Workshop:
Integrating Samsung Mobile SDK into your Android App

Anna Schaller
Sr. Manager, Developer Relations
Samsung Media Solutions Center America

SAMSUNG
DEVELOPERS

MOBILE SDK

PEN

- Drawing Canvas
- Advanced Editing
- Embedded media
- Signature Recognition
- Equation and Shape Recognition
- Air View

MEDIA CONTROL

- Play media files on another device
- Play media files from a Web Server on another device
- Play media files from a search on a DLNA Media Server

PROFESSIONAL AUDIO

- Musical Instrument Creation
- Plug-ins for acoustic piano, steel guitar and drums

GESTURE

- Supplements device touch or motion events

LOOK

- WritingBuddy
- AirButton
- SmartClip
- PointerIcon

VISUAL VIEW

- Adv. Geometry
- Basic Animation
- Key-Frame Animation
- Transition Animation
- Sprite Animation
- Animation Set

MULTIWINDOW

- MultiWindow Traybar
- Multi-Instance
- Restore Paired Windows
- Resize/Reorder

IMAGE FILTER

- Choose a Filter Effect
- Adjusting Filter Levels
- Adjusting Transparency

CHORD

- Device search and group management
- Data and file transfer
- Secure channels for data transfer

MOTION

- Call Motion
- Pedometer

SAMSUNG DEVELOPERS

What's next? You decide.

SAMSUNG DEVELOPERS

What's next? You decide.

Workshop Agenda

1. Integrating MultiWindow
   - The easy way

2. Verifying support for features
   - Using the sdk-v1.0.0.jar interface file

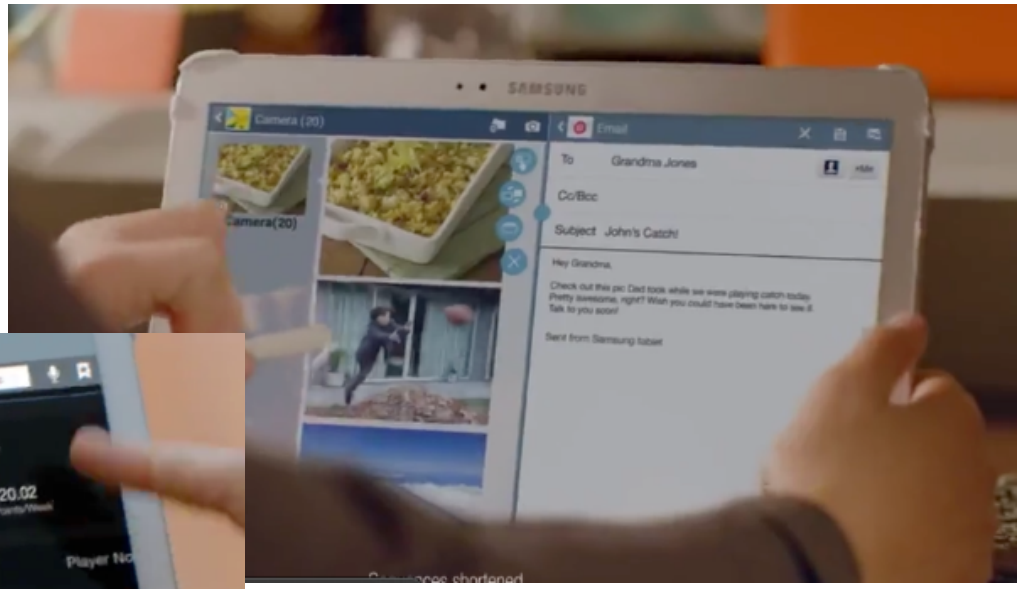
3. Integrating Gestures

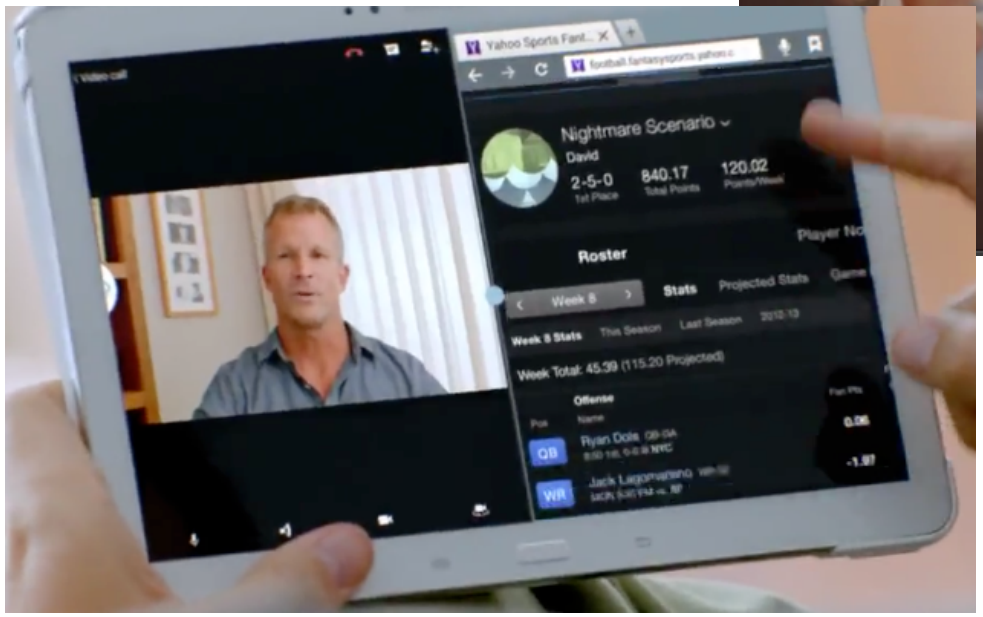
NOTE: you must have the Mobile SDK installed
http://developer.samsung.com/samsung-mobile-sdk

MultiWindow

MultiWindow – What is it?

