Android Development

Lecture AD 0
Android SDK & Development Environment

Lecture Summary

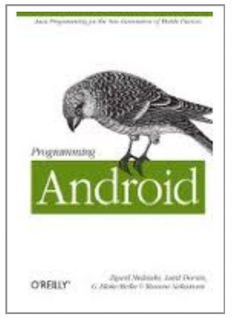
- Android Module Overview
- The Android Platform
- Android Environment Setup
  - SDK
  - Eclipse & ADT
  - SDK Manager
  - Android Emulator
  - ADT Additional Tools
- First Android Application

Android Module Overview

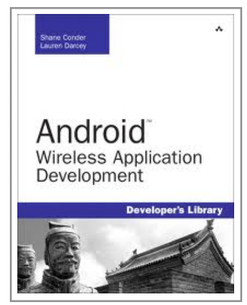
- Android Introduction & Dev. Environment Setup
- Android Platform
- Android Graphical User Interface 1
- Android Graphical User Interface 2
- Location and Mapping
- Data Persistence
- Asynchronous Tasks, Thread and Handlers
- Services
- Networking
- TBD (Bluetooth, Photo/Video, etc ...)

Book & Web References

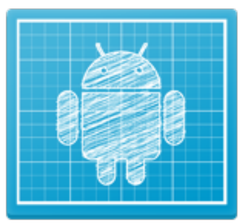
- Programming Android: Java Programming for the New Generation of Mobile Devices". Zigurd Mednieks, Laird Dornin, G. Blake Meike, and Masumi Nakamura. Pub. O'Reilly, 1st edition (Summer 2011) ISBN 9781449389697. **[PA] [Online Old Version - http://androidapps.org.ua/]**



- "Android Wireless Application Development". Lauren Darcey and Shane Conder. Addison Wesley, Second Edition (September 2011). ISBN 9780321743015. **[AWA]**



- Android Developers http://developer.android.com/index.html **[AD]**



- Android Developers Blog http://android-developers.blogspot.com/?hl=en (Updates and Tricks !)

The Android Platform

- Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the **Java** programming language.

- It is a Linux-based operating system for mobile devices such as smartphones and tablet computers. It is developed by the Open Handset Alliance led by Google.
- The Linux 2.6 kernel handles core system services and acts as hardware abstraction layer (HAL) between the physical hardware and the Android Software Stack.
- Kernel handles:
  - Application permissions and security
  - Low-level energy management
  - Process management and Threading
  - Networking
  - Display, keypad input, camera, Flash memory, audio and binder (IPC) driver access

Android SDK

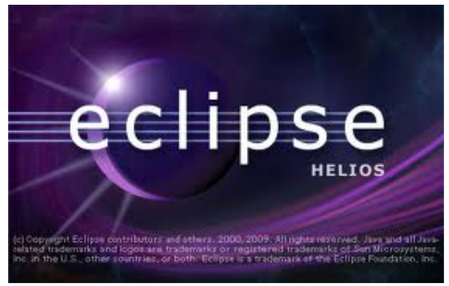
http://developer.android.com/sdk/index.html

The Android SDK has the tools, sample code, and docs needed to create Android Application.

Platform	SDK	ADT Tools (Eclipse + ADT)
Windows (32/64bit)	Android SDK r22.2.1	✅
Mac Os X (Intel 64bit)	Android SDK r22.2.1	✅
Linux (x86 - 32/64bit)	Android SDK r22.2.1	✅

