

# Multi Window in Android N

Migrating your app to  
Android N multi window compatible

Google Developers Summit

2016. 04. 28

# About

- 김태호 (커니)
- GDG(Google Developers Group)  
Korea Android: Co-organizer
- GDE(Google Developer Experts): Android
- 블로그 ‘커니의 안드로이드 이야기’
  - <http://androidhuman.com>
  - <http://kunny.github.io>

# History

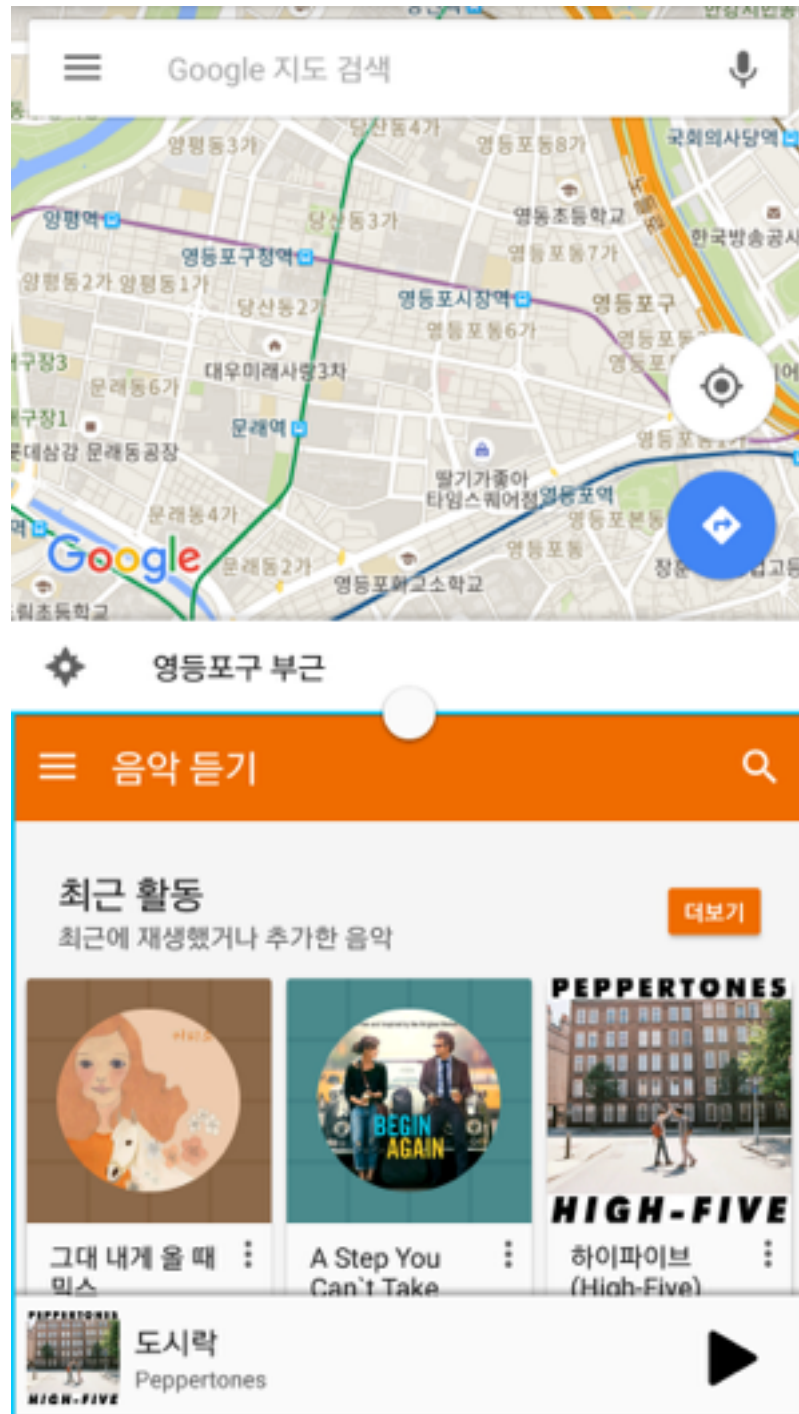
- Samsung (TouchWiz)
  - Galaxy Note 2 에 최초 도입 (Split View)
  - 이후 자사 플래그십 휴대폰 및 태블릿에서 대부분 지원
  - Window mode - 2개 이상의 애플리케이션 동시 수행 지원
- Google (AOSP)
  - Android M Developer Preview에 숨겨진 기능으로 추가

# Comparison: Samsung vs. Android N

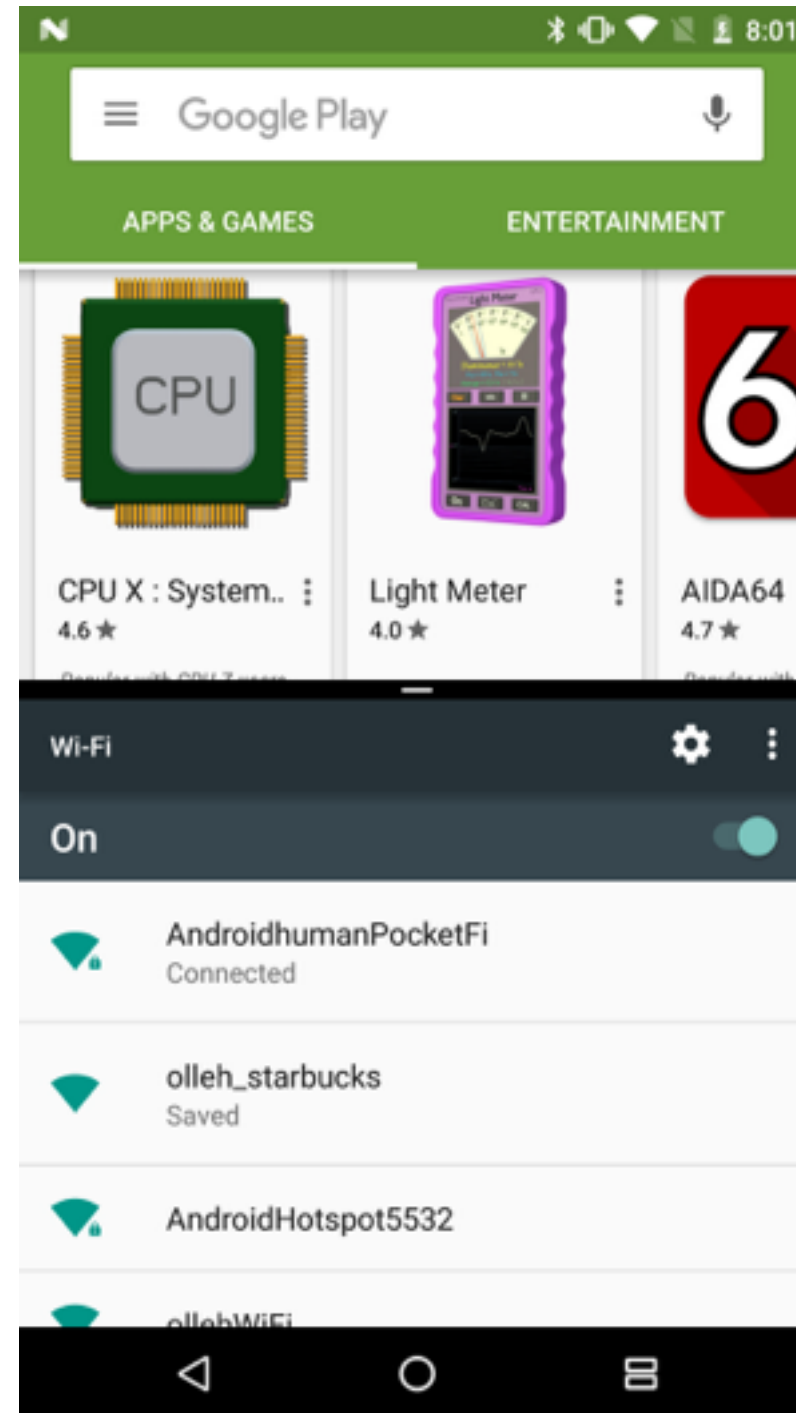
# Samsung MultiWindow SDK

- 삼성 갤럭시 단말 중 멀티윈도우를 지원하는 기기에 한정
- 지원 모드 및 기능
  - Split Screen, Window mode(Pen window)
  - Multi Instance, Drag & Drop
- 액티비티 생명주기 관리
  - 액티비티 단독 실행시와 동일
- Google Stock App을 포함한 다수의 애플리케이션에서 사용

