Introduction to Android™ Development

Philippe Leefsma

Senior Developer Consultant, Autodesk Developer Network

Class Summary

Learn how quick and easy it is to start programming for Android devices.

We will create from scratch a simple Android application with UI elements like button and list. Then deploy this app on an Android device and show how you can debug it.

We will also illustrate how to consume REST web services from Android and present a couple of more advanced APIs.

About the Presenter

Philippe Leefsma

Autodesk Developer Technical Services
EMEA (Neuchatel, Switzerland)

Philippe has a master's degree in Computer Sciences. He carried his studies in Paris at I.S.E.P and in USA, at Colorado School of Mines.

He joined Autodesk 6 years ago where he works as developer consultant for Autodesk Developer Network. He supports various products APIs such as AutoCAD®, AutoCAD Mechanical®, and Autodesk® Inventor®. He likes to travel, meet developers from around the world to work with them around challenging programming, CAD and manufacturing topics.

Learning Objectives

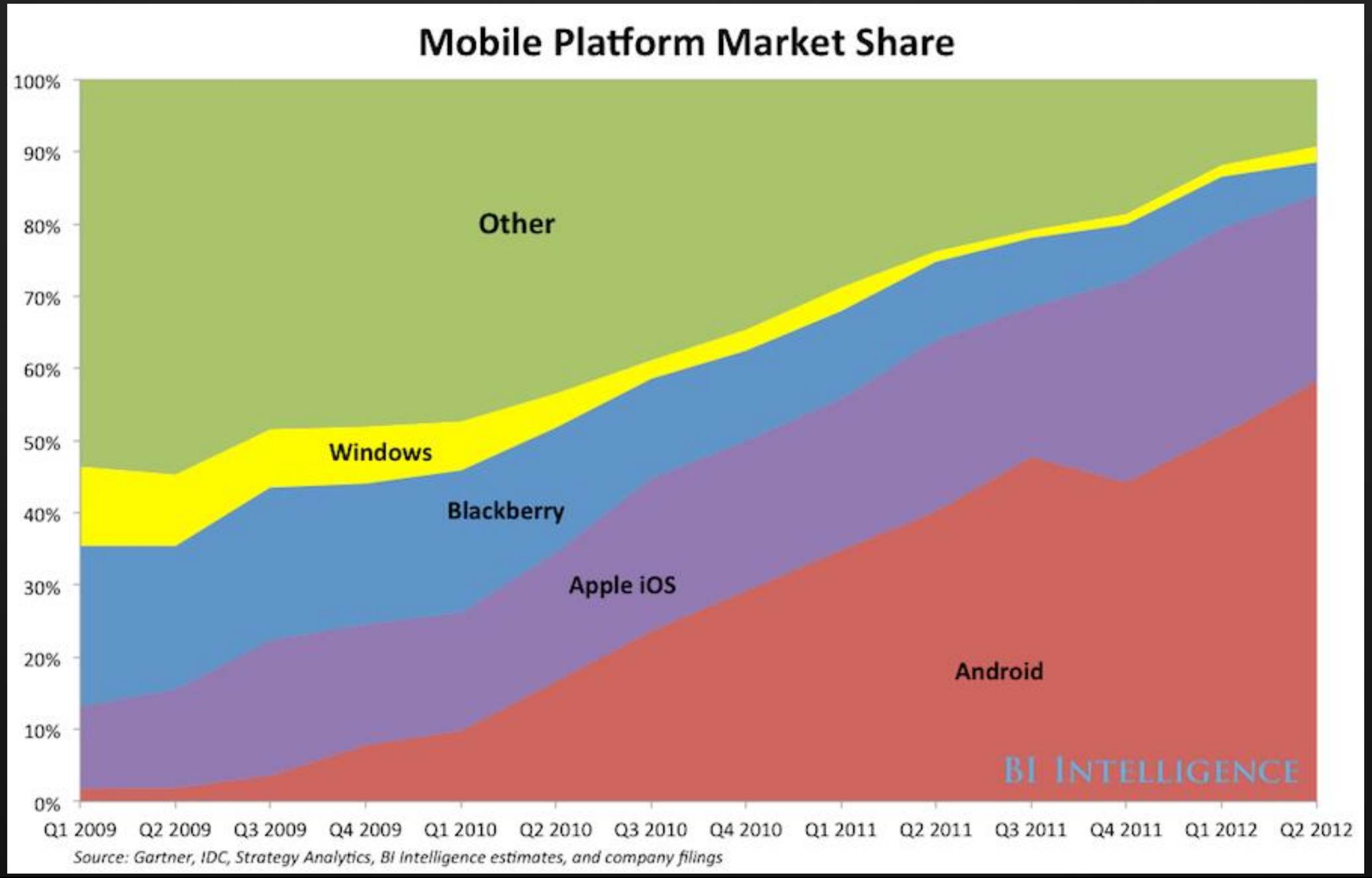
At the end of this class, you will be able to:

