1. Density application error - The layout is broken

A. Case

In some applications, the Canvas size of the Drawing area is not properly recognized and a broken layout screen is displayed as a result.

B. Guideline

The Android platform provides DIP (Device Independent Pixels), which represents the size of the screen, as well as the pixel size. This is mapped 1:1 to the dots on the screen, so that the resources are displayed on the device screenin the same size, regardless of device whichisusing the DIP information.

```
 E.g. For the Galaxy S2 (Density value: 240)
         The actual screen size of the device: 800 × 480 px
         The screen size calculated in DIP: 533 × 320 DIP
```

In an application, you can acquire the size of the current screen using the following code.

```
Density = getResources().getDisplayMetrics().density;

         DisplayMetrics dm = getResources().getDisplayMetrics();

         screen_width  = dm.widthPixels;
         screen_height = dm.heightPixels;
```

The acquired screen size is determined according to the application's Density value setting in the AndroidManifest.xml file.

```
E.g. For the Galaxy S2 (Density value: 240)
         Density = getResources().getDisplayMetrics().density;
         DisplayMetrics dm = getResources().getDisplayMetrics();

         // When the Density value is 1.0
         screen_width  = dm.widthPixels;          // Result: 533
         screen_height = dm.heightPixels;         // Result: 320

         // When the Density value is 1.5
         DisplayMetrics dm = getResources().getDisplayMetrics();
         screen_width  = dm.widthPixels;          // Result: 800
         screen_height = dm.heightPixels;         // Result: 480
```

The screen size values are internally applied through the Activity or View in an application. You can acquire the current settings as follows.

```
E.g. For the Galaxy S2 (Density value: 240)
         protected void onDraw( canvas paramCanvas )
         {
               canvas_width = paramCanvas.getwidth();   // Result: 800
```

```
            canvas_height = paramCanvas.getheight(); // Result: 480
```

However, in the current version of ICS, the size values described above may be incorrectly configured to some of the settings in the AndroidManifest.xml file.

For example, if the Attributes for the User SDK information such as Min SDK version, Target SDK version from the supported SDK range settings in the AndroidManifest.xml file are missing, or if the <uses-sdk> attribute settings are located in the wrong location, such as between the <application> and the </application> tags in the AndroidManifiest.xml file, then the following result will be resulted.

```
E.g. For the Galaxy S2 (Density value: 240)

        // When the SDK settings are missing or incorrect in the AndroidManifest.xml file.
        DisplayMetrics dm = getResources().getDisplayMetrics();
        Density = getResources().getDisplayMetrics().density;

        // Density result is 1.0
        screen_width  = dm.widthPixels;      // Result: 533
        screen_height = dm.heightPixels;     // Result: 320

        // In the Galaxy S2 GB(Gingerbread) version,
        // the same size as that of the screen is returned.
       protected void onDraw( canvas paramCanvas )
       {
            canvas_width = paramCanvas.getwidth();   // Result: 533 (Match)
            canvas_height = paramCanvas.getheight(); // Result: 320 (Match)
             ...

        // In the Galaxy S2 ICS updated version, a different size
        // from that of the screen is returned.
        protected void onDraw( canvas paramCanvas )
        {
            canvas_width = paramCanvas.getwidth();   // Result: 800 (Mismatch)
            canvas_height = paramCanvas.getheight(); // Result: 480 (Mismatch)
         ...
         ....
```

As a result, although the screen size in the current application is set to the 533 ×320 DPI coordinate system, some internal components are improperly applied to the 800 ×480 px coordinate system in the ICS environment. Due to the differences in screen sizes, various types of errorsmay occur in the layout.

By correctly setting the attributes for the User SDK in the AndroidManifest.xml file,screen size mismatch errors can be avoided.</uses-sdk>

2. Resource (Layout, Drawable) selection error - The layout is broken

A. Case

i) In some applications, when the layout for a dialog box or menu construction is loaded, a layout that does not match the current display resolution is loaded and the layout of the screen ends up being broken. e.g. the 480 × 320 layout is loaded on the Galaxy S2 with the 800 ×480 resolution

ii) In the current GBenvironment, a Galaxy Note with a 1280 x 800 resolution is recognized as a large sized screen at present. In the ICS environment, the Galaxy Note with a 1280 x 800 resolution is recognized as a normal sized screen. So, afterthe ICS update, the Galaxy Note screen size may be changed from large to normal, andmaynot be displayed correctly.

