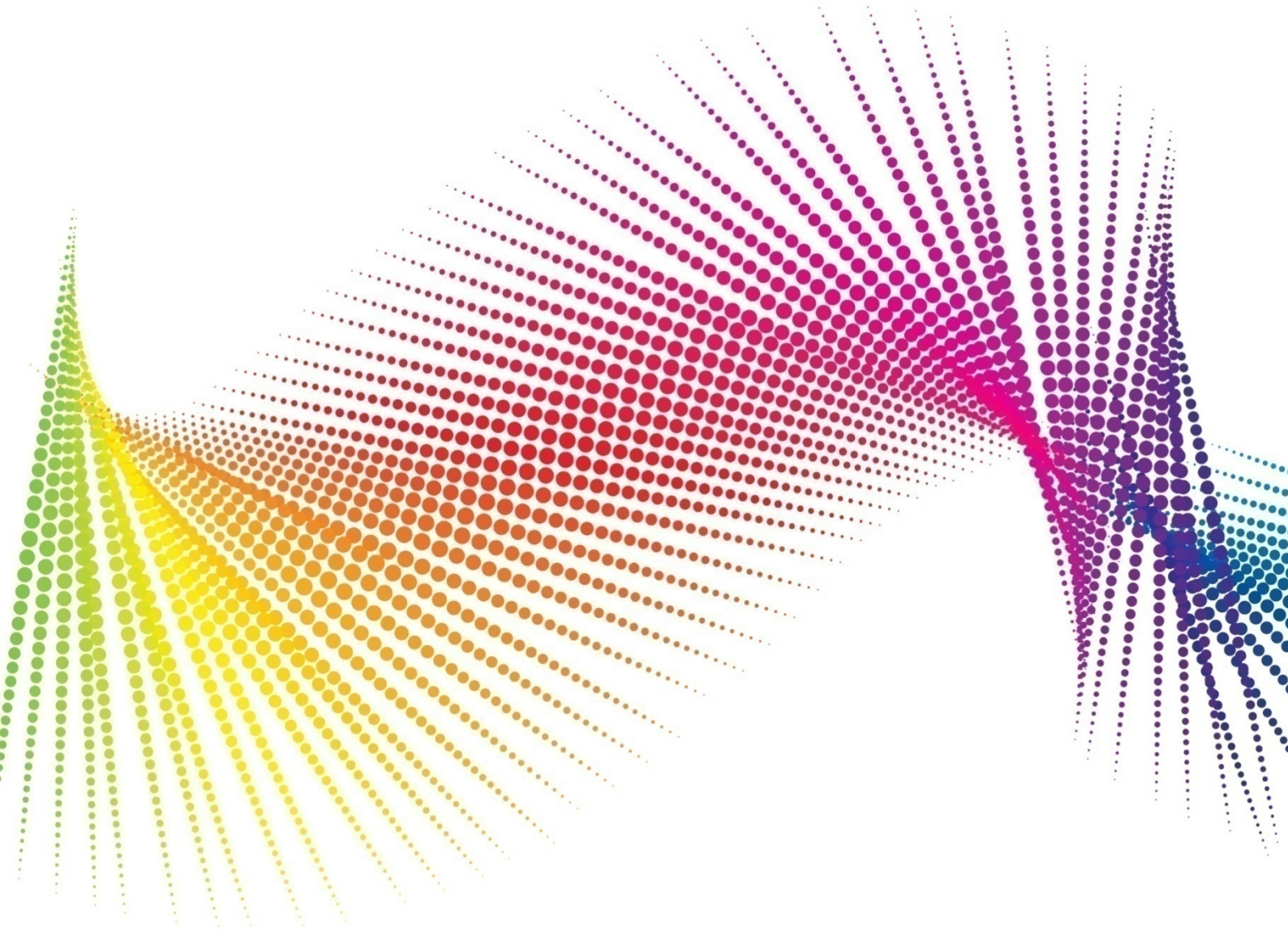
Computação Móvel

Aula 11



Aula 11: Interface com usuário android

Objetivo: Demonstrar as formas de construção de uma interface com usuário android, iniciando com o entendimento da estrutura de layouts fornecida por ele, suas widgets, os eventos associados às interfaces, concluindo com a criação dos diversos tipos de menus disponíveis do android.

