Android-разработка: как начать, лучшие практики и интересные фишки

Автор: Константин Слисенко,
Software Engineer in JazzTeam

О чём сегодня расскажу

- Splash activity
- Uncaught exception handling + logging
- Кастомизация билд-системы android-проекта
- Автоматизированное UI-тестирование (Selenium like)
- Декомпиляция apk-файлов
- ***Начинающим разработчикам***

Splash activity

Задача

- при запуске приложения загрузить какие-либо данные с сервера, либо синхронизироваться с сервером

Splash activity

- входная точка приложения, выполняет эти операции, потом переходим на главный скрин приложения

Splash activity

```
public class Splash extends GenericActivity {
    @Override
    public void onCreate() {
        // Можем показывать пользователю картинку либо progress bar пока  идёт загрузка
        setContentView(R.layout.splash);

        new AsyncTask<Void, Void, Void> () {
      @Override
      protected void doInBackground() {
            synchronizeWithServer();
            loadData();
            // Из AsyncTask можно обновлять UI с  помощью
            // publishProgress(int progress);
      }
      @Override
      protected void onPostExecute() {
            // После выполнения синхронизации запускаем
            // главный скрин приложения
            startActivity(MainActivity.class);
      }
        }.execute();
    }
}
```

Uncatched exception handling + logging

Логирование в файл во время работы приложения

Обработчик для uncaught исключений. Отправка e-mail с логами при возникновении исключения. Удобно для отладки.

```
Thread.setDefaultUncaughtExceptionHandler(new DebugExceptionHandler());

public class DebugExceptionHandler implements UncaughtExceptionHandler {
    @Override
    public void uncaughtException(Thread thread, Throwable throwable)
    {
        // Логируем ошибку, пишем в лог-файл
        Logger.e(getClass(), getStackTrace(throwable));
        // Отправляем отчёт на почту
        Logger.sendReport();
        // Удаляем log-файл
        Logger.clear();
    }
}
```

Кастомизация билд-системы андроид проекта

Задача

приложение с большим объёмом графики, нужна поддержка устройств с различными разрешениями, для которых много графики различается

Версии

320x240, 480x320, 800x480, 1024x600 (android 2 и 3), 1280x768 (android 2 и 3)

Объём графики: 4 Мб для каждой версии
Общая графика: 1.5 Мб
Графика для каждого разрешения: 2.5 Мб

Итого: 19 Мб!

в основном не нужная пользователю графика
пользователь не захочет скачивать приложения из-за большого объёма

При добавлении новых разрешений устройств, объём приложения будет только расти

Кастомизация билд-системы андроид проекта

Решение

- отдельная apk для каждого разрешения только со своей графикой

Реализация

- Сборка приложения ant-скриптом с подставлением нужной графики,
- bat-файл для сборки семи apk

Apache Ant

Утилита для автоматизации сборки приложений (аналог утилиты make в UNIX)

Процесс сборки описывается в XML-файле build.xml

http://ant.apache.org/

Apache Ant

Targets (функции)

- Откомпилировать весь проект, очистить временные файлы, …
- Между ними можно устанавливать зависимости

