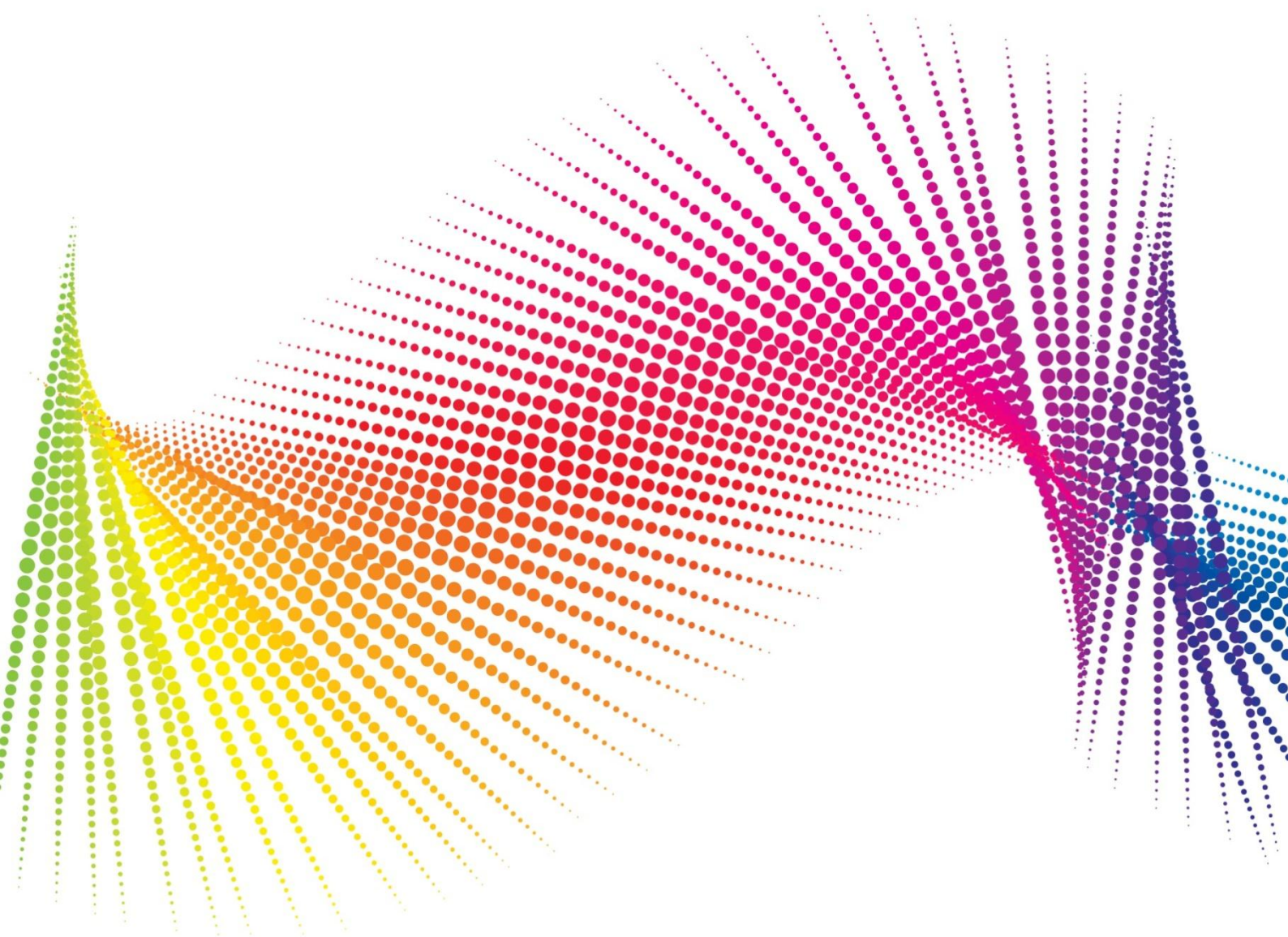


Computação Móvel

Aula 06



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 06: Ambiente de Programação IDE Eclipse, Ferramentas e Testes do Aplicativo Móvel

Objetivo: Demonstrar o ambiente integrado de desenvolvimento fornecido pelo eclipse para realização da programação de aplicativos móveis, as ferramentas presentes neste ambiente que facilitam a visualização do aplicativo, bem como o software emulador que permite que testes sejam feitos sem que haja a necessidade de um equipamento físico.