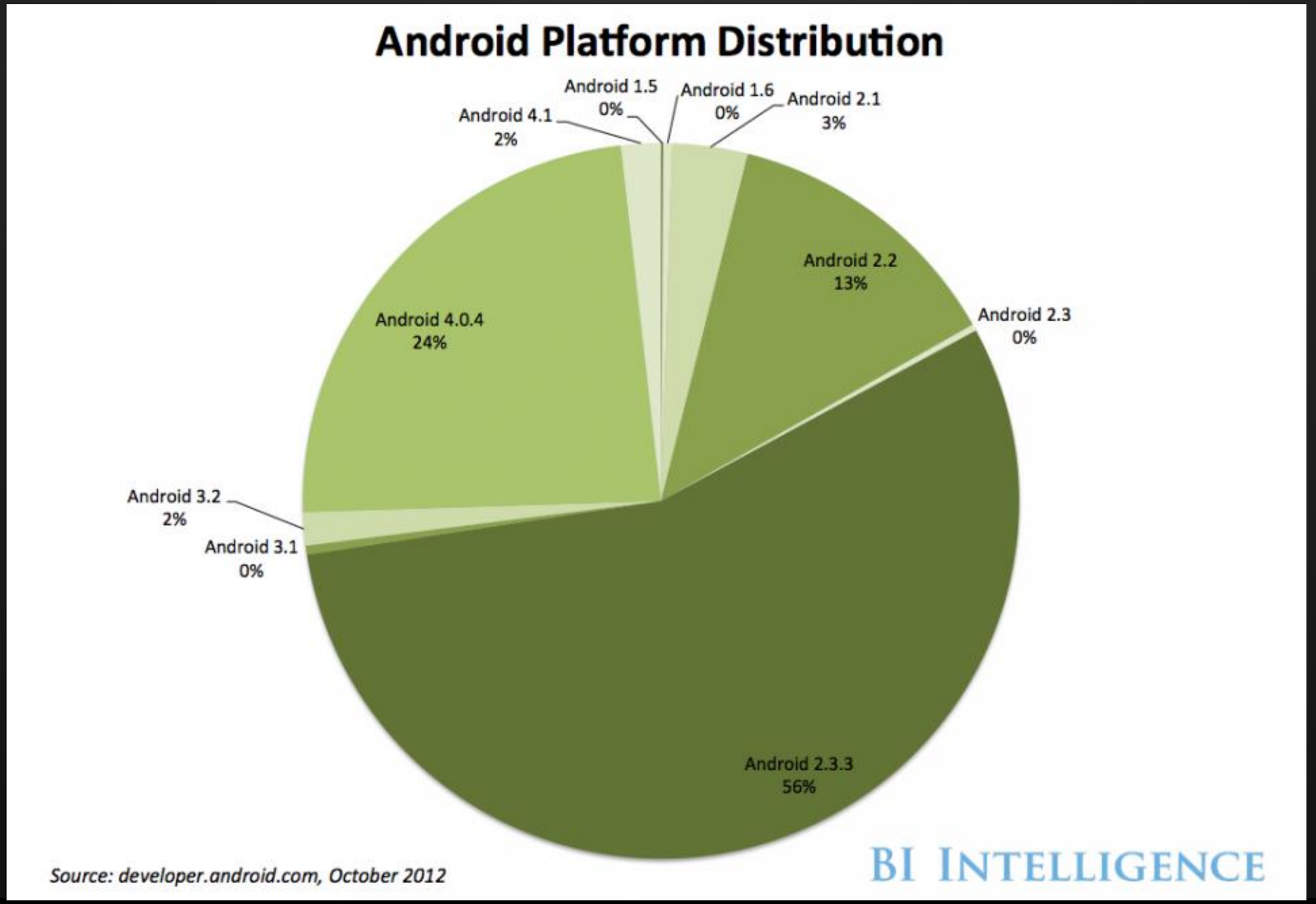
- Understand the basic components of the Android OS
- Start developing applications for Android devices
- Create apps with simple UI
- Consume REST web services from Android
- Get a taste of some more advanced parts of the API

