



PLAN.NET

AGENTURGRUPPE FÜR INTERAKTIVE KOMMUNIKATION

AGENTURGRUPPE FÜR INTERAKTIVE KOMMUNIKATION



Mobile Apps

with Adobe® Flash® CS5

May 2010

**Getting your App
from the idea
through development
and testing
into the store**



PLAN.NET

AGENTURGRUPPE FÜR INTERAKTIVE KOMMUNIKATION



Some facts about me

- Working with Flash and ActionScript since 10 years
- Head of Technology and Development at PLAN.NET
- Participant of Flash CS5 alpha and beta test
- BlackBerry user



Some facts about this session

- Flash CS5 has a Packager for iPhone apps
- New iPhone Developer Program License Agreement disallows Flash apps
- Adobe has discontinued development of this feature
- There will be a Packager for Android apps



New iPhone Developer Program License Agreement

- 3.3.1 Applications may only use Documented APIs in the manner prescribed by Apple and must not use or call any private APIs. Applications must be originally written in Objective-C, C, C++, or JavaScript as executed by the iPhone OS WebKit engine, and only code written in C, C++, and Objective-C may compile and directly link against the Documented APIs (e.g., Applications that link to Documented APIs through an **intermediary translation or compatibility layer or tool** are prohibited).



Agenda

1. Preparing – What do I need?

1.1. Developer Account

1.2. Certificates, emulators and devices

2. Developing – As easy as 1-2-3?

2.1. Best practices and code samples

2.2. Keep in mind: performance, filesize and API limitations

3. Testing and ditribution – How to?

3.1. Device testing

3.2. Ad hoc Distribution

4. Publishing – Bringing your App to the store



Agenda

1. Preparing – What do I need?

1.1. Developer Account

1.2. Certificates, emulators and devices

2. Developing – As easy as 1-2-3?

2.1. Best practices and code samples

2.2. Keep in mind: performance, filesize and API limitations

3. Testing and ditribution – How to?

3.1. Device testing

3.2. Ad hoc Distribution

4. Publishing – Bringing your App to the store



Developer Account

- With an Apple Developer Account (\$99 or \$299 per year) you can:
 - create certificates
 - create mobilprovision files
 - manage UDIDs of test devices
 - submit your app for review
 - generate reports
- With an Android Developer Account (\$25 once) you can:
 - submit your app to the Android marketplace
 - generate reports



Agenda

1.

Preparing – What do I need?

1.1.

Developer Account

1.2.

Certificates, emulators and devices

2.

Developing – As easy as 1-2-3?

2.1.

Best practices and code samples

2.2.

Keep in mind: performance, filesize and API limitations

3.

Testing and ditribution – How to?

3.1.

Device testing

3.2.

Ad hoc Distribution

4.

Publishing – Bringing your App to the store



Certificates, emulators and devices

- For your Android app create a certificate with the Java Keytool utility
 - Has to be valid vor 25 years
- For your iPhone app create a certificate in the developer center
 - Via CSR with „keychain“ on your Mac, will be valid for 1 year
- For your Android app install an emulator (or connect a device via USB)
 - Manage applications with command line tool „adb“
- For your iPhone app connect your device with your computer
 - Manage devices via developer center
 - Manage applications via drag&drop in iTunes



Agenda

1. Preparing – What do I need?

1.1. Developer Account

1.2. Certificates, emulators and devices

2. Developing – As easy as 1-2-3?

2.1. Best practices and code samples

2.2. Keep in mind: performance, filesize and API limitations

3. Testing and ditribution – How to?

3.1. Device testing

3.2. Ad hoc Distribution

4. Publishing – Bringing your App to the store



Best practices and code samples

- Optimize your assets, until you have squeezed out every byte, do not embed fonts (if possible)
- Use lightweight tweening engines like TweenNano from Greensock
- Avoid at using timers or onenterframes
- Avoid using new variables inside a loop and “delete” global objects, “null” variables and local objects
- Use feedback on buttons because there’s no "rollover" in the mobile world
- Don’t use "alpha=0" or "visible=false", remove objects from displaylist
- Try to avoid large XML data or use &name=value pair instead and load data only on demand
- Use as much devices as possible to test on
- Avoid Flex framework and classes for little apps because they consume memory
- Crash test in real scenarios: home/bar/restaurant/bus/metro/train



Best practices and code samples

- Detect screen orientation (default, rotatedLeft, rotatedRight or upsideDown):

```
import flash.events.StageOrientationEvent;
```

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,  
                       orientationChangeListener);
```

```
function orientationChangeListener (evt:StageOrientationEvent) : void {  
    trace ("beforeOrientation:" + evt.beforeOrientation + "\n");  
    trace ("afterOrientation:" + evt.afterOrientation);  
}
```



Best practices and code samples

- Detect acceleration (tilt of your device):

```
import flash.sensor.Accelerometer;
import flash.events.AccelerometerEvent;

var acc:Accelerometer = new Accelerometer();
acc.addEventListener(AccelerometerEvent.UPDATE, updateHandler);

function updateHandler(evt:AccelerometerEvent):void {
    trace ("acceleration X: " + evt.accelerationX + "\n");
    trace ("acceleration Y: " + evt.accelerationY + "\n");
    trace ("acceleration Z: " + evt.accelerationZ);
}
```



Best practices and code samples

- Detect geolocation of your device:

```
import flash.events.GeolocationEvent;
import flash.sensors.Geolocation;

if (Geolocation.isSupported) {
    var geo:Geolocation = new Geolocation();
    geo.setRequestedUpdateInterval(100);
    geo.addEventListener(GeolocationEvent.UPDATE, geolocationUpdateHandler);
}

function geolocationUpdateHandler (evt:GeolocationEvent) : void {
    trace ("latitude:" + evt.latitude.toString() + "°\n");
    trace ("longitude:" + evt.longitude.toString() + "°\n");
    trace ("horizontal accuracy:" + evt.horizontalAccuracy.toString() + " m");
}
```



Best practices and code samples

- Recieve touch events and gestures:

```
import flash.ui.Multitouch;
import flash.ui.MultitouchInputMode;
import flash.events.TransformGestureEvent;

Multitouch.inputMode = MultitouchInputMode.GESTURE;
mySprite.addEventListener (TransformGestureEvent.ROTATE, rotateHandler);

function rotateHandler(evt:TransformGestureEvent): void {
    evt.target.parent.rotationZ += evt.target.rotation;
}
```



Best practices and code samples

- Add images to the gallery of your device:

```
import flash.media.CameraRoll;
import flash.events.Event;

var cr:CameraRoll = new CameraRoll();
cr.addEventListener(Event.COMPLETE, completeHandler);

function completeHandler(evt:Event):void {
    trace("bitmapData successfully added");
}

if (cr.supportsAddBitmapData) {
    cr.addBitmapData(bitmap);
}
```



Agenda

1. Preparing – What do I need?

1.1. Developer Account

1.2. Certificates, emulators and devices

2. Developing – As easy as 1-2-3?

2.1. Best practices and code samples

2.2. Keep in mind: performance, filesize and API limitations

3. Testing and ditribution – How to?

3.1. Device testing

3.2. Ad hoc Distribution

4. Publishing – Bringing your App to the store



Performance, filesize and API limitations

- Performance of devices is as good (or bad) as of a ten years old computer
 - only 600 to 800 MHz and 256 MB of RAM
 - Weak graphic processor, however high resolution
- Only iPhone apps < 10 MB could be downloaded without WiFi access
 - Minimum filesize of IPAs built with Flash: 3,5 MB
 - Minimum filesize of APKs built with Flash: < 0,1 MB
- On iPhones you have no access to other applications like „Maps“
- Only Android supports multitasking



Agenda

1. Preparing – What do I need?

1.1. Developer Account

1.2. Certificates, emulators and devices

2. Developing – As easy as 1-2-3?

2.1. Best practices and code samples

2.2. Keep in mind: performance, filesize and API limitations

3. Testing and ditribution – How to?

3.1. Device testing

3.2. Ad hoc Distribution

4. Publishing – Bringing your App to the store



Device testing

- Device testing with iPhones
 - Drag&drop your IPA together with your *.mobileprovision file to iTunes
 - Connect your iPhone via USB
 - Sync your iPhone with iTunes
- Device or emulator testing with Android phones
 - Connect your phone via USB or start emulator
 - Use commandline tool „adb“ to install your app



Agenda

1. Preparing – What do I need?

1.1. Developer Account

1.2. Certificates, emulators and devices

2. Developing – As easy as 1-2-3?

2.1. Best practices and code samples

2.2. Keep in mind: performance, filesize and API limitations

3. Testing and ditribution – How to?

3.1. Device testing

3.2. Ad hoc Distribution

4. Publishing – Bringing your App to the store



Ad hoc Distribution

- With an iPhone enterprise account you are able to distribute your app inside your company without submitting it to the iTunes store
- Differences between ad hoc distribution and device testing
 - Your app can be installed on up to 100 devices
 - Your app is valid for 1 year and not just for 1 month
 - Your app can't be debugged



Agenda

1. Preparing – What do I need?

1.1. Developer Account

1.2. Certificates, emulators and devices

2. Developing – As easy as 1-2-3?

2.1. Best practices and code samples

2.2. Keep in mind: performance, filesize and API limitations

3. Testing and ditribution – How to?

3.1. Device testing

3.2. Ad hoc Distribution

4. Publishing – Bringing your App to the store



Publishing – Bringing your App to the store

- For iPhone development start here: <http://developer.apple.com>
 - Use <http://itunesconnect.apple.com> for app submission
- For Android development start here: <http://developer.android.com>
 - Use <http://market.android.com> for app submission



What's next?

- Hopefully Adobe and Apple will be best friends
- Currently there are only 2 devices supporting AIR:
 - "Motorola Milestone" (EUR 450,-)
 - "Nexus One" (EUR 650,-)
- But more than 20 will be available at the end of this year
 - most of them in Q3 2010
 - i.e. from Acer, Dell, HTC, LG, Motorola, Samsung and SonyEricsson
 - a handful of tablet computers in addition
 - i.e. from Dell, HardKernel or Velocity Micro
- „Shibuya“ will come – a monetization service for AIR applications



How to contact me

- @ PLAN.NET:
 - <http://www.plan-net.de>
 - f.feiler@plan-net.de

- Personal Blog:
 - <http://www.florian-feiler.de>
 - info@florian-feiler.de

