cada opção de seleção em uma lista vertical, como no exemplo a seguir:



Para criar cada opção de seleção, crie um CheckBox em seu layout, pois um conjunto de opções de caixa de seleção permite ao usuário selecionar vários itens. Cada caixa de verificação é gerenciada separadamente e você deve registrar um verificador de pressionamento para cada uma.

Respondendo ao evento click

Quando o usuário seleciona uma caixa de verificação o objeto CheckBox recebe um evento on-click.

Para definir o manipulador de eventos de clique para uma caixa de verificação, adicione o atributo android:onClick para o elemento <CheckBox> em seu layout XML. O valor para este atributo deve ser o nome do método que você deseja chamar, em resposta a um evento de click.

Por exemplo, aqui estão um par de objetos CheckBox em uma lista.


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <CheckBox android:id="@+id/checkbox_meat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/meat"
        android:onClick="onCheckboxClicked"/>
    <CheckBox android:id="@+id/checkbox_cheese"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/cheese"
        android:onClick="onCheckboxClicked"/>
</LinearLayout>
```

Dentro da atividade que hospeda esse layout, o método a seguir manipula o evento. Clique para ambas as opções:

```
public void onCheckboxClicked(View view) {
    boolean checked = ((CheckBox) view).isChecked();
    switch(view.getId()) {
        case R.id.checkbox_meat:
            if (checked)
                // Put some meat on the sandwich
            else
                // Remove the meat
            break;
        case R.id.checkbox_cheese:
            if (checked)
                // Cheese me
```

```
else
    // I'm lactose intolerant
    break;
// TODO: Veggie sandwich
}
```

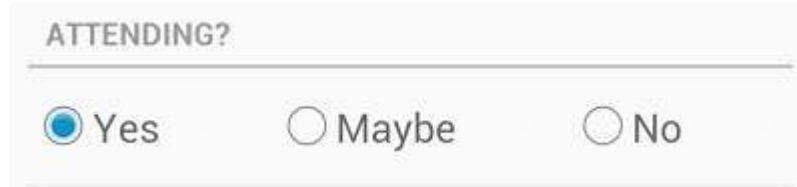
EXERCÍCIOS
Agora, veja os exercícios disponíveis acessando o AVA, ou via QR Code*. Não deixe de visualizar esses exercícios, pois eles fazem parte da sequência desta aula e, portanto, são essenciais para a aprendizagem.



* O QR Code é um código de barras que armazena links às páginas da web. Utilize o leitor de QR Code de sua preferência para acessar esses links de um celular, tablet ou outro dispositivo com o plugin Flash instalado.

Botões de seleção

Botões de seleção permitem que o usuário selecione uma opção de um conjunto. Devem-se usar os botões de seleção para conjuntos opcionais que são mutuamente exclusivos se o usuário precisar para ver todas as opções disponíveis, lado a lado. Se não for necessário mostrar todas as opções de lado a lado, use um objeto spinner. Em seguida está um exemplo de botões de seleção.



ATTENDING?

☒ Yes ☐ Maybe ☐ No

Para criar cada opção de botão de seleção, crie um objeto `RadioButton` em seu layout. No entanto, em razão dos botões de seleção serem mutuamente exclusivos, é necessário agrupá-los dentro de um objeto `RadioGroup`. Agrupando-os, o sistema garante que apenas um botão de opção pode ser selecionado de cada vez, como no exemplo a seguir:

```
<?xml version="1.0" encoding="utf-8"?>
<RadioGroup xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <RadioButton android:id="@+id/radio_pirates"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/pirates"
        android:onClick="onRadioButtonClicked"/>
    <RadioButton android:id="@+id/radio_ninjas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ninjas"
        android:onClick="onRadioButtonClicked"/>
</RadioGroup>
```

```
public void onRadioButtonClicked(View view) {
    boolean checked = ((RadioButton) view).isChecked();
    switch(view.getId()) {
        case R.id.radio_pirates:
            if (checked)
                // Pirates are the best
    }
```

```
        break;
    case R.id.radio_ninjas:
        if (checked)
            // Ninjas rule
            break;
    }
}
```

Resumo

Nesta aula apresentamos os objetos que podem compor um layout de interface para o usuário no sistema android, e como executar eventos associados a estes objetos de modo a melhorar o desempenho da interface.

Próxima aula

Finalmente chegou a hora, vamos começar o desenvolvimento de aplicações – **Aula 11** – no qual continuaremos aprendendo os conceitos e como criar uma interface voltada para o usuário. Para tanto, discutiremos de layouts a menu, de modo explicaremos tudo que o ambiente nos permite.

REFERÊNCIAS

LECHETA, Ricard R. *Android: aprenda a criar aplicações para dispositivos móveis com o android SDK*. 2. ed. São Paulo: Novatec, 2010.

ROGERS, Rick; et al. *Desenvolvimento de aplicações android*. São Paulo: Novatec, 2009.