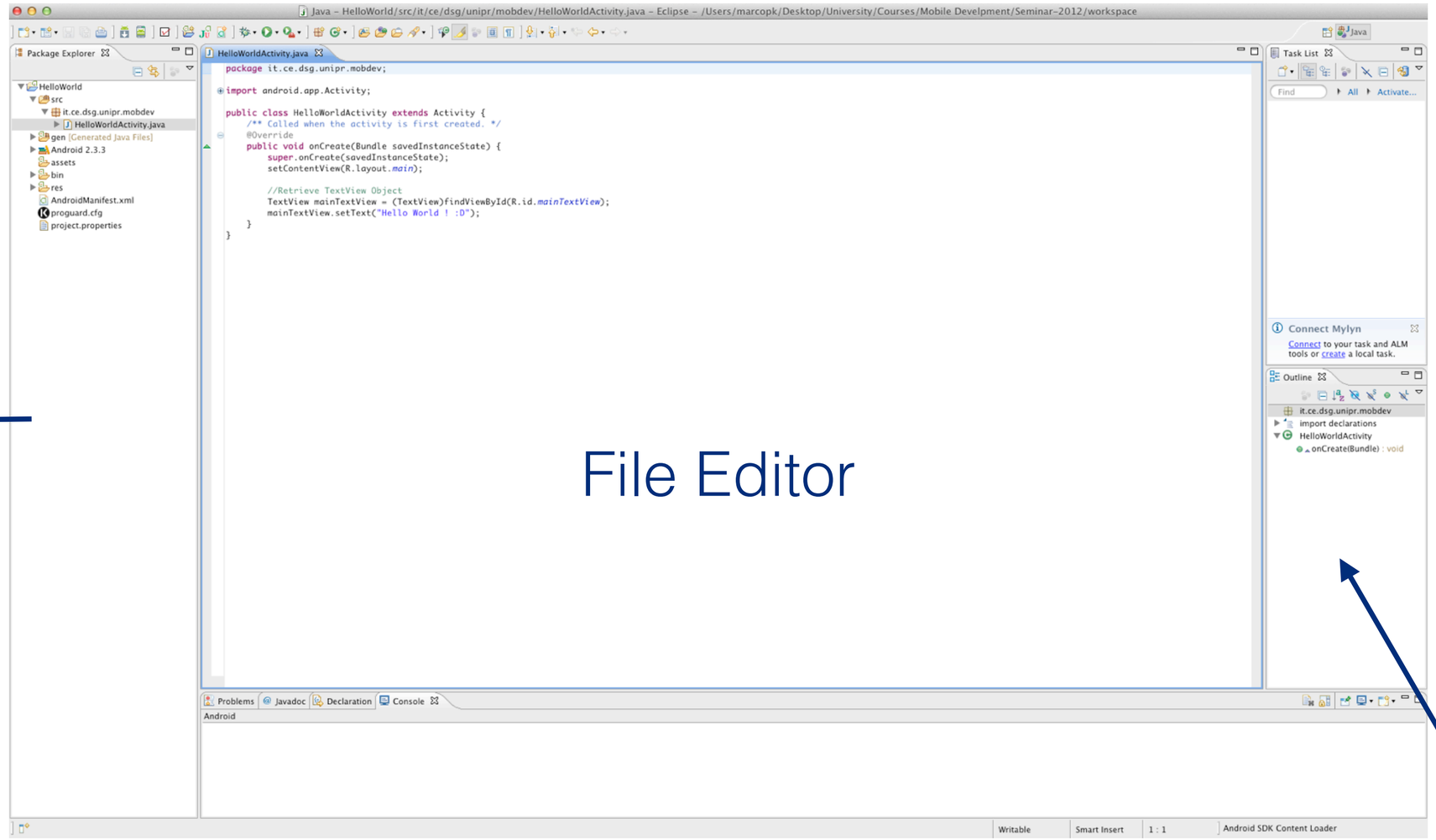
Prepare Your Development Computer

- Android SDK Supported OS:
  - Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7/8 (32- or 64-bit)
  - Mac OS X 10.5.8 or later (x86 only)
  - Linux
  - Java Development Kit (JDK)
  - Not compatible with Gnu Compiler for Java (gcj)
- Eclipse IDE (3.6 Helios o greater)
  - part of the Eclipse Platform, it is a multi-language software development environment based on extensible plug-in system.
- Android Development Tools (ADT):
  - A complete IDE for the Android development based on Eclipse
  - is a plugin for the Eclipse IDE