Componentes do eclipse

O Eclipse provém um ambiente que permite a criação de tipos específicos de projetos, o que inclui projetos do tipo Java. Um plug-in do tipo ADT adiciona ao eclipse a habilidade de criar e utilizar projetos do tipo Android. Deste modo, ao criar um novo projeto Android, o ADT adiciona a hierarquia de arquivos no projeto e todos os arquivos necessários para um projeto Android.

O ADT tem diversos componentes que podem ser utilizados separadamente, apesar de um “plug-in” dar a ideia de ser somente uma pequena modificação. Ele é composto de uma quantidade considerável de funções. Para entendermos melhor a estrutura e o funcionamento do ADT, a seguir, serão apresentadas as funcionalidades de cada uma de suas partes e como elas serão utilizadas no Eclipse no desenvolvimento de aplicações para Android.

Android Layout Editor

A criação de layouts para interfaces de usuário em aplicativos Android podem ser feitos utilizando-se o XML. Com o ADT será adiciona um editor que proporcionara uma visualização do layout proposto, o que auxiliará na composição e da aplicação para o Android. Ao abrir um arquivo de layout, o ADT iniciará automaticamente o editor que comporá a visualização do arquivo.

Nas versões anteriores do SDK do Android, o editor de layout possuía limitações quando se utilizava esta funcionalidade. Atualmente, entretanto, deve-se considerar a ferramenta de visualização de layouts para o Android como sendo a melhor maneira de agilizar o desenvolvimento de uma aplicação, sendo que com a automação da especificação dos layouts, aumenta significativamente a possibilidade de que os layouts gerados funcionem na maior parte dos dispositivos Android.

Android Manifest Editor

Todos os projetos Android devem possuir um arquivo de manifesto, que acompanhará a aplicação e os recursos que estarão contidos no projeto quando ele é desenvolvido e que informa ao Android como instalar e utilizar o software. Ele está em XML e um editor desta extensão especializado é fornecido pelo ADT com a finalidade de auxiliar desenvolvedores nesta tarefa.

Editores XML para outros arquivos XML do Android

Existem outros arquivos XML em um projeto Android que contêm informações como montagem de menus, recursos de strings, ou que armazenam recursos gráficos para o aplicativo; existem editores específicos que inicializam sempre que estes tipos de arquivos forem abertos.

Compilação de aplicativos Android

Sempre que solicitamos a execução de um Projeto no Eclipse, ele é compilado automaticamente. Em razão disso, não será possível encontrar um local individual que possua a função de transformar o código-fonte e seus respectivos recursos em um programa executável ou, até mesmo, um instalador. Ao invés disso, o Android necessita de passos específicos para gerar a compilação de um arquivo que possa ser instalado em um emulador ou dispositivo móvel, sendo que o ADT fornece o software responsável pela execução desses passos. Como resultado de uma compilação de um projeto Android, será gerado um arquivo do tipo .apk, em que se encontrará na subpasta bin na hierarquia de arquivos do projeto na sua área de trabalho do Eclipse.

Compiladores específicos do Android são fornecidos com o ADT, permitindo que se utilize Java como a linguagem para geração de aplicativos para o Android, ao mesmo tempo, executa esses aplicativos numa máquina virtual Dalvik que em contrapartida executa seus próprios bytecodes.

Execução e depuração de aplicativos Android

Quando se executa ou se depura uma aplicação Android no Eclipse, o arquivo .apk dessa aplicação será invocado e inicializado em um AVD ou dispositivo Android. Durante este processo, será utilizado o ADB e o DDMS para comunicação com o AVD ou o dispositivo e, também, com o ambiente de execução da máquina virtual Dalvik que executa o código da aplicação. O ADT adiciona um conjunto de funções que permitirão ao Eclipse realizar essa execução.

Gerenciador de AVD e SDK

Um QEMU é um emulador de propósito geral. Enquanto o SDK do Android permite que se configure o QEMU, o que constitui uma funcionalidade interessante quando se criam emuladores que tem por objetivo executar aplicações para Android. Os gerenciadores do AVD e do SDK fornecem a interface do usuário, permitindo obter o controle do dispositivo virtual com base no QEMU.

Hierarchy Viewer

O visualizador de hierarquias mostra, permitindo que se analise a hierarquia da atividade atual ou de um dispositivo selecionado. Isso permite que se encontre e diagnostique problemas na hierarquia de suas visualizações, mesmo quando o aplicativo está sendo executado; também permite que se visualize uma apresentação ampliada da tela da aplicação, com orientações de alinhamento que permitem identificar problemas em layouts.

Layoutopt

Layoutopt é uma ferramenta que analisa os arquivos de layout XML e que pode diagnosticar problemas nos layouts do Android.

