

Android DarkTheme

+ CodeLab



안성용 SOUP

soupyong@gmail.com

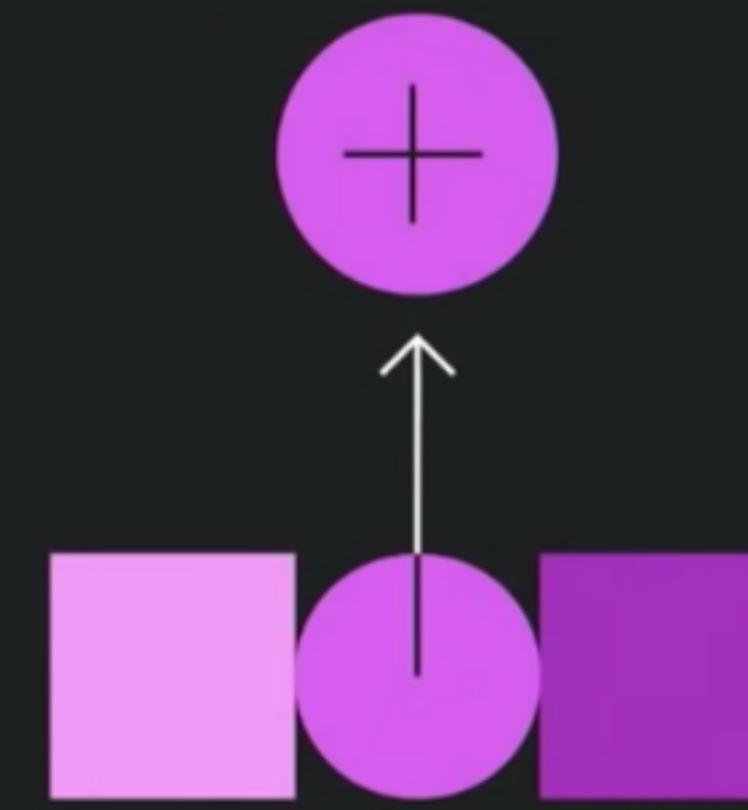
Why? 🤔



Battery

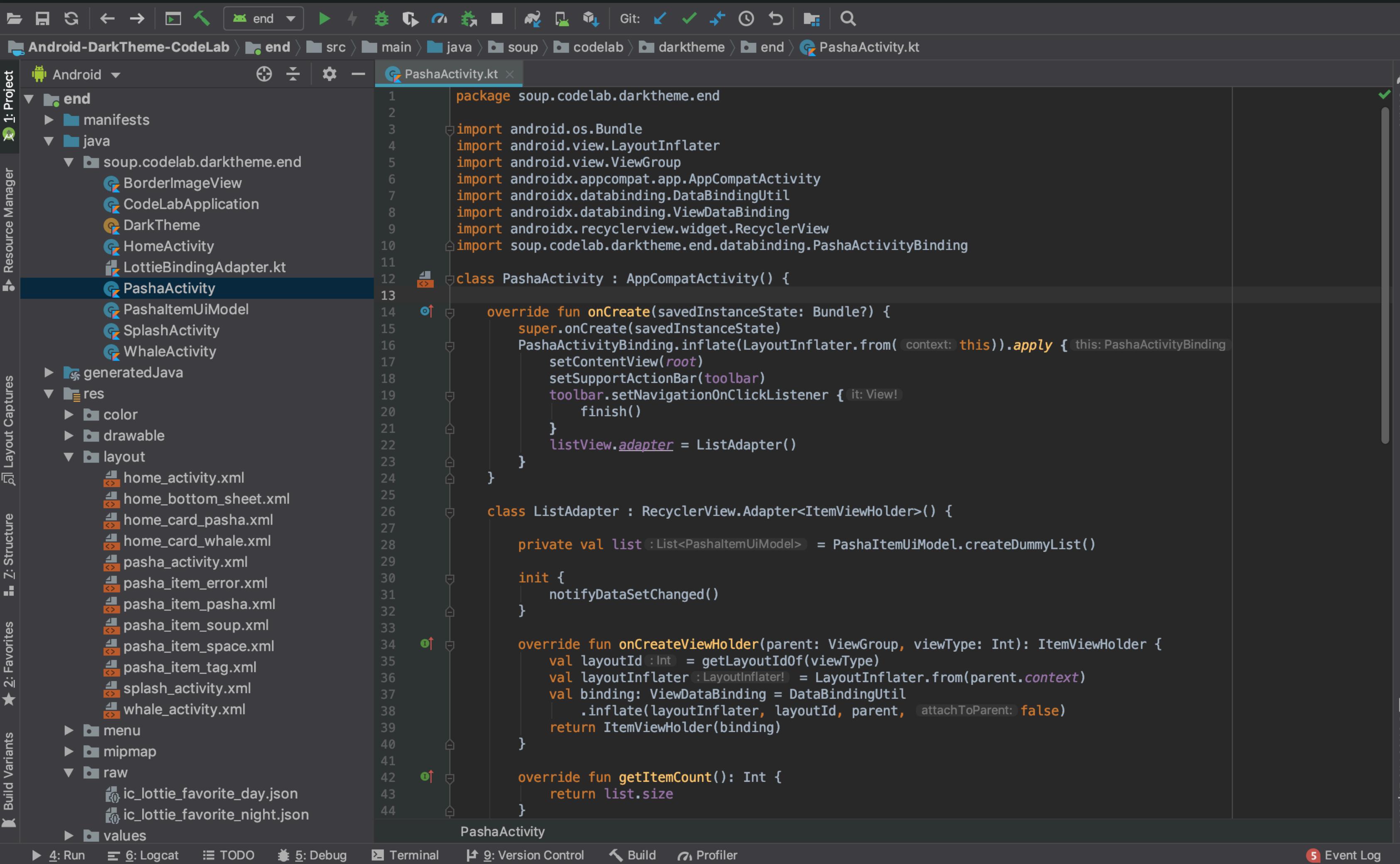


Accessibility



Environment

Why? 😊 ✨



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "Android-DarkTheme-CodeLab". The "end" module contains Java files under "src/main/java/soup.codelab.darktheme.end". The "PashaActivity" file is selected in the Project Manager.
- Code Editor:** The "PashaActivity.kt" file is open. It defines a class `PashaActivity` that extends `AppCompatActivity`. The `onCreate` method inflates a layout from XML and applies a binding. The `ListAdapter` class implements `RecyclerView.Adapter` and uses `DataBindingUtil` to bind data to a list of `PashalItemUiModel` objects.
- Toolbars and Status Bar:** The top bar shows standard Android Studio icons and the current file path. The status bar indicates "end" and "PashaActivity.kt".
- Sidemenu:** The left sidebar includes sections for "Resource Manager", "Layout Captures", "Structure", "Favorites", and "Build Variants".
- Bottom Navigation:** The bottom navigation bar includes tabs for Run, Logcat, TODO, Debug, Terminal, Version Control, Build, Profiler, and Event Log.