# Split Screen

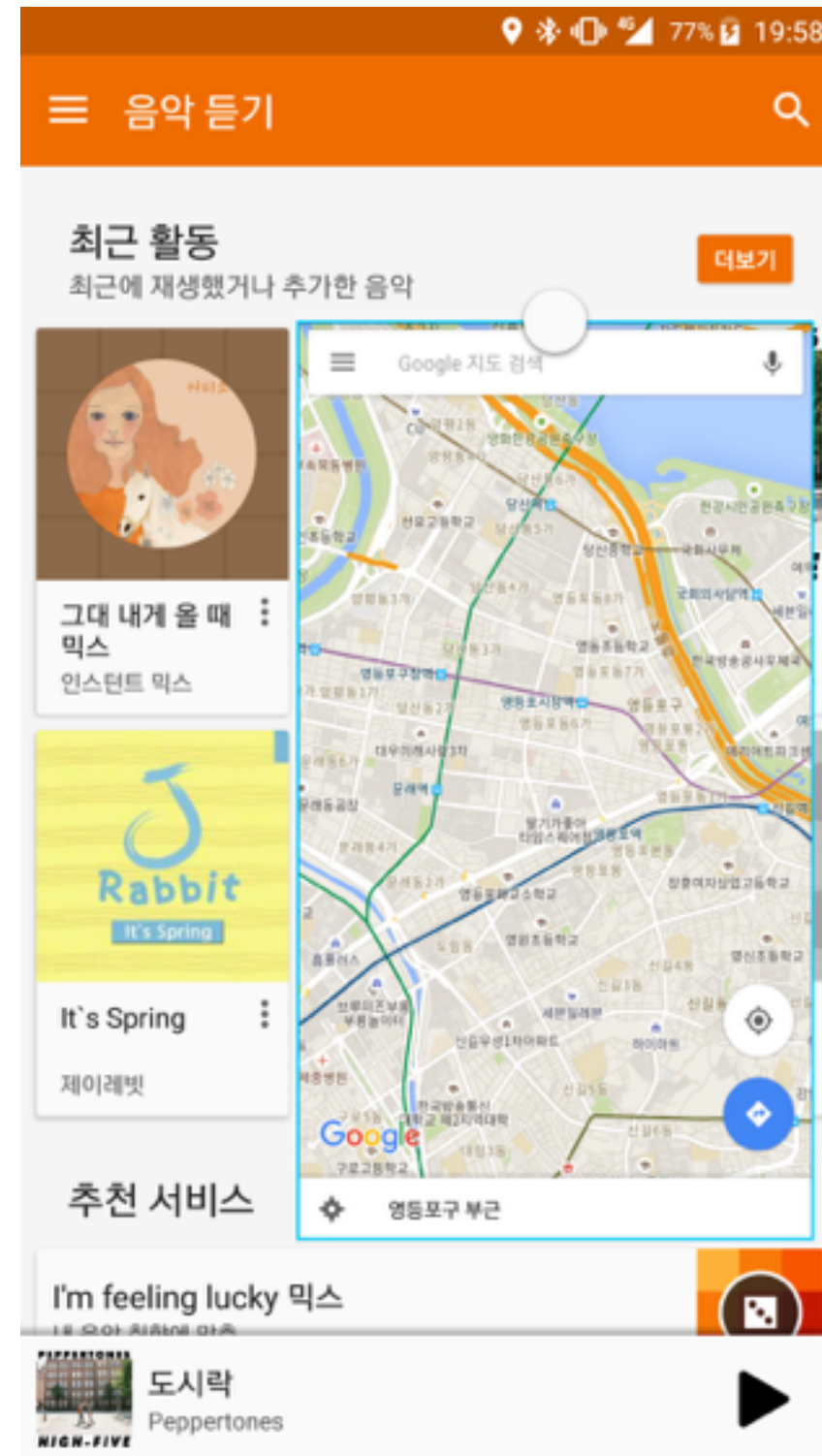
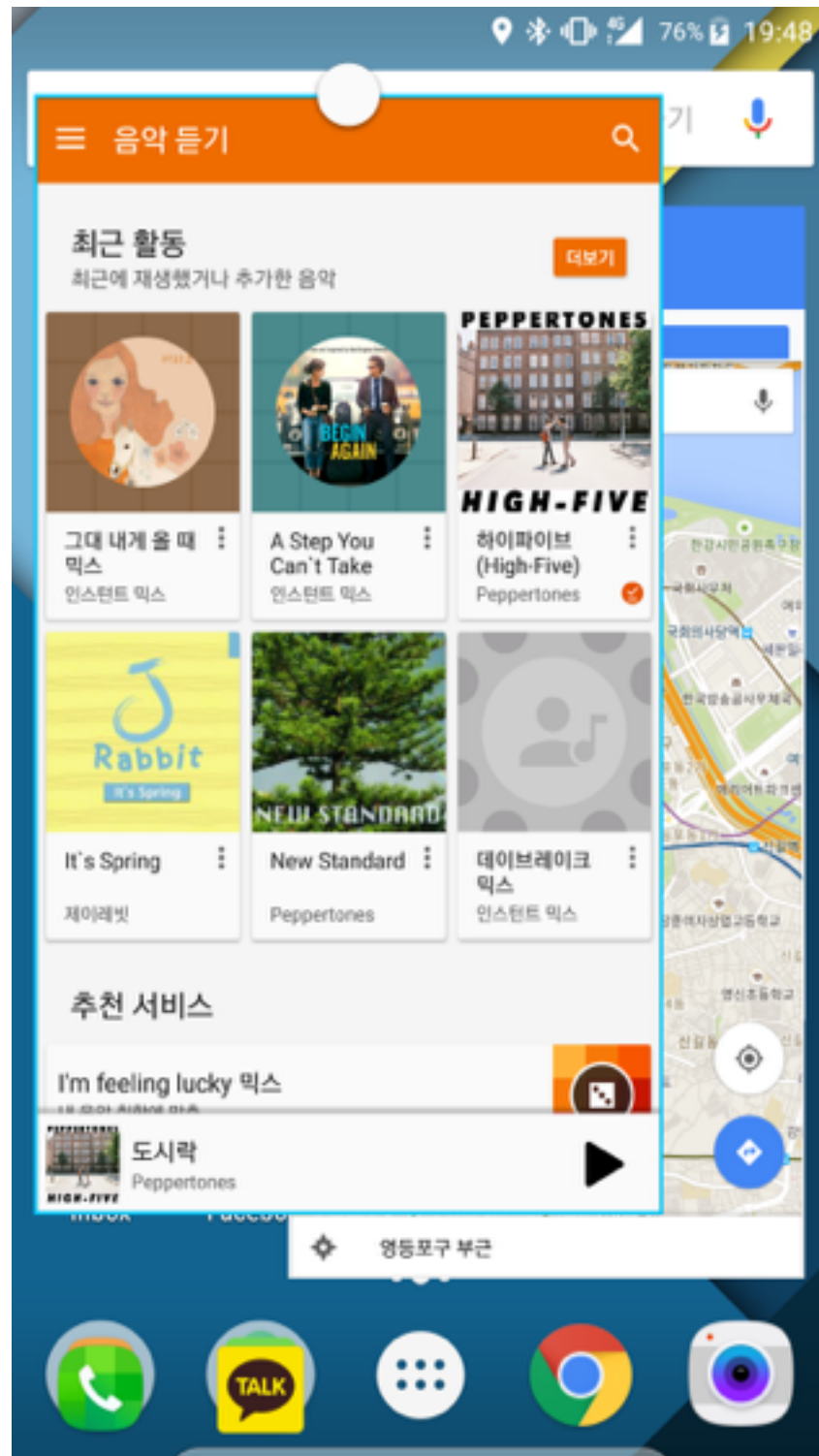


