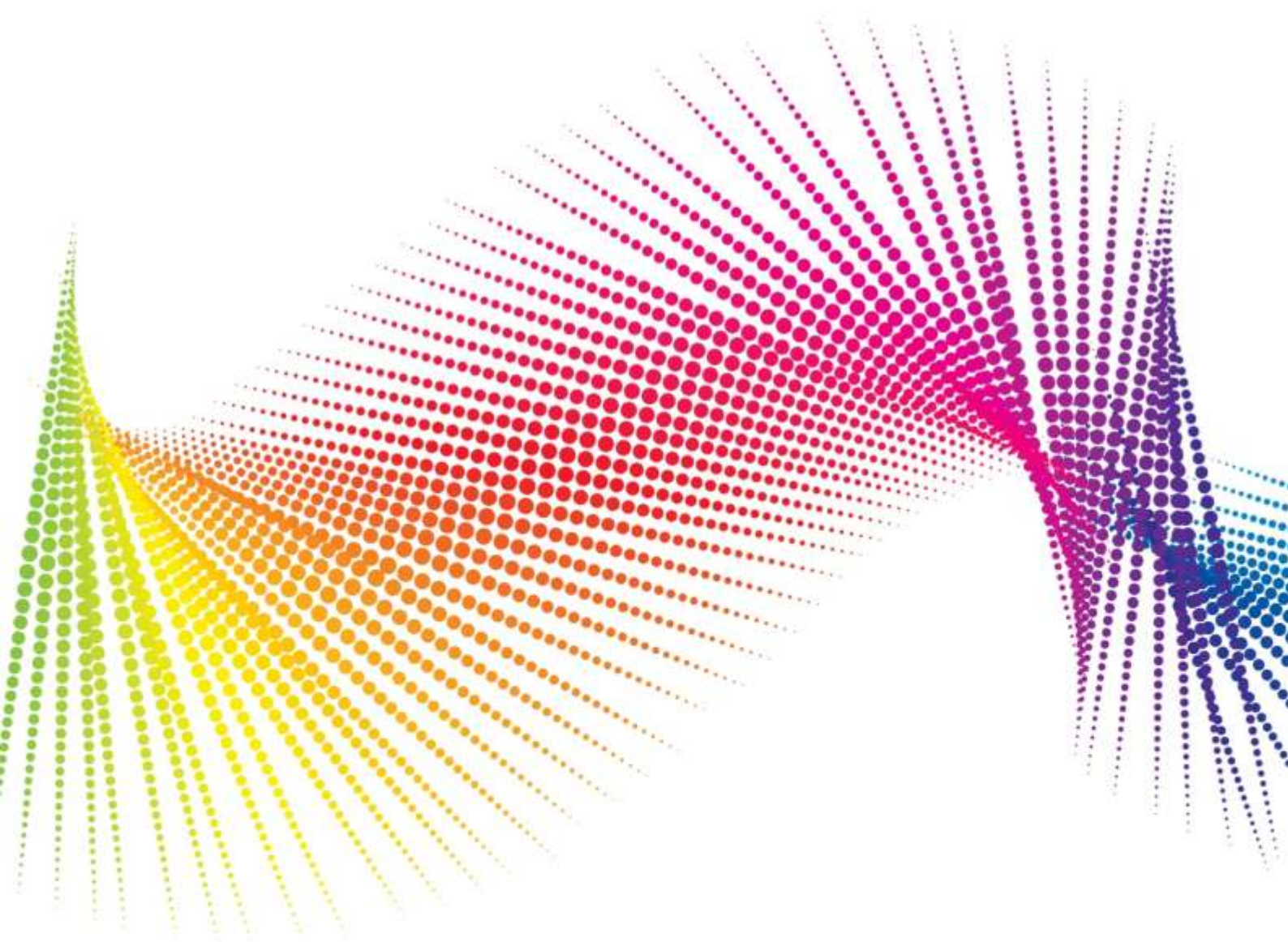


Computação Móvel

Aula 18



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 18: Compatibilidade Android com dispositivos e telas

Objetivo: Demonstrar o processo utilizado para realizar as definições de telas quando aplicações são preparadas para trabalhar em vários tipos de dispositivos, surgindo múltiplos tipos telas, que variam em tamanho e densidade.