OS Setting



- **Android Q**

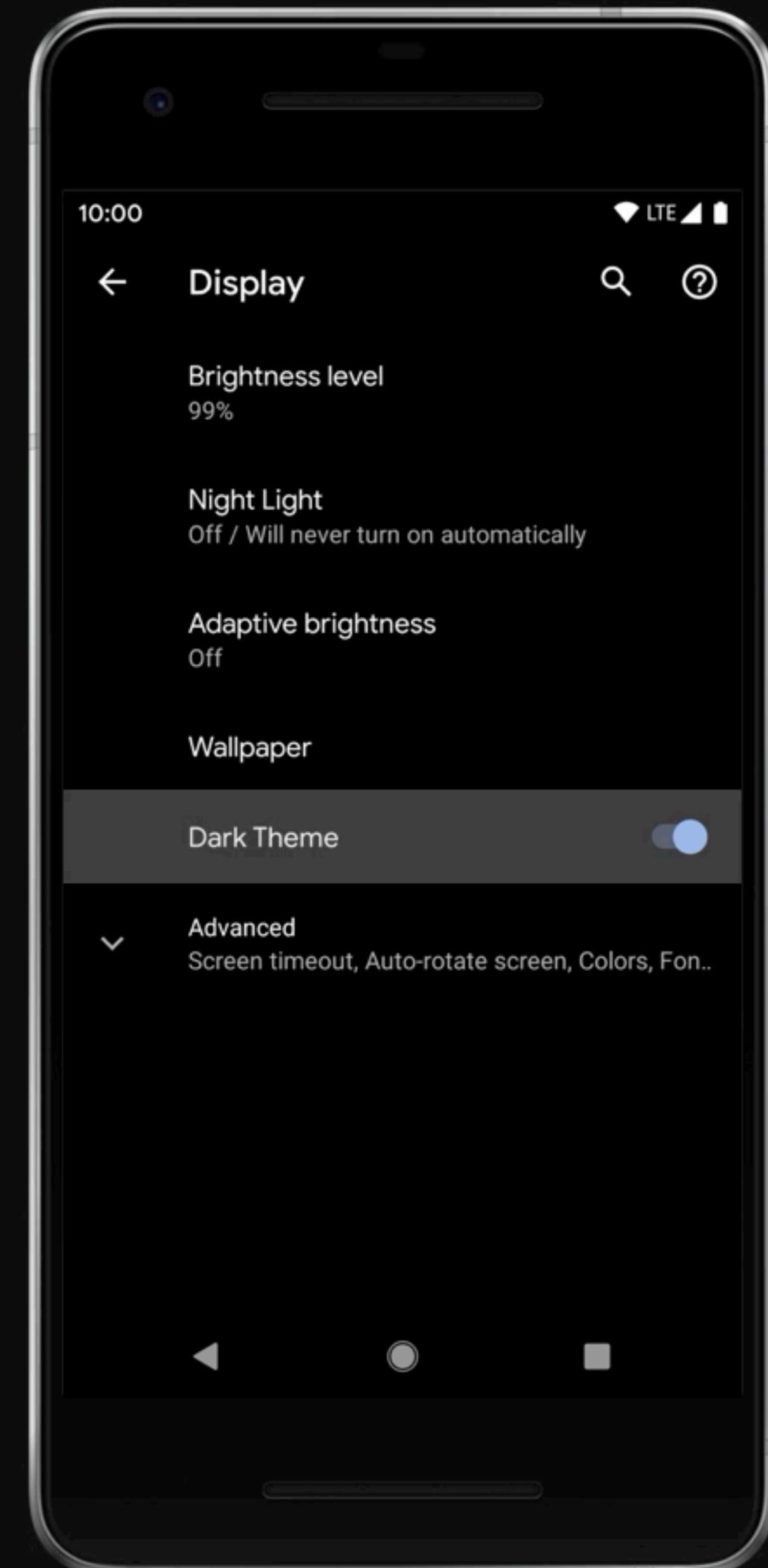
Settings > Display > Dark Theme

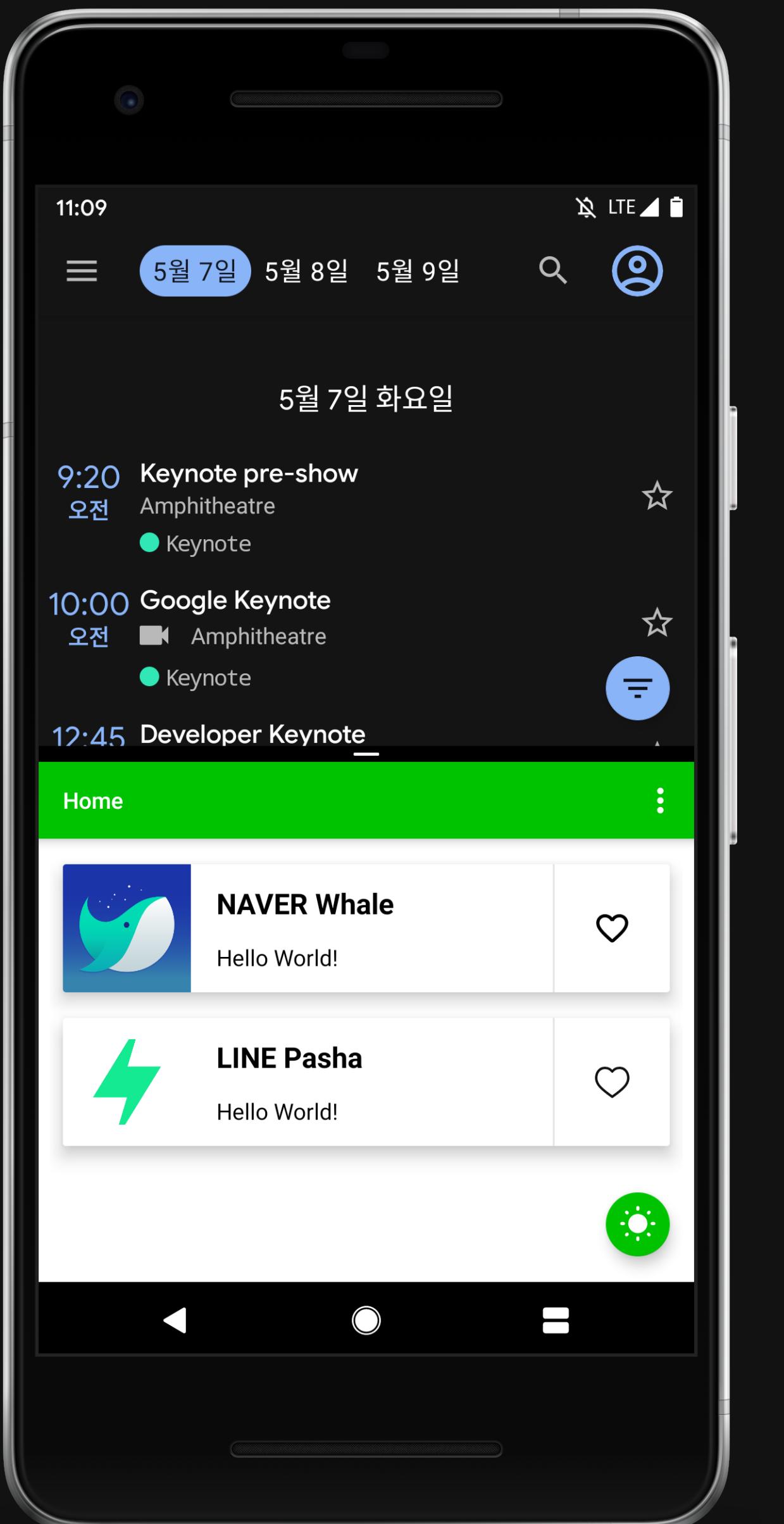
('어두운 테마' QuickSetting도 추가됨)