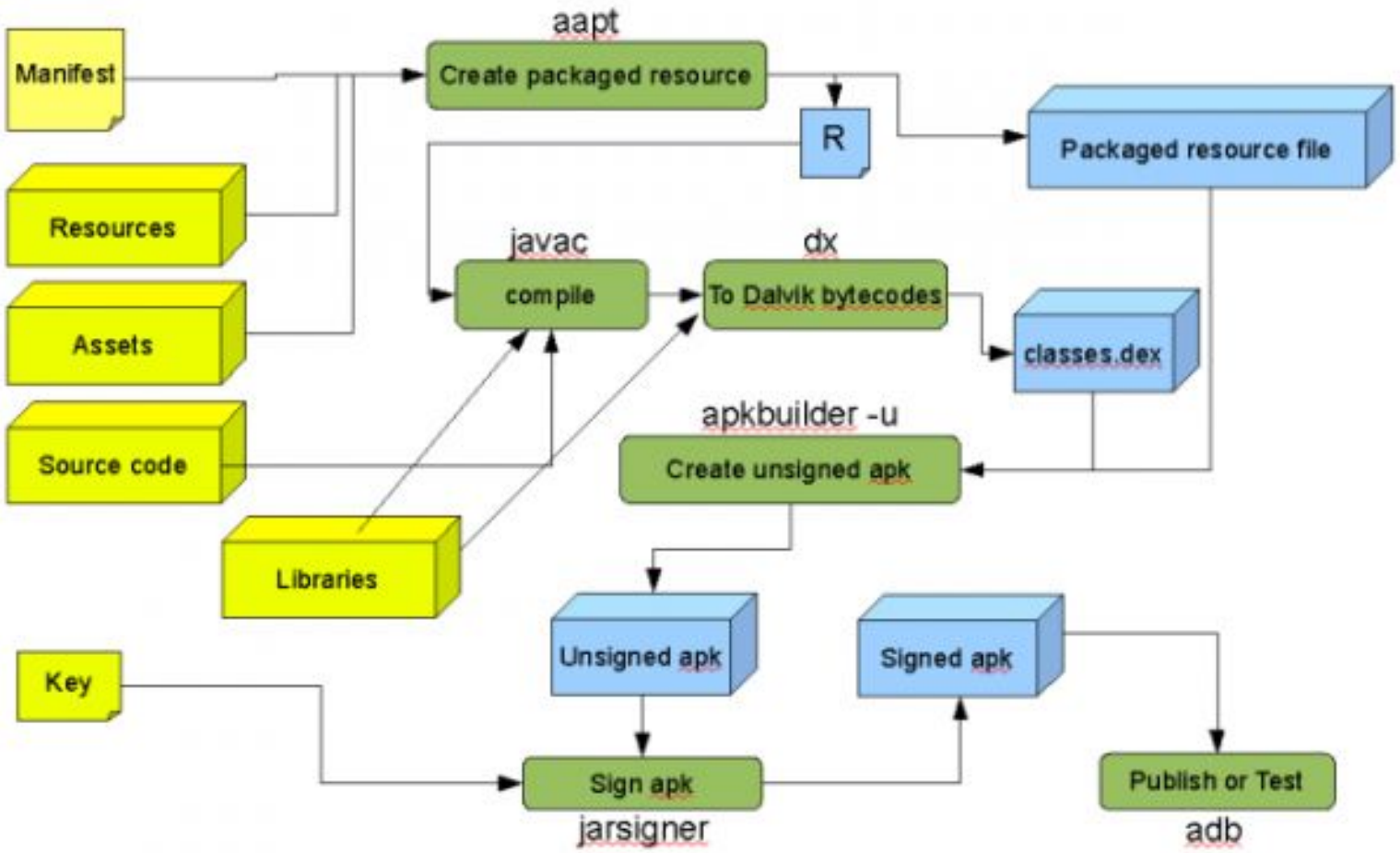
Tasks (элементарные действия)

- javac – откомпилировать java-код
- copy – скопировать файлы
- delete – удалить файлы
- exec – вызвать программу и т.д.

Пример Ant сценария

```
<?xml version="1.0"?>
<project default="build" basedir=".">
<property name="src.dir" location="${basedir}/src"/>
<property name="res.dir" location="${basedir}/res"/>
<property name="classes.dir" location="${basedir}/classes"/>
<property name="dst.dir" location="${basedir}/build"/>
<property name="jar.name" location="application"/>

<target name="build" depends="clean" description="Builds the application">
  <mkdir dir="${dst.dir}"/>
  <mkdir dir="${classes.dir}"/>

  <javac srcdir="${src.dir}« destdir="${classes.dir}‟ debug="false« deprecation="true« optimize="true" ></javac>

  <copy todir="${classes.dir}">
    <fileset dir="${src.dir}" includes="**/*.*" excludes="**/*.java"/>
  </copy>

  <jar jarfile="${dst.dir}/${jar.name}.jar">
    <fileset dir="${classes.dir}"/>
  </jar>
</target>

<target name="clean" description="Removes all temporary files">
  <delete dir="${classes.dir}"/>
  <delete dir="${dst.dir}"/>
</target>
</project>
```

Сборка android-приложения



http://asantoso.wordpress.com

Дорабатываем стандартные скрипты

Скрипты находятся в папке с SDK
.../Android/android-sdk/tools/ant

Типы скриптов:

- main_rules (приложения)
- test_rules (тестовые проекты)
- lib_rules (библиотеки)

Targets, которые можно переопределить

-pre-build (перед началом сборки)

