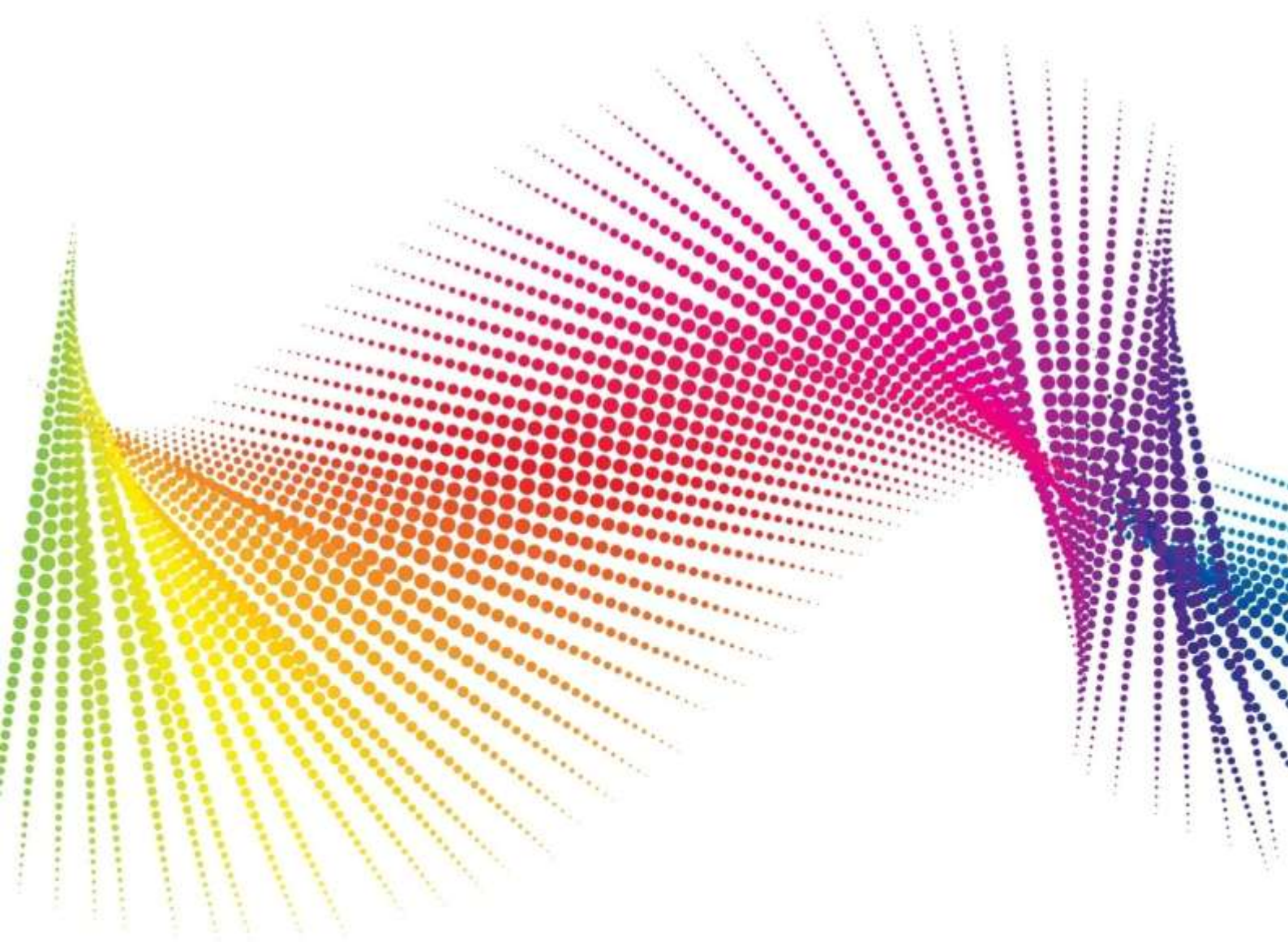


Computação Móvel

Aula 17



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 17: Mídias em android: gráficos, áudio, vídeo e mapas

Objetivo: Introduzir a utilização de recursos de mídia, como gráficos, arquivos de áudio e vídeo e mapas no enriquecimento de aplicações android.

Animação no android

O sistema de animação é propriedade de uma estrutura robusta que permite animar praticamente qualquer coisa. Pode-se definir uma animação para alterar qualquer propriedade do objeto ao longo do tempo, independentemente se ele chama a tela ou não.