- Google Hangouts
- Chrome
- Email



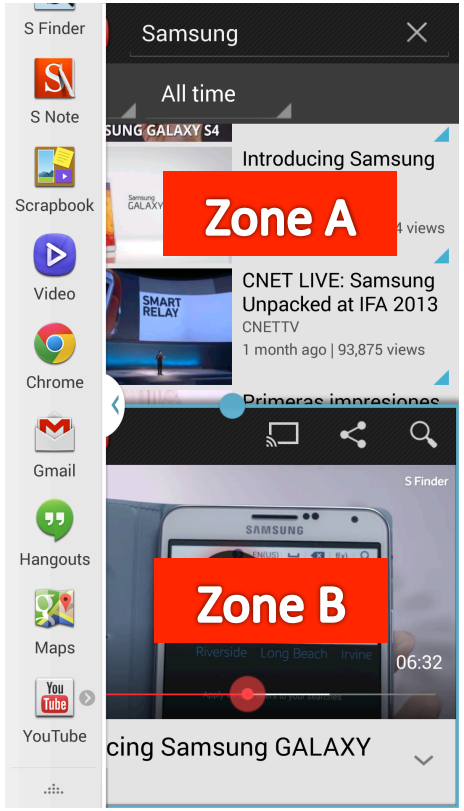


MultiWindow Features

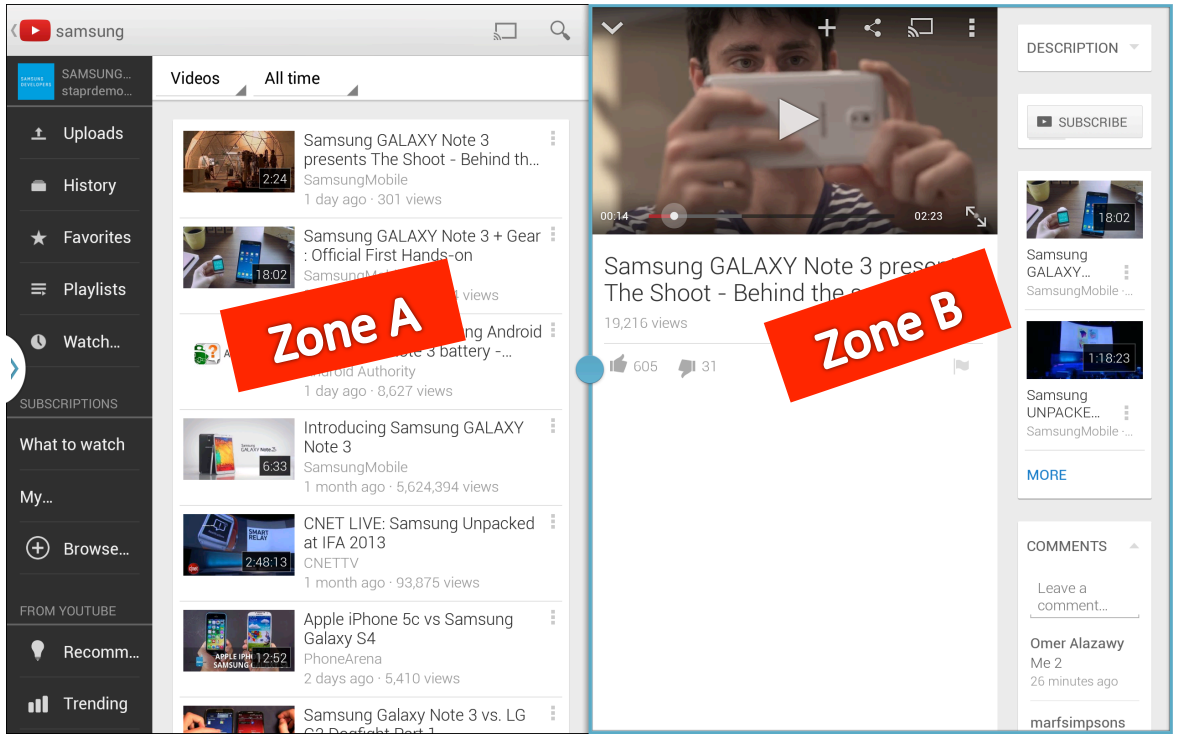
- Zones
  - Supports 2 separate applications running side-by-side
- Style
  - Split Window, Normal, Pen Window
- MultiWindow Tray
  - Navigation bar for multiwindow-enabled apps
- MultiInstance MultiWindow
  - Supports same application running in 2 different windows
- Drag-n-drop
  - Between windows