Layouts

Layouts são as interfaces do usuário em uma activity. Eles definem a estrutura de visualização e possuem os elementos que aparecem em uma activity. Um layout pode ser declarado de duas maneiras:

- Pode-se declarar os elementos da interface utilizando o XML, pois o android fornece um XML simplificado que representa às classes e subclasses de visualização.
- Pode-se instanciar elementos para um layout de modo dinâmico, ou seja, o aplicativo pode criar Views e objetos ViewGroup durante a execução do programa.

Criando uma interface com usuário com XML

Quando utilizamos o XML do android, podemos criar rapidamente interfaces com usuário e os elementos de tela nela contida, da mesma forma que criaríamos uma página da Web utilizando HTML.

Toda interface com usuário deve conter somente um elemento raiz, que deve ser uma View ou um objeto ViewGroup. Uma vez que se defina o elemento raiz, você pode adicionar outros objetos para construção de um layout ou widgets que será herdado em uma hierarquia na View que define o layout. Para melhor entendimento veja o exemplo 1.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >
        <TextView android:id="@+id/text"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a TextView" />
        <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

Layouts Comuns

A classe ViewGroup fornece as maneiras de mostrar os elementos colocados na interface com usuário. A seguir estão alguns dos tipos de layout mais comuns que são construídos para a plataforma android.

Disposição linear

Quando utilizamos um layout linear ele organiza seus elementos em uma única linha horizontal ou vertical, criando uma barra de deslocamento, caso o comprimento necessário a todos os elementos da janela de exceder o comprimento da tela.

Disposição relativa

Permite especificar a localização dos elementos secundários em relação a um elemento primário, ou alinhada com a parte superior do elemento primário.

Web view

Exibe páginas da web.

Construção de layouts dinâmicos

Quando se deseja construir um layout dinâmico ou não pré-determinado, pode-se usar uma classe AdapterView para realizar o preenchimento do layout em tempo de execução. Essa classe usa um adaptador para ligar dados aos seus elementos, o qual se comporta como um intermediário entre a fonte de dados e o AdapterView atualizando os dados, como uma matriz ou uma consulta do banco de dados, e converte cada entrada numa visualização que pode ser adicionados ao AdapterView.

Widgets

Widgets são aplicativos em miniatura que podem ser embutidos em outras aplicações e receber atualizações periódicas.

Para criar um aplicativo Widget, precisa-se do objeto AppWidgetProviderInfo, o qual deve ser definido em XML, descrever os metadados de uma Widget App, declarar seu layout, sua frequência de atualização e a classe AppWidgetProvider.

Eventos da interface

No android, existem vários meios de interceptar os eventos do usuário com o aplicativo. Considerando os eventos que compõem a interface do usuário, teremos a necessidade de capturar os eventos do objeto View que o usuário estiver utilizando, sendo que esta classe fornecerá o método para fazê-lo como descrito a seguir.

onTouchEvent() – ocorrerá sempre que o usuário encostar em um elemento da View.

Listeners de eventos

Um listener de evento é uma interface da classe View que contém um método de retorno para uma chamada. Estes métodos serão chamados pela estrutura android quando a View receber alguma ação realizada pelo usuário em algum elemento da interface.

Os métodos associados a um listener de eventos estão listados a seguir:

- onClick() – associado ao View.OnClickListener. Este evento é chamado quando o usuário pressiona o item.
- onLongClick() – associado ao View.OnLongClickListener. Este evento é chamado quando o usuário pressiona o item e o mantém pressionado.
- onFocusChange() – associado ao View.OnFocusChangeListener. Este evento é chamado quando o usuário retira um objeto de evidência.
- onKey() – associado ao View.OnKeyListener. Este evento é chamado quando o usuário está focado num item e pressiona ou libera uma tecla.

- OnTouch () – associado ao View.OnTouchListener. Este evento é chamado quando o usuário executa a ação “toque” no aplicativo.
- onCreateContextMenu() – associado ao View.OnCreateContextMenuListener. Este evento é chamado quando um menu de contexto está sendo construído.

Para melhor entendimento veja o exemplo 2.