Eclipse/ADT IDE

Project and File Explorer



File Editor

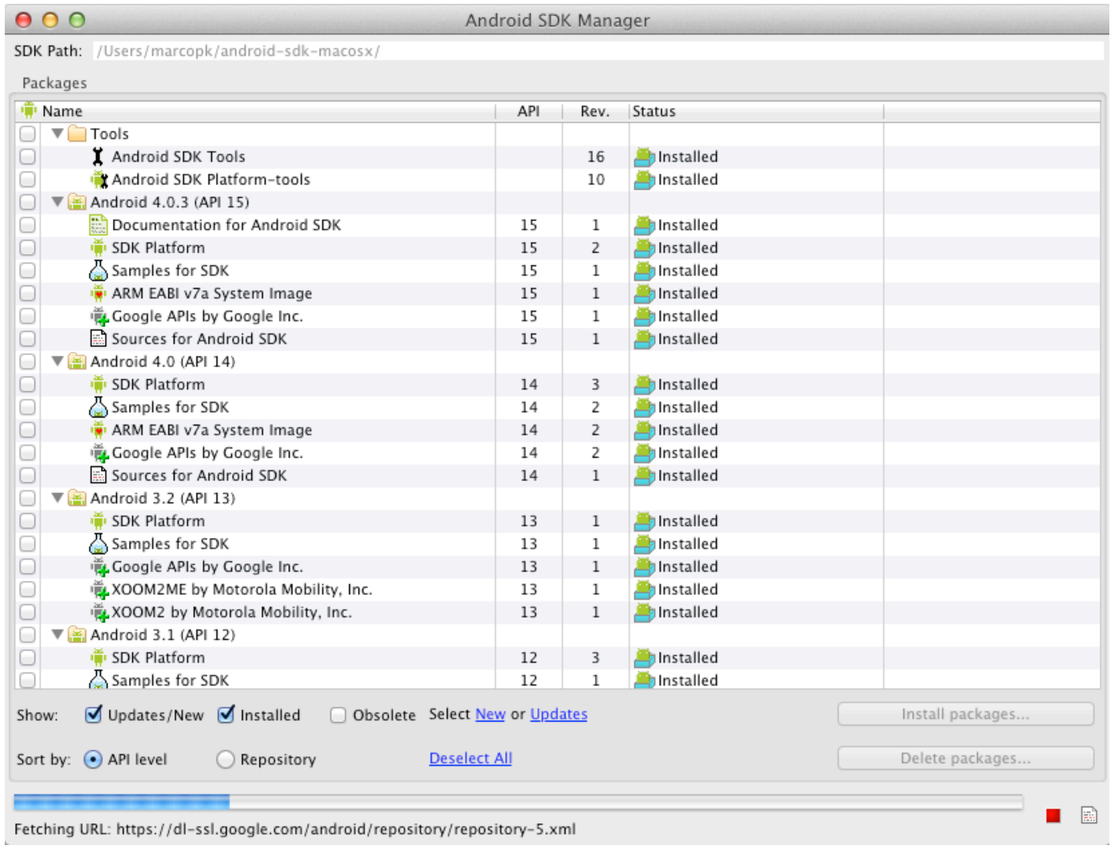
Console and additional Views

Android Developer Toolkit - ADT

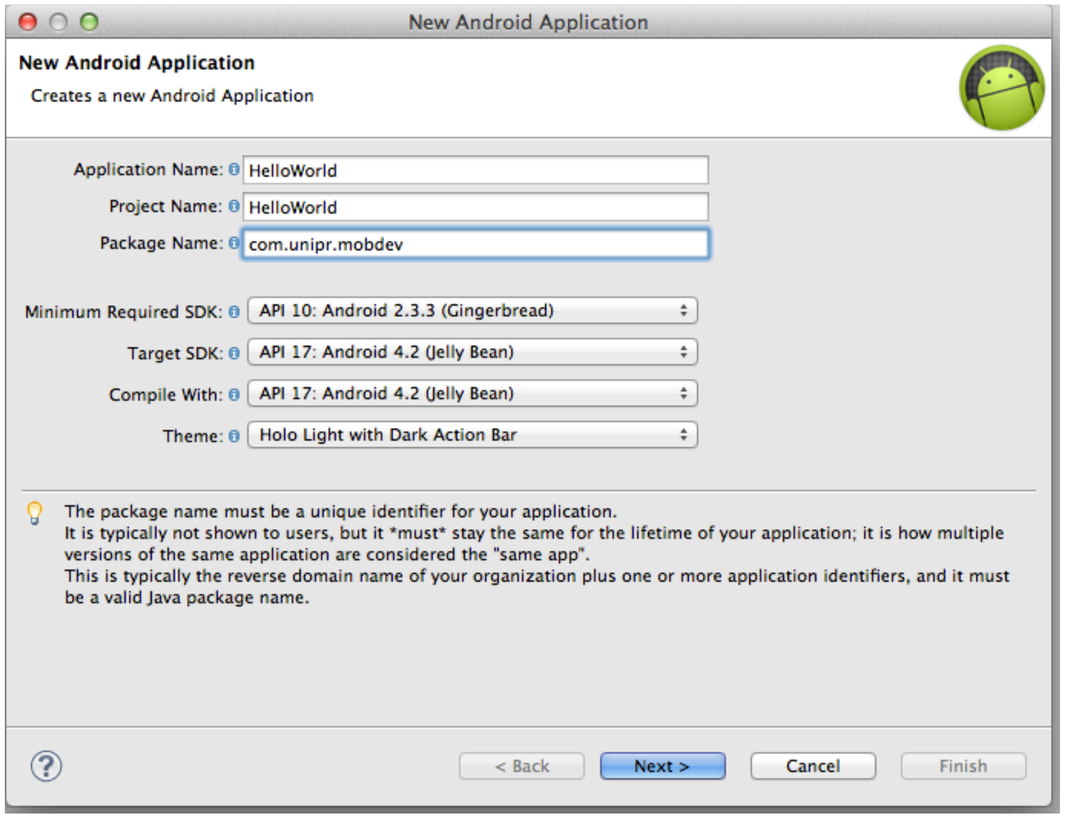
- Android Development Tools (ADT) is designed to give you a powerful, integrated environment in which to build Android applications.
- ADT extends the capabilities of Eclipse to let you:
  - quickly set up new Android projects
  - create an application UI
  - add components based on the Android Framework API
  - create Android emulator with multiple options and available configurations
  - test applications on real or emulated devices
  - debug your applications using the Android SDK tools
  - export signed (or unsigned) .apk files in order to distribute your application.

