MULTI-WINDOW AND YOUR APP

ANDROID 7.0 AND CHROME OS

FORMS OF MULTI-WINDOW

- Split-screen
- Picture-in-picture: mostly for Android TV
- Freeform
- Chrome OS: freeform kinda sorta but not

GETTING YOUR APP IN MULTI-WINDOW

- Generally, do nothing unusual
- Your app winds up in windows... whether you like it or not
- Older `targetSdkVersion` may give a warning `Toast` to the user

GETTING YOUR APP OUT OF MULTI-WINDOW

THE THEORY

- Use `android:resizeableActivity="false"` on `<activity>`
- Older `targetSdkVersion` and `android:screenOrientation`

GETTING YOUR APP OUT OF MULTI-WINDOW

THE REALITY

- Root activity of the task determines the window behavior
- Exported activities attempting to break out of multi-window need to ensure they run in a task other than default
  - Some third-party apps might specifically launch you in their own task, to force you back into a window
- Changes in task behavior may impact other things (BACK behavior, overview screen entries)

ABOUT THE SIZING

- User can change the size of the window (e.g., drag split bar in split-screen)
- Your window might be as small as 220dp in either direction
- Bonus: if you can handle this, you can handle `small`-rated devices
- Can specify a minimum size, but then your UI is cropped

LIFECYCLE METHODS

PAUSE FOR A MOMENT AND THINK ABOUT `onPause()`

- Paused = visible but not receiving user input
- Stopped = no longer visible
- Outside of multi-window, being only paused is unusual
- With multi-window, being only paused is very common
- Are you doing the right thing when you are paused?

CONFIGURATION CHANGES

- By default, activity may be destroyed and recreated due to multi-window
  - User enters multi-window with your activity visible
  - User resizes your window (if big enough change)
  - User exits multi-window with your activity focused
- `android:configChanges` now "out of the doghouse"
- Note: your orientation flips when entering/exiting split-screen mode

NOT ALL HOMES ARE CREATED EQUAL

- We expect to get called with `onStop()` when the user presses HOME
- Still true in single-window mode, plus split-screen mode if you are in bottom/right pane
- Not true if you are in top/left pane of split-screen mode
  - Supposed to be a transient state
  - May cause problems if you hold wakelocks, continue playing media, etc.

LAUNCHING ANOTHER WINDOW

- Add `FLAG_ACTIVITY_LAUNCH_ADJACENT`, along with appropriate task flags, to `Intent`
- Ignored if device is not in multi-window mode
- "This flag is only used in split-screen multi-window mode" (???)
- If activity exported, third-party apps can add this flag, force you into multi-instance state

DETECTING MULTI-WINDOW

WHAT PROBABLY WORKS: `onMultiWindowModeChanged()`

- Override on your `Activity` or `Fragment`
- Do something useful (e.g., enable/disable "launch adjacent" action bar item)

DETECTING MULTI-WINDOW

THEN, THERE IS `isInMultiWindowMode()`

- User exits multi-window mode; your activity may be destroyed and recreated
- In `onCreate()`, `isInMultiWindowMode()` returns `true` (???) for a while (?!?) after which it returns `false`, even if you never left the main application thread (?!?!?)
- IOW, a race condition ... but one that is "working as intended"

DRAGGIN' AND DROPPIN'

- `startDrag()`, `OnDragListener`, and kin
- Ill-used, but around since API Level 11
- Works across activities, if both are visible
- Cross-app behavior
  - Opt-in to initiate a drag event that can be dropped in another app's window
  - Cannot avoid cross-app drop events

HEY, WHAT ABOUT FREEFORM?

Manufacturers of larger devices can choose to enable freeform mode, in which the user can freely resize each activity. If the manufacturer enables this feature, the device offers freeform mode in addition to split-screen mode.

HEY, WHAT ABOUT FREEFORM?

- Docs say that you should test on freeform
- Officially, it does not exist
- Unofficially, `adb shell settings put global enable_freeform_support 1` and reboot

ANDROID ON CHROME OS

- Available for Acer Chromebook R11/C738T, ASUS Chromebook Flip, and 2015 Google Chromebook Pixel
- Rollout to everyone else... someday

CHROME OS AND YOUR APP

- By default, most apps will ship to Chrome OS devices that support touchscreens
  - Unclear what percentage of Chrome OS market has touchscreens
  - Other app features might still block availability on these Chromebooks
- By default, most apps will *not* ship to Chrome OS devices that lack touchscreens

WHAT'S DIFFERENT ON CHROME OS

WINDOWS

- Landscape
- Portrait
- Full-screen
- Future: N freeform?

WHAT'S DIFFERENT ON CHROME OS

STUFF CHROME OS DOESN'T LIKE

- `android.software.input_methods`: for implementing your own IME
- `android.software.app_widget`: for apps whose primary purpose is to publish an app widget
- `android.software.live_wallpaper`: for apps whose primary purpose is to publish some live wallpaper
- `android.software.home_screen`: for home screen replacement apps

WHAT'S DIFFERENT ON CHROME OS

NOT ALWAYS A TOUCHSCREEN

- Many Chromebooks, all Chromeboxes/Chromebits lack touchscreens
- Need to say that touchscreen is not required in the manifest to be distributed to such devices
- To do *that*, need to adequately test your app with keyboard/mouse combination
- Do not assume a trackpad!

WHAT'S DIFFERENT ON CHROME OS

OTHER FIDDLY BITS

- Notifications: basics work, but get converted into Chrome OS-style notifications
- `Theme.Translucent`: still gets a window
- External displays: owned by Chrome OS, no `Presentation`

WHAT'S DIFFERENT ON CHROME OS

OH, AND THEN THERE ARE BUGS

- Hardcoded support to go to Play Store for certain requests (e.g., `ACTION_VIEW` of PDF) if no app is installed
- URL-handling `<intent-filter>` elements ignored by browser
- `ACTION_IMAGE_CAPTURE` causes Chrome OS reboot
- Changing font scale in Settings does not work
- And so on

TESTING ON CHROME OS

ENABLING SIDE-LOADING

- Move your device into "developer mode" (details vary by device)
- Prepare to press Ctrl-D on every reboot (or wait 30 seconds)
- Enable "unknown sources" in Android's Settings app

TESTING ON CHROME OS

adb

- Seriously nasty set of instructions, affecting Chrome OS device and your development machine
- End result: use `adb connect [IP]:22` to connect to Chrome OS device, run apps from Android Studio

WHAT YOU SHOULD WORRY ABOUT NOW

ANDROID 7.0 MULTI-WINDOW

- Does your app fit within a small window?
- Does your opt-out strategy really work?
- Are you handling `onPause()` properly?
- Do you need to optimize configuration changes?
- Is your drag-and-drop safe?

WHAT YOU SHOULD WORRY ABOUT NOW

ANDROID ON CHROME OS

- Grab a developer device, do some light testing
- Nothing much more, unless Chrome OS support is strategic, since no idea when this will widely ship
- Consider hacky opt-out (e.g., require `android.software.app_widget`) if seeing problems or wish to avoid lots of testing work