I - The Android Operating System

A bit of History…

- **2003** - Android, Inc. is founded by Andy Rubin, Rich Miner, Nick Sears and Chris White with the goal of developing smarter mobile devices
- **2005** - Google seeing a large growth of Internet use and search on mobile devices, acquires Android Inc.
- **2007** - Apple introduces the iPhone with some ground-breaking ideas:
  - *Multi-touch screen, Open market for applications*
  - *Android is quickly adapted to include these features & other distinctions*
- **2008** - First Android-powered phone sold (*HTC Dream T-Mobile G1*)
- **2010** - Android becomes world's leading smartphone platform, overtaking Symbian





What Android is and isn’t

- An embedded OS that relies on Linux kernel for core services but it is NOT embedded Linux
- Standard Linux utilities (X-windows, GNU C libraries) NOT supported
- Applications use Java framework but some standard libs such as Swing are not supported
- Several libs replaced by Android libs > optimized for resource-constrained embedded environment

Android Images

The Android OS is organized into the following images:

- Boot image
  Kernel and RAMdisk
- Bootloader
  Initiates loading of the boot image during startup
- System image
  Android operating system and apps
- Data image
  User data saved across power cycles
- Recovery image
  Files used for rebuilding or updating the system
- Radio image
  Files of the radio stack

Application Components

Functionality	Android Base Class	Example
Focused thing user can do	***Activity***	Edit note, play game
Background process	***Service***	Play music, update weather icon
Receive messages	***BroadcastReceiver***	Trigger alarm upon event
Store and retrieve data	***ContentProvider***	Open phone contact