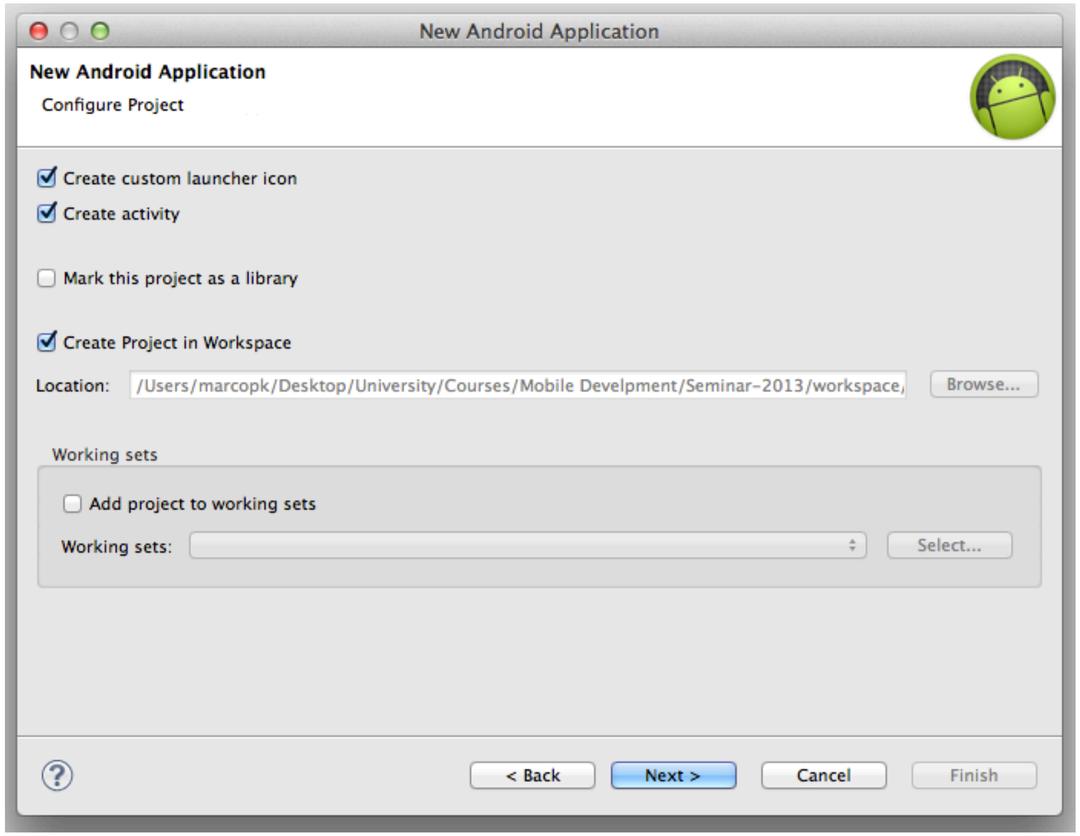
Android SDK Manager

- Before you can build an Android application, or even create a project you must install one ore more building target.
- By clicking on Android SDK Manager Button you can select platforms' components that you would like to install on you development machine.
- There are several packages related to different API levels.
- In our course we will use API level 4.x and 2.3.x in order to see latest platform release and compile developed application for available real devices.