Monkey

Monkey é um programa que pode ser executado, tanto em um emulador como em um dispositivo, e gerando um fluxo pseudoaleatório de eventos do usuário, como cliques, toques, ou gestos, assim como uma série de eventos a respeito do sistema. Você pode usar o Monkey para realizar testes de estresse em aplicativos que estão em desenvolvimento, de uma maneira aleatória, ainda repetível.

Visão global

O Monkey inclui uma série de opções que se dividem em quatro categorias principais:

- Opções de configuração básica, como a definição do número de eventos que devem ser testados e a quantidade de tentativas para cada evento.

- Constrangimentos operacionais, como a restrição do teste a um único pacote.

- Tipos de eventos e frequências.

- Opções de depuração.

Quando o Monkey for executado, serão gerados eventos a serem enviados ao sistema. Feito isso, ele observará o comportamento de três condições do sistema que estiver em teste:

- Se o Monkey estiver para ser executado em uma ou mais aplicações específicas, ele verificará, especialmente, as tentativas de navegar entre as aplicações e seus respectivos bloqueios.

- Caso a aplicação produza algum tipo de falha ou receba qualquer tipo de exceção não tratada, o Monkey parará e reportará a falha ou o erro encontrado.

- Dependendo do nível de detalhamento selecionado, serão mostrados relatórios sobre o progresso do Monkey e os eventos que foram gerados e testados por ele.

Uso básico do Monkey

Você pode iniciar o Monkey usando uma linha de comando na máquina de desenvolvimento ou um script, porque o Monkey pode ser executado no emulador ou no dispositivo móvel, deve-se inicializá-lo a partir de um shell no local em que se deseja executá-lo.

A sintaxe básica de inicialização é:

- `$ Adb shell monkey [options] <event-count>[opções] < evento - contagem >`

Sem nenhuma opção específica, o Monkey vai iniciar em modo não detalhado e irá enviar eventos para todas as aplicações instaladas no dispositivo onde estiver instalado. Aqui está uma linha de comando mais típica, que iniciará o pedido e enviará 500 eventos pseudoaleatórios a uma aplicação:

- `$ Adb shell monkey-p your.package.name-v 500- p seu . pacote . nome - v 500`

Resumo

Nesta aula, tivemos uma breve visualização do eclipse, uma ferramenta de desenvolvimento voltada para linguagens como Java e C, enfocando, especialmente, necessidades e facilidades que são produzidas no desenvolvimento de aplicativos para a plataforma Android. Demonstraram-se, também, os recursos de edição herdados do ADT, bem como a estrutura de testes provida pelo software Monkey.

Para termos certeza de que entendemos a aula, vamos realizar uma atividade sobre ambiente de programação IDE. Acesse o AVA e desenvolva os exercícios propostos.

Próxima aula

Agora que temos conhecimento sobre uma das maneiras de se editar um projeto da plataforma Android, estamos prontos para dar o próximo passo: aula 7, na qual começaremos a desvendar o processo de funcionamento e de programação de uma aplicativo da plataforma Android.

Vale a pena conferir

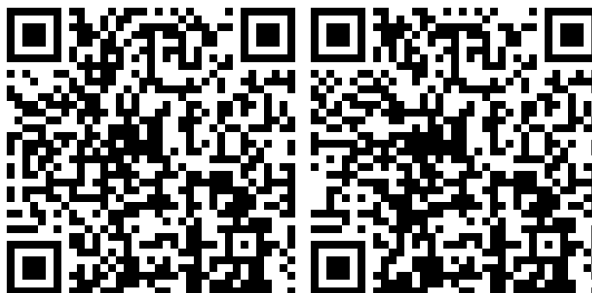
1. Para informações mais detalhadas sobre o Hierarchy Viewer acesse o seguinte endereço. Disponível em: <http://developer.android.com/guide/developing/tools/hierarchy-viewer.html>. Acesso em: 1º mar. 2013.
2. Para informações detalhadas sobre o Layoutopt, acesse o seguinte endereço. Disponível em: <http://developer.android.com/guide/developing/tools/layoutopt.html>. Acesso em: 1º mar. 2013.
3. Informações detalhadas sobre o Monkey podem ser encontradas no endereço a seguir. Disponível em: <http://developer.android.com/guide/developing/tools/monkey.html>. Acesso em: 1º mar. 2013.
4. Para informações sobre a Instalação do SDK do Android, acesse o link a seguir. Disponível em: <http://www.davidroid.com.br/site/?p=698>. Acesso em: 1º mar. 2013.



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*. 1. ed. São Paulo: Editora Novatec, 2009.