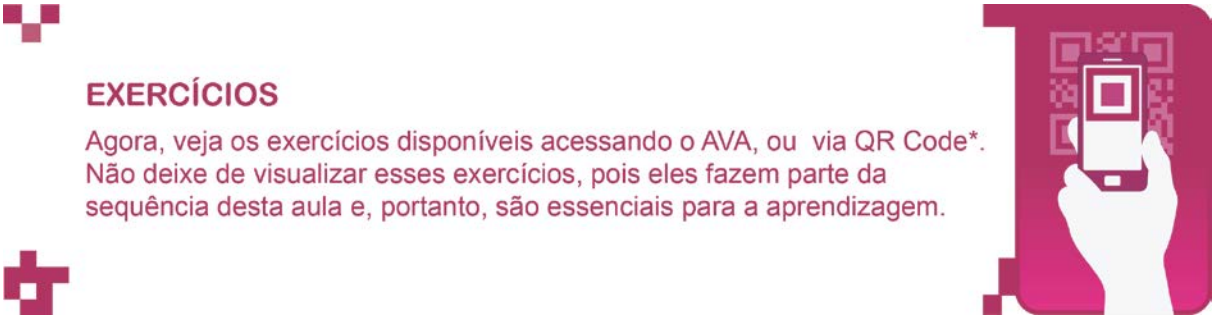
```
// Create an anonymous implementation of OnClickListener
private OnClickListener mCorkyListener = new OnClickListener() {
    public void onClick(View v) {
      // do something when the button is clicked
    }
};

protected void onCreate(Bundle savedValues) {
    ...
    // Capture our button from layout
    Button button = (Button)findViewById(R.id.corky);
    // Register the onClick listener with the implementation above
    button.setOnClickListener(mCorkyListener);
    ...
}
```

Manipuladores de eventos

Caso seja necessário personalizar uma View, será preciso definir métodos de retorno de vários usados como manipuladores de eventos padrão, como segue:

- onKeyDown(int, KeyEvent) – Chamado quando se pressiona uma tecla.
- onKeyUp(int, KeyEvent) – Chamado quando se solta uma tecla.
- onTrackballEvent(MotionEvent) – Chamado quando um evento de movimento ocorre trackball.

- onTouchEvent(MotionEvent) – Chamado quando se movimenta o dedo pela tela.
- onFocusChanged(boolean, int, Rect) – Chamado quando o elemento em destaque é alterado.



***** 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.

Menus

Os menus são um componente comum na interface do usuário, pois eles estão em muitos tipos de aplicações. Afim de oferecer uma experiência familiar ao usuário, você deve-se utilizar Menus para apresentar as ações e opções a serem tomadas por ele em suas atividades.

Menu de opções

O menu de opções é representado pelo menu principal da atividade. É onde colocamos as ações que têm um impacto global sobre o aplicativo, como "Pesquisa", "Compor e-mail," e "Configurações".

Menu de contexto

Um menu de contexto é um menu flutuante que aparece quando o usuário realiza um longo clique em um elemento. Ele fornece ações que afetam o conteúdo selecionado ou quadro de contexto.

Menu pop-up

Um menu pop-up exibe itens em uma lista vertical que está posicionado em na View que chamou o menu. As ações de um menu pop-up devem não afetar diretamente o conteúdo. Em vez disso, ele é para ações que se relacionam com regiões de sua atividade.

Definição de um menu em XML

Para todos os tipos de menus, o android fornece um formato padrão XML para definir seus itens. Em vez de construir um menu de código de sua atividade, você deve definir um menu e todos os seus itens em um XMLresorce de menu.

Para defini-lo, deve-se criar um arquivo XML dentro do seu projeto res/menu/diretório e montar o menu com os seguintes elementos:

<menu> – Define um menu, que serve como recipiente para itens de menu.
<item> – Cria um MenuItem, o que representa um único item em um menu.

Um recipiente, opcional invisível para <item> elementos, permite categorizar os itens de menu para que eles compartilhem propriedades, tais como estado ativo e visibilidade. Para melhor entendimento, veja o exemplo 3.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
   <item android:id="@+id/new_game"
        android:icon="@drawable/ic_new_game"
        android:title="@string/new_game"
        android:showAsAction="ifRoom"/>
```

```
    <item android:id="@+id/help"
        android:icon="@drawable/ic_help"
        android:title="@string/help" />
</menu>
```

Para usar o menu em sua atividade, você precisa inserir o recurso de menu usando MenuInflater.inflate () .

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.

Resumo

Nesta aula apresentamos os layouts que o android disponibiliza para interface do usuário, os eventos associados a esta interface e o processo de criação de menus em uma aplicação.

Próxima aula

Na próxima aula continuaremos a nos familiarizar com as interfaces do usuário – **Aula 12** – em que serão mostrados os processos de criação de notificação para o usuário, bem como estilos, temas e visualizações desta interface.

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.