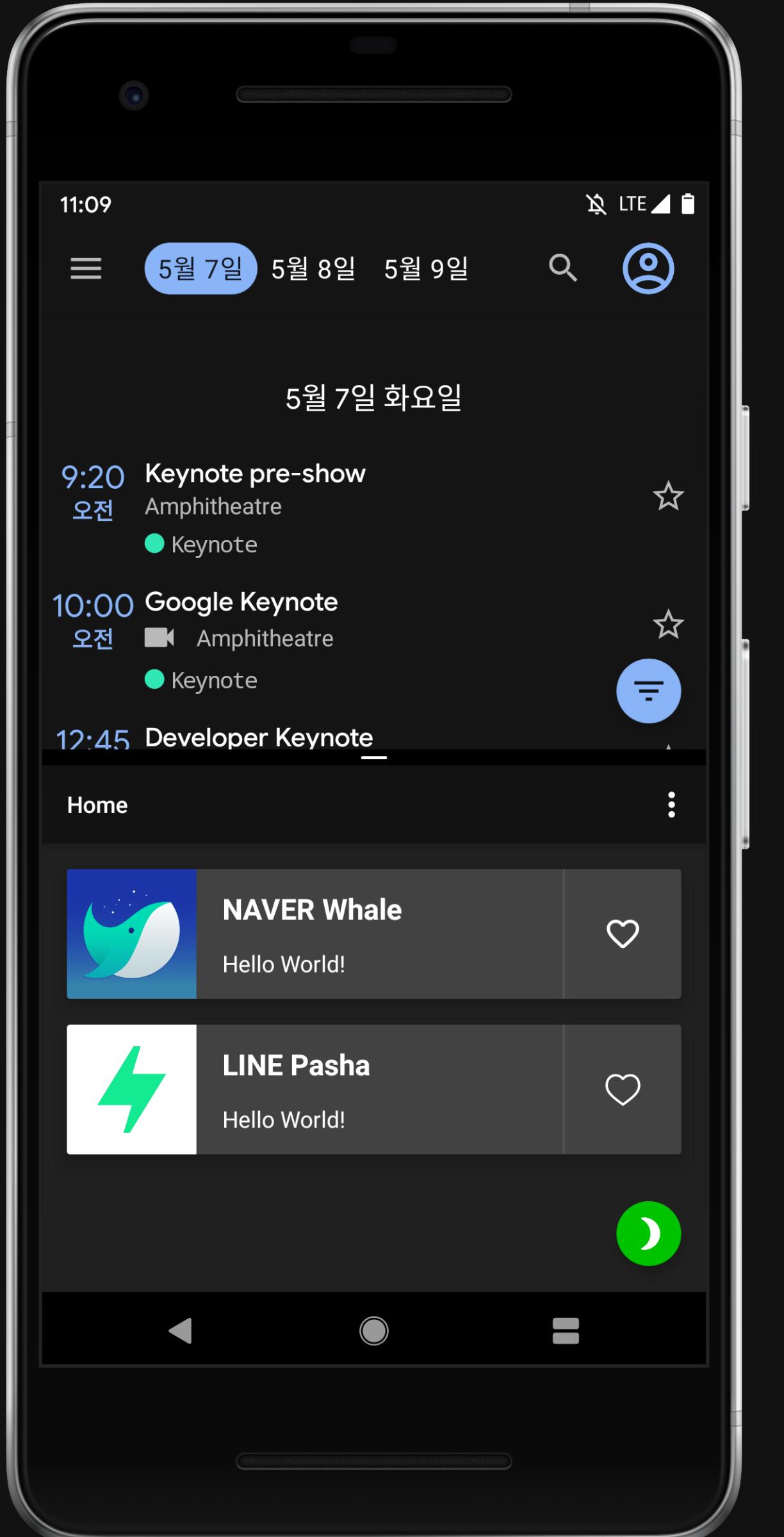
- **Android P**

Settings > System > Developer options > Night mode

(삼성 단말은 '야간 모드' QuickSetting 제공)







목표

디자인 리소스를 변경하지 않고,
다크테마 미리 적용해보기

ForceDark

- 앱을 다크 테마로 자동 변환하는 기능
- API 29 (Q)에서만 사용 가능

10:10

Schedule

May 8 May 9 May 10

9:20 AM Keynote pre-show
Keynote 40 minutes / Amphitheatre

10 AM Google Keynote
Keynote 1 hour / Amphitheatre

11:30 AM [Office Hour] Firebase Remote Config and A/B testing
Firebase 1 hour / Office Hours tent - C

[Office Hour] Android Bazel
Android & Play 1 hour / Office Hours tent - B

[Office Hour] Building websites with AMP and PWA
Web 1 hour / Office Hours tent - D

[App Review] Google accessibility
Accessibility 1 hour / App Reviews tent - F

[Office Hour] Google Mobile Ads SDK
Ads 1 hour / Office Hours tent - E

New in Q

Force Dark

Schedule		
	May 8	May 9 May 10
9:20 AM	Keynote pre-show 40 minutes / Amphitheatre	★
	Keynote	
	Google Keynote 1 hour / Amphitheatre	★
	● Keynote	
11:30 AM	[Office Hour] Firebase Remote Config and A/B testing 1 hour / Office Hours tent - C	★
	● Firebase	
	[Office Hour] Android Bazel 1 hour / Office Hours tent - B	★
	● Android & Play	
	[Office Hour] Building websites with AMP and PWA 1 hour / Office Hours tent - D	★
	● Web	
	[App Review] Google accessibility 1 hour / App Reviews tent - F	★
	● Accessibility	
	[Office Hour] Google Mobile Ads SDK 1 hour / Office Hours tent - E	★
	● Ads	

Schedule		
	May 8	May 9 May 10
9:20 AM	Keynote pre-show 40 minutes / Amphitheatre	★
	● Keynote	
10 AM	Google Keynote 1 hour / Amphitheatre	★
	● Keynote	
11:30 AM	[Office Hour] Firebase Remote Config and A/B testing 1 hour / Office Hours tent - C	★
	● Firebase	
	[Office Hour] Android Bazel 1 hour / Office Hours tent - B	★
	● Android & Play	
	[Office Hour] Building websites with AMP and PWA 1 hour / Office Hours tent - D	★
	● Web	
	[App Review] Google accessibility 1 hour / App Reviews tent - F	★
	● Accessibility	
	[Office Hour] Google Mobile Ads SDK 1 hour / Office Hours tent - E	★
	● Ads	