B. Guideline

Android provides a method to construct a different layout and drawable icon resources by applying size and density filters so that the resources can be used in various device displays.

```
E.g. For the Galaxy S2
                    layout-normal           // normal screen size (4.x inch)
                    drawable-hdpi           // hpi density  (240 DPI)

          For the Galaxy Note
                    layout-large             // large screen size (5.x inch)
                    drawable-hdpi          // xhpi density  (320 DPI)
```

As above, while finding resources that are the most appropriate for the device display,differences between GB andICSlead tounintended resource usage. The layoutbecome broken and other related errors may occur.

For example, if the layout of the application that runs on a Galaxy S2 with a 800 x 480 resolution is set as follows,

```
layout
          layout-480x320
          layout-1024x600
```

and the application attempts to load an item of the layout that isoptimized to the screen size in the res folder, the result differs depending on the platformversion that is installed on the current device.

```
LinearLayout    loadingLinearLayou
                            = (LinearLayout)((LinearLayout)LayoutInflater.from(this)
                                .inflate(PackageName.R.layout.Item, null) ).getChildAt(0);

  // Galaxy S2 GB version:
     The developer finds and loads the item in the "layout" folder.

  // Galaxy S2 ICS updated version:
     Unlike the GB version, the developer finds and loads the item in the "layout-480x320"
folder.
```

```
  For your information, with the conditions above, the Galaxy Nexus ICS version with a 1280
x 720
  resolution selects the layout-1024x600 folder, while the Galaxy Nexus GB version with a
1280 x 720
  resolution selects the layout folder.
```

If developers make "layout" and "layout-1280x720" folders then Galaxy Note with GB will choose "layout" folder, and Galaxy Note with ICS will choose "layout-1280x720" folder.But when we add athird folder "layout-1280x800" then both devices choose this layout. This probalby happensbecausethe GB system looks for theexactresolution of the device for layout, and when the system cannot find it, then it uses thedefault layout.On the other hand,the ICS systemprobably looks for the "nearest" resolution.

1.As above, if the same device calls different resources depending on whether the application is developed for the Android GB or ICS version, various errorsmay be occur.
In the ICS environment, when you construct the layout reflectingvarious attributes such as resolution, size, density of the resources, you have to apply the correct values and priorities.
For example, in the previous example Galaxy S2 ICS updated version, the problem is easily solved by adding the layout-800x480 folder to the res folder.

2.Before Android API 9 and 10, the xlarge screen size and xhdpi density support were not considered and there was ambiguity in the standardsin etermining whether the screen size is normal or large depending on whether the device is a smartphone or a tablet. Therefore, some applications resulted inlayout errors in the ICS environment of some devices with special screens.
e.g. Galaxy Tab 7 inch, Galaxy Note

To solve this problem, selecting resources by distinguishing the screen size as small, normal or large was deprecated after Android 3.0 and new ways to distinguish between and select resources were introduced on the Android developer site.

When you develop an application on a device that has an unusual screen size, you have to construct resourcesby distinguishing them on the basis of the latest guide introduced in the Android developer guide.

C. Related links

1. The android developer site provides various guides to cope with cases where applications running on a device fail to choose resources that are intended by the developer.

- Providing Alternative Resources
  http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources
- New Tools For Managing Screen Sizes
  http://android-developers.blogspot.com/2011/07/new-tools-for-managing-screen-sizes.html

2. The Android developer site has announced that there could be an error distinguishing normal and large sized screens because there is an ambiguity in the screen size standards in Android versions earlier than Android 3.0.

Range of supported screens
http://developer.android.com/guide/practices/screens_support.html#range

- xlarge screens are at least 960dp x 720dp
- large screens are at least 640dp x 480dp
- normal screens are at least 470dp x 320dp

•small screens are at least 426dp x 320dp

These minimum screen sizes were not as well defined prior to Android 3.0, so you may encounter some devices that have been misclassified between normal and large.

3. Functions in some game engines fail in the ICS platform

A. Case

i) When running a game,an image is broken in the splash screen or game is abnormally terminated whenrotating the device screen.