MultiWindow Features -- Zones

- Note 3

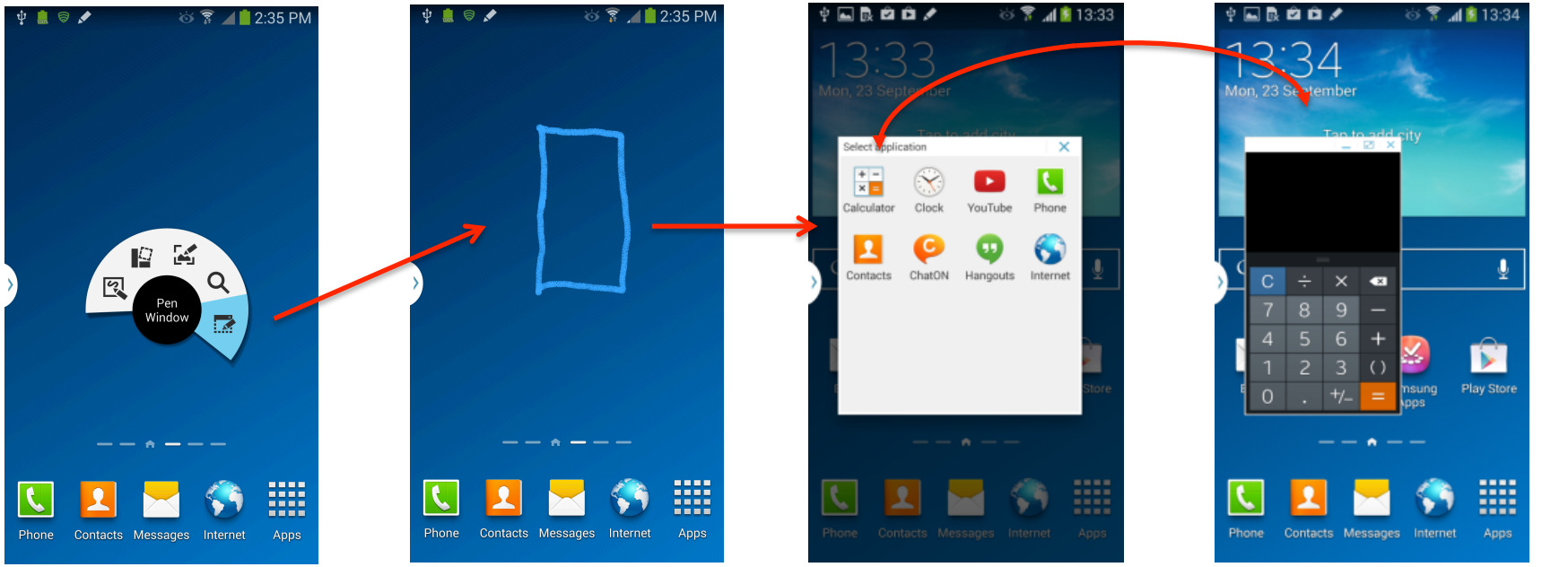


- 2014 Note 10.1



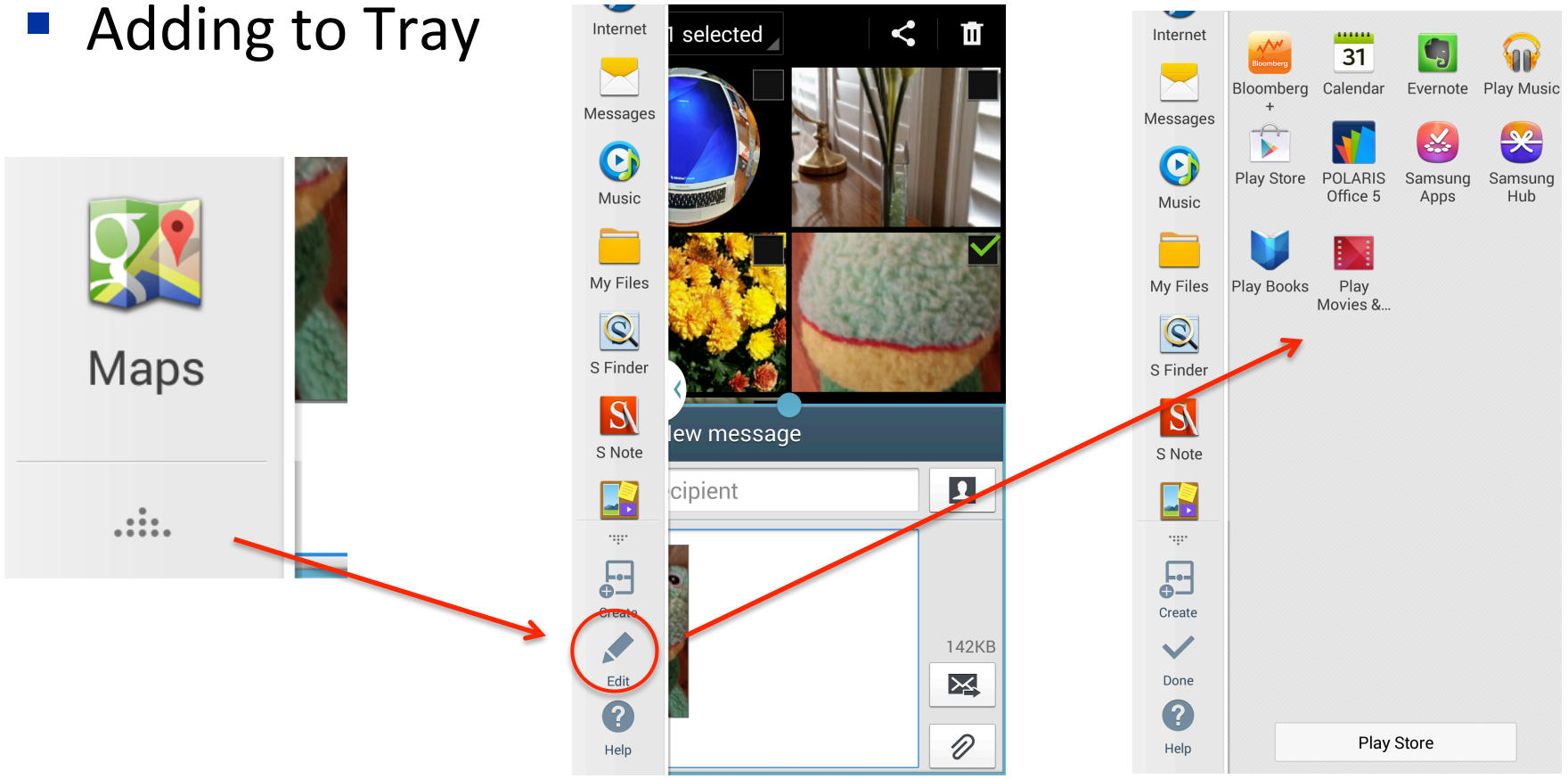
MultiWindow Features -- Styles

- Side-by-side
- Pen Window

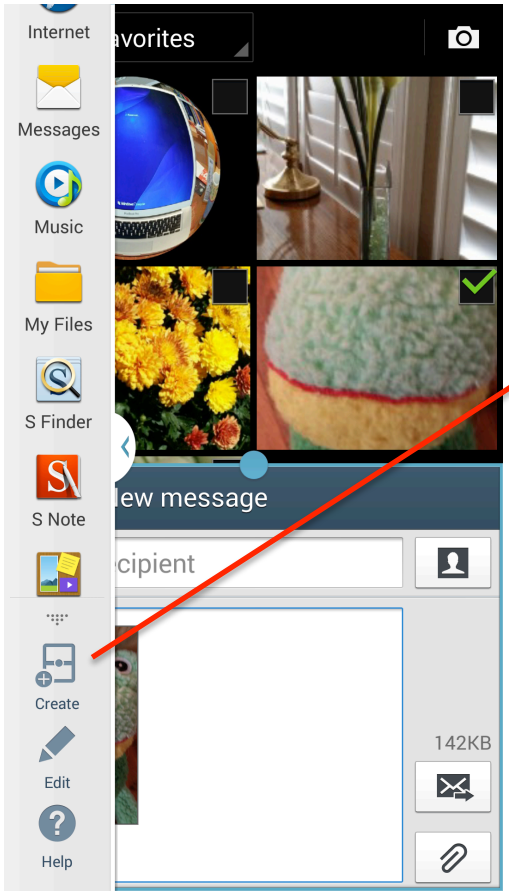


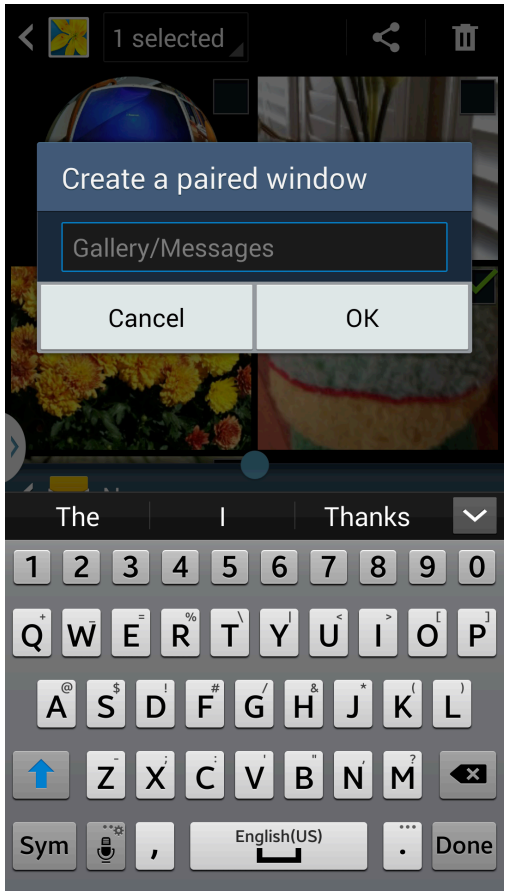
MultiWindow Features -- Tray

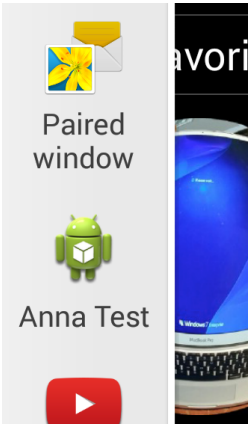
- Adding to Tray



MultiWindow Features – Tray Paired Windows



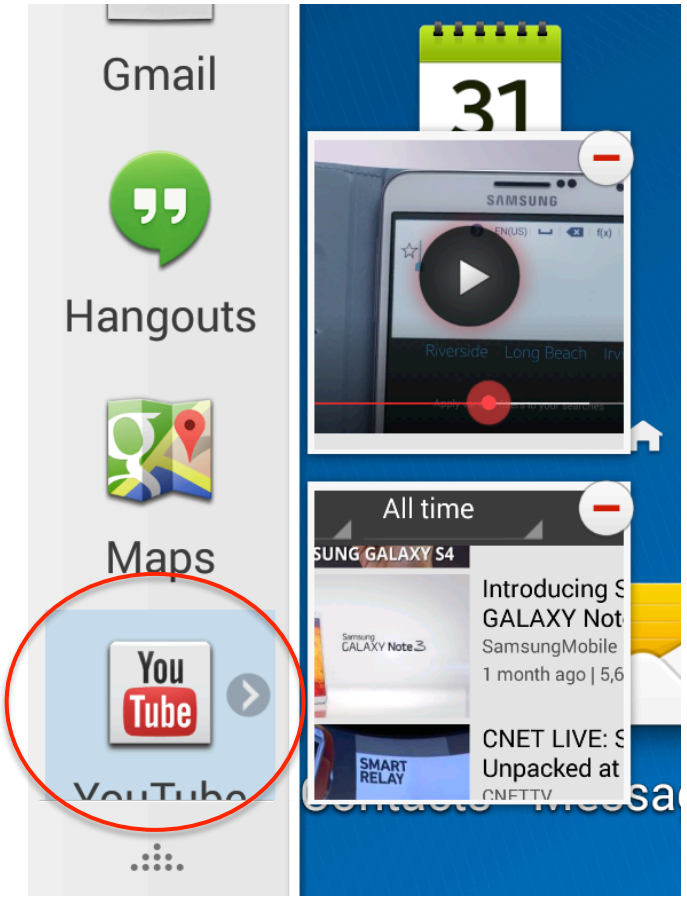




MultiWindow Features -- MultiInstance

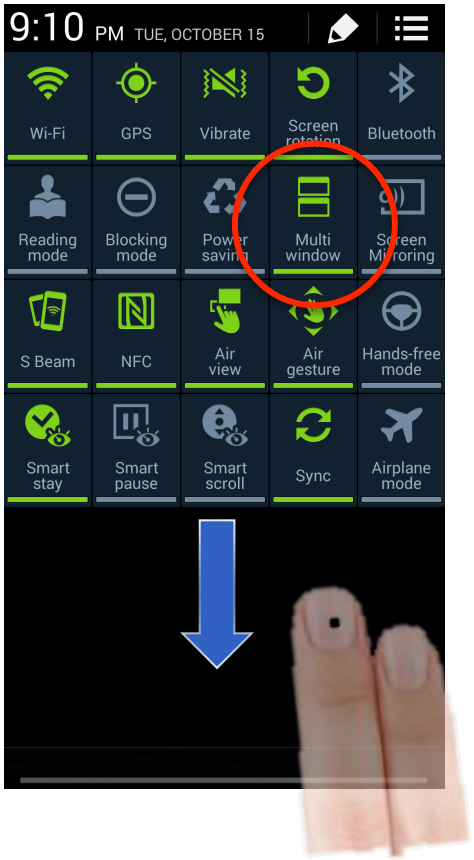
- MultiInstance MultiWindow
  - Up to 2 separate instances of your app running in their own instance of Dalvik
  - MultiInstance-enabled apps have arrow indicating more options

  - 2 possible zones to open in



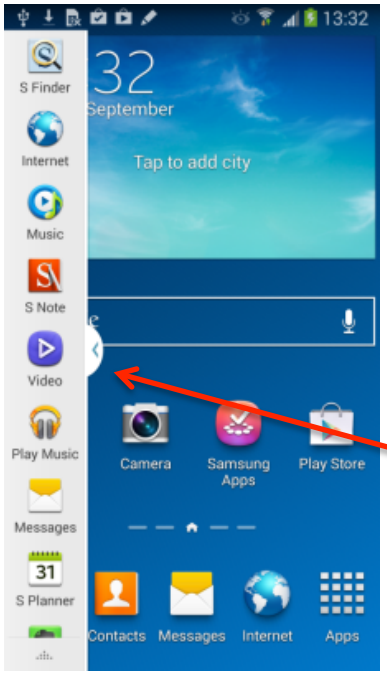
Implementing: Easy as 1 – 2 – 3

- Turn on MultiWindow
- Add <meta-data> tags to the AndroidManifest.xml file
- Add 1 new <category> tag to the AndroidManifest.xml file



AndroidManifest.xml: <MultiWindow> mode – Step 1

```
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
```




```
<meta-data
    android:name="com.samsung.android.sdk.multiwindow.enable"
    android:value="true"/>
```

Enables multiwindow and displays on multiwindow traybar

AndroidManifest.xml: <MultiWindow> category – Step 2

```
<activity
    android:name="com.example.annatest.MainActivity"
    android:label="@string/app_name" >

    <meta-data . . . >

    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />

        <category
            android:name="android.intent.category.MULTIWINDOW_LAUNCHER" />

    </intent-filter>
```