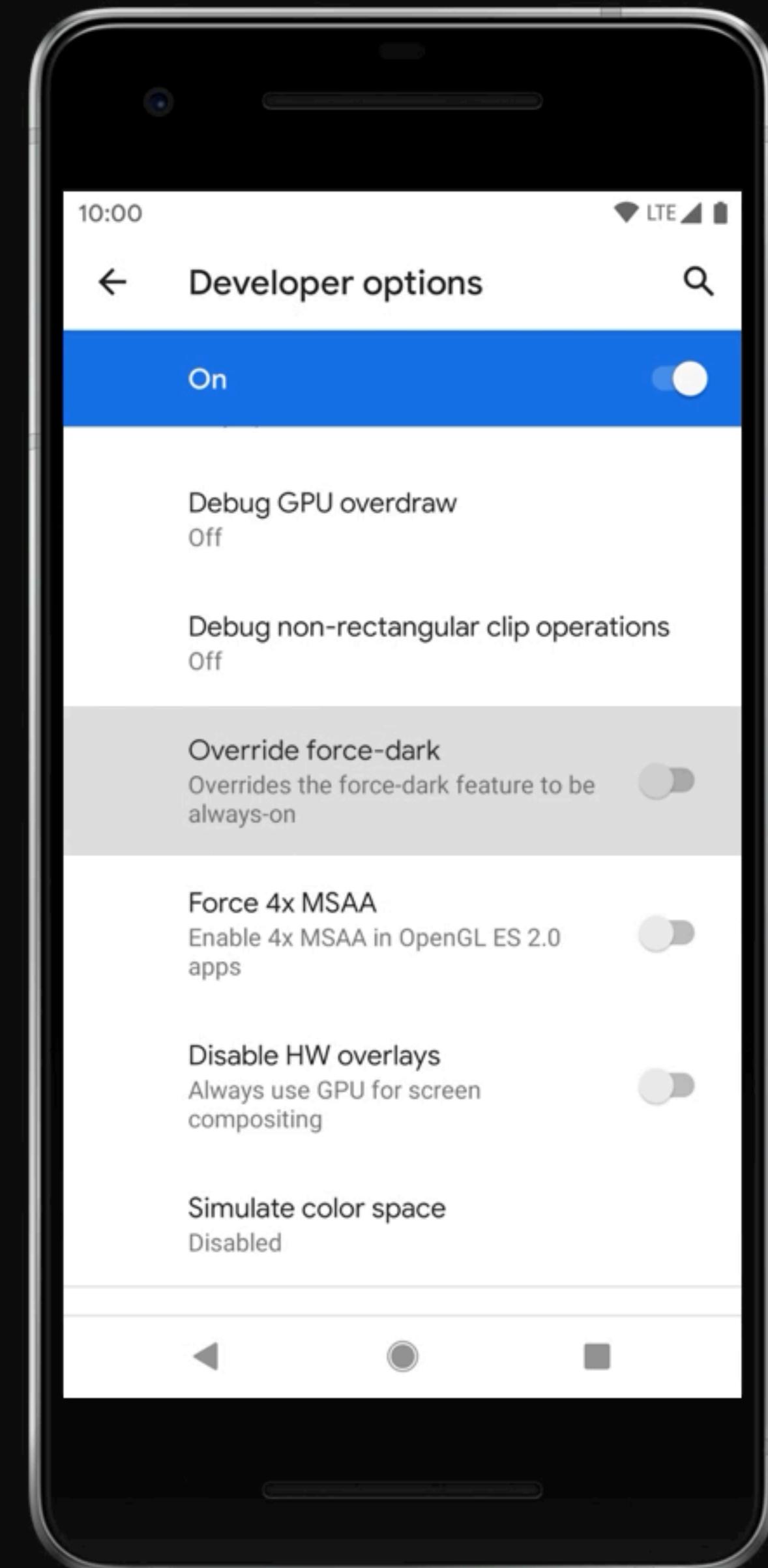
FORCE DARK

Override ForceDark

- 모든 앱에 force-dark를 적용하는 개발자 모드
- **Android Q**

Settings > System > Developer options > Override force-dark

- 하지만 Q Beta4부터 **동작하지 않는다.** 🤦



FORCE DARK

Enable ForceDark

- 앱 테마의 `forceDarkAllowed` 속성을 `true`로 설정하면 활성화된다.

```
<style name="AppTheme"
    parent="@style/Theme.MaterialComponents.Light"
    <item name="android:forceDarkAllowed">true</item>
</style>
```