•Some cases using the Unity game engine

ii) In the ICS environment, some game applicationsrandomly stopwhile running.

•Some cases using the Marmalade game engine

iii) When running some games, pressing the Back button returns to the initial game screen and not to the previous screen.

•Some cases using the Corona game engine

B. Guideline

i) When running some gamesusing the Unity game engine, the splash images are broken and incorrectly displayed in both the Galaxy S2 ICS updated versionand Galaxy Nexus devices.

Our tested games displayed the screen using the splash.PNG file of the same resource path and an error occurs during this step.

In this case, there is no problem in the GB environment and with the resources. We presume that this is due to an internal problem of the Unity engine on the basis of the following log message output by the Unity engine while the gameis running.

```
.....
WARN / Unity (4476): image error       // Unity engine's log
.....
```

In addition, in the Unity engine developer forum, some issues regarding the splash screen settingssuch astermination, frozen screen have been reported and the solution is being prepared through an engine update.

On the basis of the above facts, we believe that there is an incompatibility with the ICS for some functions in some versions of the Unity engine or some screen settingsselected by developers using the Tool provided by the Unity engine that causes a problem.

In addition, some developers reported that there is an error due to the setting for the rotation function that has been changedin API 13 of the Unity engine.

The general setting method for checking screen orientation has been changed in ICS.

```
In the Androidmanifest.xml file:  android:configChanges="orientation"
```

However, for some games that are developed using the Unity engine versions to which this change is not applied, the gameis terminated abnormally when the screen orientation is changed, according to the reports ofsome developers on the Unity developer forum site.

ii) Some games developed using the Marmalade game engine, randomly stop while running. Since the symptom could not be reproduced in the test of the sample application even after multiple attempts,or after disconnecting and then reconnecting the Wi-Fi connection, we presume that the reason for this is an internalmemory management problem or a network data processing thread handling problem of the application.

Since the Marmalade engine supports multiple platforms and games are developed using the APIs provided by the engine, it still needs to be checked if there areany ICS compatibility problems in the APIs of the Marmalade engine.

Recently, a few runtime errors have been reported on the support boardof the Marmalade engine developer forum and the errors are being solved by an engine update.

iii) In some games that are developed using the 2D game engine Corona, pressing the Back button returns to the initial game screen and not to the previous screen.

This game engine internally uses the CoronaGLSurfaceView class. And, they are internally handling the GL thread and GL surface destruction , restoration processes following the Back key input. Some errors may be occurs in the internal process. The version of the Corona engine used by games that have a problem is 1.0 and seems to be developed on the basis of API 8. Further issues can be identifiedin the ICSenvironment.

C. Related links

i) The Android developer site provides information about the attribute change according to the orientation change.

http://developer.android.com/guide/topics/manifest/activity-element.html

> If your application targets API level 13 or higher
> (as declared by the minSdkVersion and targetSdkVersion attributes),
> then you should also declare the "screenSize" configuration, because it also changes when a device switches between the portrait and landscape orientations.

The articles regarding thesome issues posted on the Unity forum are as follows.

http://answers.unity3d.com/questions/64632/unity-android-splash-screen-hangs-on-tegra-2-motor.html http://answers.unity3d.com/questions/64632/unity-android-splash-screen-hangs-on-tegra-2-motor.html http://answers.unity3d.com/questions/63384/android-testing-screen-error.html http://answers.unity3d.com/questions/132073/android-game-crash.html http://answers.unity3d.com/questions/197400/unity-3d-crashes-on-android.html http://forum.unity3d.com/threads/116250-ICS-Orientation-issue-(Ice-Cream-Sandwish

The articles regarding the ICS issue posted on the Marmalade forum are as follows.

http://www.madewithmarmalade.com/devnet/forum/7152 http://www.madewithmarmalade.com/devnet/forum/7046

4. Using a deprecated function or inappropriate thread handling

A. Case

i) In some news and magazine applications using HTTP communication, a NetworkOnMainThreadException error occurs and the application is terminated abnormally while data is received over the network.

ii) In some applications, an abnormal case occurs when exiting by pressing the Back key during a normal operation.

iii) In some news and magazine applications using HTTP communication, a broken layout is displayed, a NullPointExeption error occurs and the application is terminated abnormally during or just after receiving data over the network.