Samsung



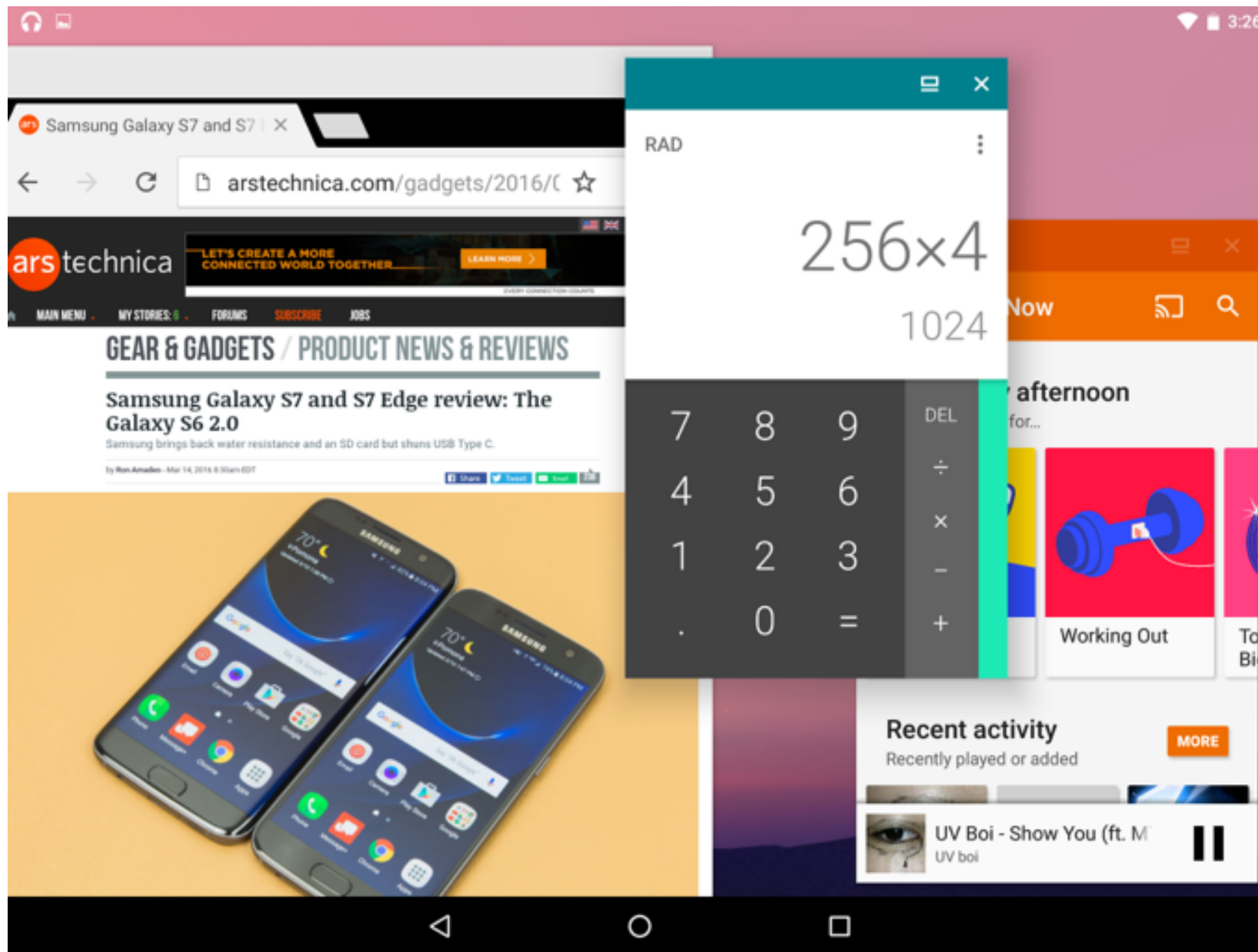
Android N

# Window mode (Samsung)





# Freeform mode (Android N)



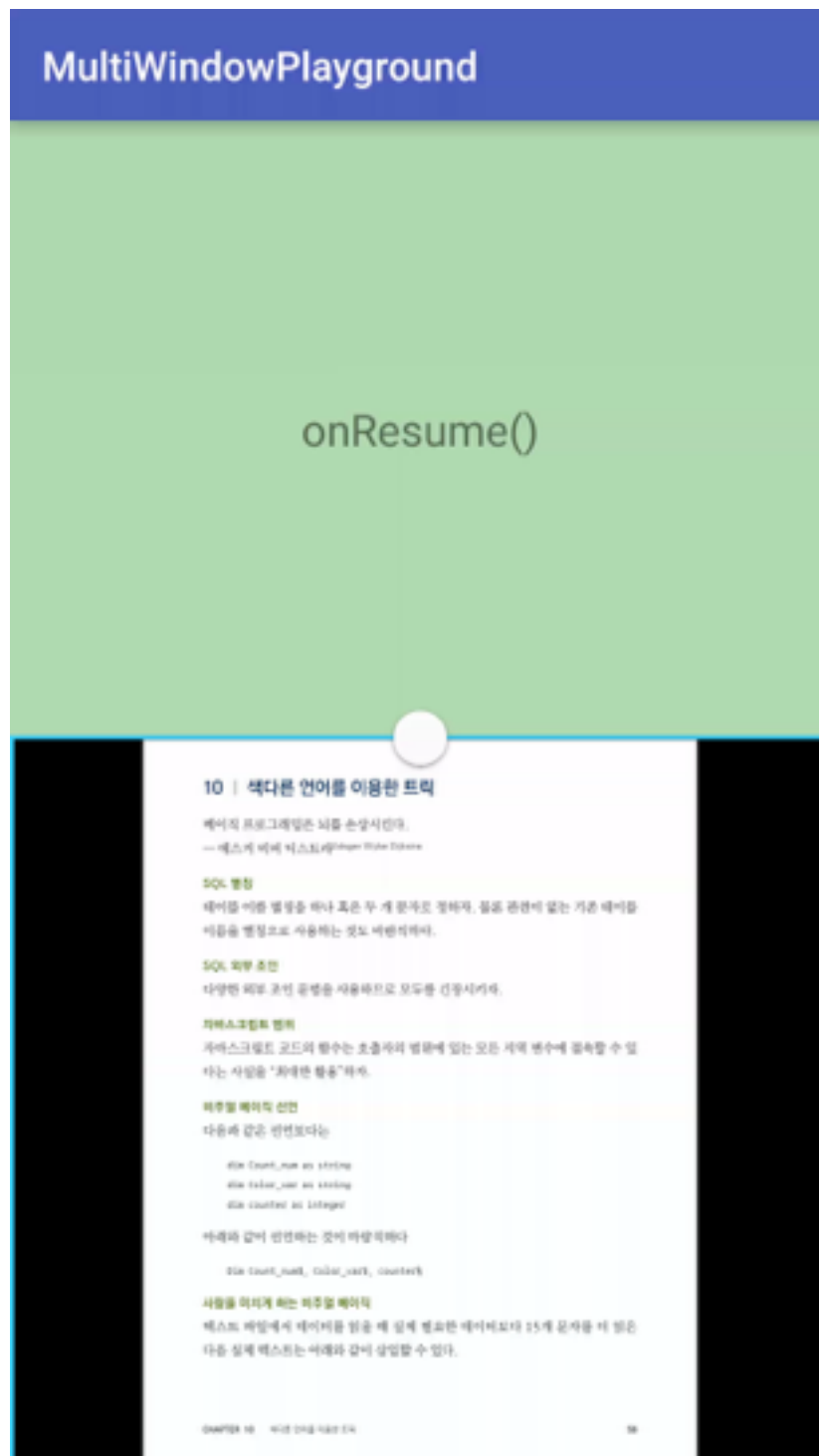
source: <http://arstechnica.com/gadgets/2016/03/this-is-android-n-freeform-window-mode/>



# Lifecycle handling (Samsung)

- 액티비티가 단독으로 실행될 때와 생명주기가 동일
- 화면에 표시되고 있다면 활성 상태 보장
- 멀티윈도우를 고려하여 작성한 앱이 아니더라도 어느 정도의 호환성 보장

# Lifecycle handling (Samsung)

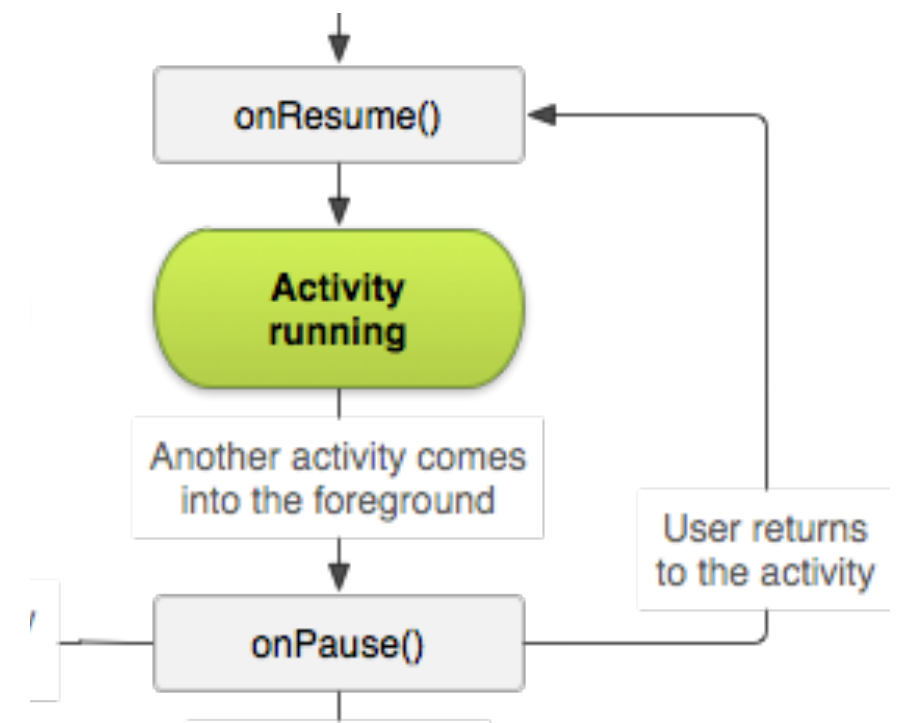


# Lifecycle handling (Android N)

- 한 번에 단 하나의 액티비티만 활성 상태로 존재 가능
- 활성 상태가 아닌 애플리케이션은 일시중지 상태로 전환
- 포커스를 받으면 다시 활성 상태로 전환
- 액티비티 생명주기에 따른 작업을 올바르게 처리한 애플리케이션은 큰 수정 없이 멀티윈도우 지원 가능

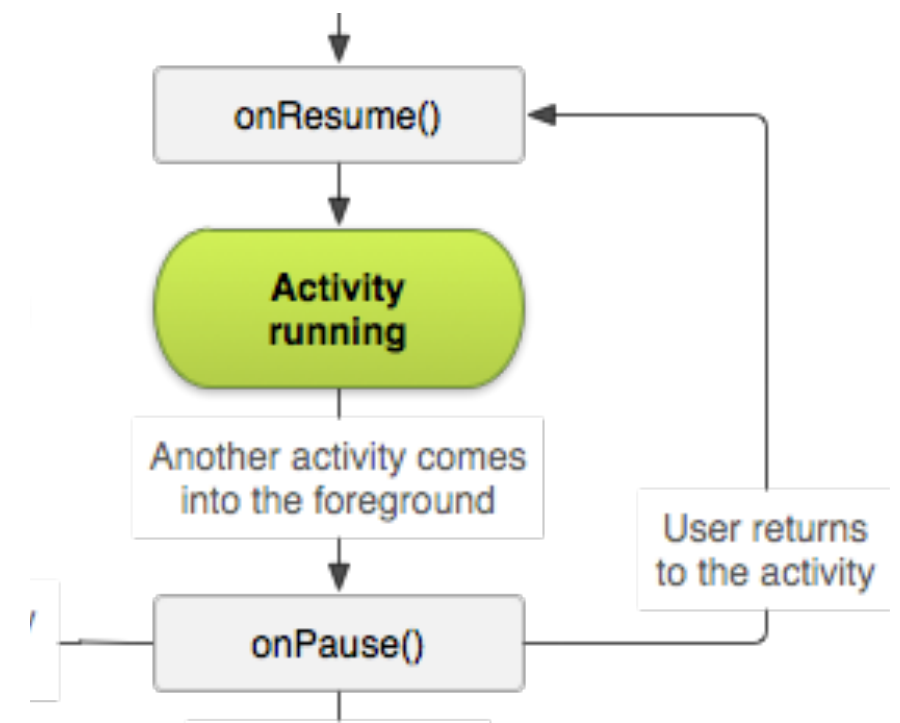
# Lifecycle handling (Android N)

*The foreground lifetime of an activity  
... During this time the activity is  
in front of all other activities and  
interacting with the user.*




# Lifecycle handling (Android N)

*The foreground lifetime of an activity  
... During this time the activity is  
in front of all other activities and  
interacting with the user.*



<http://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle>

# Lifecycle handling (Android N)



## MultiWindowPlayground

onResume()

디바이스 드라이버를 바꿀 방법이 없으므로 선택의 여지는 없다.

**커스텀 스크립트 언어**  
우리의 클라이언트/서버 응용프로그램에서 실행 중에 바이트로 컴파일되는 스크립팅 명령어 언어를 포함해야 한다.

**컴파일러 종속 코드**  
컴파일러나 인터프리터 버그를 발견했으면 이 버그를 이용해 우리 코드가 제대로 동작하게 만들자. 이제 우리 프로그램을 사용하는 모든 이는 다른 컴파일러를 사용할 수 없게 된다.

**실생활 예제**  
스승님께서 작성하신 실생활 예제를 보여주겠다. 하나의 C 함수에 그가 사용한 여러 기법을 살펴보자.

```
void* Realocate(void*buf, int os, int ns)
{
    void*temp;
    temp = malloc(os);
    memcpy((void*)temp, (void*)buf, os);
    free(buf); buf = malloc(ns);
    memset(buf, 0, ns);
    memcpy((void*)buf, (void*)temp, ns);
    return buf;
}
```

1. 자세히 보면 Realocate의 철자가 부정확하다. 창의적인 철자법의 힘을 알리지 말라.
2. 아무런 이유 없이 입력 버퍼를 임시 복사본을 만든다.
3. 이유 없이 형을 변환한다. Malloc()은 (void\*)형을 매개변수로

