MEDIAS W N-05E

アプリ作成ガイドライン

第 1.0 版

2013 年 4 月 18 日

株式会社 NTT ドコモ

目次

商標について

改版履歴

版	項目	種別	内容
0.5	—		初版作成
0.6	全般	修正	全般的に表現を修正しました。
0.9	4.2	修正	参照元誤りを修正しました。
1.0	3 章	追加	非アクティブ時の動作について追加しました。
	4 章	修正	アプリケーション実装時の注意点について誤記を修正しました(「画像保存後のファイル表示」を「画像保存前のファイル表示」に修正)。
	4.1	追加	図 19 の補足説明を追加しました。
	4.3	修正	図 22 の `matrix.postRotate()`の引数の誤りを修正しました。

1章　はじめに

本ドキュメントは、2013 年春モデルとして発売する、「docomo NEXT series MEDIAS W N-05E」（以降、「N-05E」と表記します）に向けてアプリケーションを提供するにあたっての注意事項を解説したものです。

このドキュメントは、N-05E の特有の振舞いを考慮した Android アプリケーション開発について解説しています。したがって、本ドキュメントは Android アプリケーション開発の基礎的な知識を有する読者を対象としています。

このドキュメントでは、2 章にてN-05Eの画面表示に関する仕様を解説します。3 章以降では、アプリケーション開発時に特に注意すべき事柄について解説します。

2章　N-05E 仕様

本章では、N-05E 独自の拡張仕様のうち、画面表示に関する部分について解説します。

2.1. 3 つの表示モード

N-05E は 2 つの物理ディスプレイを搭載しています。N-05E には、この 2 つの物理ディスプレイを有効に利用できるように 3 つの表示モードが搭載されています。

- シングルモード
  端末が閉じられた状態で、1 つの画面のみを利用するモードです。
  このモードではアプリケーションは通常のスマートフォンと同様に動作し、またユーザも通常のスマートフォンと同様に操作することができます。
  シングルモード時に画面表示されるディスプレイを以降「メインディスプレイ」と表記します。また、もう一方のディスプレイを「サブディスプレイ」と表記します。
- ダブルモード
  端末が開かれた状態で、2 つのアプリケーションがそれぞれの画面上で動作するモードです。
  このモードでは、メインディスプレイ上で任意のアプリケーションを実行することができます。
  サブディスプレイには常に N-05E 固有のアプリケーションである「Utility Apps」のみが表示されます。
- フルスクリーンモード
  端末が開かれた状態で、2 つの画面を 1 つの大きな画面として利用できるモードです。
  このモードでは、メインディスプレイとサブディスプレイを足し合わせた 1 画面として、通常のスマートフォンと同様に利用できます。

なお、N-05E には上記以外に、端末を 90 度程度に開くことで 2 つのディスプレイに同じ画面を映し出す「W ムービー」や、音声着信時、カメラ起動時など、端末を閉じた状態でサブディスプレイに画面を表示するアプリケーションもありますが、アプリケーション開発者が同様の動作を行うアプリケーションを作成することはできません。

ブラウザアプリを起動した状態を例に、3 つの表示モードを図 1に示します。
ダブルモードとフルスクリーンモードでは、サブディスプレイにダブルモードとフルスクリーンモードを切り替えるアイコン(拡大・縮小アイコン)が表示されます。

図 1 表示モード一覧

2.2. 表示モードの遷移

前節で解説した 3 つの表示モードがどのように変化するかを図 2に示します。

N-05E には、端末を開いた際にダブルモードで表示するかフルスクリーンモードで表示するかを選択・設定する機能が搭載されています。デフォルトの設定値は「直前のモードで起動」です。

ダブルモードとフルスクリーンモードの切り替えは、サブディスプレイに表示される拡大・縮小アイコンをタップすることで行います。

図 2 表示モード遷移

2.3. アクティブの見え方・切り替え方法

N-05E では、Android 互換性を保つため、端末内でアクティブ(Resumed)状態のアプリは 1 つのみ存在できるようにしており、2 つのアプリがそれぞれのディスプレイに表示されているダブルモードでは、どちらか一方のみがアクティブ状態となり、他方のアプリは非アクティブ(Paused)状態となります。
N-05E では、どちらの画面に表示されているアプリがアクティブな状態かを判別できるように、アプリがアクティブな状態の画面の下部に、以下の 2 種類が目印として表示されます。

- アクティブな状態であることを示すためのライン(アクティブライン)
- バックキー

これらの目印の表示によって、ユーザはどちらのアプリがアクティブ状態かを判別することができます。非アクティブ状態のアプリの表示画面をタップすることで、その画面がアクティブ状態となり、他方の画面が非アクティブ状態に切り替わります。

なお、シングルモードではアクティブな状態であることを示すアクティブラインは表示されません。また、フルスクリーンモードではメインディスプレイとサブディスプレイ双方にアクティブラインが表示されます。

図 3 アクティブ画面表示

2.4. ディスプレイサイズと傾き

N-05E に搭載されている 2 つのディスプレイは共に同じ画面解像度、スクリーン密度、ディスプレイサイズです。

項目	仕様
画面解像度(ピクセル数)	540x960
スクリーン密度(dpi)	hdpi(240dpi)
ディスプレイサイズ	約 4.3 インチ

N-05Eを開いた際、2 つのディスプレイを合わせると画面解像度は 1080x960 ピクセルになります。そのためフルスクリーンモードでは、N-05E本体の向きを変えずに端末を開くと、画面の向き(Orientation)の縦(Portrait)、横(Landscape)が入れ替わります(図 4)。
N-05E を開いた際にシングルモードからフルスクリーンモードに遷移する場合においては、画面の向きが変更されたという理由で、`Activity` はライフサイクルプロセスに従い一旦終了し、その後 `Activity` が再生成されます。また、同時に画面サイズも変更になるため、`Activity` の再生成を回避するには、AndroidManfest.xml の `activity` 要素の `android:configChanges` 属性に"`orientation`"および"`screenSize`"を追加する必要があります。