Adding MultiWindow -- Exercise 1

1. Start with “feels” project and open AndroidManifest.xml file
2. Locate the <application> node. Add the following line

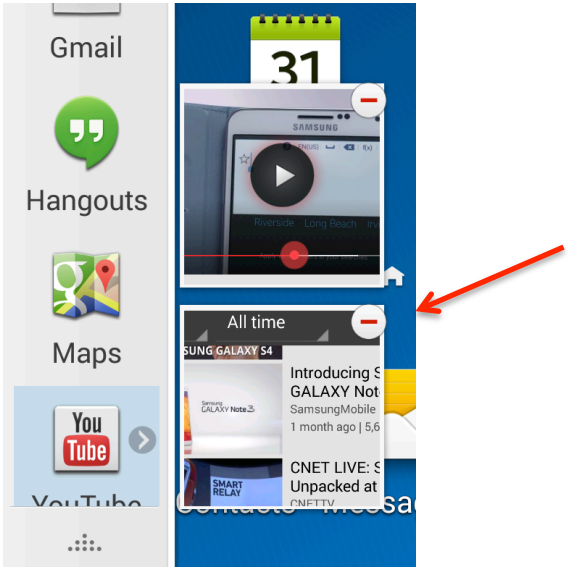
```
<meta-data
    android:name="com.samsung.android.sdk.multiwindow.enable"
    android:value="true"/>
```

3. Locate <intent-filter> tag under <activity> node

[. . .]

Extra Credit -- <MultiInstance> mode

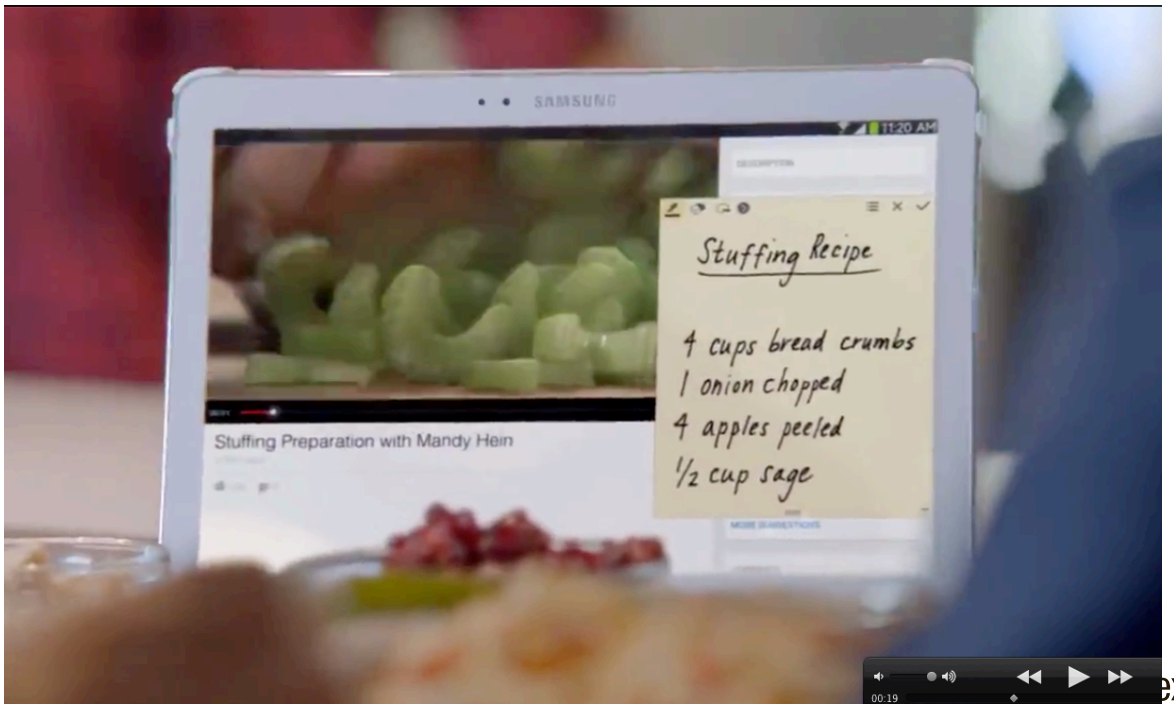
```
<application>
    <meta-data
        android:name="com.samsung.android.sdk.multiwindow.multiinstance.enable"
        android:value="true"/>
```



To support MultiInstance, lauchMode in AndroidManifest.xml should be defined as standard or singleTop

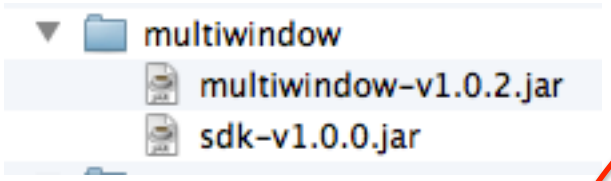
Extra Credit -- <PenWindow> mode

```
<application>
    <meta-data
        android:name="com.samsung.android.sdk.multiwindow.penwindow.enable"
        android:value="true"/>
```



ext? You decide.

Coding for Additional Control



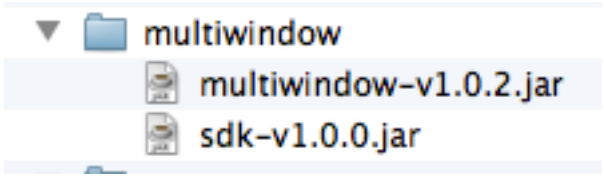
Class Summary

Class	Description
SMultiWindow	Provides capability and version code api.
SMultiWindowActivity	SMultiWindowActivity class for MultiWindow.

Methods

Modifier and Type	Method and Description
int	`getVersionCode()` Return SDK version of multiwindow package,
String	`getVersionName()` Return SDK version name of multiwindow.
void	`initialize(Context arg0)` If the device is not supported, Exception(SsdkUnsupportedException) will be occured.
boolean	`isFeatureEnabled(int type)` Checks whether the feature is supported.

Coding for Additional Control



Class Summary

Class	Description
SMultiWindow	Provides capability and version code api.
SMultiWindowActivity	SMultiWindowActivity class for MultiWindow.

Methods

Modifier and Type	Method and Description
Rect	getRectInfo() Get size and location of MultiWindow
int	getZoneInfo() Get the window zone Information
boolean	isMultiWindow() Check whether window mode is multi window
boolean	isNormalWindow() Check whether current window mode is normal window
static Intent	makeMultiWindowIntent(Intent intent, int zoneInfo) Make an intent for starting Activity in MultiWindow Should use only one of the rect and zoneinfo.
void	normalWindow() Changed window state to normal.
boolean	setStateChangeListener(SMultiWindowActivity.StateChangeListener listener) set listner for detecting changing window mode and info.