O sistema de animação permite que se definam as seguintes características de uma animação:

- **Duração:** pode-se especificar a duração de uma animação. O comprimento padrão é de 300 ms.
- **Interpolação tempo:** pode-se especificar como os valores da propriedade são calculados em função do momento atual da animação decorrido.
- **Repetir contagem e comportamento:** pode-se especificar se quer ou não ter uma repetição de animação quando se chega ao fim de uma duração e quantas vezes ela repetir. Também pode-se especificar se deseja que a animação reproduza em sentido inverso. Defini-lo para reverter reproduz a animação para a frente depois para trás várias vezes, até que o número de repetições é atingido.
- **Animador:** pode haver animações em conjuntos lógicos que jogam juntos ou sequencialmente ou após atrasos especificados.
- **Quadro de atualização:** pode-se especificar a frequência para atualizar quadros de sua animação. O padrão é definido para atualizar a cada 10 ms, mas a velocidade em que seu aplicativo pode atualizar quadros, o que em última análise, depende de como o sistema está ocupado.

Visualizando uma animação

Podemos usar o sistema de animação vista para executar animação interpolada em exibições. O Tween Animation calcula a animação com informações como o ponto de início, ponto final, tamanho, rotação, e outros aspectos comuns de uma animação.

A animação de interpolação pode executar uma série de transformações simples (posição, tamanho, rotação e transparência) sobre o conteúdo de um objeto View. Então, se temos um objeto TextView, pode-se mover, girar, aumentar ou reduzir o texto. Se ele tem uma imagem de fundo, ela será transformada com o texto. O pacote de animação fornece todas as classes usadas em uma animação de interpolação.

Uma sequência de instruções de animação define a animação por XML ou código do android. Tal como acontece com a definição de um layout, um arquivo XML é recomendado porque é mais legível, reutilizável e flexível que embutir a animação. No exemplo a seguir usaremos XML.

As instruções de animação para definir as transformações que se deseja que ocorra, quando eles ocorrem, e quanto tempo eles devem esperar para aplicar. As transformações podem ser sequencial ou simultânea – por exemplo, podemos ter o conteúdo de um movimento TextView da esquerda para a direita, e depois girar 180 graus, ou podemos ter o movimento de texto e rodar simultaneamente.

Cada transformação é um conjunto de parâmetros específicos para a transformação (tamanho inicial e final de tamanho para mudança de tamanho, o ângulo inicial e final para o ângulo de rotação, e assim por diante), e também um conjunto de parâmetros comuns (por exemplo, hora de início e duração). Para fazer com que várias transformações ocorram em simultâneo, é necessário dar a mesma hora de início para tornar sequencial o cálculo do tempo de início mais o tempo de transformação anterior.

A animação arquivo XML pertence no diretório res/anim/ de seu projeto android. O arquivo deve ter um único elemento raiz: o qual será único <alpha>, <scale>, <translate>, <rotate>, interpolador de elemento, ou elemento <set> que mantém grupos desses elementos. Por padrão, todas as instruções de animação são aplicadas simultaneamente. Para ocorrer em sequência, deve-se especificar o atributo startOffset.

O XML a seguir de um dos ApiDemos é usado para alongar, então, simultaneamente girar e girar um objeto View, exemplo 1.

Exemplo 1

```
<set android:shareInterpolator="false">
```

```
  <scale
```

```
    android:interpolator="@android:anim/accelerate_decelerate_interpolator"
```

```
    android:fromXScale="1.0"
```

```
    android:toXScale="1.4"
```

```
    android:fromYScale="1.0"
```

```
    android:toYScale="0.6"
```

```
    android:pivotX="50%"
```

```
    android:pivotY="50%"
```

```
    android:fillAfter="false"
```

```
    android:duration="700" />
```

```
<set android:interpolator="@android:anim/decelerate_interpolator">
```

```
  <scale
```

```
    android:fromXScale="1.4"
```

```
    android:toXScale="0.0"
```

```
    android:fromYScale="0.6"
```

```
    android:toYScale="0.0"
```

```
    android:pivotX="50%"
```

```
    android:pivotY="50%"
```

```
    android:startOffset="700"
```

```
    android:duration="400"
```

```
    android:fillBefore="false" />
```

```
  <rotate
```

```
    android:fromDegrees="0"
```

```
        android:toDegrees="-45"  
        android:toYScale="0.0"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:startOffset="700"  
        android:duration="400" />  
  
</set>  
  
</set>
```

Exibindo uma mídia

O android framework multimídia inclui suporte para reprodução de diversos tipos de mídia comuns, de modo que podemos facilmente integrar vídeo, áudio e imagens em suas aplicações. Podemos reproduzir áudio ou vídeo a partir de arquivos de mídia armazenados em recursos do seu aplicativo (recursos-primos), a partir de arquivos autônomos no sistema de arquivos, ou a partir de um fluxo de dados que chega através de uma conexão de rede, todas as APIs usando MediaPlayer.

Mostraremos como escrever um aplicativo para exibir uma mídia que interage com o usuário e o sistema, a fim de obter um bom desempenho e uma experiência de usuário agradável.

Básico

As seguintes classes são utilizadas para reproduzir som e vídeo no sistema do android:

- MediaPlayer: é a API primária para reproduzir som e vídeo.
- AudioManager: gerencia as fontes de áudio e saída de áudio em um dispositivo.