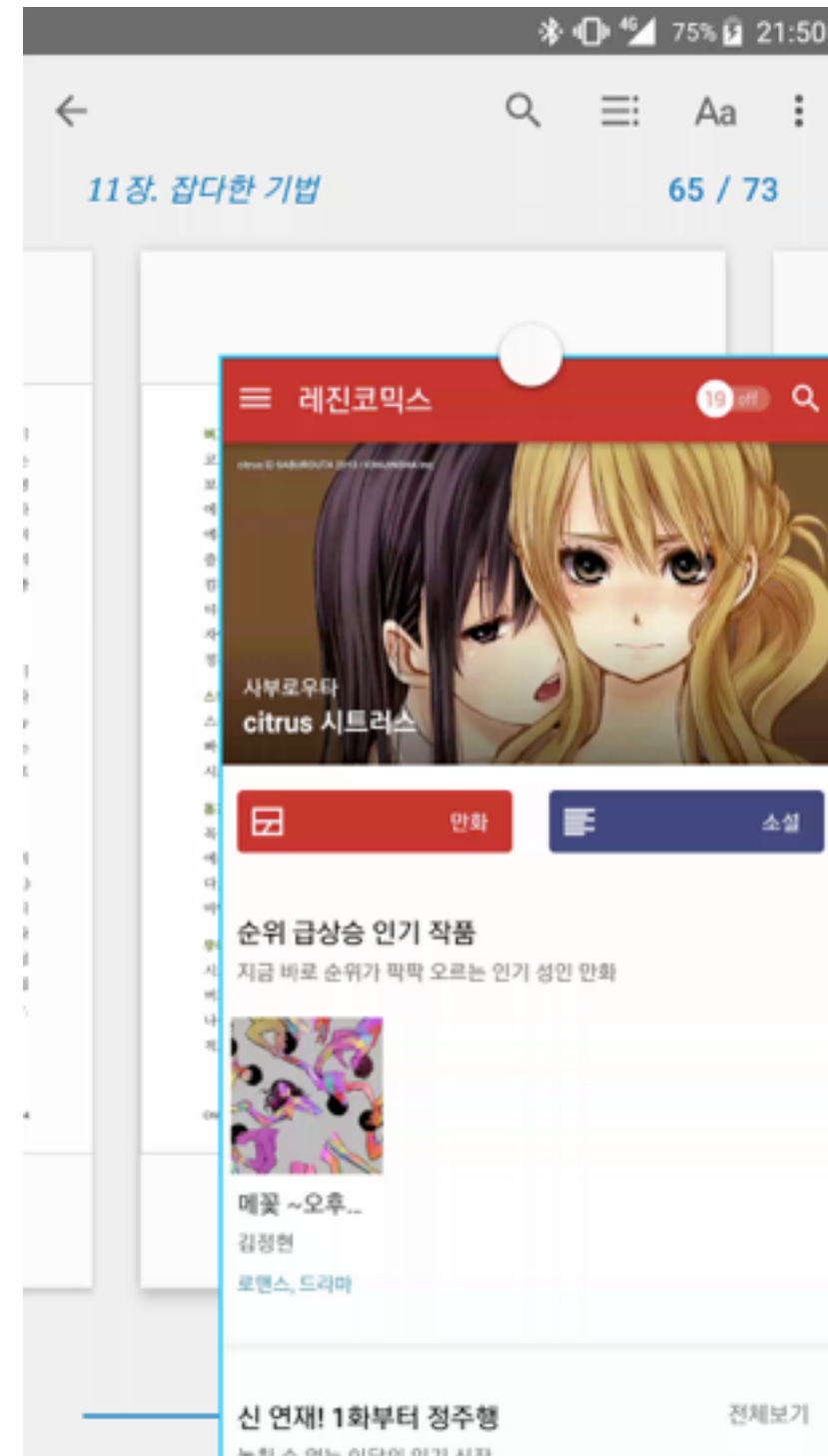
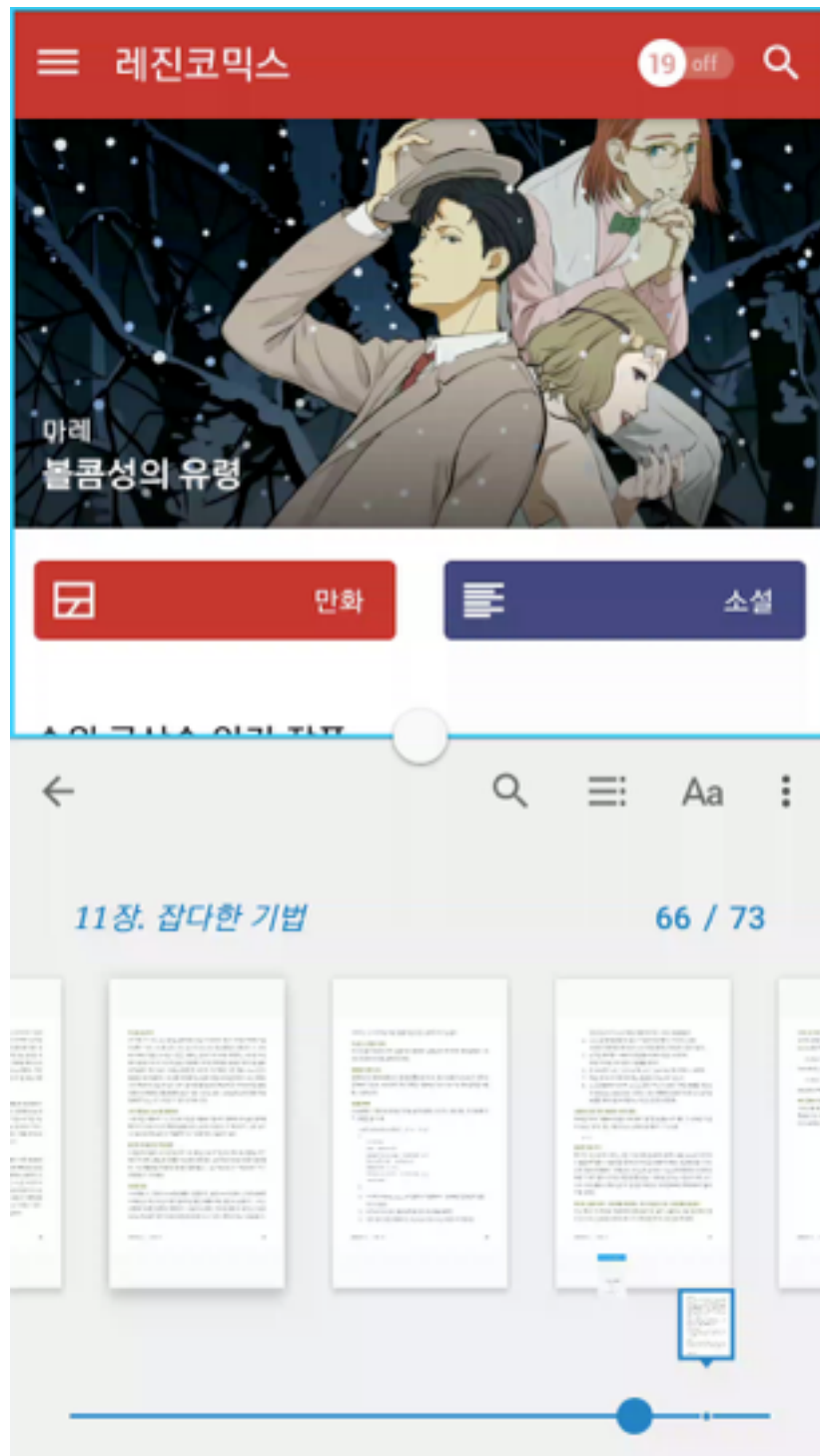
CHAPTER 11 잡다한 기법 86

# Configuration changes (Samsung)

- 액티비티 단독 실행시와 동일하게 동작 (화면 회전 등)
- 액티비티가 표시되는 화면 크기 변경은 액티비티 상태에 영향을 주지 않음



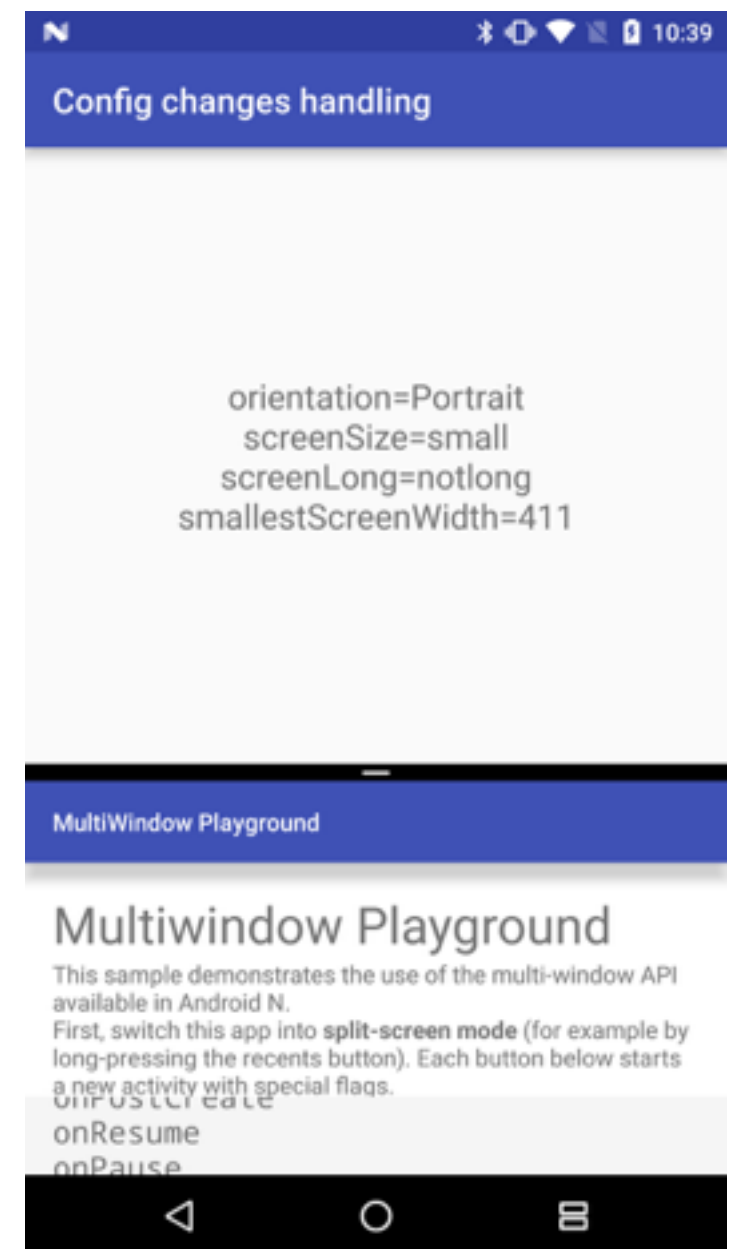
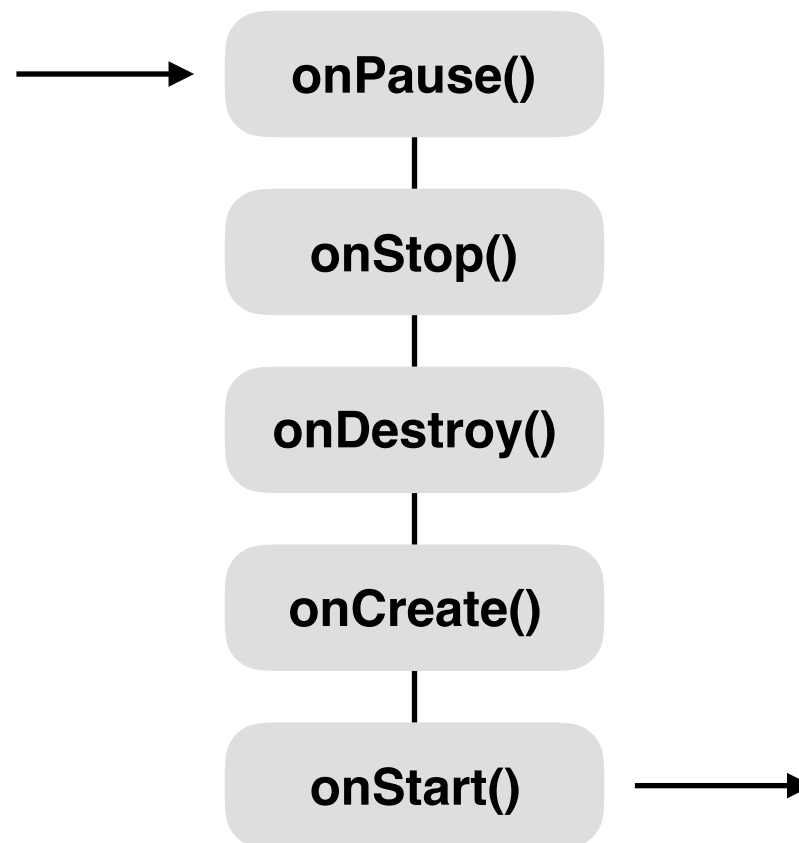
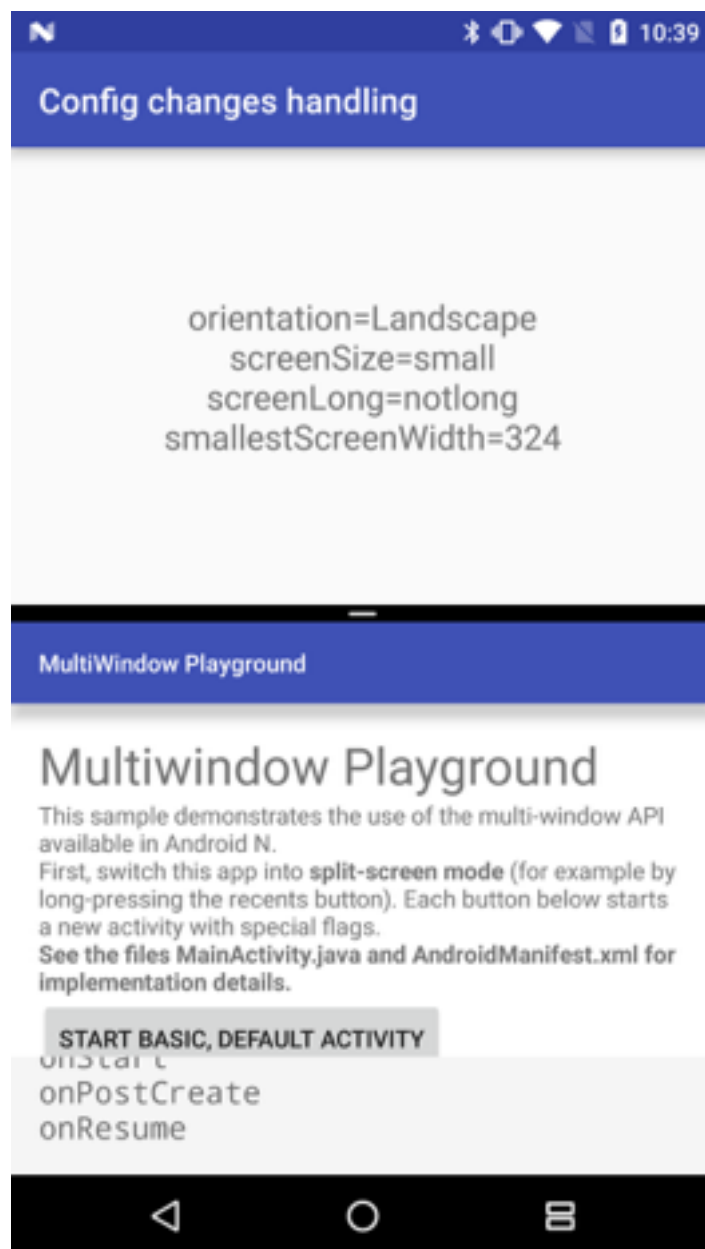
# Configuration changes (Samsung)