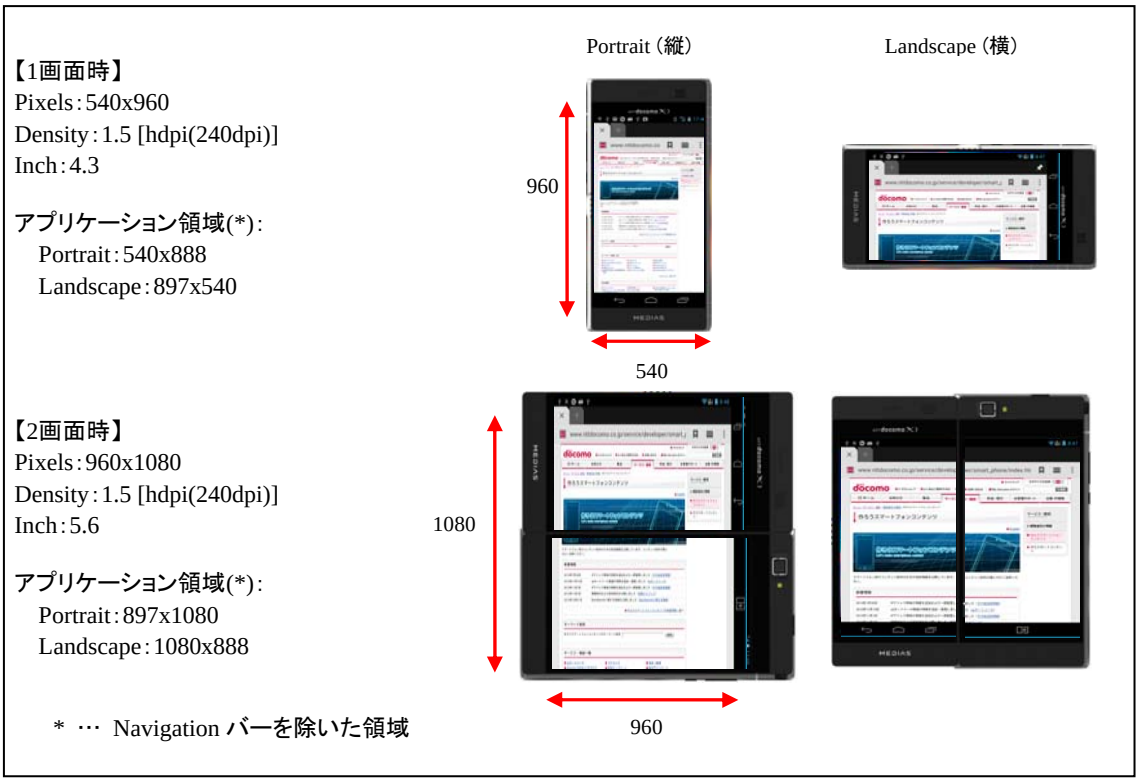


図 4 ディスプレイ仕様

3章　Activity のライフサイクル

ダブルモードやフルスクリーンモードであっても、Activity のライフサイクルは Android 仕様に従います。そのため、N-05E でも、アプリケーションは端末の表示モードを意識することなく動作することができます。
ダブルモード中は、それぞれのディスプレイに表示されているActivityのうち、一方がアクティブ(Resumed)状態となり、他方は非アクティブ(Paused)状態となります。ダブルモード、フルスクリーンモード間の遷移とActivityのライフサイクルの関係を図 5に示します。

図 5では、メインディスプレイで最前面に表示されるActivityを「Activity A」、バックグラウンドで生存しているActivityを「Activity B」としています。

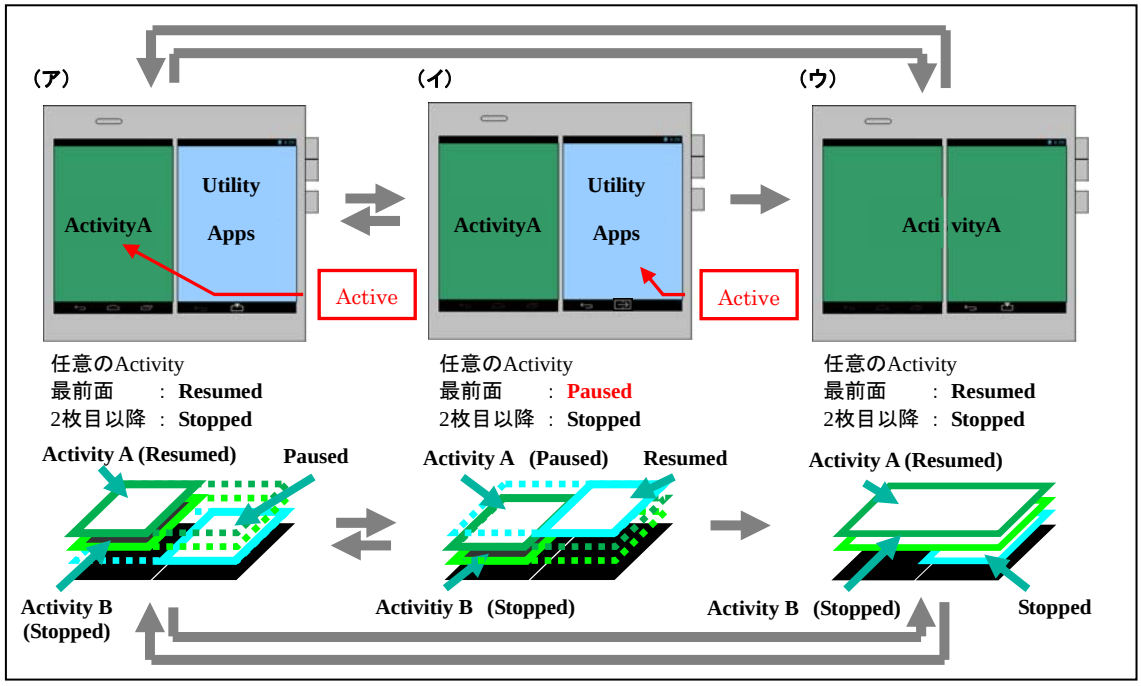


図 5 表示モード遷移とアプリ状態

- 図 5の(ア)
  表示モードがダブルモードで、メインディスプレイに表示されているActivity Aがアクティブ(Resumed)状態です。このとき、サブディスプレイのUtility Appsは画面全体が表示されてはいますが非アクティブ(Paused)状態です。
  また、最前面に表示されていないActivity Bは、通常のスマートフォンと同様、停止(Stopped)状態です。
- 図 5の(イ)
  表示モードがダブルモードで、サブディスプレイに表示されているUtility Appsがアクティブな

状態です。このとき、Activity Aは非アクティブ状態で、Activity Bは(ア)と同様、停止状態です。

- 図 5の(ウ)
  フルスクリーンモードでActivity Aがアクティブ状態です。Activity Bは停止状態で、シングルモードと共に、通常のスマートフォンと同様です。

以降では、ユーザ操作による表示モードや開閉状態の変化によって、メインディスプレイに表示される Activity の状態がどのように変化するかを解説します。

○非アクティブの画面をタップした場合

表示モードがダブルモード、かつActivity Aがアクティブ状態でサブディスプレイがタップされると、Activity AのonPause( )メソッドが呼び出され、Activity Aは非アクティブ状態に遷移します(図 6 (ア))。