Callbacks for the State Change Listener

```
public interface StateChangeListenter( . . . )
```

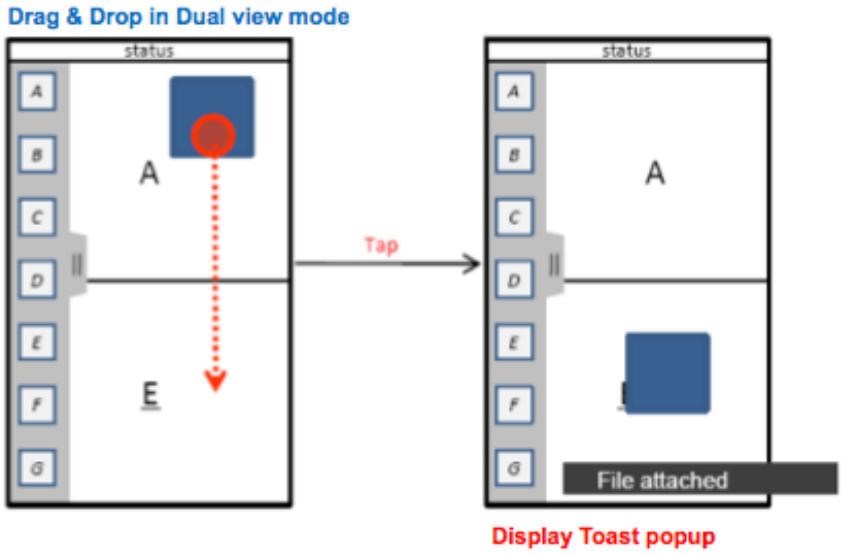
Methods

Modifier and Type	Method and Description
`void`	**`onModeChanged`**`(boolean isMultiWindow)` Called when activity is changed the multiwindow and normal window.
`void`	**`onSizeChanged`**`(`**`Rect`**` rectInfo)` Called when activity is changed the multiwindow size information.
`void`	**`onZoneChanged`**`(int zoneInfo)` Called when activity is changed the multiwindow zone information.

Drag and Drop in MultiWindow

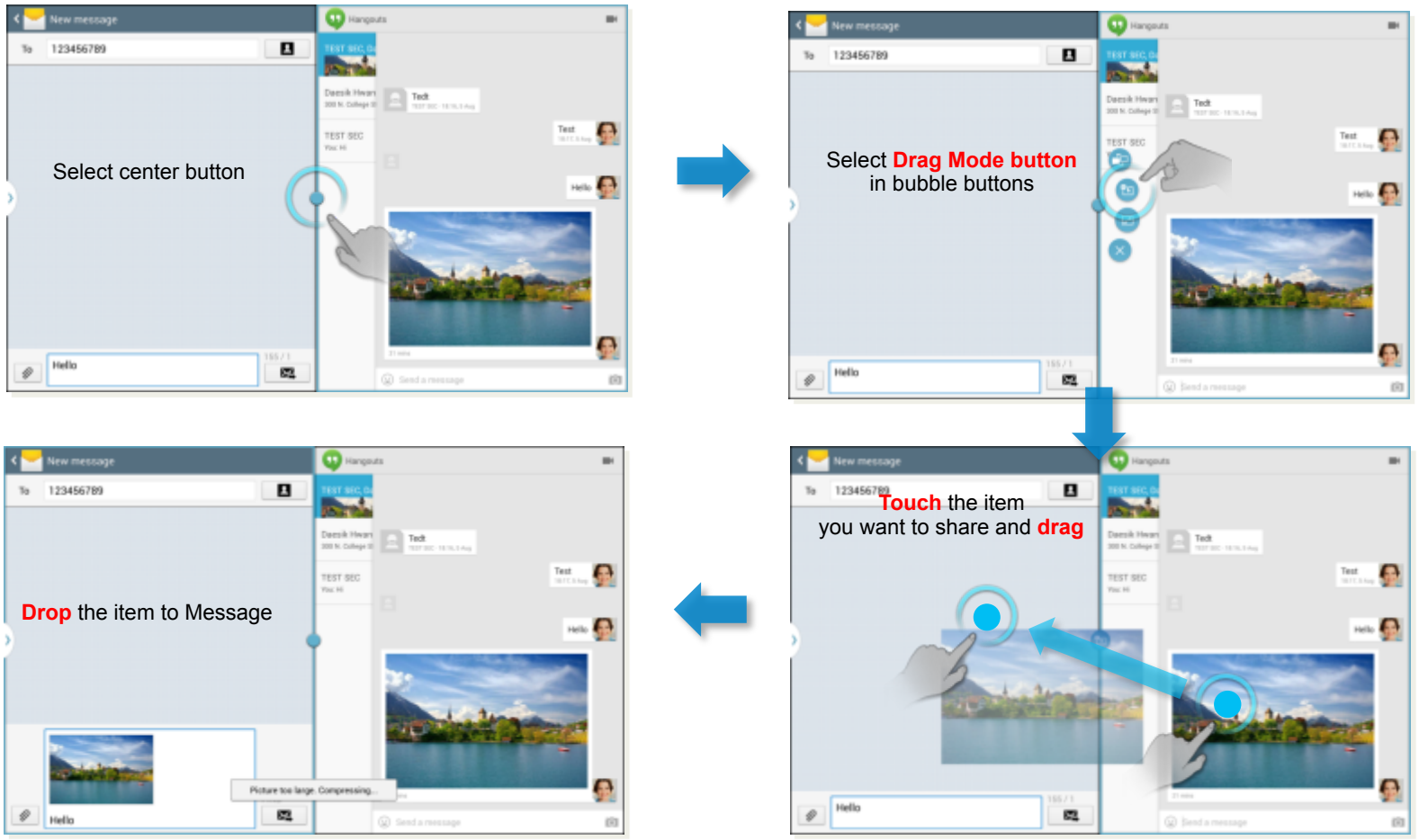
- 3 supported methods for Drag and Drop
  - Long click on Image or File
  - Long click on Text
  - Drag Mode in center button

- Drag Apps ?



- Drop Apps ?