# Configuration changes (Android N)

- 액티비티가 표시되는 영역 크기 변경이 액티비티 상태 영향을 줌
  - 일반적으로 액티비티가 재시작되며 변경된 환경에 맞게 UI를 재구성
- Possible configuration changes
  - `smallestScreenSize`
  - `screenLayout`
    - `screenSize`: small, normal, large, xlarge
    - `ratio`: long, notlong
  - `orientation`

# Configuration changes (Android N)

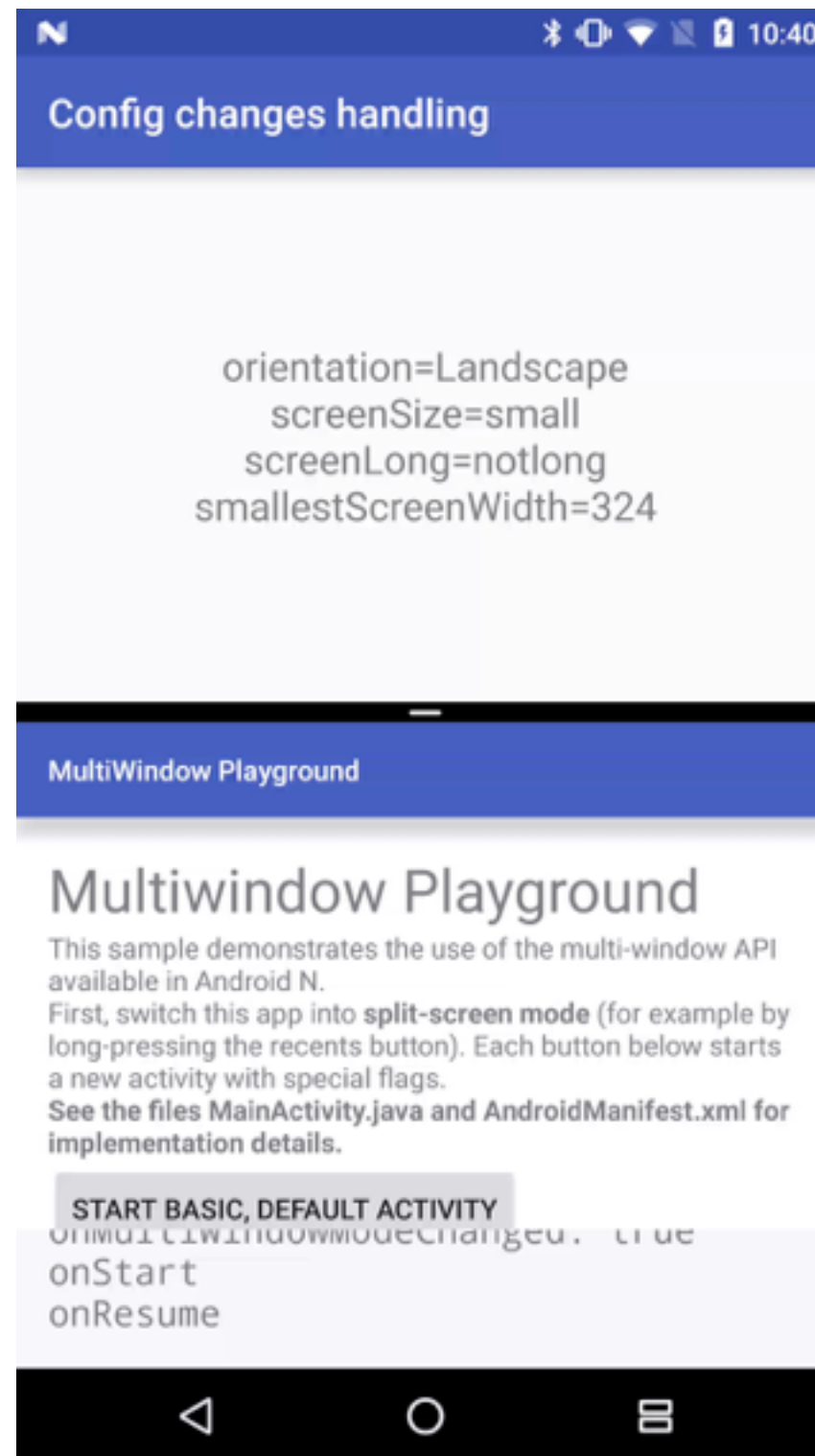


# Configuration changes (Android N)

- 액티비티 재시작 없이 직접 처리하는 방법
  - android:configChanges 사용
  - onConfigurationChanged()에서 처리

```
<activity  
    android:name=".MainActivity"  
    android:configChanges="screenSize|  
smallestScreenSize|screenLayout|orientation"/>
```

# Configuration changes (Android N)



# Supporting Multi Window without targeting Android N

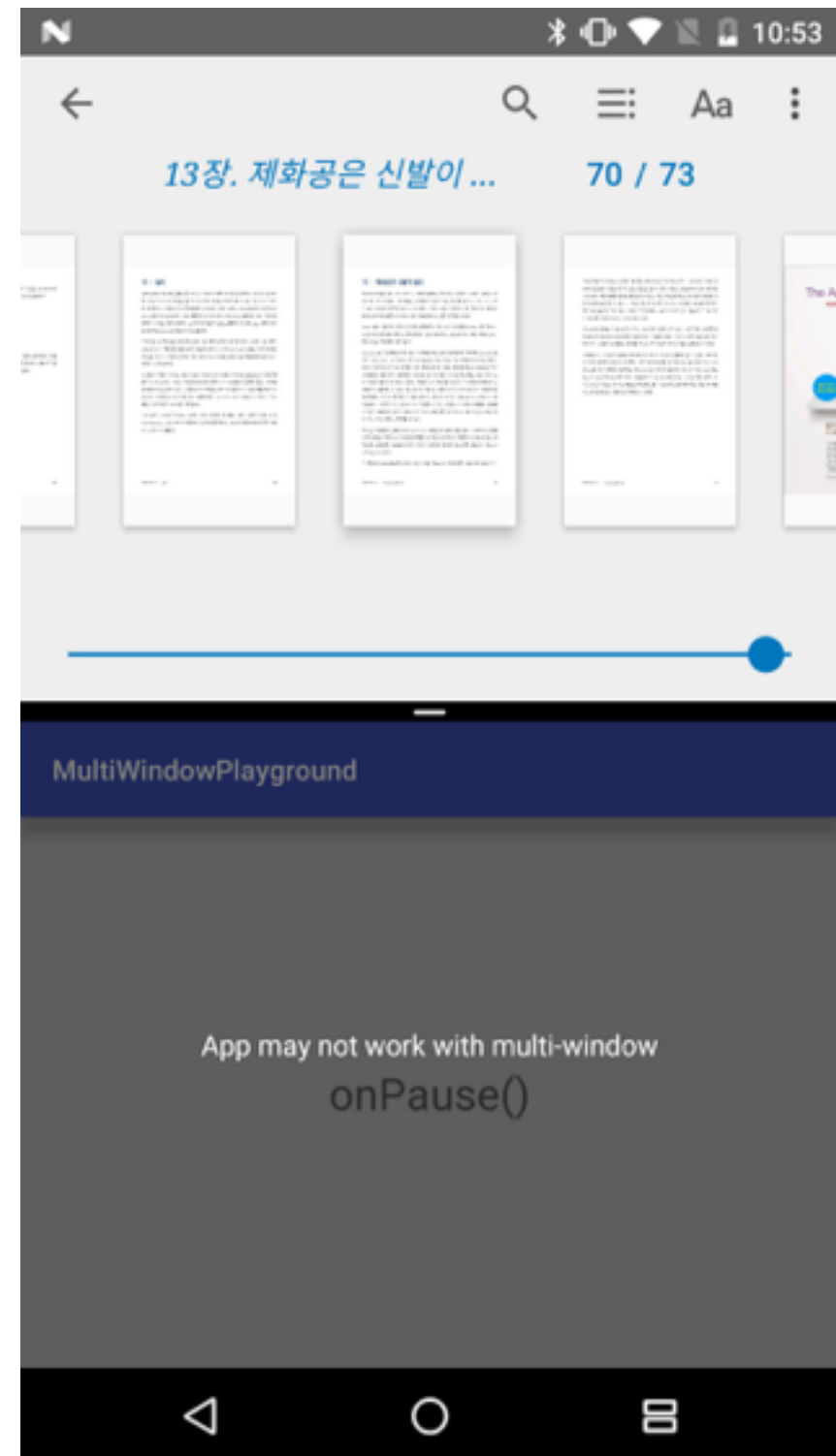
# targetSdkVersion < 'N'