その後、メインディスプレイがタップされると、Activity AのonResume( )メソッドが呼び出され、アクティブ状態に遷移します(図 6 (イ))。

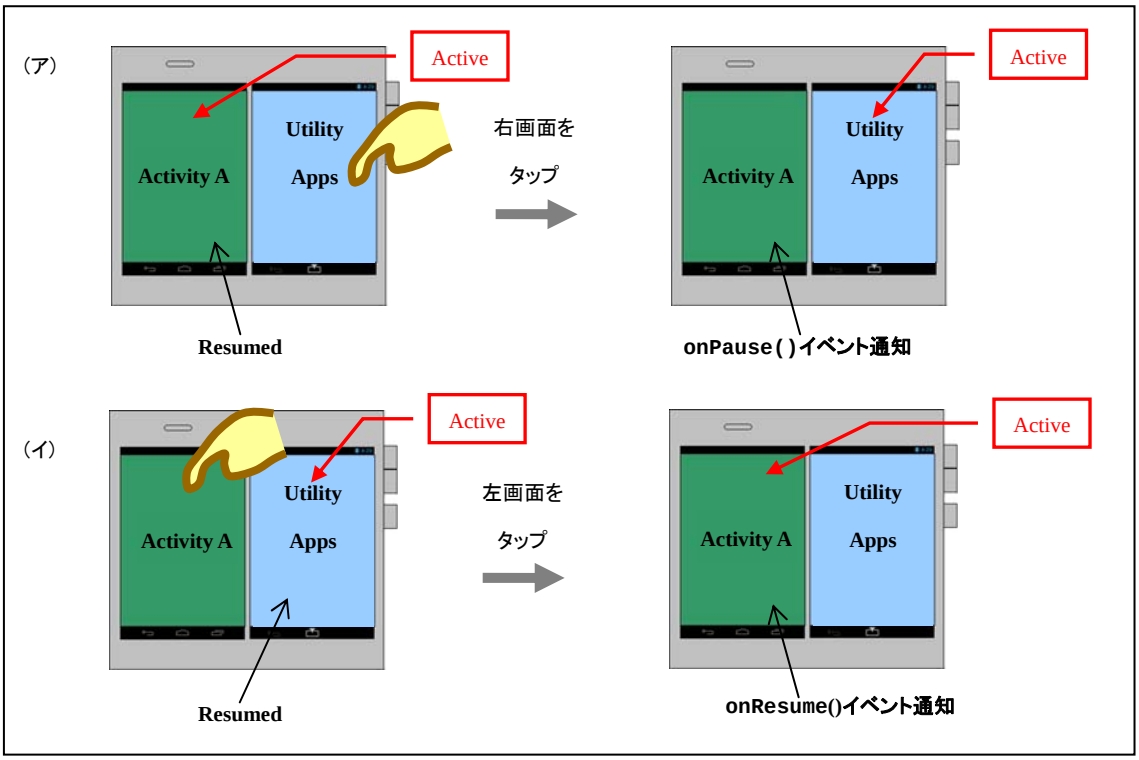


図 6 非アクティブの画面をタップした場合

○アクティブ状態で端末を閉状態にした場合

表示モードがダブルモード、かつ Activity A がアクティブ状態で端末が閉状態になっても、Activity A の状態はアクティブ状態から変化しません。

この遷移の様子を図 7に示します。

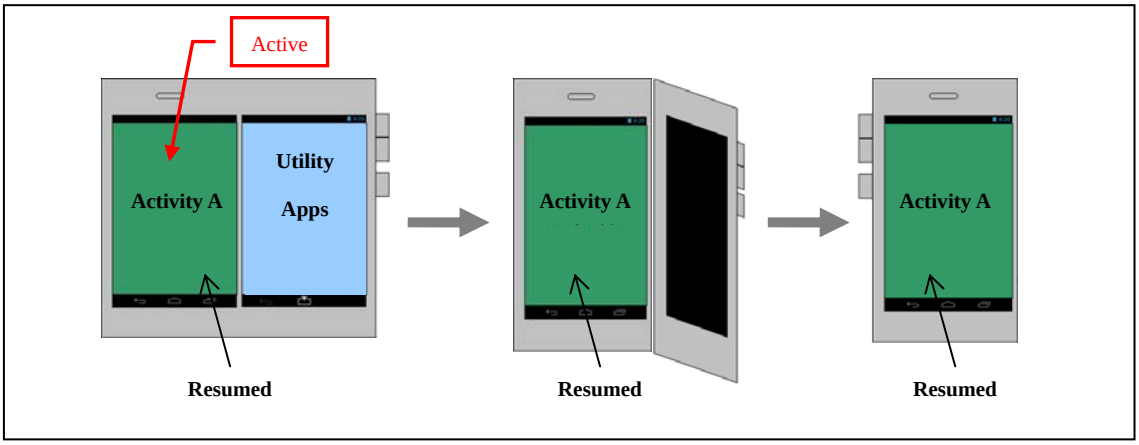


図 7 アクティブ状態で端末を閉状態にした場合

○非アクティブ状態で端末を閉状態にした場合

表示モードがダブルモード、かつ Activity A が非アクティブの状態で端末が閉状態になると、Activity A の onResume()メソッドが呼び出され、アクティブ状態に遷移します。

この遷移の様子を図 8に示します。

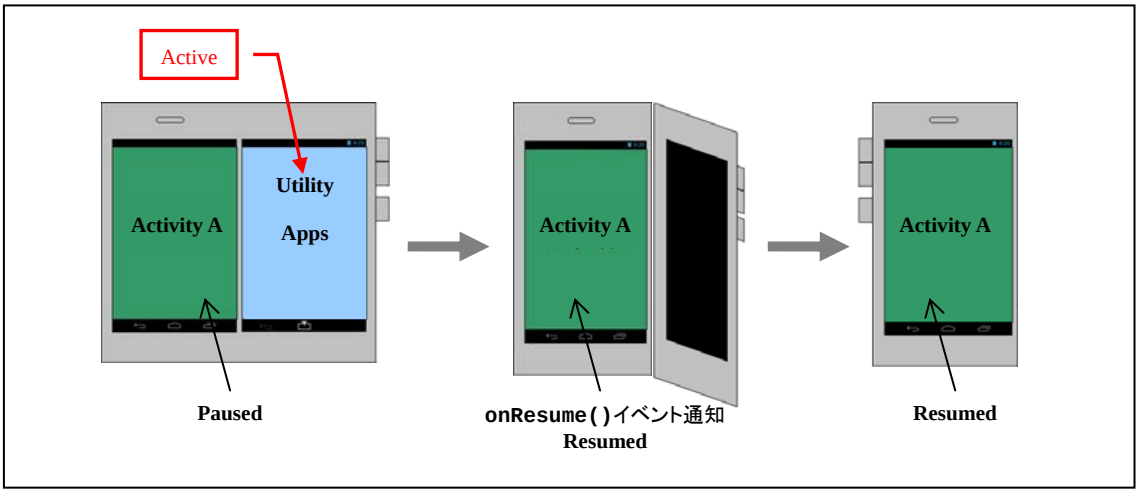


図 8 非アクティブ状態で端末を閉状態にした場合

〇端末を開状態にしてダブルモードに遷移する場合

Activity A がシングルモードで動作中に、端末が開状態になっても、Activity A の状態に変化はなく、ダブルモードに遷移します。

この遷移の様子を図 9に示します。

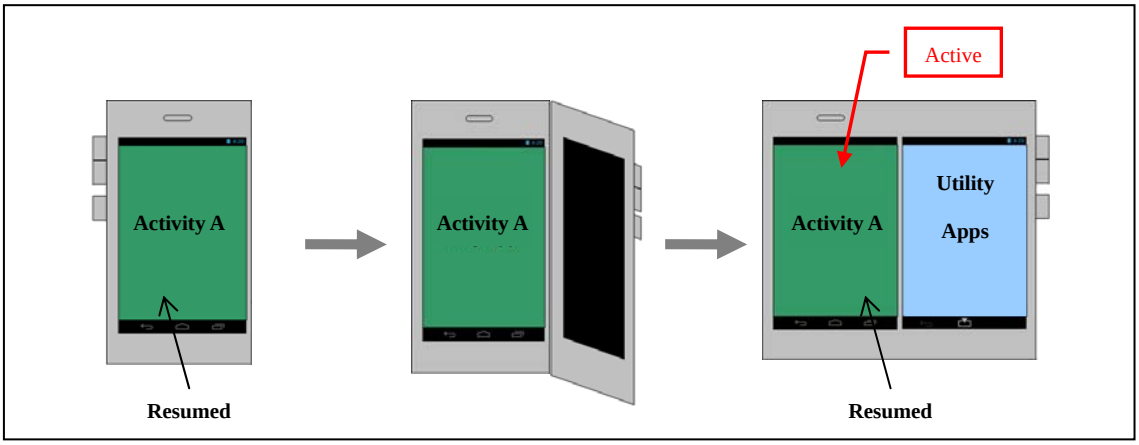


図 9 開状態にしてダブルモードに遷移する場合

ただし例外として、端末の電源 ON 直後など、Utility Apps が新たに起動されなおす場合はサブディスプレイ側の Utility Apps がアクティブになります。そのため、Activity A の `onPause()`メソッドが呼び出され、Activity A は非アクティブ状態に遷移します。

〇端末を閉状態で Utility Apps 起動中に開状態にした場合

Utility Apps がシングルモードで動作中に、端末が開状態になりダブルモードに遷移する場合、Utility Apps のバックグラウンドで停止状態の Activity A に対して、以下のメソッドが順に呼び出され、アクティブ状態に遷移します。