II - Getting Started with Android programming

Set up Programming Environment

- Install Eclipse

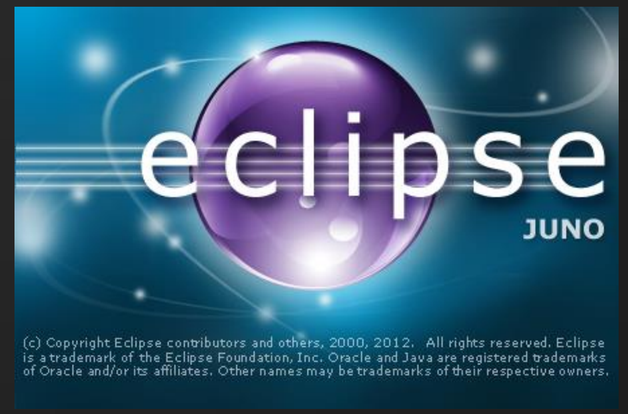
  http://www.eclipse.org/downloads/

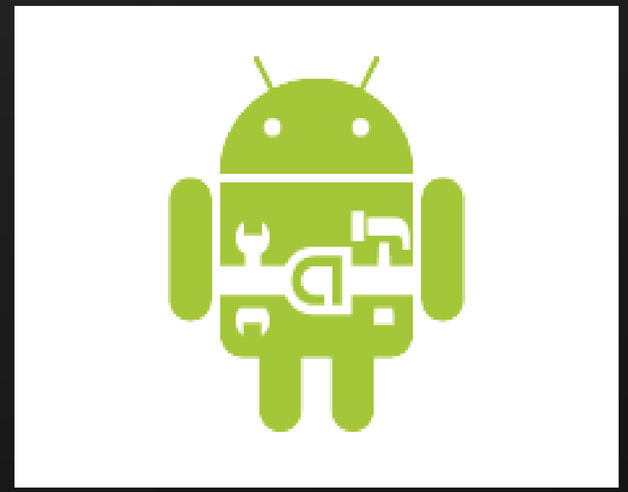
- Install Android SDK

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

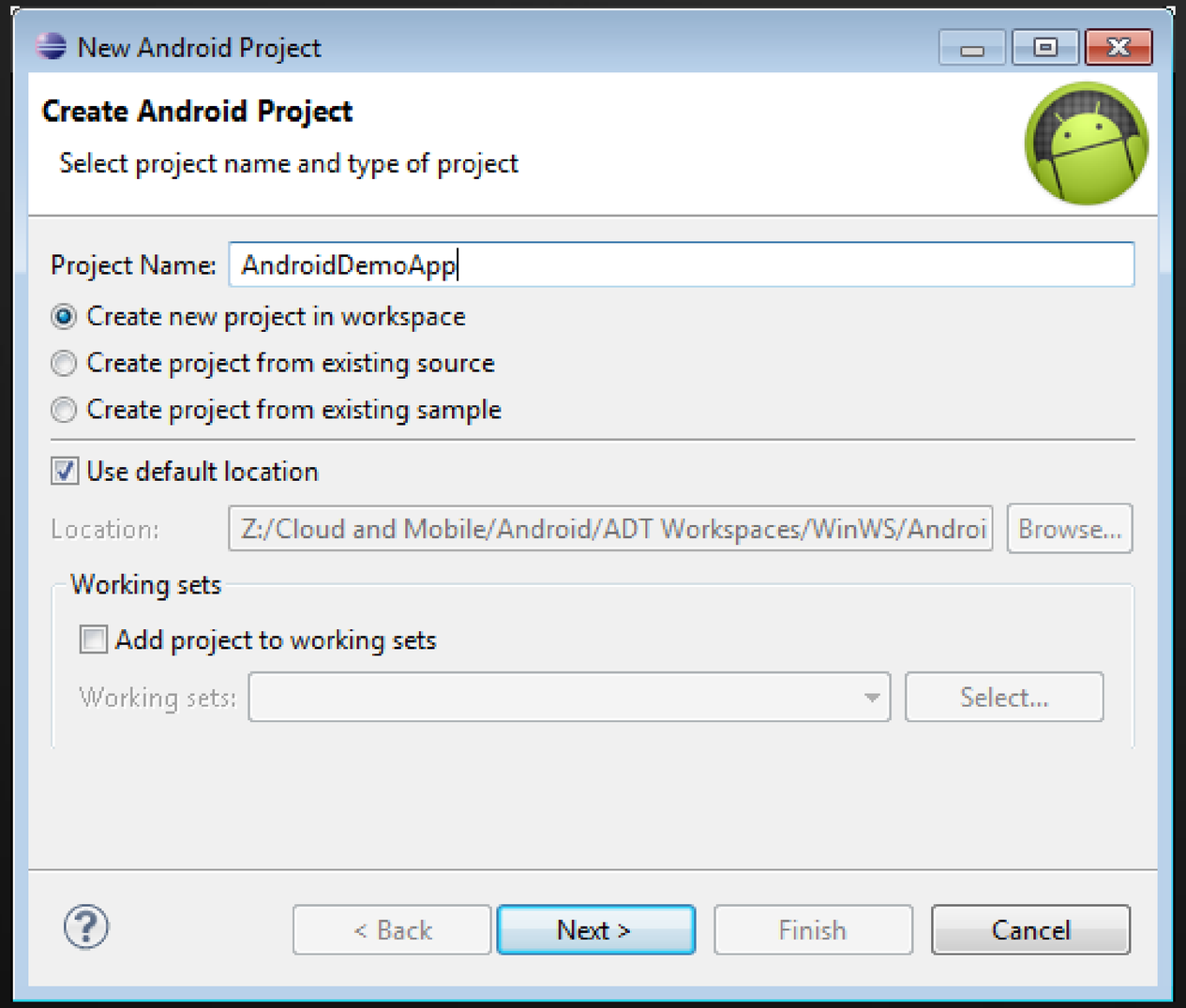
- Get started with Android development

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





My first Android App in Eclipse



Android Project Structure

- AndroidManifest.xml
- src/
- res/
    - drawable-XXXX/
    - layout/
    - values/

Debugging

- GPU-Enabled Emulator
  - http://developer.android.com/tools/devices/emulator.html
- Connecting device with USB
  - Requires installation of device specific driver for Windows
  - Immediate on Mac OS (setting to be enabled on device)
- Connecting device via wifi/tcp
  - Device needs to be rooted
  - Using **adb** utility: *adb connect device.ip*
- Virtualization solution
  - *http://www.android-x86.org*
  - Connect through adb same way as wifi