Declarações de manifesto

Antes de iniciar o desenvolvimento de um aplicativo usando MediaPlayer, verifique se o arquivo de manifesto tem as declarações apropriadas para permitir o uso de recursos relacionados.

Permissão internet: se estivermos usando MediaPlayer para fluxo de rede baseada em conteúdo, o aplicativo deve solicitar acesso à rede.

```
<uses-permission android:name="android.permission.INTERNET" />
```

Permissão de visualização é quando o aplicativo precisa apagar a tela ou o processador hibernar. Usamos os métodos MediaPlayer.setScreenOnWhilePlaying() ou MediaPlayer.setWakeMode() e devemos solicitar essa permissão.

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

Usando MediaPlayer

Um dos componentes mais importantes do quadro de mídia é a classe MediaPlayer. Um objeto desta classe pode buscar, decodificar e reproduzir áudio e vídeo com configuração mínima. Ele suporta várias fontes diferentes de mídia, tais como: recursos locais, URIs internos, como um que se possa obter de um provedor de conteúdos e URLs externos (streaming).

Aqui está um exemplo de como reproduzir o áudio que está disponível como um recurso primo local (salvo na aplicação em res/raw/directory):

```
MediaPlayer mediaPlayer = MediaPlayer.create (contexto,  
R.raw.sound_file_1);  
mediaPlayer.start ();
```

E aqui está como executar uma mídia a partir de um URI disponível localmente no sistema:

```
Uri myUri = .... // inicializar Uri aqui
MediaPlayer mediaPlayer = new MediaPlayer ();
mediaPlayer.setAudioStreamType (AudioManager.STREAM_MUSIC);
mediaPlayer.setDataSource (getApplicationContext (), myUri);
mediaPlayer.prepare ();
MediaPlayer.start ();
```

Para se reproduzir a partir de uma URL remota via HTTP streaming:

```
String url = "http:// ....."; // sua URL aqui
MediaPlayer mediaPlayer = new MediaPlayer ();
mediaPlayer.setAudioStreamType (AudioManager.STREAM_MUSIC);
mediaPlayer.setDataSource (url);
mediaPlayer.prepare ()
MediaPlayer.start ();
```

Localização, mapas e mapas de localização

- **Serviços de localização**

Aplicativos de localização baseados em mapas oferecem uma experiência fascinante em dispositivos móveis. Podemos construir esses recursos usando as classes do pacote `android.location` e o Google Maps API `android`.

O `android` permite o acesso de aplicações aos serviços de localização suportados pelo dispositivo por meio de classes do pacote `android.location`. O componente central da estrutura local é o serviço de sistema `LocationManager`, que oferece APIs para determinar a localização e o rumo do dispositivo subjacente (se disponível).

Tal como acontece com outros serviços do sistema, não é possível instanciar um `LocationManager` diretamente. Em vez disso, deve-se solicitar uma instância do sistema chamando `getSystemService (Context.LOCATION_SERVICE)`. O método retorna um identificador para uma instância `LocationManager`.

Uma vez que sua aplicação tem uma `LocationManager`, o aplicativo é capaz de fazer três coisas:

- Consultar a lista de todos os LocationProviders para o local último usuário conhecido.
- Registrar/cancelar o registro para atualizações periódicas de localização atual do usuário a partir de um provedor de localização (especificado quer por critérios ou nome).
- Registrar/cancelar o registro para uma determinada localização se o dispositivo estiver nas proximidades (especificado por raio em metros) de uma dada latitude/ longitude.

Com o Google Maps API android, pode-se adicionar mapas no aplicativo que são baseados em dados do Google Maps. A API lida automaticamente e possui acesso aos servidores do Google Maps, os dados de download, visualização do mapa e gestos de toque no mapa.

A classe de chave no Google Maps API android é MapView. A MapView exibe um mapa com os dados obtidos a partir do serviço do Google Maps. Quando o MapView tem foco, ele irá capturar teclas pressionadas e gestos de toque para deslocar e fazer zoom no mapa automaticamente, incluindo manipulação de solicitações de rede para mapas adicionais. Ele também fornece todos os elementos da interface do usuário necessários para que eles controlem o mapa. O aplicativo também pode usar métodos de classe MapView para controlar o mapa de programação e desenhar uma série de sobreposições na parte superior do mapa.

O Google Maps Android APIs não está incluído na plataforma Android, mas está disponível em qualquer dispositivo com a Loja Google Play.

Para integrar o Google Maps em um aplicativo, será preciso instalar os serviços do Google Play no android SDK.

Exercício

Para termos certeza de que entendemos a aula, vamos realizar uma atividade sobre os diversos tipos de mídias que podem ser manipulados pelos sistemas android.

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.

Próxima aula

Agora que possuímos conhecimento sobre o funcionamento do sistemas android deveriam nos passar uma pergunta pela cabeça, “Isso funcionara em todos os dispositivos do mercado?”. Este é o assunto da próxima aula – **Aula 18** – em que aprenderemos sobre as compatibilidades possíveis com o sistema android.

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.