-pre-compile

-post-compile

Структура файлов

MyAndroidApplication
- */android2*
- */res_320x240*
  - ***/drawable***
  - ***/values***
- */res_480x320*
- */res_800x480*
- */res_1024x700*
- */res_1280x768*
- */src*
- */assets*
- */bin*
- */build*
- */gen*
- */src*
- */res*
- ***/drawable***
- ***/values***
- */lib*
- *AndroidManifest.xml*
- *build.xml*

Скрипт копирует содержимое res_320x240 в res и выполняет сборку приложения

Bat-файл сборки нескольких версий

```
call ant release -Dresolution.current=320x240 -Dsdk.current=android2
call ant release -Dresolution.current=480x320 -Dsdk.current=android2
call ant release -Dresolution.current=800x480 -Dsdk.current=android2
call ant release -Dresolution.current=1024x600 -Dsdk.current=android2
call ant release -Dresolution.current=1280x768 -Dsdk.current=android2
call ant release -Dresolution.current=1024x600 -Dsdk.current=android3
call ant release -Dresolution.current=1280x768 -Dsdk.current=android3
```

Собираются семь apk-файлов за один запуск!

Автоматизированное UI-тестирование андроид-приложений

Robotium

Robotium

- Как Selenium, только для Android
- Пишем тест-кейсы, фреймворк прокликивает приложение
- Не обязательно иметь исходный код приложения, достаточно apk

http://code.google.com/p/robotium

Пример использования Robotium

```
public class MyApplicationUITest extends ActivityInstrumentationTestCase2<MyActivity> {
	private Solo solo;
	public NotePadTest() {
	 super("com.mycompany.myapp.activity", MyActivity.class);
	}

	public void setUp() throws Exception {
	 solo = new Solo(getInstrumentation(), getActivity());
	}

	// Тут пишем тесовые методы

	@Override
	public void tearDown() throws Exception {
	 try {
		//Robotium will finish all the activities that have been opened
		solo.finalize();
	 } catch (Throwable e) {
		e.printStackTrace();
	 }
	 getActivity().finish();
	 super.tearDown();
	}
}
```

Пример использования Robotium

Пример тестового метода

```
@Smoke
public void testAddNote() throws Exception {
// Go to help page
ImageView navigation = (ImageView) solo.getView(R.id.view_navigation);
// View about screen
solo.clickOnView(navigation);
solo.assertCurrentActivity("help", HelpActivity.class);

// Return from help to main screen
TextView title = (TextView) solo.getView(R.id.view_navigation_title);
solo.clickOnView(title);

solo.assertCurrentActivity("MyActivity", MyActivity.class);
}
```

Robotium: Black box и White box

Black box

- не знаем кода приложения и id элементов интерфейса

White box

- есть исходный код, знаем id элементов пользовательского интерфейса
- сложней, если id меняются
- больше возможностей

Robotium: основные методы

- assertCurrentActivity(String message, Class expected)
- clickLongOnScreen(int x, int y)
- clickOnScreen(int x, int y)
- ArrayList<Activity> getAllOpenedActivities()
- sleep(int time)
- clickOnView(View view)
- enterText(EditText editText, String text)

В документации ещё много интересных штук!

- drag(float fromX, float toX, float fromY, float toY, int stepCount)

Декомпиляция APK-файлов

Хочу посмотреть, как у них сделано…

1. Распаковываем apk-файл обычным winrar-ом
   classes.dex, xml: binary, картинки

2. Декомпилируем classes.dex
   dex2jar (open source) http://code.google.com/p/dex2jar

3. Просмотр jar-файлов
   JDGui (open source)
   http://code.google.com/p/android-apktool/

4. Декомпиляция xml-файлов
   Apktool (open source)
   http://code.google.com/p/android-apktool/

Начинающим разработчикам

Enviroment

Android SDK, Eclipse + ADT plug-in, драйвера на телефон

Что почитать

developer.android.com – туториалы, документация
vogella.de/android.html – хорошие туториалы
stackoverflow.com – часто пользуюсь этим сайтом

Базовые понятия

Activity, Intent, layouts (LinearLayout, RelativeLayout)
работа с ресурсами (string, dimen, images, layouts)
AndroidManifest.xml

Core Java

ООП, паттерны проектирования, unit-тестирование (JUnit 3,4)

Девайс или эмулятор

Вопросы?

Константин Слисенко,
Software Engineer in JazzTeam

Спасибо за внимание!

kslisenko@gmail.com
konstantin@jazzteam.org