1. `onRestart()`
2. `onStart()`
3. `onResume()`

この遷移の様子を図 10に示します。

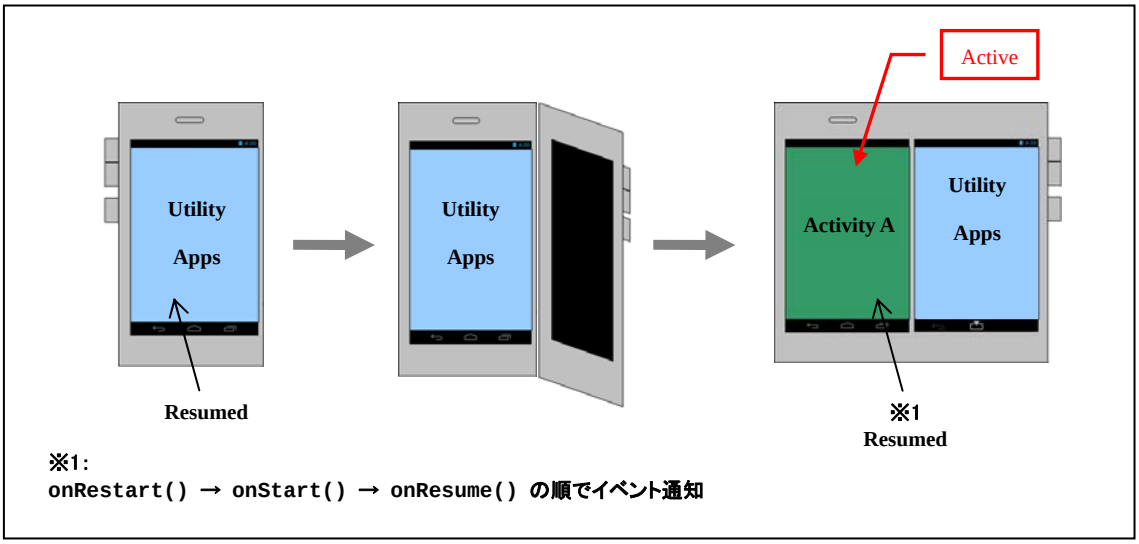


図 10 閉状態で Utility Apps 起動中に開状態にした場合

○端末を開状態にしてフルスクリーンモードに遷移する場合

Activity A がシングルモードで動作中、端末が開状態になりフルスクリーンモードに遷移する場合、表示モードが変化することに加えて、画面の向きも変化します。また、フルスクリーンモードで端末が閉状態になるとシングルモードになることに加えて、画面の向きも変化します。

シングルモードとフルスクリーンモードとの間で表示モードが切り替わる際、ActivityInfo クラスの CONFIG 定数のうち、以下のものの変化が生じ、Activity A が再生成されます。

- CONFIG_ORIENTATION
- CONFIG_SCREEN_LAYOUT
- CONFIG_SCREEN_SIZE
- CONFIG_SMALLEST_SCREEN_SIZE

また、一般的な端末と同様に、Activity A の再生成の過程で、Activity A に対して以下のメソッドが順に呼び出されます。

1. onPause()
2. onStop()
3. onDestory()
4. onCreate()
5. onStart()
6. onResume()

N-05E でシングルモードとフルスクリーンモードとの間で表示モードが切り替わる際の Activity の再生成を抑止するには、以下を android:configChanges 属性に指定します。

- orientation
- screenSize

なお、android:configChanges 属性の指定によって Activity の再生成を抑止している場合、Activity が再生成されるかわりに、Activity.onConfigrationChanged()メソッドが呼び出されます。

この遷移の様子を図 11に示します。

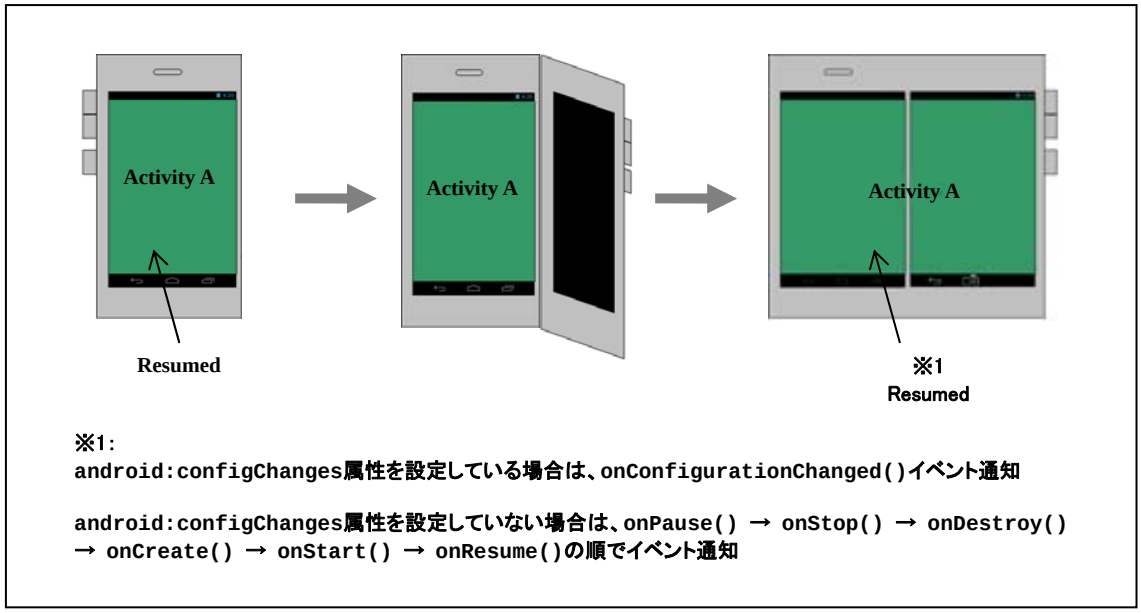


図 11 開状態にしてフルスクリーンモードに遷移する場合

○端末が開状態で拡大・縮小アイコンをタップした場合

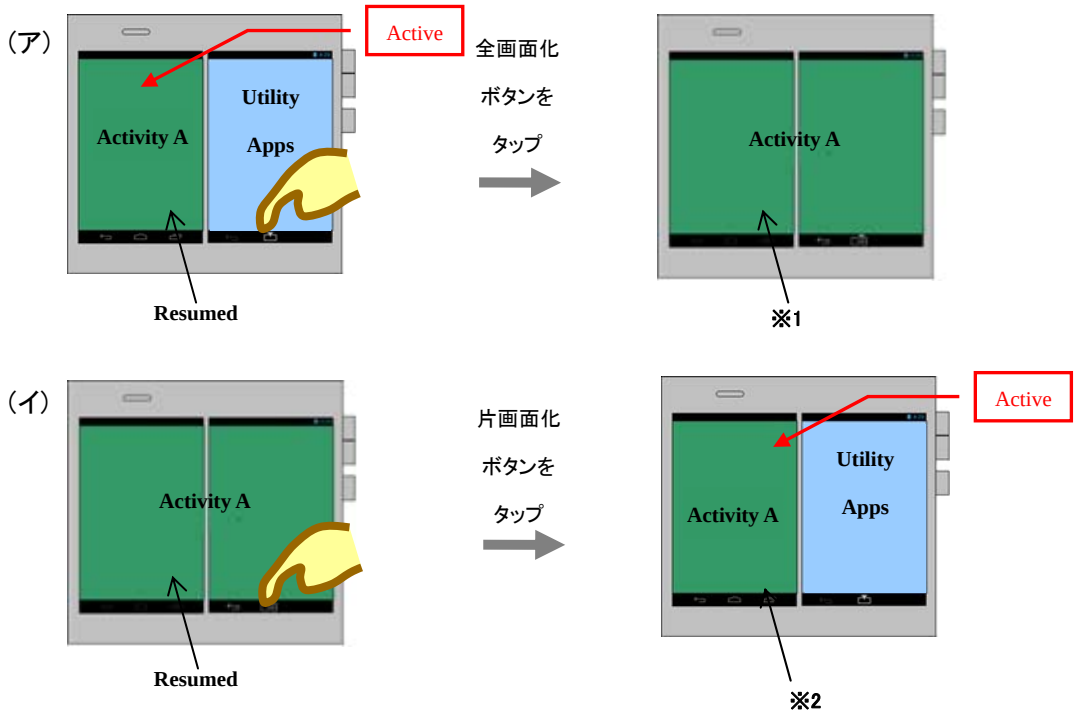
Activity Aがダブルモードで動作中、拡大・縮小アイコンがタップされた場合、表示モードがフルスクリーンモードになります(図 12 (ア))。また、表示モードがフルスクリーンの状態で拡大・縮小アイコンがタップされた場合、表示モードはダブルモードになります(図 12 (イ))。
ダブルモードとフルスクリーンモードとの間で表示モードが切り替わる際、シングルモードとフルスクリーンモード間の切り替え時と同様に、ActivityInfo クラスの CONFIG 定数のうち、以下のものの変化が生じ、Activity A が再生成されます。

- CONFIG_ORIENTATION
- CONFIG_SCREEN_LAYOUT
- CONFIG_SCREEN_SIZE
- CONFIG_SMALLEST_SCREEN_SIZE

ダブルモードとフルスクリーンモードとの間で表示モードが切り替わる際の Activity の再生成を抑止したい場合、シングルモードとフルスクリーンモード間の切り替え時の Activity 再生成の抑止方法と同様、以下を android:configChanges 属性に指定します。