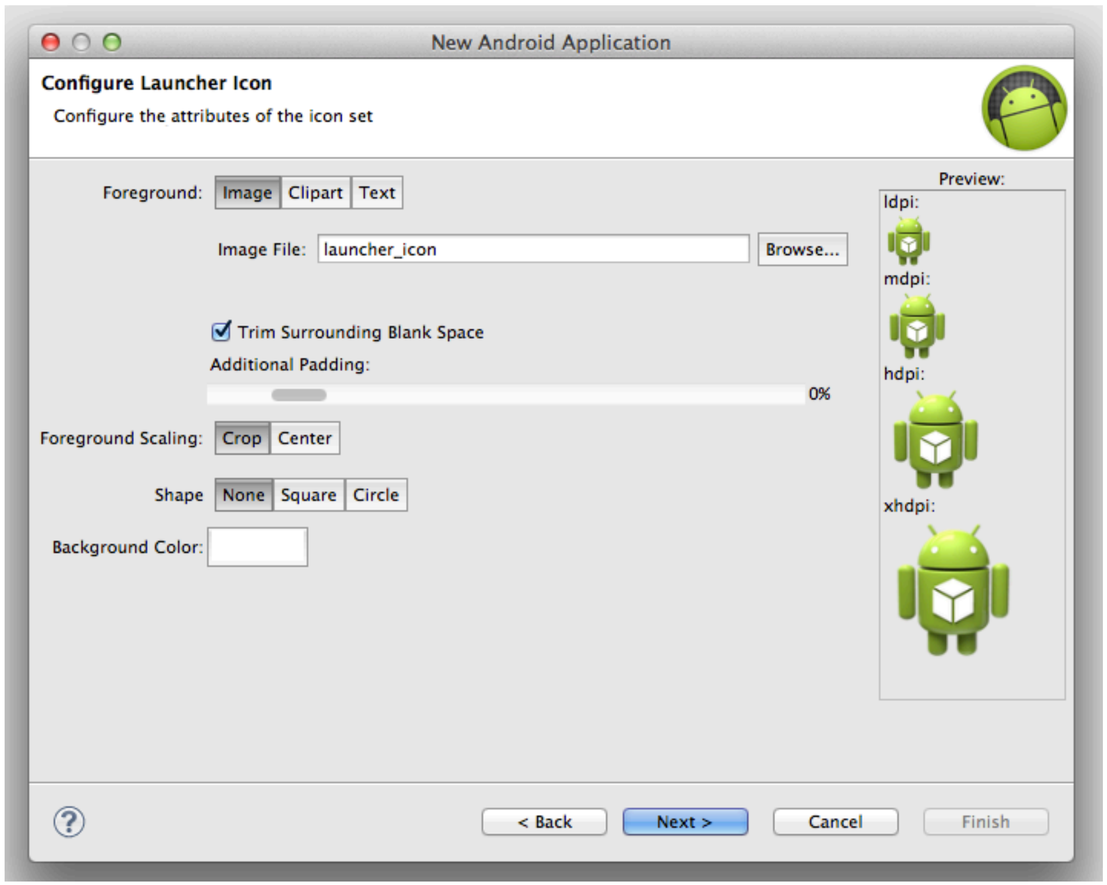


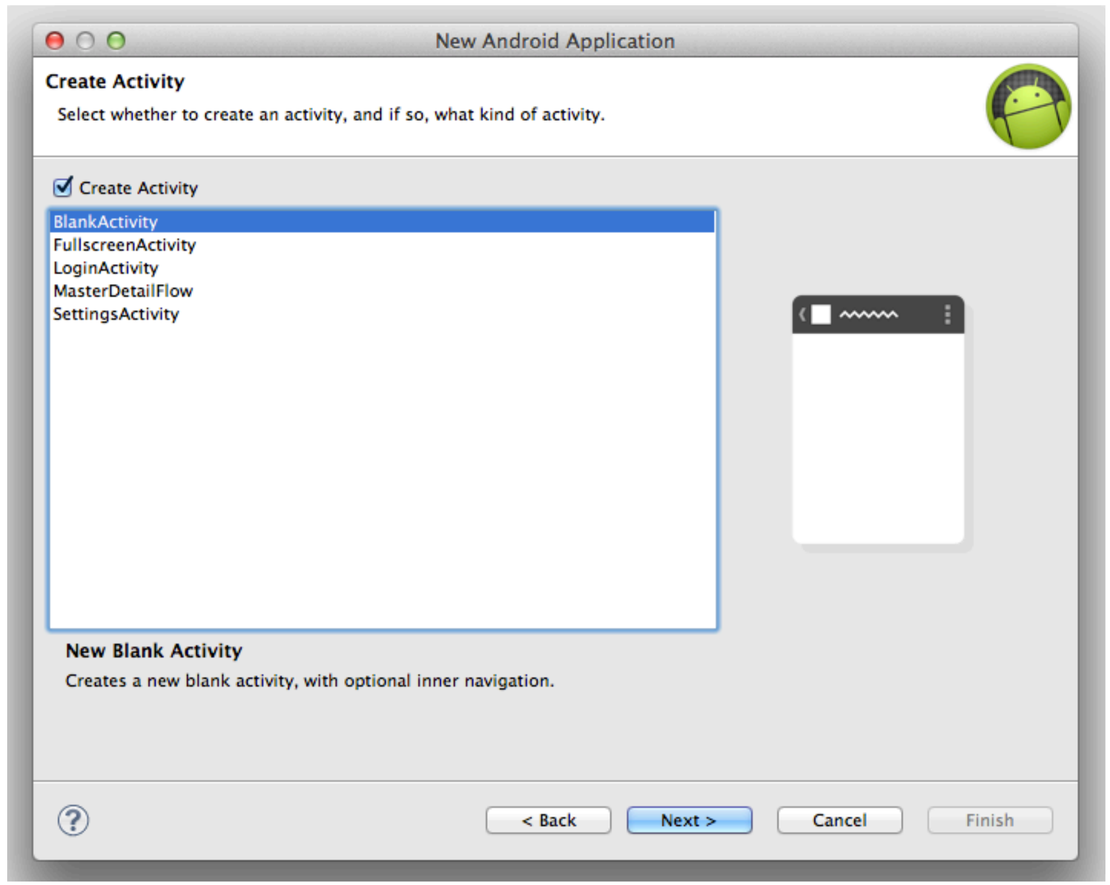
Hello World Application



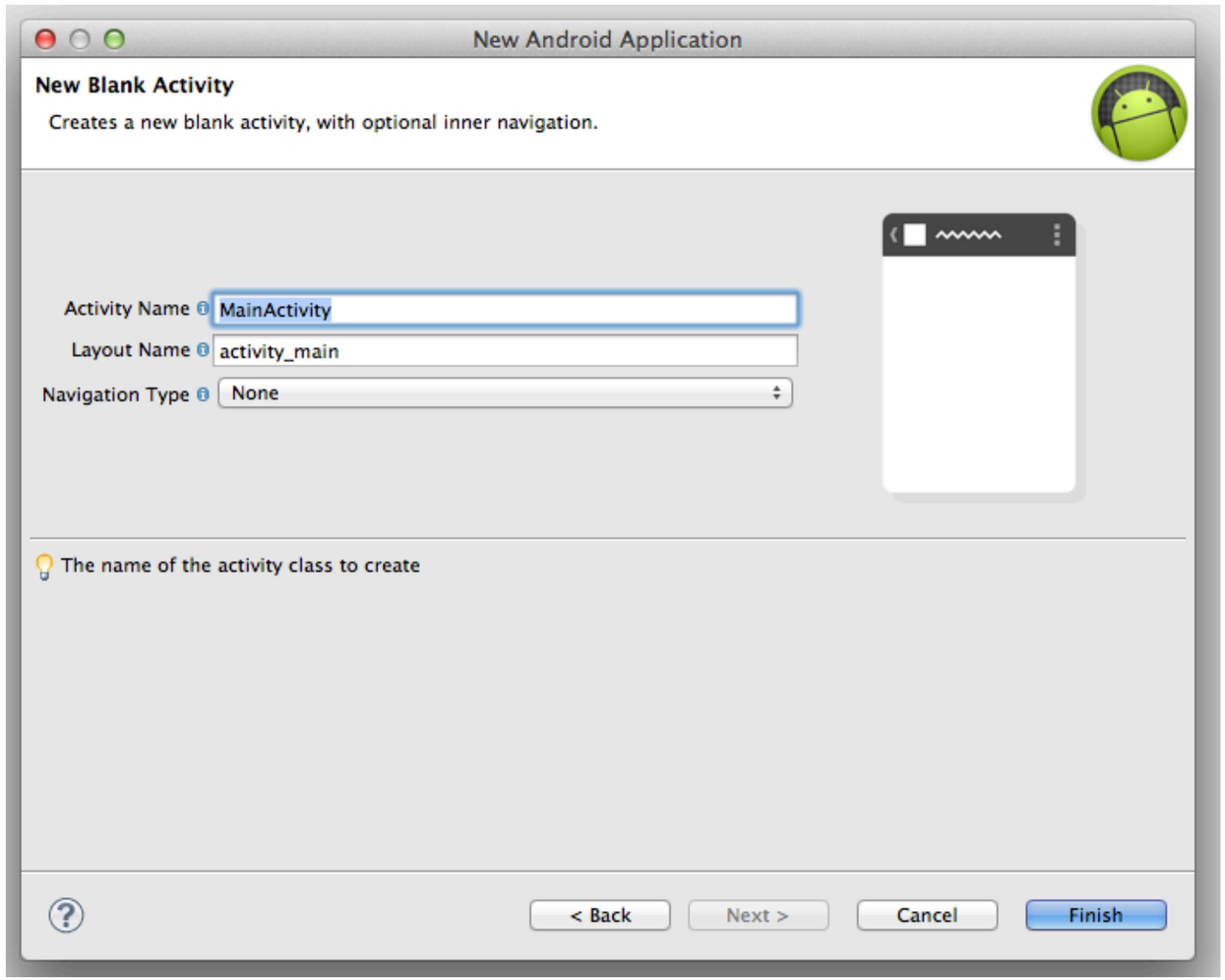


Hello World Application



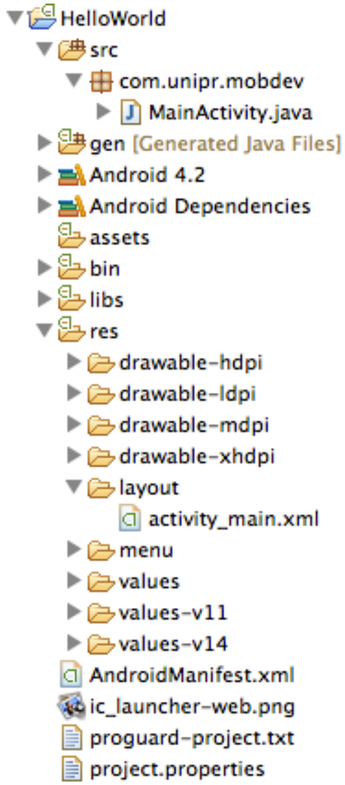


Hello World Application



Hello World Application

Package Explorer