Suporte as múltiplas telas

Permite especificar os tamanhos de tela que darão suporte à aplicação e ativar o modo de compatibilidade para telas maiores do que o que seu aplicativo suporta. É importante que se use sempre este elemento em um aplicativo para especificar os tamanhos de tela que o aplicativo suporta.

Uma aplicação "suporta" determinado tamanho de tela que se redimensiona corretamente para preencher a tela inteira. O redimensionamento normal aplicado pelo sistema funciona bem para a maioria das aplicações e não temos que fazer qualquer trabalho extra para fazer a aplicação funcionar em telas maiores do que um aparelho celular, no entanto, muitas vezes é importante otimizar UI para diferentes tamanhos de tela, fornecendo recursos de formatos alternativos. Por exemplo, podemos querer modificar o layout de uma atividade quando está em um tablet comparado a quando rodando em um aparelho celular.

No entanto, se o aplicativo não funcionar bem quando redimensionado para caber em tamanhos de tela diferentes, podemos usar os atributos do elemento `<supports-screens>` para controlar se deve ser distribuído para telas menores ou ter sua interface ampliada ("zoom") para caber em telas maiores, usando o modo de compatibilidade do sistema. Quando não projetamos o aplicativo para tamanhos de tela maior e o redimensionamento normal não alcança os resultados adequados, o modo de compatibilidade de tela irá escalar sua interface emulando uma tela de tamanho normal e média densidade, em seguida, ampliando de forma a preencher toda a tela. A seguir, o formato do elemento `<supports-screens>`.

```
<supports-screens android:resizeable=["true" | "false"]
    android:smallScreens=["true" | "false"]
    android:normalScreens=["true" | "false"]
    android:largeScreens=["true" | "false"]
    android:xlargeScreens=["true" | "false"]
    android:anyDensity=["true" | "false"]
    android:requiresSmallestWidthDp="integer"
    android:compatibleWidthLimitDp="integer"
    android:largestWidthLimitDp="integer"/>
```

O Android pode ser executado em uma variedade de dispositivos que oferecem diferentes tamanhos de tela e densidades. Para os aplicativos, o sistema Android fornece um ambiente de desenvolvimento consistente em todos os dispositivos e lida com a maior parte do trabalho para ajustar cada interface de usuário do aplicativo para a tela em que é exibido. Ao mesmo tempo, o sistema fornece APIs que permitem o controle da interface do usuário do aplicativo para tamanhos de tela e densidades específicos, a fim de aperfeiçoar o design da interface do usuário para configurações de telas diferentes. Por exemplo, podemos querer uma interface para tablets, que é diferente da interface do usuário para telefones celulares.

Embora o sistema execute dimensionamento e redimensionamento para fazer o aplicativo de trabalho em telas diferentes, devemos fazer um esforço para otimizá-lo para diferentes tamanhos e densidades de tela. Ao fazer isso, maximizamos a experiência do usuário para todos os dispositivos, fazendo com que acreditem que sua escolha foi realmente feita para os seus dispositivos em vez de simplesmente esticada para caber na tela de seus dispositivos.

Visualizando o suporte de telas

Agora daremos uma visão geral do suporte para telas múltiplas do Android, incluindo: uma introdução aos termos e conceitos usados na API, um resumo das configurações de tela que o sistema suporta, com uma visão geral da API e características de compatibilidade de tela subjacente.

Termos e conceitos

Tamanho da tela, tamanho físico real, medida diagonal da tela e para simplificar o agrupamento de todos os tamanhos de telas Android, elas foram divididas em quatro tipos: pequeno, normal, grande e extra grande.

Densidade tela

A quantidade de pixels dentro de uma área física da tela, geralmente dada em dpi (pontos por polegada). Por exemplo, uma tela de densidade "baixa" tem menos pixels dentro de uma determinada área física, em comparação com uma tela de densidade "normal" ou "elevada", para simplificar, os grupos de todas as densidades de tela do Android, elas foram divididas em quatro: alta, baixa, média e extra.

Orientação

A orientação da tela a partir do ponto de vista do utilizador. Esta é paisagem ou retrato, o que significa que a proporção de tela de aspecto é mais larga ou alta, respectivamente.