- orientation
- screenSize

この遷移の様子を図 12に示します。



※1:
android:configChanges属性を設定している場合は、onConfigurationChanged()イベント通知

android:configChanges属性を設定していない場合は、onPause() → onStop() → onDestroy() → onCreate() → onStart() → onResume()の順でイベント通知

※2:
android:configChanges属性を設定している場合は、onConfigurationChanged()イベント通知

android:configChanges属性を設定していない場合は、onPause() → onStop() → onDestroy() → onCreate() → onStart() → onResume()の順でイベント通知

図 12 開状態で拡大・縮小アイコンをタップした場合

非アクティブ(Paused)状態の動作について

前述したとおり、N-05E ではダブルモードで動作中にサブディスプレイをタップすることにより、メインディスプレイ側のアプリケーションは非アクティブ(Paused)状態となります。
非アクティブ状態中は描画処理など動作を停止するようなアプリケーションの場合、サブディスプレイ側の操作が行われている間、メインディスプレイ側にアプリケーション画面全体が表示されたまま、アプリケーション動作が停止しているように見えます。これがアプリケーションとしては意図通りの挙動であっても、この状態がユーザに見え続けることが望ましくない場合があります。
このような状態を回避するには、非アクティブ状態中であることをユーザが判別できるような表示を行うといった方法があります。

また、別のアプローチとして、以下のような方法も考えられます。

- onResume()メソッドで処理は行わない。onResume()メソッドでの処理は onStart()メソッド内の既存の処理を行った後の位置に移動する。
- onPause()メソッドでは処理は行わない。onPause()メソッドでの処理は onStop()メソッド内の既存処理を行う前の位置に移動する。

この方法により、アプリケーションは非アクティブ状態へ遷移しても、アクティブ状態と同等の動作を継続するようになります。反面、アクティブ状態から非アクティブ状態へ状態遷移したときにアプリケーションの動作を停止する契機を失うことになります。そのため、たとえばゲームアプリケーション実行中にアラームのポップアップなどの割り込みが発生した場合、ユーザが一時的にゲームアプリケーションの操作ができなくなるにもかかわらず、ゲームアプリケーションは停止せずに進行してしまうといったことが発生する可能性があります。この点を考慮にしたうえで対処の実施要否を検討してください。

4章　カメラアプリにおける注意事項

本章では、N-05E にカメラアプリを提供する際の注意事項について解説します。特にフルスクリーンモードでカメラ機能を利用する場合のアプリケーションの振舞いには注意する必要があります。

理解の手助けのため、本章内の解説で使用する座標系を以下のように定義します。
端末がシングルモードかつPortraitの状態を基準とし、x方向はディスプレイ面を左から右に向かう方向、y方向はディスプレイ面を下から上に向かう方向、z方向はディスプレイの奥(サブディスプレイ側)から手前に向かう方向とします。また、端末の向きにあわせて座標軸の向きも変化するものとします(図 13)。

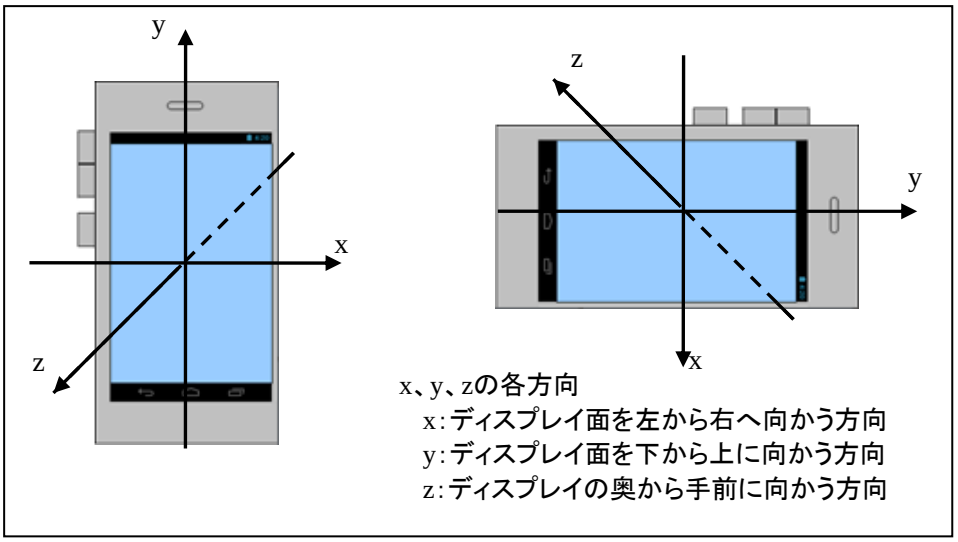


図 13 座標軸の定義

y軸が地面に垂直になるように端末を置いた際、シングルモードやダブルモードでは表示領域の解像度は 540x960 で縦長(Portrait)です。しかし、y軸の向きを固定したままフルスクリーンモードにすると解像度が 1080x960 と横長(Landscape)になります。このように画面の向き(Orientation)は表示モードによって変化します(図 14)。

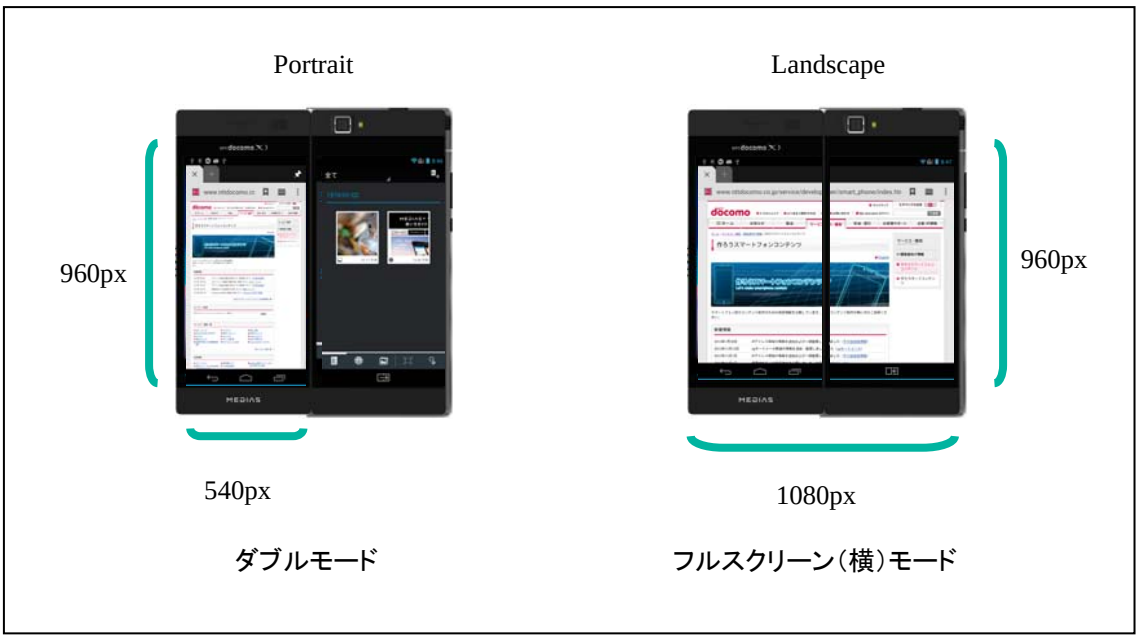


図 14 表示モードと画面の向き

N-05E の持つ、このような特性によって、アプリケーションの実装によっては、フルスクリーンモードでプレビューした際、画像が意図しない向きに回転した状態で表示されたり、画像が画面に表示される際のアスペクト比が実際に撮影した画像のアスペクト比と異なったりする場合があります。

このような振る舞いにならないようにアプリケーションを実装するための注意点には以下のようなものがあります。

- カメラビューの向きに対するカメラセンサの回転補正
- プレビュー表示のアスペクト比補正
- 画像保存前のファイル表示

以降、本章では上記注意点を順に解説します。

なお、Android 標準カメラアプリでは、N-05E の持つ特性に対しても正しく動作するように考慮されていますので、本解説とあわせて標準カメラアプリのソースコードも参照することを推奨します。
Android の標準カメラアプリのソースコードは以下のサイトより取得可能です。