Drag and Drop – MultiWindow Button



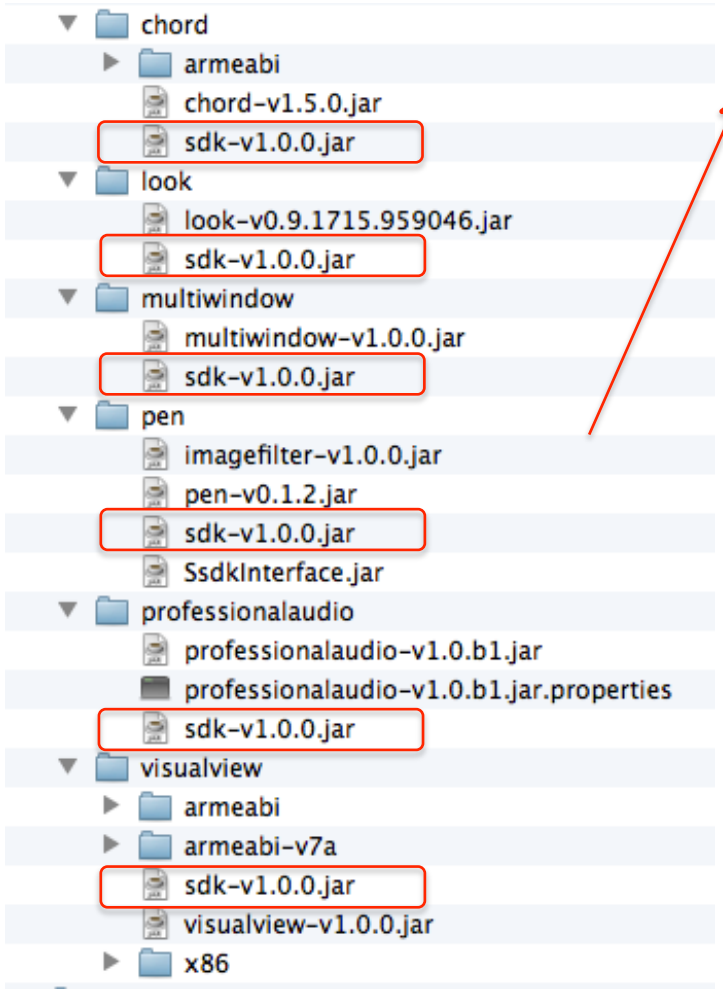
Implementing Drag and Drop

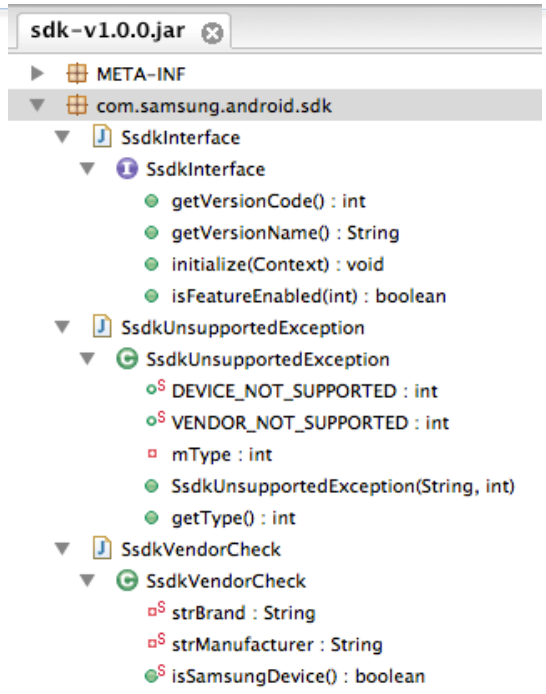
- Uses standard Android drag and drop listeners/ callbacks
  - http://developer.android.com/guide/topics/ui/drag-drop.html

- Uses standard Android clipboard to copy and paste data
  - http://developer.android.com/reference/android/content/ClipData.html

Verifying features

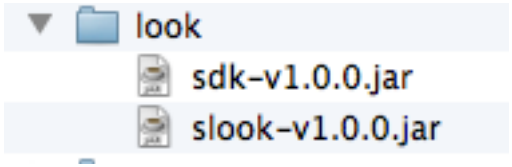
Samsung Mobile SDK Libraries





- New APIs to check for presence and properties of technology

Example -- Implementing Look Features

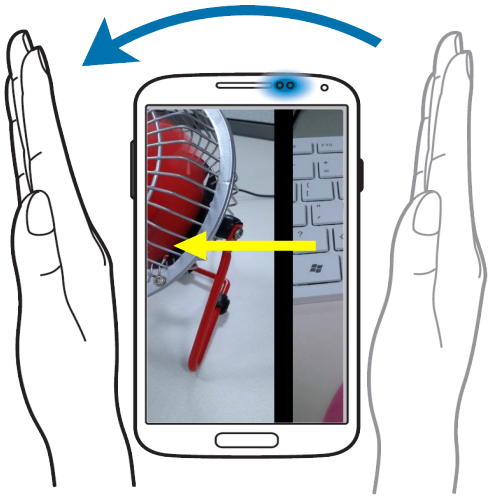



```
@Override
public void onCreate(Bundle savedInstanceState) {

  Slook slook = new Slook();
  try {
     slook.initialize(this);
   } catch (SsdkUnsupportedException e) {
     // notify user device does not support
     return;
  }
  if (slook.isFeatureEnabled(Slook.AIRBUTTON)
  if (slook.isFeatureEnabled(Slook.SPEN_HOVER_ICON)
  if (slook.isFeatureEnabled(Slook.SMARTCLIP)
  if (slook.isFeatureEnabled(Slook.WRITINGBUDDY) {
     Slook<class> variable = new Slook<class> }
```

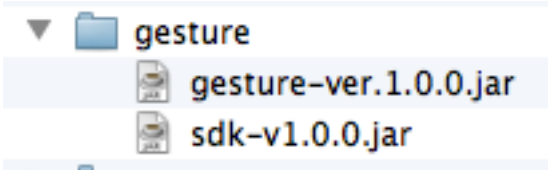
Gesture

What is Gesture?

- Gesture allows you to use events generated by user's hand movements in your application
- Recognizes hand movements using data from the gesture sensor to the right of the device speaker
- Requirements:
  - Gesture sensor
  - Android 4.3 (Jelly Bean API level 18) or higher



Gesture Package



Interface Summary

Interface	Description
SgestureHand.ChangeListener	This listener interface provides callback methods for SgestureHand events.