B. Guideline

i) From Android 3.0, new exceptions have been added to prevent an ANR from occurring when applications are located in the UI thread.
If a network thread is used in the wrong location, an Android.os.NetworkOnMainThreadException error may occur.

The system control has been reinforced so that if the UI main thread performs a thread task that requests network access in an application, the system recognizes this as a critical error and may terminate the program depending on the error level.
Therefore, no applications should access the network in a UI thread and should access the network using a separate thread or AsyncTask to prevent this.

ii) On the android platform, the Back key is used to exit an application. Some applications attempt to terminate a thread using the deprecated stop() operation in the exiting part and this causes an exception when the application is being terminated.

```
E.g,  ...
       m_Thread.stop()     // Use of the thread stop() function has been
                                   //  deprecated because of possible errors.
       or
       m_Thread = null;
       ...

       // The thread should be terminated using Interrupt() as shown below.
       ...
       if ( m_Thread!= null && m_Thread.IsAlive() )
            m_Thread.Interrupt();
```

iii) In some applications, a NullPointerException error occurs during or just after downloading data via HTTP. The most critical reason for application errors in the ICS environment, unlike the GB environment, has been identified as a failure to parse data received through an HTTP connection in the application through an analysis of the log data.

Since this is not a data reception problem through an HTTP connection but is an application problem where received data is not parsed in the ICS environment, a run-time error occurs because null or another incorrect value is included in the contents after the parsing operation.

For example, in some news and magazine applications, it has been identified through the log file that after downloading the article data via HTTP, only the start tag is added and the end tag is not processed in the HTML parsing process.

In this case, even if 100 articles are downloaded over the network, only one article is not properly displayed on the screen.

Although there could be several reasons for the incorrect parsing operation, some classes and APIs used for the parsing operation could be the reason for this.

The StringBufferInputStream class that was used for parsing in some applications that caused the problem has been identified as not being able to properly convert text strings in the buffer data and has been deprecated.

```
....
  XmlPullParser parser = Xml.newPullParser();
  StringBufferInputStream sbi = new StringBufferInputStream( sBuffer );
  parser.setInput(sbi, "utf-8");
  ....
```

For your reference, since the StringBufferInputStream class generates a NullPointerException error when there is null in the input parameters, you can refer to the location of the occurrence of the NullPointException error to identify the location of the problem.

C. Related links

i) The Android developer site provides a developer guide about using threads.

ii) The Android developer site provides information about the deprecated stop() operation for Threads.

http://developer.android.com/reference/java/lang/Thread.html

```
void  stop()
         This method has been deprecated, because stopping a thread in this
         manner is unsafe and can leave your application and the VM in
         an unpredictable state.
```

iii) In the Android developer site, you can view the deprecated StringBufferInputStream class information.

http://developer.android.com/reference/java/io/StringBufferInputStream.html

5. Others

A. Case

i) Some applications using the camera preview display are abnormally terminated in the ICS environment.

ii) In some ICS devices, errors occur in some functions that perform the auto-scaling operation according to the display density of devices. The auto-scaling operation of the image is not performed and the original image is displayed as is.

B. Guideline

i) Specifying null as the argument for the PreviewDisplay method to use the camera preview data without the overlay function on the Android platform that was allowed until Honeycomb is not allowed in the ICS environment.

Therefore, if the following code is used in some applications using the camera, the applications are abnormally terminated.

```
 camera.setPreviewDisplay( null );
 camera.StartPreview();
```

From now on, applications using the camera preview function should use a valid View as the argument.

ii)Due to a setting in the Androidmanifest.xml, an API does not work.
For example, if the TargetSdk value is set to 14 in the Androidmainfest.xml file and the setMeasuredDimension method is executed, which is applied to the auto-scaling operation of a bitmap image, the method does not work on some devices.

```
...
final DisplayMetrics metrics = getResources().getDisplayMetrics();

setMeasuredDimension( mBitmap.getScaledWidth( metrics ),
                                        mBitmap.getScaledHeight( metrics )  );
...
```

ref:http://developer.samsung.com/android/technical-docs/Guideline-for-resolving-frequent-errors-when-porting-apps-to-Android-4-0-1