Android Open Source Project - http://source.android.com/

4.1. カメラセンサの回転補正

これまでに解説したとおり、ダブルモードとフルスクリーンモードでは画面の向き(Orientation)が異なります。
ダブルモードでは Portrait 時における天の向きは y 軸正方向ですが、フルスクリーンでは Portrait 時における天の向きは x 軸負方向となります。
しかし、カメラセンサの基準方向(`CameraInfo.orientation`で取得できる値が 0 となる方向)は、ダブルモードでもフルスクリーンモード(Landscape時)でもy軸正方向で固定です(図 15)。

※`CameraInfo.orientation` には、現在の表示モードにおける Portrait 時の天の向きがカメラセンサの基準方向と一致するのに必要な角度が設定されます。

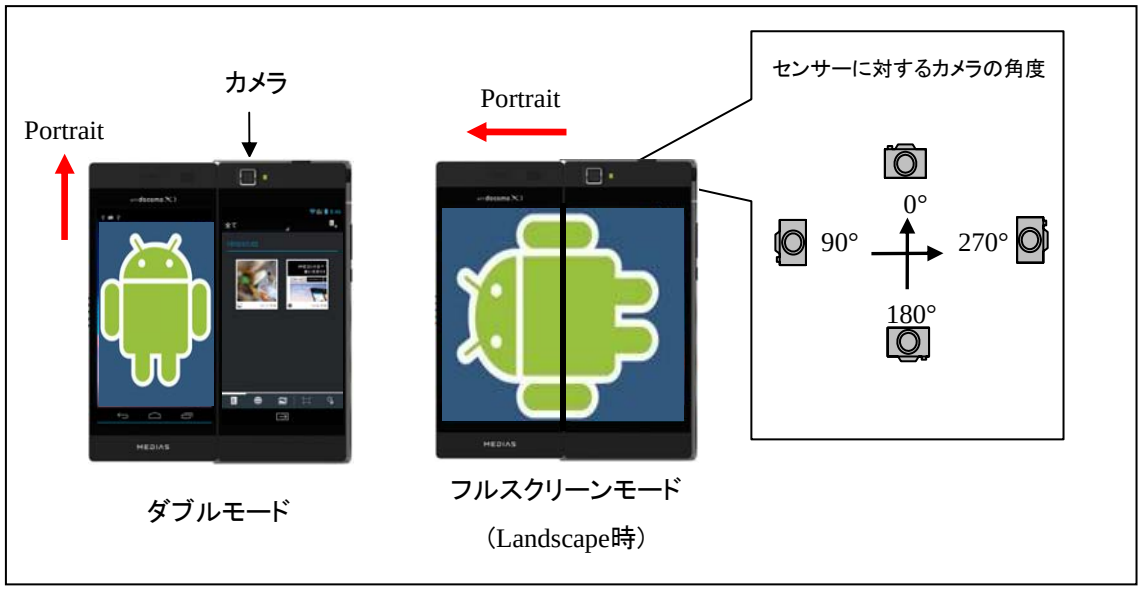


図 15 画面の向きとカメラの向きの関係

このため、「画面の向き(カメラビューの向き)」と「カメラの向き(カメラセンサが検知する向き)」を考慮せずにカメラプレビューを表示すると、図 16のように常に反時計方向に 90 度回転した状態となります。

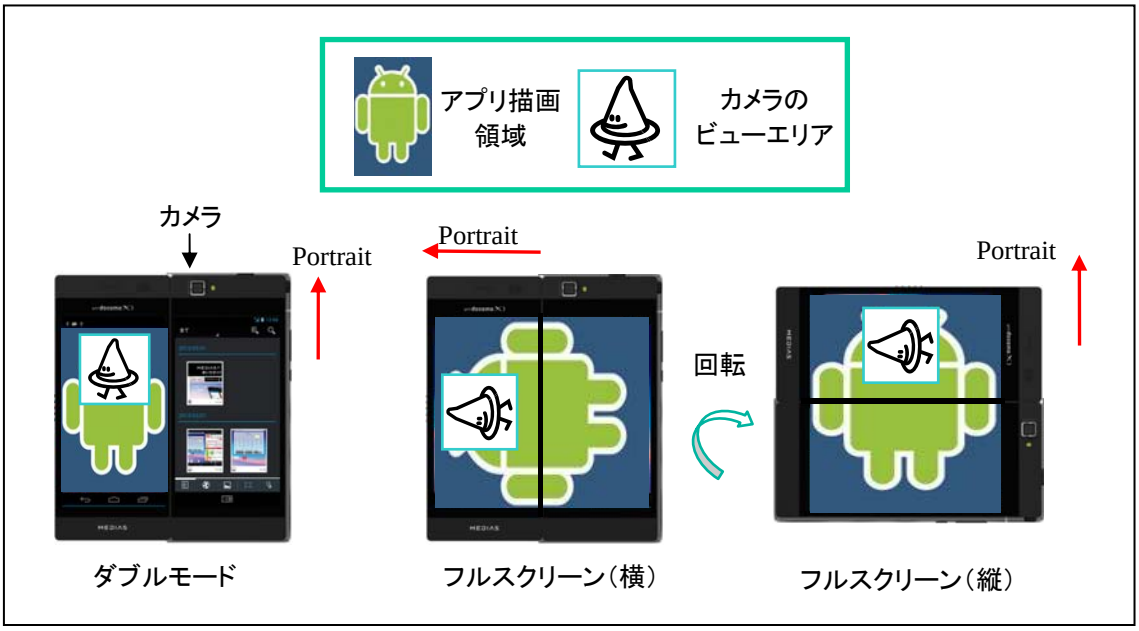


図 16 表示モードとカメラビューの描画

このような挙動を回避するには、画面の向きとカメラの向きから、カメラプレビューに適切な回転角度を計算し、カメラデバイスに設定する必要があります。
適切な回転角度を取得するには、以下の 3 つの処理をアプリに実装します。

1. CameraInfo を取得する。
2. カメラビュー(Activity)の向きを取得する。
3. これら 2 つの情報から適切な角度を決定する。

1. CameraInfo を取得する。

CameraInfoオブジェクトを生成し、Cameraクラスの getCameraInfo( )メソッドを使用して、そのオブジェクト内にカメラ情報を取得します(図 17)。

```
Camera.CameraInfo info = new Camera.CameraInfo();
int cameraId = 0; // N-05E のカメラは 1 つ。
Camera.getCameraInfo(cameraId, info);
```

図 17 CameraInfo 取得

2. カメラビュー(Activity)の向きを取得する。

カメラビューを持つActivityの向きを取得します(図 18)。
Display クラスの getRotation( )メソッドは、回転角度を表す Surface クラスの ROTATION 定数のいずれかを返します(a)。この定数を角度に変換した値を degrees という変数で保持します(b)。

```
int rotation = getWindowManager().getDefaultDisplay().getRotation(); // (a)
int degrees = 0;
switch (rotation) {   // (b)
    case Surface.ROTATION_0:   degrees =   0; break;
    case Surface.ROTATION_90:  degrees =  90; break;
    case Surface.ROTATION_180: degrees = 180; break;
    case Surface.ROTATION_270: degrees = 270; break;
}
```

図 18 カメラビューの向きの取得

3. CameraInfo と degrees から適切な角度を決定する。

CameraInfoとdegreesを元に角度を決定します(図 19)。
カメラレンズが画面と同じ方向を向いているか、反対を向いているのかを確認した上で(a)、適切な角度(result)を決定し(b)、Cameraクラスの setDisplayOrientation()メソッドを呼び出して、カメラデバイスに設定しています(c)。

```
int result;
if (info.facing == Camera.CameraInfo.CAMERA_FACING_FRONT) {  // (a)
    result = (info.orientation + degrees) % 360;
    result = (360 - result) % 360; // compensate the mirror  // (b)
} else { // back-facing
    result = (info.orientation - degrees + 360) % 360;       // (b)
}
camera.setDisplayOrientation(result);        // (c)
```