- 단, **Light** 테마만 **ForceDark**가 적용될 수 있다.

```
<style name="Theme.AppCompat.Light" parent="...">
    <item name="isLightTheme">true</item>
</style>
```

FORCE DARK

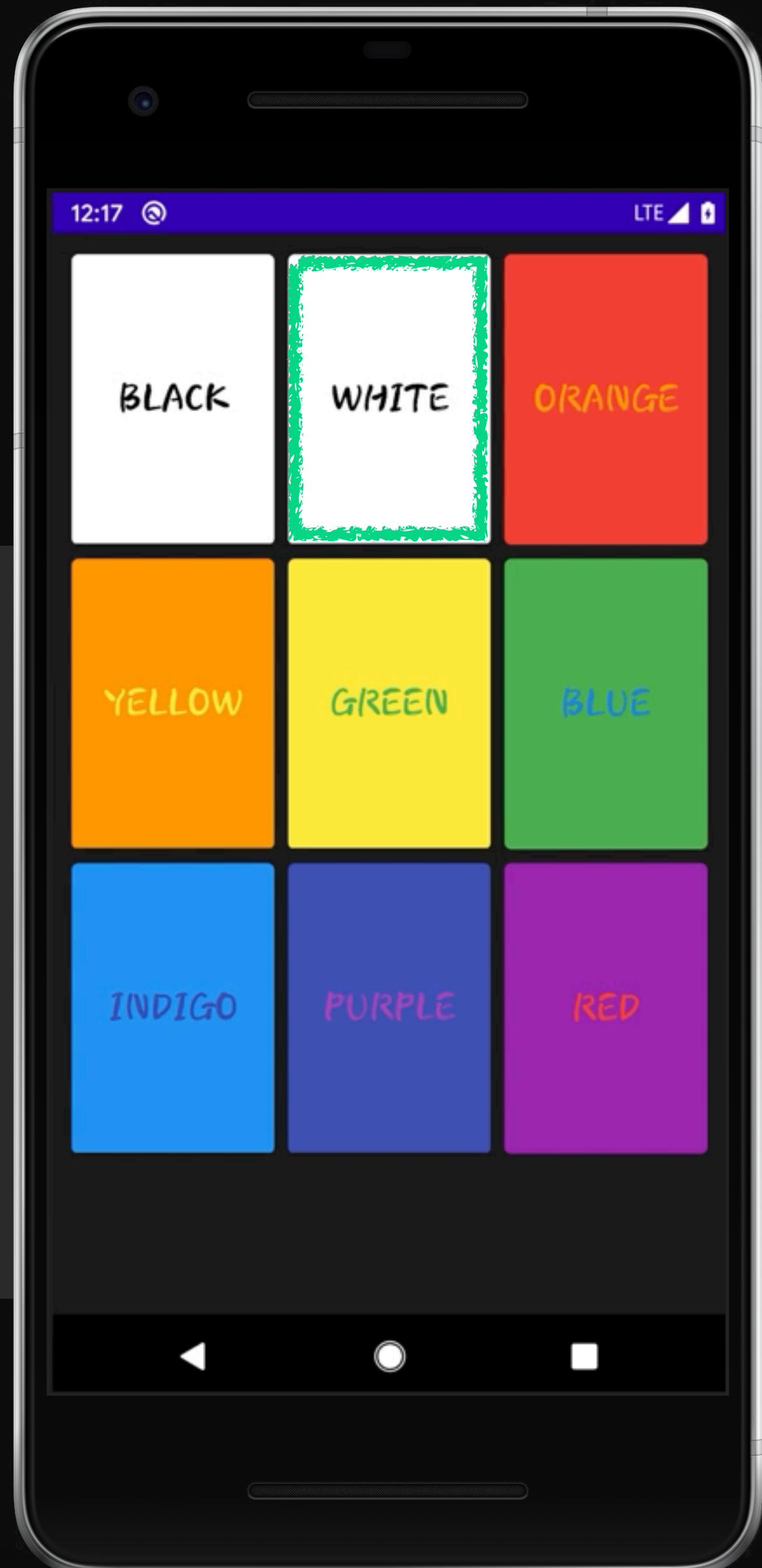
Disable ForceDark (Partially)