III - Web Services & Other APIs

REST/GET on Android

```
HttpClient httpclient = new DefaultHttpClient();

HttpGet httpget = new HttpGet(address);

HttpResponse response =
    httpclient.execute(httpget);

HttpEntity entity = response.getEntity();

InputStream instream = entity.getContent();
String jsonMsg= ConvertStreamToString(instream);
instream.close();
```

Json library on Android

Jackson and ***Gson*** are the most complete Java JSON packages compatible with Android regarding actual data binding support

- http://jackson.codehaus.org/
- https://sites.google.com/site/gson/Home

```
Gson parser = new Gson();

Type collectionType = new TypeToken< T >(){}.getType();

T data = (T)parser.fromJson(jsonMsg, collectionType);
```

```
String jsonMsg = parser.toJson(data);
```

REST/POST on Android

```
HttpClient httpclient = new DefaultHttpClient();

HttpPost httppost = new HttpPost(address);

httppost.setHeader("Accept", "application/json");
httppost.setHeader("Content-type", "application/json");

//Requires exception handling...

StringEntity strEntity = new StringEntity(jsonMsg);

strEntity.setContentEncoding(
    new BasicHeader(HTTP.CONTENT_TYPE, "application/json"));

httppost.setEntity(strEntity);

HttpResponse response = httpclient.execute(httppost);

HttpEntity entity = response.getEntity();
```

OpenGL on Android

- OpenGL ES 1.0/1.1
  - javax.microedition.khronos.opengles - Standard implementation of OpenGL ES 1.0/1.1
  - android.opengl - Better performance than javax.microedition.khronos
- OpenGL ES 2.0
  - android.opengl.GLES20 - Interface to OpenGL ES 2.0 and available from Android 2.2 (API Level 8)

Touch API

```
public class TouchListener
   implements OnTouchListener {
   public boolean onTouch(View arg0, MotionEvent event)
   {
      switch (event.getAction() & MotionEvent.ACTION_MASK)
      {
          case MotionEvent.ACTION_DOWN:
              break;

          case MotionEvent.ACTION_POINTER_DOWN:
              break;

          case MotionEvent.ACTION_MOVE:
              break;

          case MotionEvent.ACTION_UP:
              break;

          case MotionEvent.ACTION_POINTER_UP:
              break;
      }
      return true;
   }
}
```

Sensors API

```
public class OrientationManager
    implements SensorEventListener
{
    public void startListening()
    {
        _sensorManager = (SensorManager) _activity.getSystemService(Context.SENSOR_SERVICE);

        _accelerometer = _sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

        _magnetometer = _sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);

        _sensorManager.registerListener(this, _accelerometer, SensorManager.SENSOR_DELAY_FASTEST);

        _sensorManager.registerListener(this, _magnetometer, SensorManager.SENSOR_DELAY_NORMAL);
    }
    public void onSensorChanged(SensorEvent event)
    {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
            _gravity = event.values;
        if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD)
            _geomagnetic = event.values;
        if (_gravity != null && _geomagnetic != null)
            SensorManager.getRotationMatrix(R,  I, _gravity, _geomagnetic);
    }
}
```

Android NDK - Native Development Kit

- NDK is a toolset that allows implementing parts of your app using native-code: C/C++
  - Reuse existing libraries
  - Potentially increases performances

- NDK will **NOT** benefit most apps
  - Native code on Android generally does not result in noticeable performance
  - Increases your app complexity
  - Do not use NDK because you simply prefer to program in C++

- Good candidates for the NDK
  - Self-contained, CPU-intensive operations that don't allocate much memory

Wrap Up

- We exposed basic components of the Android OS
- Developed and debugged an application on Android
- Created simple UI
- Consumed REST web service
- Got a taste of some more advanced part of the API

Material

- CP3100 - Introduction to Android™ Development.pptx
- CP3100 - Introduction to Android™ Development.pdf

- Sample Apps
  - AndroidDemoApp
  - AdnAndroidMaterial
  - AdnDroidView
  - InvBluetoothConnector
  - HelloOpenGLES10
  - HelloOpenGLES20

Resources for Android developers

- Android Developer Center

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

- Android Cookbook

  http://androidcookbook.com/home.seam

- ADN Cloud And Mobile DevBlog

  http://adndevblog.typepad.com/cloud_and_mobile

- Forums

  http://stackoverflow.com

  http://www.codeproject.com/KB/android

Q & A