図 19 適切な角度の決定

このような実装とすることで、図 16のようにフルスクリーンモードでカメラプレビューが不自然に回転することなく正しい向きで表示されるようになります(※)。

※図 19の 4 行目(“compensate the mirror”のコメントのある行)で鏡像表示を考慮した回転補正を行っていますが、これはN-05E固有の実装となります。N-05E以外の端末で動作させる場合、この回転補正の実装は不要です。

なお、Android の標準カメラアプリでも同様の処理が実装されており、本節で解説した内容は

Android アプリに推奨される実装方法の一つです。本節で解説した処理は、標準カメラアプリでは以下の箇所で実装されています。

処理	標準カメラアプリでの実装箇所
CameraInfo の取得	Util.getDisplayOrientation()
カメラビュー(Activity)の向きの取得	Util.getDisplayRotation()
適切な角度の決定	Util.getDisplayOrientation()

また、Android の android.hardware.Camera クラスの setDisplayOrientation()メソッドの API リファレンスでも同様な実装が説明されていますので、こちらもあわせて参照することを推奨します。

4.2. プレビュー表示のアスペクト比補正

N-05Eで動作させるカメラアプリでは、4.1節で解説したカメラプレビュー画像の回転補正に加え、フルスクリーンモードにおけるカメラプレビュー画像のアスペクト比補正に留意する必要があります。

シングルモードでは、カメラプレビュー画像の向きは実際にカメラレンズを通して見えている画像と同じ状態です（図 20 (ア)）。

フルスクリーンモードではカメラの縦方向はx軸方向、画面の向き(Orientation)はy軸方向であるため、カメラプレビュー画像とカメラプレビューの描画エリア、それぞれの縦横サイズが異なっている場合、実際にディスプレイに表示されるカメラプレビュー画像は誤ったアスペクト比で表示されることになります（図 20 (イ)）。

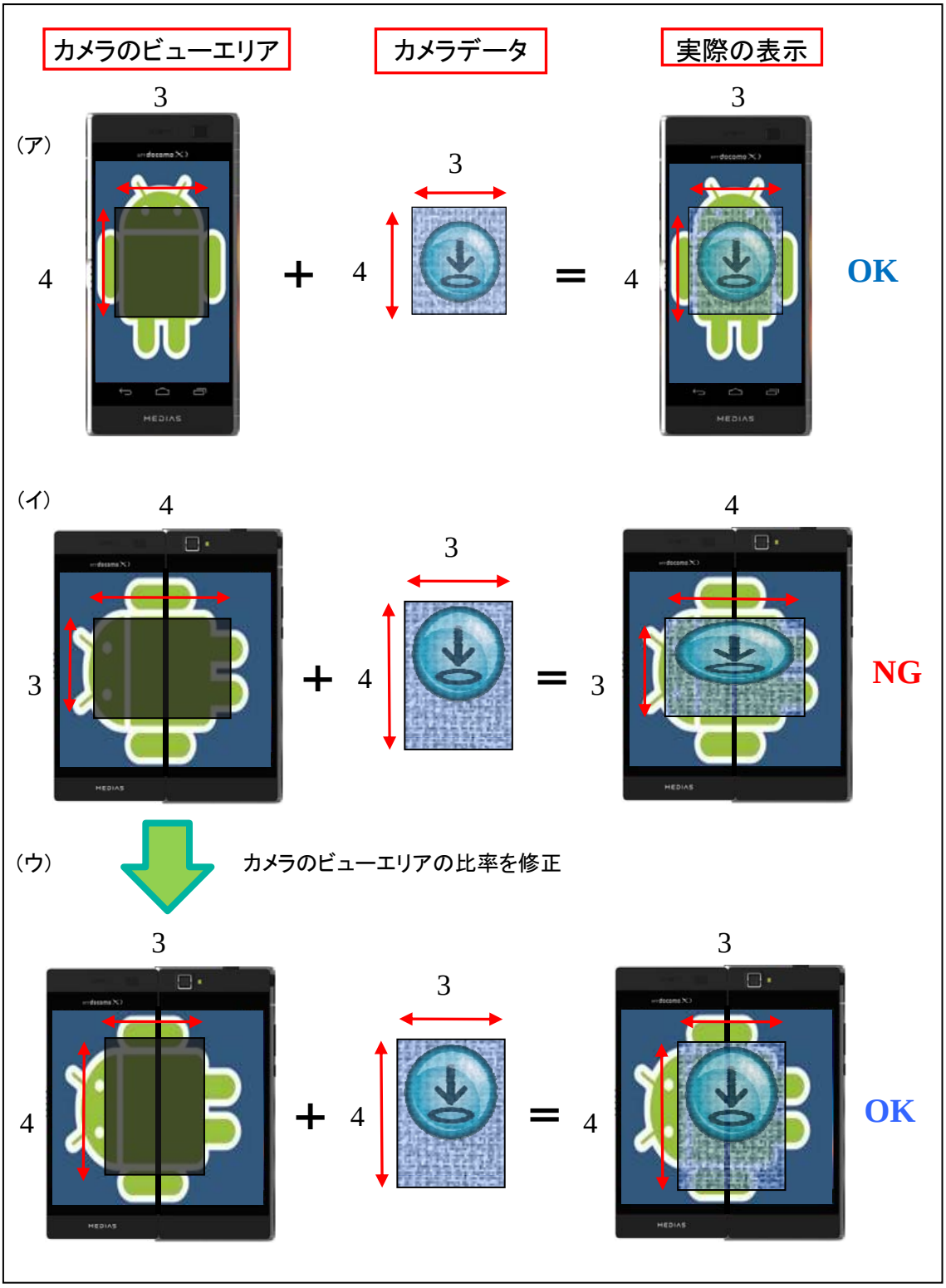


図 20 表示モード遷移とアプリ状態

このような問題を回避するには、カメラプレビューの描画エリアの比率がカメラプレビュー画像と同じになるように、描画エリアの縦横サイズを入れ替える必要があります。この処理ロジックの例を図 21に示します。

```
// 4.1 節の適切な角度の設定
camera.setDisplayOrientation(result);

// プレビューのサイズを取得する。
Size size = camera.getParameters().getPreviewSize();

if (result % 180 == 0) {
    // 角度が 0、あるいは 180 であれば縦横サイズの入れ替えはしない。
    cameraPreview.setSize(size.width, size.height);
} else {
    // 角度が 90、あるいは 270 であれば補正のため縦横サイズを入れ替える。
    cameraPreview.setSize(size.height, size.width);
}
```

図 21 描画エリアの比率変更

このような実装とすることで、図 20 (ウ)のように正しいアスペクト比で表示されるようになります。Android の標準カメラアプリでも、Camera クラスの startPreview()メソッド内で同様の処理が実装されています。

4.3. 保存前の画像表示

本節では、カメラ撮影画像をアプリ内で表示する際の注意点を解説します。

ダブルモードで撮影した場合、撮影画像をアプリ内で表示する際にも特に意識しなくても正しい向きで表示されます(図 23 (ア))。

しかしフルスクリーンモードで撮影した画像は、アプリケーションの実装によっては反時計回りに 90 度回転して表示される場合があります。カメラデバイスから渡される画像データは、カメラセンサの基準方向にのみ従い、撮影時の画面の向きは考慮されていません。そのため、フルスクリーンモードで撮影した画像をそのまま保存すると、反時計回りに 90 度回転した状態となります(図 23 (イ))。