- 가로/세로 모드를 모두 지원하도록 설정
  - 화면 높이가 일정 수준 이하로 작아지면 가로 모드로 전환됨
- 액티비티 생명주기가 적절히 처리되었는지 확인
- 모든 환경에 대한 리소스 구비 여부 확인



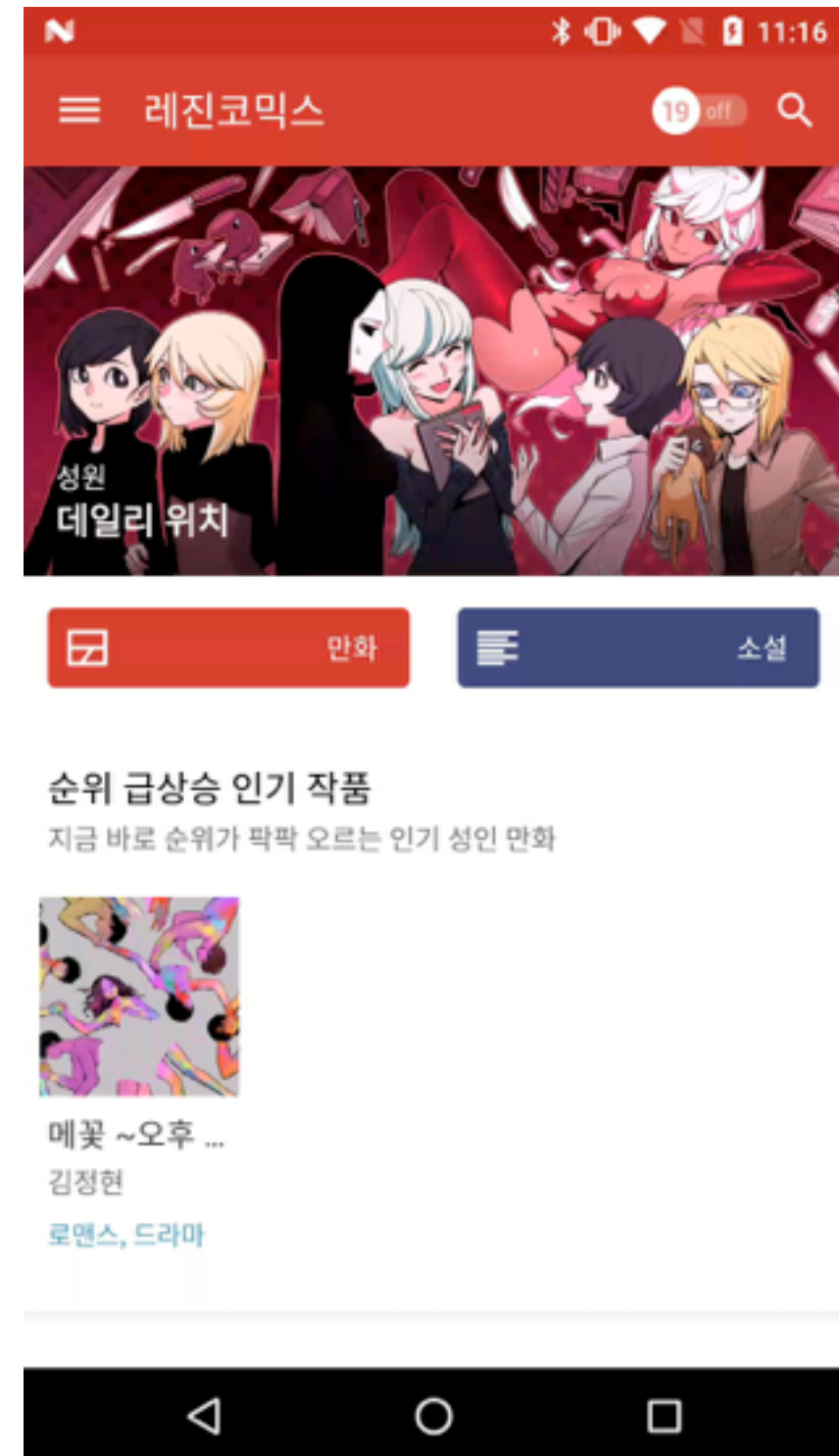
# targetSdkVersion < 'N'

- 타겟 버전이 낮을 경우,  
멀티윈도우 모드에서  
앱 실행시 주의 문구가  
함께 표시됨



# targetSdkVersion < 'N'

- 특정 화면 방향  
(portrait/landscape)로  
레이아웃이 고정된 경우  
멀티윈도우 모드  
전환 불가



# targetSdkVersion < 'N'

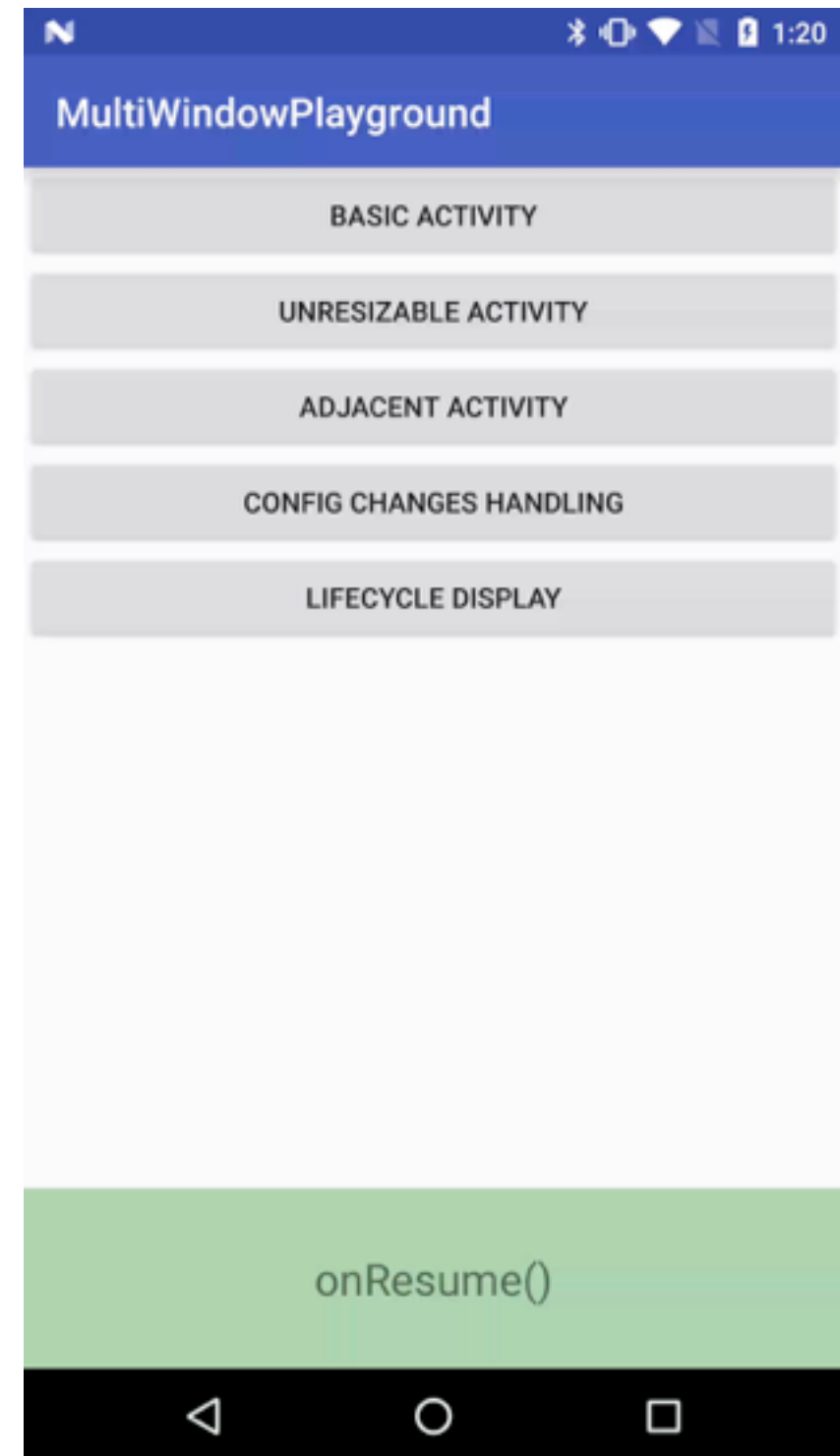
- Overview 화면에서 멀티윈도우 미지원 여부 표시
- 멀티윈도우 상태에서 미지원 앱 실행시 멀티윈도우 모드 해제