Class Summary

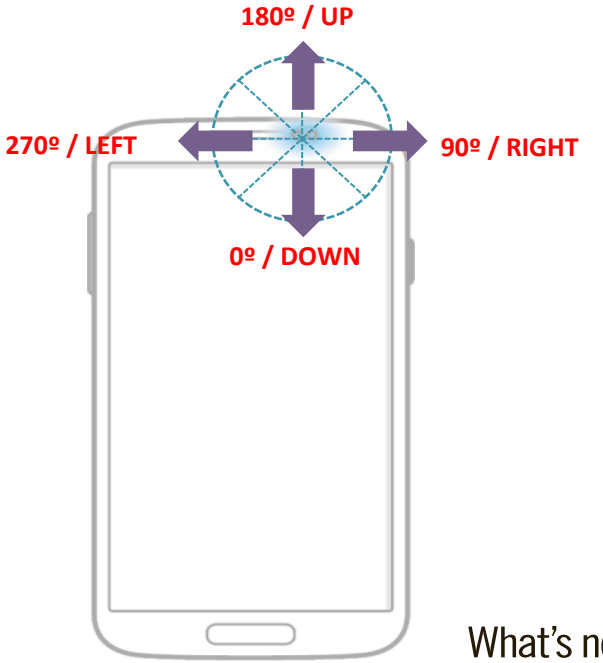
Class	Description
Sgesture	This class provides support for recognizing hand movements.
SgestureHand	This class provides the functionality.
SgestureHand.Info	This class contains SgestureHand event information and provides methods to access the information.

Implementation -- Check for support of feature

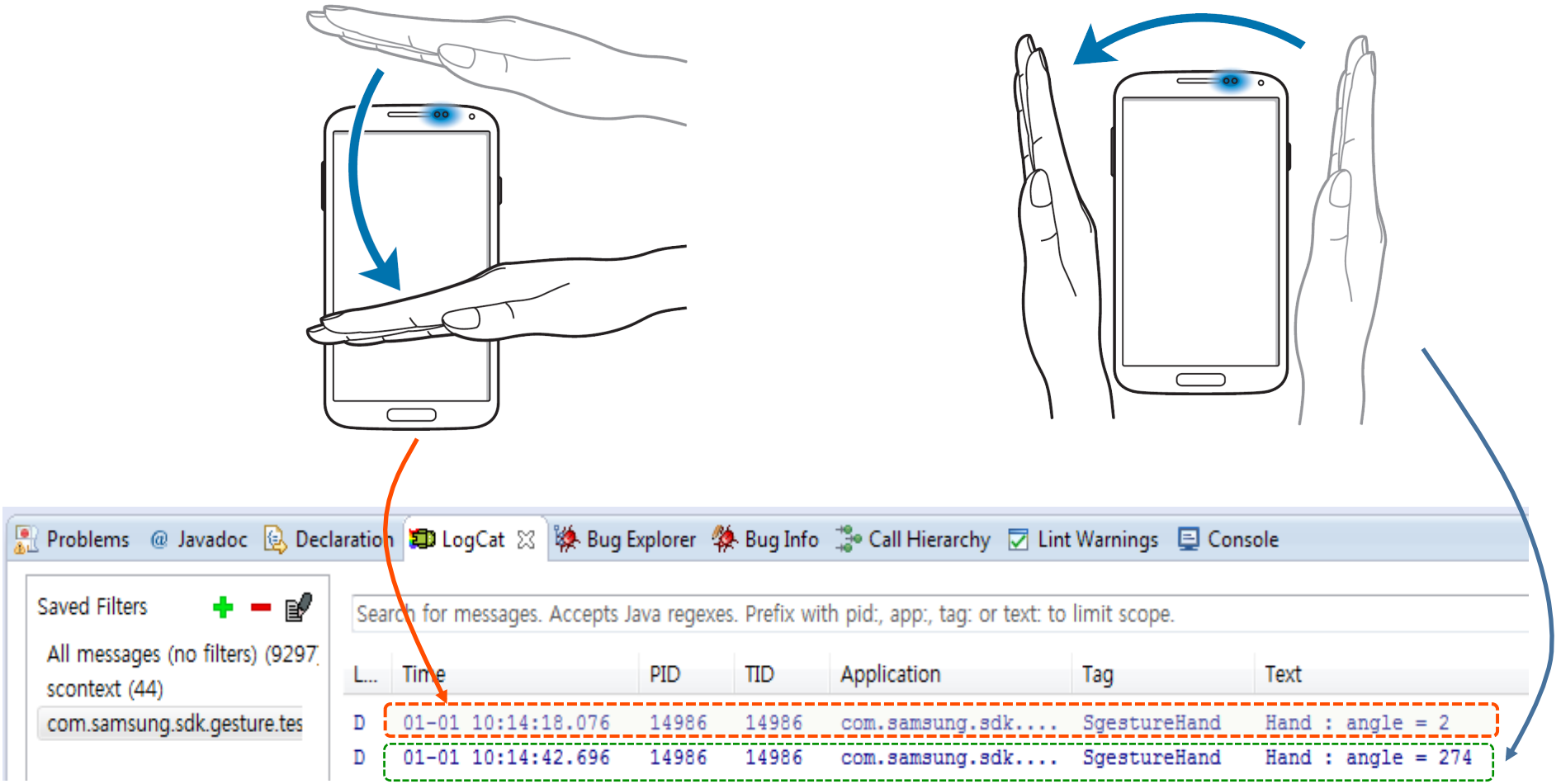
```
public void onCreate() {
    private Sgesture mGesture;
    private SgestureHand mHand;

    mGesture = new Sgesture();

    try {
        mGesture.initialize(this);
    } catch (IllegalArgumentException e) {
        //Error Handling
    } catch (SsdkUnsupportedException e) {
        //Error Handling
    }

    if(mGesture.isFeatureEnabled(Sgesture.TYPE_HAND_PRIMITIVE)){
        mHand = new SgestureHand(Looper.getMainLooper(), mGesture);
        mHand.start(Sgesture.TYPE_HAND_PRIMITIVE, mChangeListener);
    }

}
```

Register Listener

```
private final SgestureHand.ChangeListener mChangeListener =
    new SgestureHand.ChangeListener() {
        @Override
        public void onChanged(Info info) {
            Log.d(TAG, "Hand : angle = " + info.getAngle());
            // do something with data
        }
};


public void onStop() {
    mGestureHand.stop();
}
```



Reporting Angles



Adding Gestures – Exercise 2

1. Add gesture libraries to your project
2. Modify existing app to create global variables and routines
3. Add gesture

[. . .]

Thank You!

License Notices

Except where noted, sample source code written by Samsung Electronics Co. Ltd and provided to you is licensed as described below.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Samsung Electronics Co. Ltd nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
- THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Other source code displayed in this presentation may be offered under other licenses.

Apache 2.0

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0.
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Creative Commons 3.0 Attribution License

Portions of this presentation are reproduced from work created and shared by Google (http://code.google.com/policies.html) and used according to terms described in the Creative Commons 3.0 Attribution License (http://creativecommons.org/licenses/by/3.0/).