正しい向きで撮影画像を表示するには、画面の向きや撮影時のカメラの向きを元にして、画像の向きを補正する必要があります(図 23 (ウ))。

向きの補正の方法の一つに、画像撮影時のカメラセンサの回転補正値(図 19のresult値)だけ画像を回転させて表示する方法があります。図 22にカメラセンサの回転補正値だけ画像を回転させる例を示します。

```
// 図 19 で決定したカメラセンサの回転補正値
// int result = ・・・;
// ※なお、インカメラの場合は図 19 の鏡像表示を考慮した回転補正を行う前の値を使用します。
//   result = (info.orientation + degrees) % 360;

// 画像を表示する View を取得する。
ImageView imageView = (ImageView) findViewById(R.id.imageView);

// 画像を読み込む。
Bitmap bitmap = loadBitmap();

//時計回りに回転させる Matrix を用意する。
Matrix matrix = new Matrix();
matrix.postRotate(result); // 回転補正値を設定する。

// Matrix を使って画像を回転させる。
Bitmap rotatedBitmap = Bitmap.createBitmap(
     bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight(), matrix, false);

// 回転させた画像を ImageView にセットする。
imageView.setImageBitmap(rotatedBitmap);
```

図 22 撮影画像表示時の向き補正例

固定の角度で回転させるのではなく、図 22のように、図 19で算出した補正値を使用して回転させることで、撮影時のカメラの向きや画面の向きに依存しない処理として記述できます。

なお、保存後の画像を N-05E 以外の端末でも正しく表示するには、以下のような対応が必要になります。

- 撮影画像を保存する際に撮影時の端末の向きを表す Exif 属性を付与して保存する。また、画像表示時にこの Exif 属性を参照する。
- 撮影画像を保存する際に期待する向きに補正して保存する。

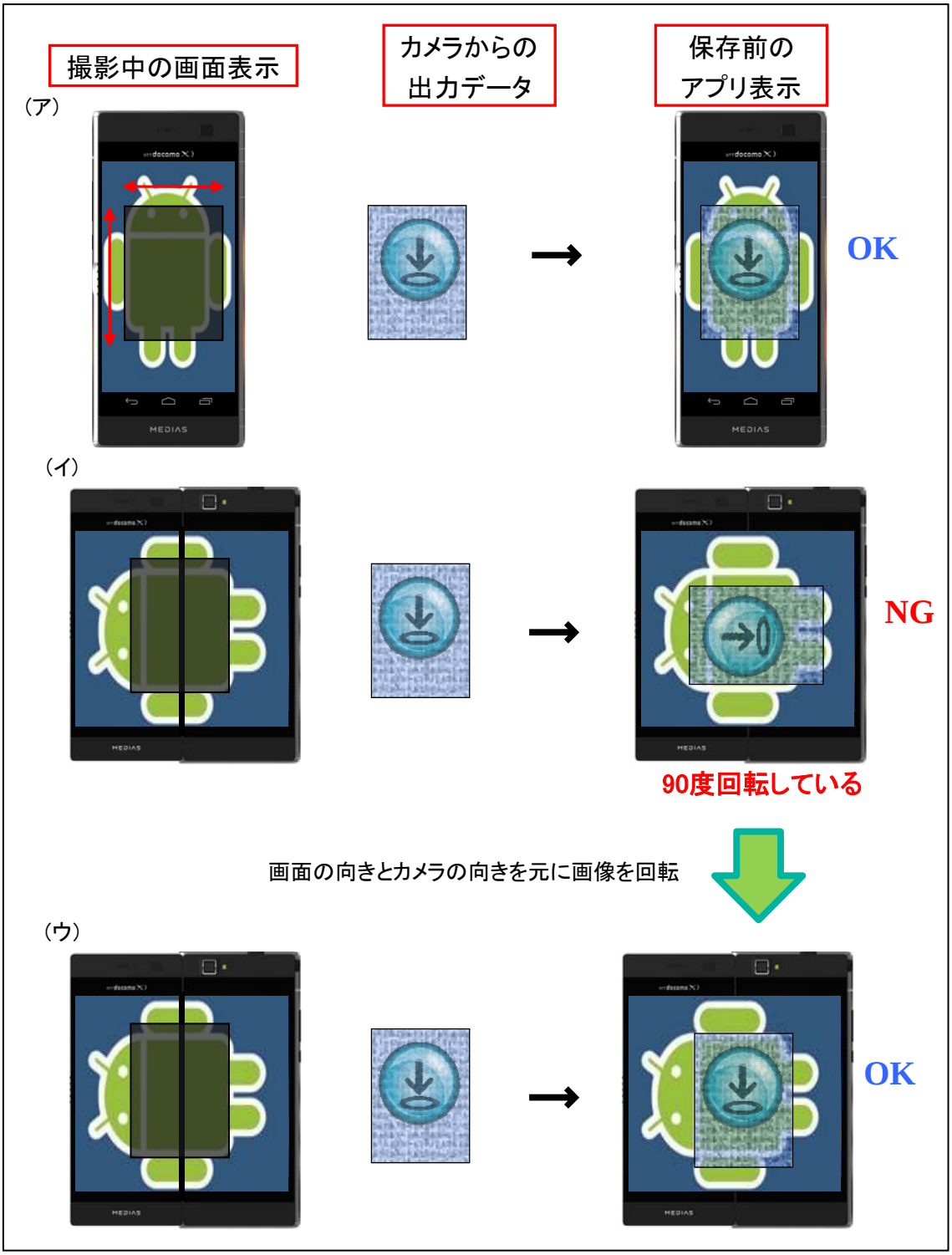


図 23 画像保存前のファイル表示

5章　ホームアプリにおけるウィジェット表示

N-05E のホームアプリは独自の仕様拡張が行われており、同程度な画面密度を持つ他の一般的なスマートフォンと比較して、グリッドのセルサイズが異なります。
端末の開閉状態と向きごとの、N-05Eのホームアプリのグリッド仕様を表 1に示します。

表 1 端末状態毎のグリッド仕様

	開閉状態	画面の向き	グリッド数 (縦 x 横)	1 セルのサイズ (縦 x 横) [dp]	セル間の スペースサイズ [dp]
(ア)	閉状態	Portrait	4 x 4	100 x 88	0
(イ)	閉状態	Landscape	4 x 4	83 x 106	0
(ウ)	開状態	Portrait	8 x 4	83 x 106	0
(エ)	開状態	Landscape	4 x 8	100 x 88	0

ウィジェットアプリにおいて、appwidget-provider 要素の android:minWidth 属性、android:minHeight 属性に指定するサイズの目安は以下となります。

(Number of cells * [1 セルのサイズ]) – 2 [dp]

端末の開閉状態や画面の向きに応じた、表 1のセルサイズの縦あるいは横の値を上記の[1 セルのサイズ]に代入することで、N-05Eに適した最小サイズを導くことができます。この値を利用することで、N-05Eに適したサイズで表示することができます。

表 1の(ア)から(エ)のレイアウトの様子を図 24、図 25に示します。
なお、(ウ)のレイアウトおよび 1 セルのサイズは(イ)と同じですが、システム上の画面の向きは Portrait であることに注意してください。また、同様に(ア)と(エ)ではレイアウト、セルサイズは同じですが、両者でシステム上の画面の向きが異なることに注意してください。

表 1(ア)

表 1(イ)

図 24 閉状態のグリッドレイアウト

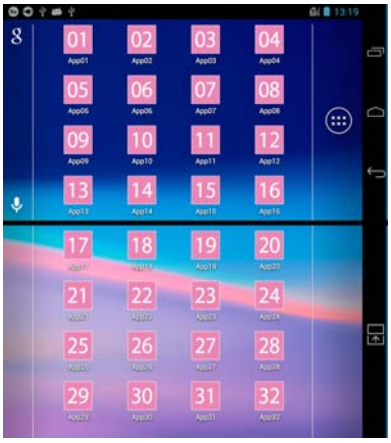


表 1(ウ)

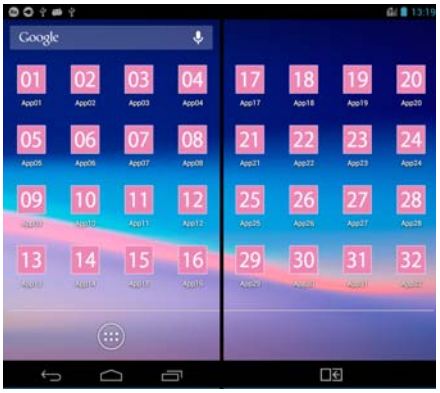


表 1(エ)

図 25 開状態のグリッドレイアウト