# Multi Window: Enabling & Disabling

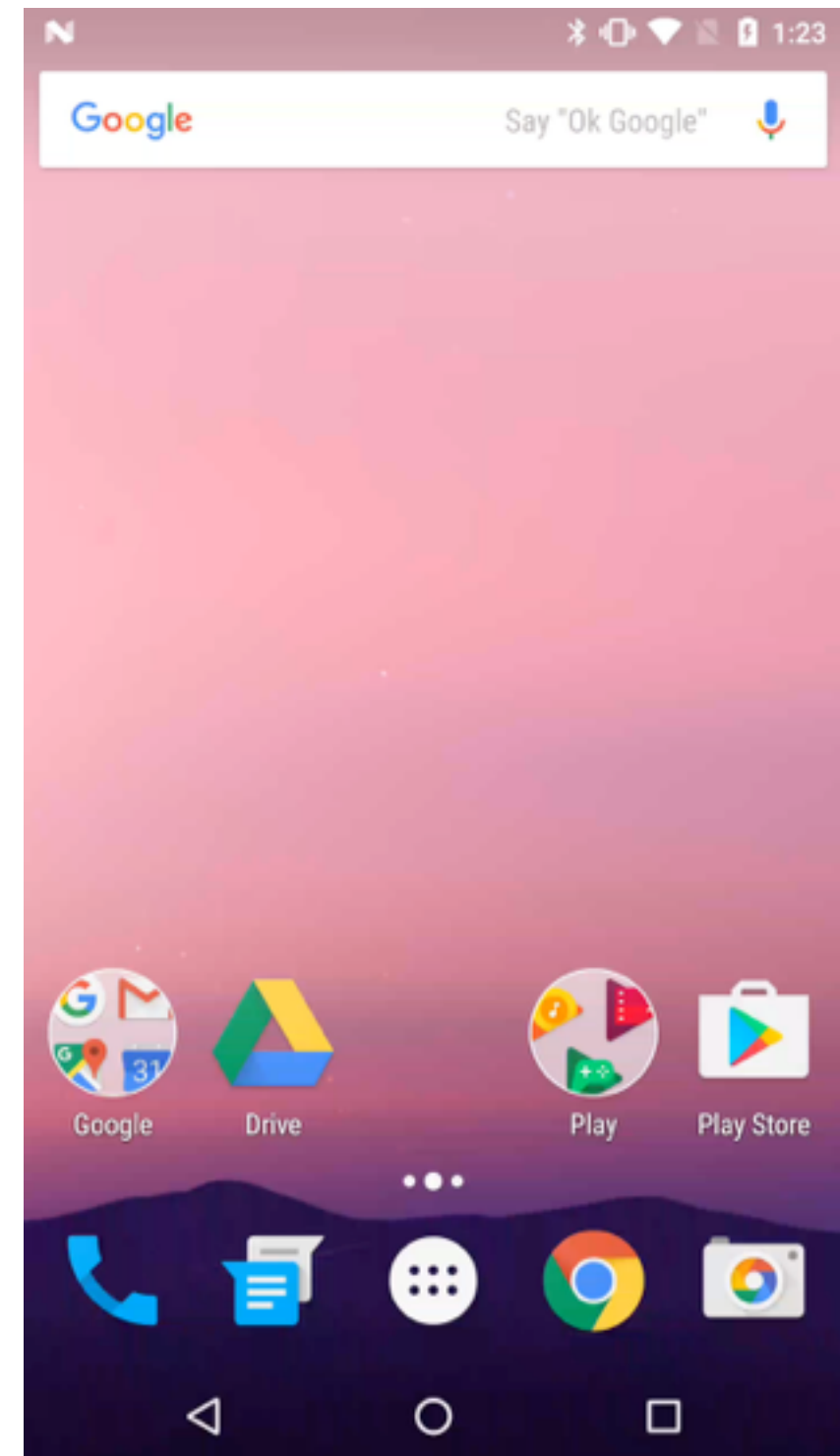
# 작업 전환 버튼

- 작업 전환 버튼을 길게 터치
  - 멀티윈도우를 지원하는 앱에서만 사용 가능
  - 멀티윈도우 활성화 상태에서 다시 길게 누를 경우 멀티윈도우 해제



# Overview 화면

- 멀티윈도우 모드로 표시할 작업 화면을 길게 터치하여 끌어놓음



# 멀티윈도우 모드 종료

- 화면에 표시되는 Divider를 위/아래로 끌어 원하는 작업을 전체 화면으로 실행





# Multi Window: APIs

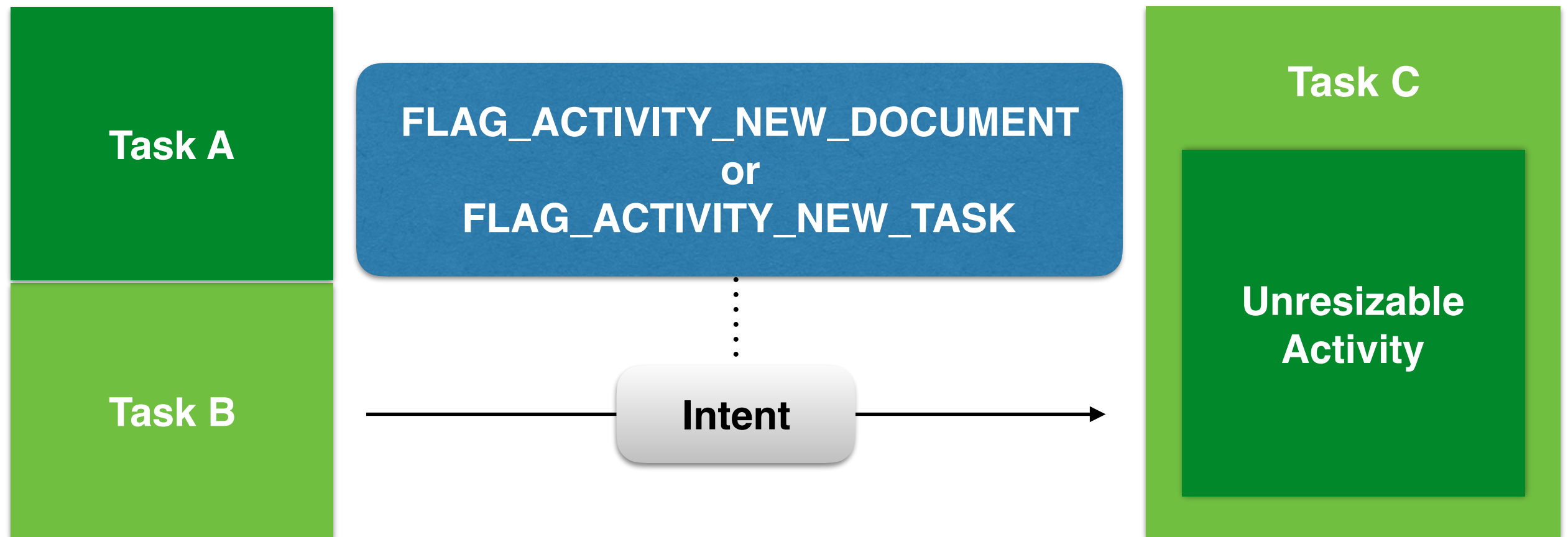
# android:resizableActivity

- 멀티윈도우 지원 여부 설정
- 매니페스트 내 <application> 및 <activity> 노드에서 설정 가능
- 타겟 레벨에 따라 기본값이 상이
  - N+: 화면 방향 고정 여부와 상관없이 기본값 true
  - <N: 화면 방향이 고정되지 않은 액티비티에 한해 true

# android:resizableActivity

- 멀티윈도우 지원 여부는 Task Root 액티비티 (태스크를 최초로 시작한 액티비티)를 따름
- 특정 액티비티에서만 멀티윈도우를 지원하지 않으려면
  - 매니페스트 내 액티비티 노드에  
android:resizeableActivity="false" 설정
  - Task Root 액티비티로 실행되도록 설정
  - Task가 분리되지 않을 경우 Root 액티비티의 설정을 따르므로 android:resizableActivity 속성이 무시됨

# android:resizableActivity



```
<activity  
    android:name=".UnresizableActivity"  
    android:resizeableActivity="false"  
    android:taskAffinity="" />
```

# 리사이즈 동작 중 표시할 색상 설정

- 화면 크기 조정 작업이 완료되기 전까지는 크기 변경 이벤트가 발생하지 않음 (Since DP2)
- 사용자가 현재 상태보다 화면 크기를 크게 조정하는 과정에서 빈 공간에 표시될 색상 값
- 기본값
  - android:windowBackground (정의되어 있을 경우)
  - android:windowBackgroundFallback  
(windowBackground가 정의되어 있지 않을 경우)

# 리사이즈 동작 중 표시할 색상 설정



..... System default

..... Customized

# 인접(Adjacent)화면에서 표시

- 멀티윈도우의 다른 창에서 화면을 표시할 때 사용
- 멀티윈도우 모드가 해제되어 있을 땐 동작하지 않음
- 동작 조건
  - 인텐트 플래그 지정
    - FLAG\_ACTIVITY\_NEW\_TASK/ FLAG\_ACTIVITY\_NEW\_DOCUMENT  
와  
FLAG\_ACTIVITY\_LAUNCH\_ADJACENT 함께 사용
  - 태스크의 Root 액티비티로 실행되어야 함
    - taskAffinity="" or launchMode="singleInstance"