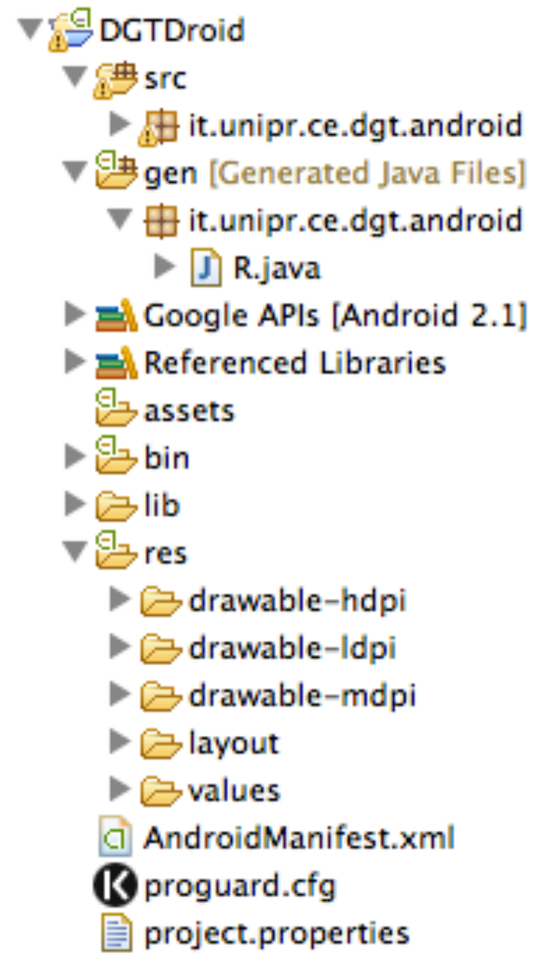


Package Explorer

```
package com.unipr.mobdev;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }

}
```

Android Application Structure

- *Android Manifest.xml*
- *res/*
  - *layout/* [application layout files]
  - *drawable/* [images, patches, drawable, xml]
  - *raw/* [data files that can be loaded as streams]
  - *values/* [xml files with strings, number values used in the code for example to localize the application in difference languages]
- *src/*
  - *java/package/directories*
- *gen/* [directory generated by Eclipse and Android SDK]