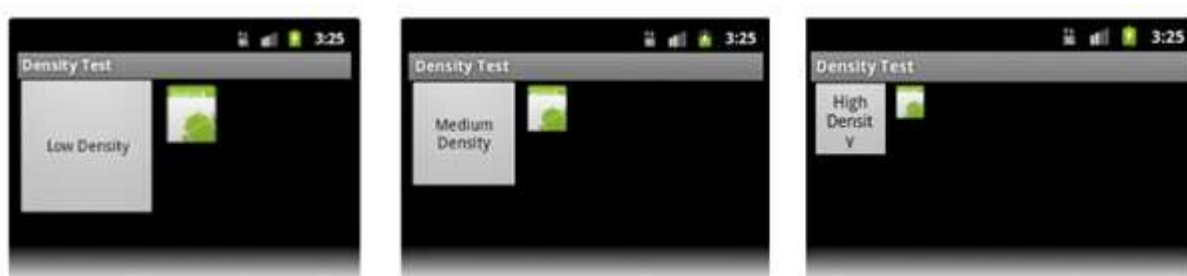
Resolução

É o número total de pixels físicos sobre uma tela. Ao adicionar o suporte para múltiplas telas, as aplicações não trabalham diretamente com a resolução, portanto, deve-se estar preocupado apenas com tamanho da tela e densidade, como especificado por tamanho generalizado e grupos de densidade.

Independência densidade

O aplicativo realiza a "independência densidade" quando se preserva o tamanho físico (do ponto de vista do usuário) de elementos de interface de usuário ao exibir telas com diferentes densidades.

Manter a independência da densidade é importante porque, sem ela, um elemento de UI (por exemplo, um botão) aparece fisicamente maior em uma tela de baixa densidade e menor numa tela de alta densidade. Tais mudanças de densidade relacionadas ao tamanho podem causar problemas no layout da aplicação e sua usabilidade. As figuras a seguir mostram as diferenças entre um aplicativo quando não fornece independência densidade e quando isso acontece, respectivamente.



Legenda: Exemplo de aplicação sem suporte para diferentes densidades, como mostrado em: baixo, médio e telas de alta densidade.



Legenda: Exemplo de aplicação com bom suporte para diferentes densidades (é independente da densidade), como mostrado em: baixo, telas de densidade média e alta.

Usando qualificadores de configuração

O Android suporta diversas configurações que permitem controlar a forma como o sistema seleciona seus recursos alternativos com base nas características da tela do dispositivo atual. Um qualificador de configuração é uma string que podemos acrescentar a um diretório de recursos no projeto Android e especificar as configurações para quais recursos são projetados.

Para utilizar um qualificador de configuração:

- Crie um novo diretório no res do seu projeto/diretório usando o formato <resources_name>-<qualifier>, no qual <resources_name> é o nome do recurso padrão (como drawable ou layout) e <qualifier> é um qualificador de configuração, conforme a tabela a seguir, especificando a configuração da tela para a qual estes recursos forem utilizados (tais como HDPI ou xlarge).

Característica	Qualifier	Característica
Size	Small	Recursos para telas de tamanho pequeno
	Normal	Recursos para telas de tamanho normal.
	Large	Recursos para telas de tamanho grande
	xlarge	Recursos para telas de tamanho extra grande.
Density	ldpi	Recursos de baixa densidade (~120 dpi).
	mdpi	Recursos de média densidade (~160dpi).
	hdpi	Recursos de alta densidade (~240 dpi).
	xhdpi	Recursos de densidade extra alta (~320dpi).
	nodpi	Recursos para todas as densidades. Estes recursos são independentes da densidade.

	tvdpi	Recursos para telas com densidade em entre mdpi e hdpi; cerca de 213 dpi.
Orientation	Land	Orientação de telas no sentido horizontal. (relação de aspecto de largura).
	Port	Orientação de telas no sentido vertical (relação de aspecto de altura).

Próxima aula

Para concluirmos os conhecimentos agregados ao sistema Android, ainda precisamos adquirir mais uma informação importante. Em nossa próxima aula – **Aula 19** –, trataremos dos aplicativos Android que necessitam realizar operações em conjunto com a web.

Agora acesse o AVA e desenvolva os exercícios propostos.



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.

REFERÊNCIAS

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

ROGERS, Rick *et al.* *Desenvolvimento de Aplicações Android*. São Paulo: Editora Novatec, 2009.