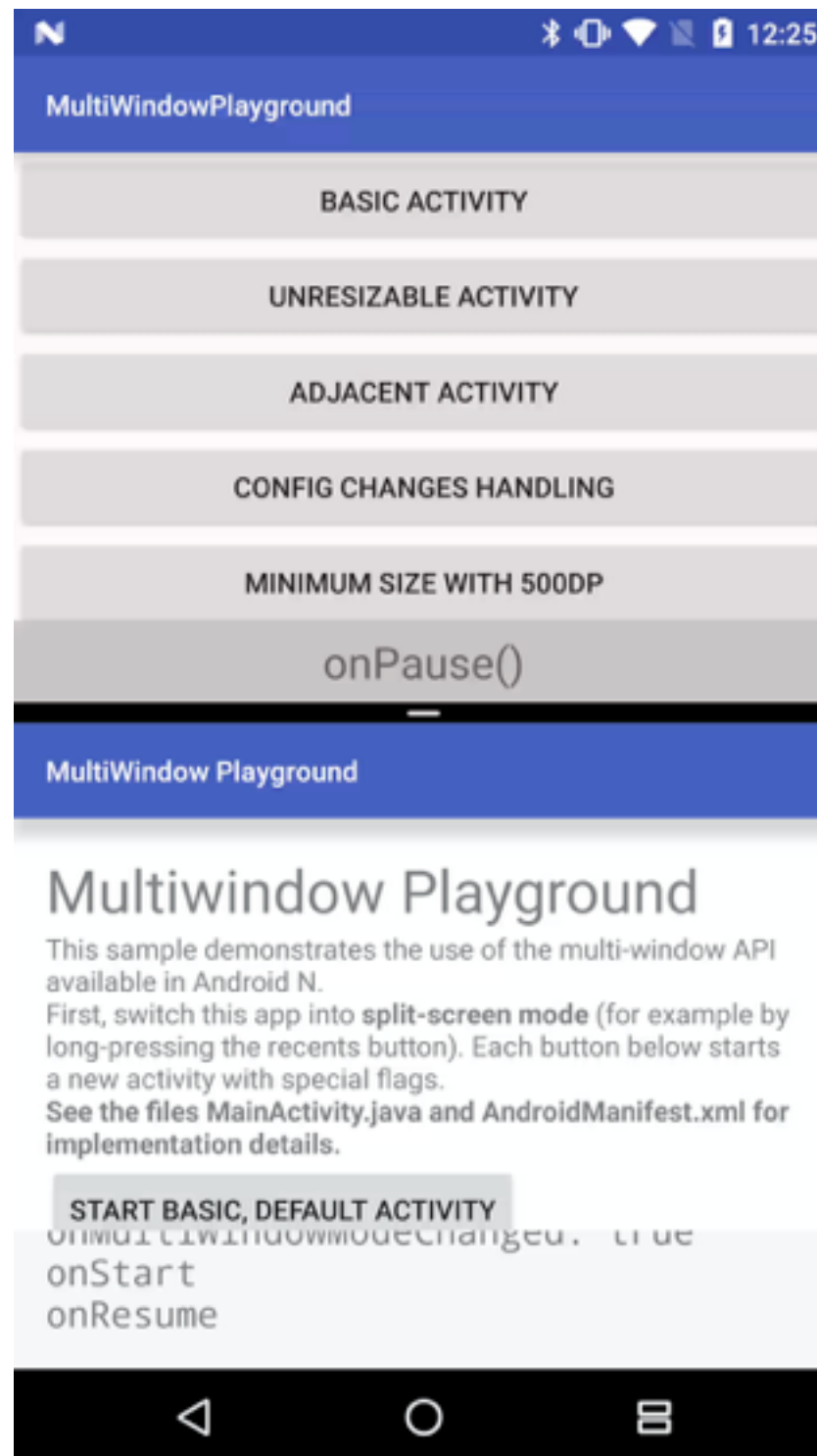


# 인접(Adjacent)화면에서 표시

```
Intent intent = new Intent(MainActivity.this, AdjacentActivity.class);  
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK  
                | Intent.FLAG_ACTIVITY_LAUNCH_ADJACENT);  
startActivity(intent);
```

```
<activity android:name=".AdjacentActivity"  
    android:label="@string/adjacent_activity"  
    android:launchMode="singleInstance"/>
```

# 인접(Adjacent)화면에서 표시



# 창 레이아웃 조절

- 기본/최소 화면 크기 및 위치 조절 지원
- 매니페스트의 <activity> 하위에 <layout> 노드를 추가하여 지정

```
<activity ..>  
    <layout android:defaultWidth="320dp"  
        android:defaultHeight="500dp"  
        android:gravity="top|start"  
        android:minimalWidth="320dp"  
        android:minimalHeight="500dp" />  
</activity>
```

# 창 레이아웃 조절

- `minimalWidth/minimalHeight`보다 창 크기가 작을 경우 화면이 크롭되어 표시
- `defaultWidth`, `defaultHeight`, `gravity` 값은 Free form 모드에서만 반영
- 태스크의 Root activity에서 지정된 값이 태스크 내 다른 액티비티에 일괄 적용됨
- 별도의 레이아웃 정책이 적용된 액티비티는 항상 태스크의 Root Activity로 실행되도록 설정 필요

# 창 레이아웃 조절



# Picture-in-Picture

- 항상 화면에 표시되어 있지만 사용자 인터랙션 불가
- `android:resizableActivity="false"`인 경우 사용 불가
- Related Methods
  - `Activity#enterPictureInPictureMode()`
  - `Activity#inPictureInPictureMode()`
  - `Activity#onPictureInPictureModeChanged(boolean)`

# 멀티윈도우 모드 상태 확인

- `Activity#isInMultiWindowMode()`
- `Fragment#isInMultiWindowMode()`
- `Activity#onMultiWindowModeChanged(boolean)`

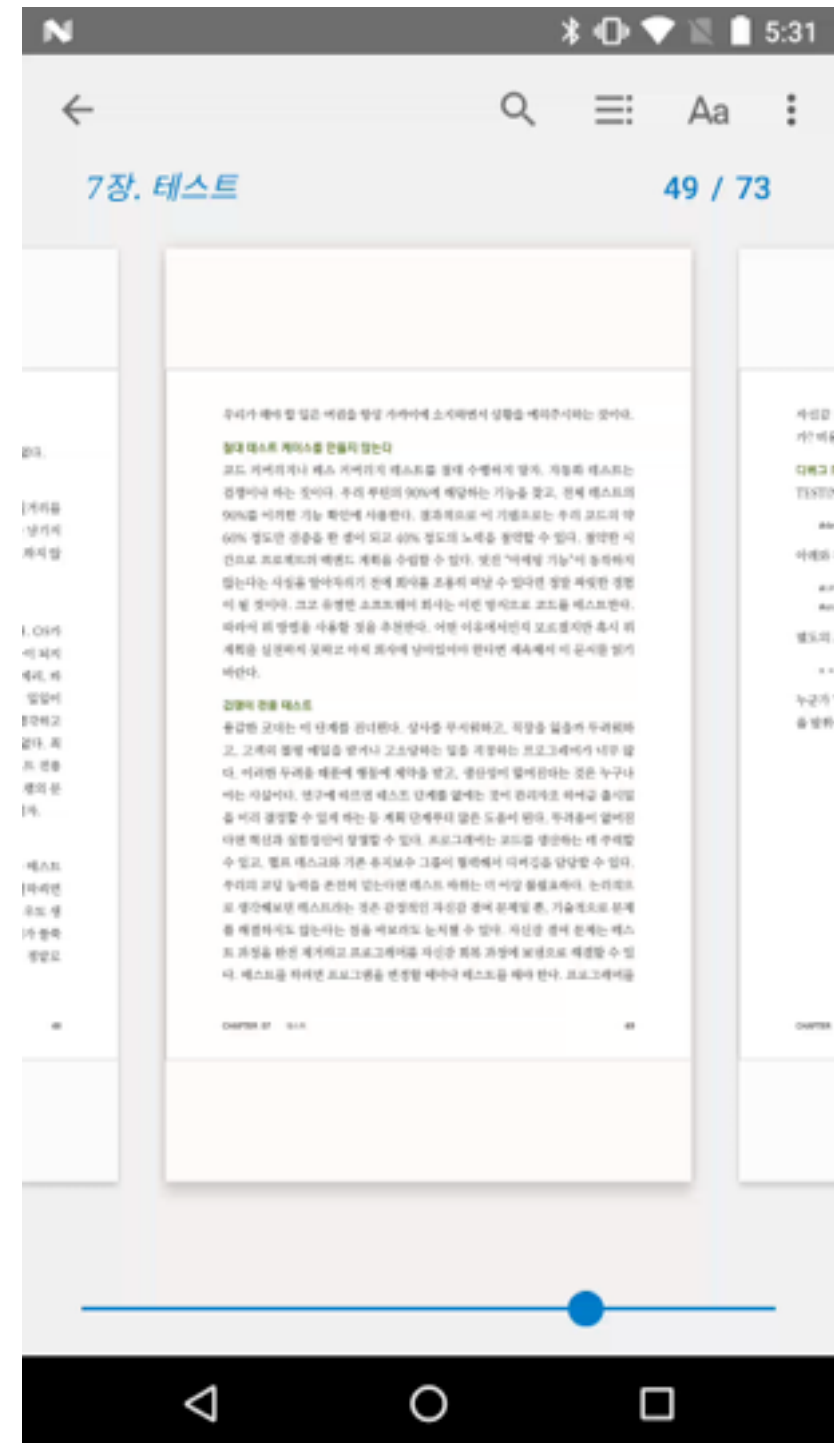
# 멀티윈도우 모드 제약사항

- Status bar 및 Navigation bar 숨기기
  - 항상 표시된 상태로 유지
- 화면 방향(portrait, landscape 등) 지정
  - 화면 방향 변경은 무시됨



# 멀티윈도우 모드 제약사항

- Immersive mode
- Status bar /  
Navigation bar를 제외한,  
해당 앱 영역에만 적용



# Multi Window: Checklist

# Multi Window: Checklist

- 액티비티 생명주기에 적절한 동작을 처리하고 있는가?
- 액티비티 재생성시 UI에 영향을 주지 않는가?
- 액티비티 재생성시 메모리 누수가 발생하지 않는가?
- 특정한 환경(예: portrait, landscape)에서만 사용되는 리소스가 존재하는가?
  - 전체 환경에서 사용 가능하도록 리소스 구조를 변경하거나, 불가피할 경우 특정 화면은 멀티 윈도우를 지원하지 않도록 처리

# Multi Window: Checklist

- 사용자 인터랙션이 가능한 상태 (활성 상태)와 인터랙션은 불가하지만 아직 화면에 보이는 상태 (일시정지 상태)는 적절히 처리되었는가?
- 예: 비디오 재생 애플리케이션은 일시정지 상태에서도 비디오 재생을 유지해야 함
- onPause()에서 할 작업과 onStop()에서 할 작업의 구분 필요

# Multi Window: Checklist

- 태스크의 Root 액티비티로 설정되거나, 액티비티 실행시 추가 플래그가 필요한 경우 적절히 처리되었는가?
  - 멀티윈도우 지원 여부 (android:resizableActivity)
  - 인접(Adjacent) 모드로 표시
  - 창 크기 및 위치 (<layout>)
- 화면이 줄어들어도 앱 사용에 지장이 없는가?
  - 스크롤 처리 추가, 메뉴 및 버튼 배치 확인

# 기타 참고자료

# Split Screen Design Spec

- 멀티윈도우를 지원하는 애플리케이션을 제작할 때 고려해야 할 사항들에 대한 가이드라인
- <https://www.google.com/design/spec/layout/split-screen.html>

# Drag and Drop

- 멀티윈도우로 표시되는 액티비티 간 Drag and Drop 이벤트 전달 가능
- 데이터 접근을 위해 권한이 필요한 경우 임시로 접근 허용 가능



# And more

- Android Developers: Multi Window
  - <http://goo.gl/2Mzmmy>
- 5 tips for preparing for Multi-Window in Android N
  - <https://goo.gl/a4d4pq>

# And more

- 구글 공식 멀티윈도우 예제
  - <https://goo.gl/Ybz1tv>
- 발표자료에서 사용한 예제
  - <https://goo.gl/uhTxPV>