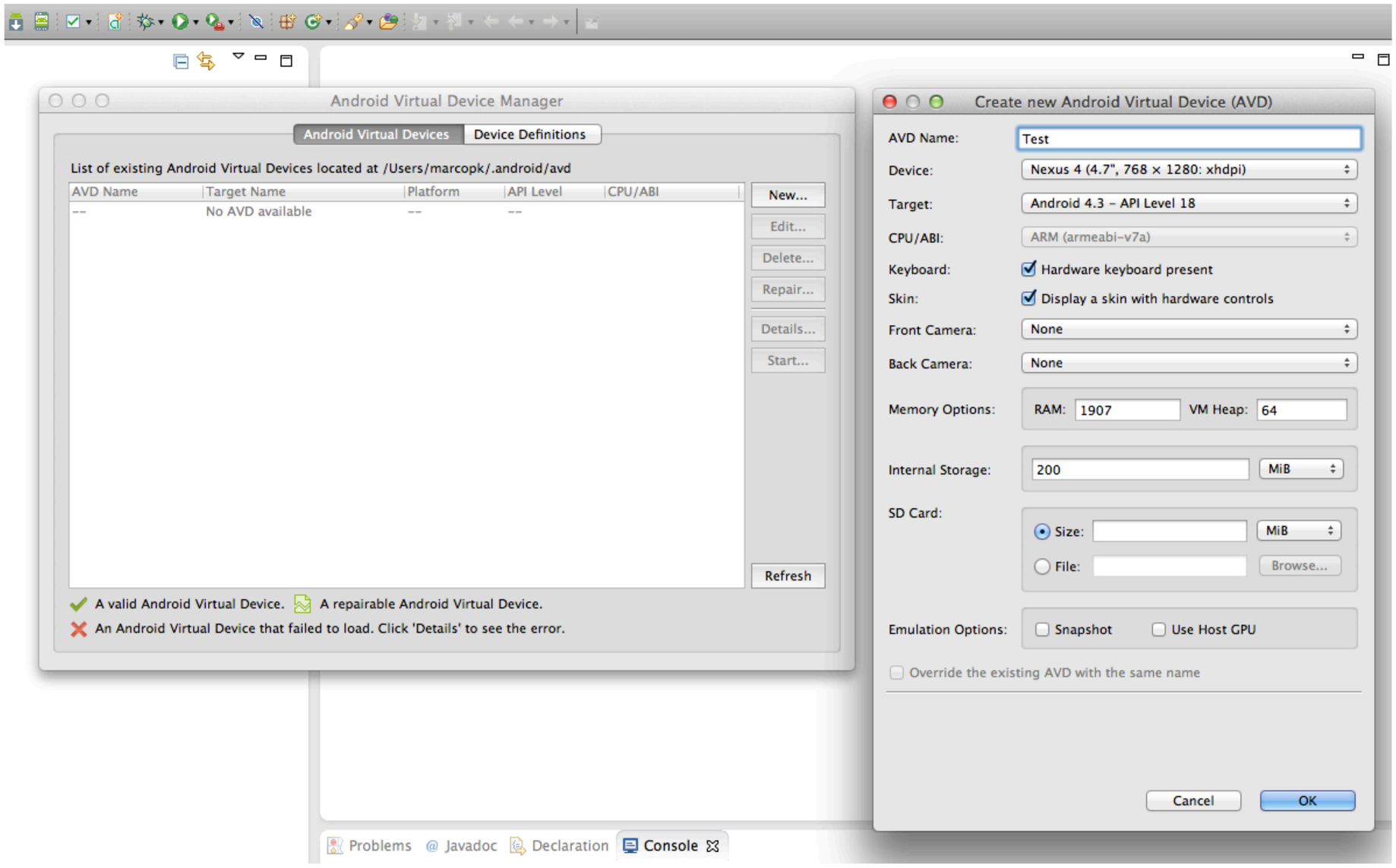
Android Emulator

- The Android SDK includes a virtual mobile device emulator that runs on your computer. The emulator lets you prototype, develop, and test Android applications without using a physical device.
- The Android emulator emulates all of the hardware and software features of a typical mobile device, except that it cannot place actual phone calls. It provides a variety of navigation and control keys, which you can "press" using your mouse or keyboard to generate events for your application. It also provides a screen in which your application is displayed, together with any other Android applications running.
- To let you model and test your application more easily, the emulator utilizes Android Virtual Device (AVD) configurations. AVDs let you define certain hardware aspects of your emulated phone and allow you to create many configurations to test many Android platforms and hardware permutations. Once your application is running on the emulator, it can use the services of the Android platform to invoke other applications, access the network, play audio and video, store and retrieve data, notify the user, and render graphical transitions and themes.
- The emulator also includes a variety of debug capabilities, such as a console from which you can log kernel output, simulate application interrupts (such as arriving SMS messages or phone calls), and simulate latency effects and dropouts on the data channel.

Android Emulator

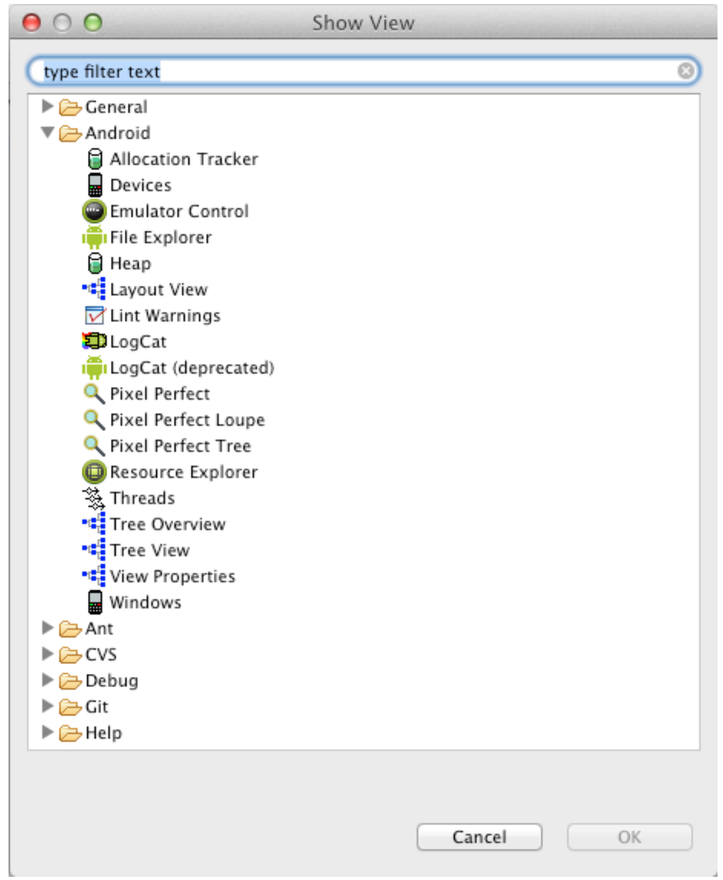
- Android emulator supports many hardware features likely to be found on mobile devices, including:
  - An ARMv5 CPU and the corresponding memory-management unit (MMU)
  - A 16-bit LCD display
  - One or more keyboards (a Qwerty-based keyboard and associated Dpad/Phone buttons)
  - A sound chip with output and input capabilities
  - Flash memory partitions (emulated through disk image files on the development machine)
  - A GSM modem, including a simulated SIM Card

Android Emulator



ADT Additional Tools

- ADT Plugin provides several useful tools to develop, control and profile Android applications, emulators and real devices
- Relevant ADT Views are:
  - Devices
  - LogCat
  - Emulator Control
  - Resource Explorer
  - Threads



Coming Up

- Next Lecture
  - The Android Platform
- Homework
  - Install Android SDK & ADT
  - Create First Android Project
  - Create a Virtual Device
  - Run the Application on a real or Virtual Device