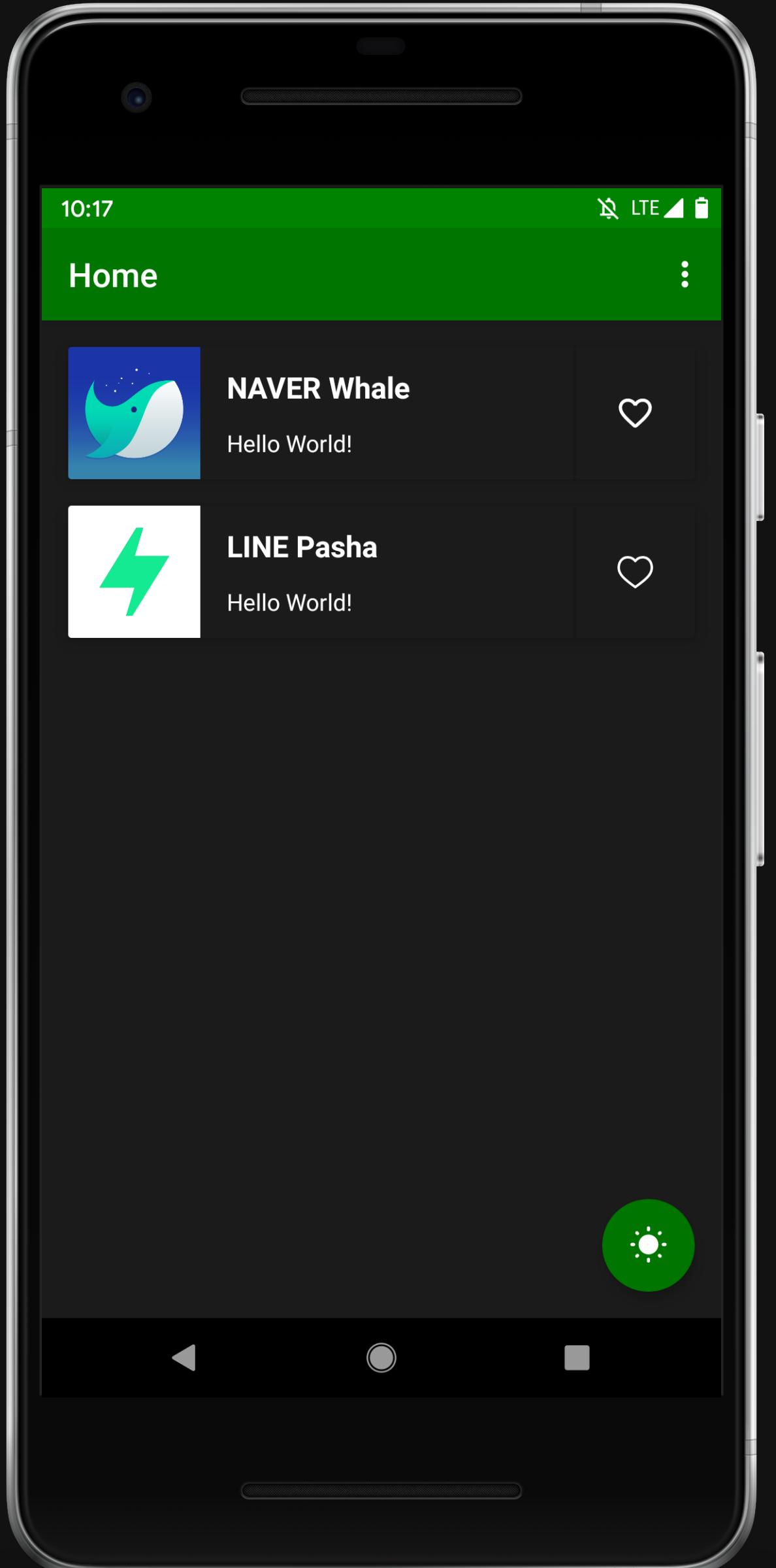
- 부분적으로 ForceDark를 비활성화할 수도 있다.

```
<!-- For Layout XML -->  
<View android:forceDarkAllowed="false" />
```

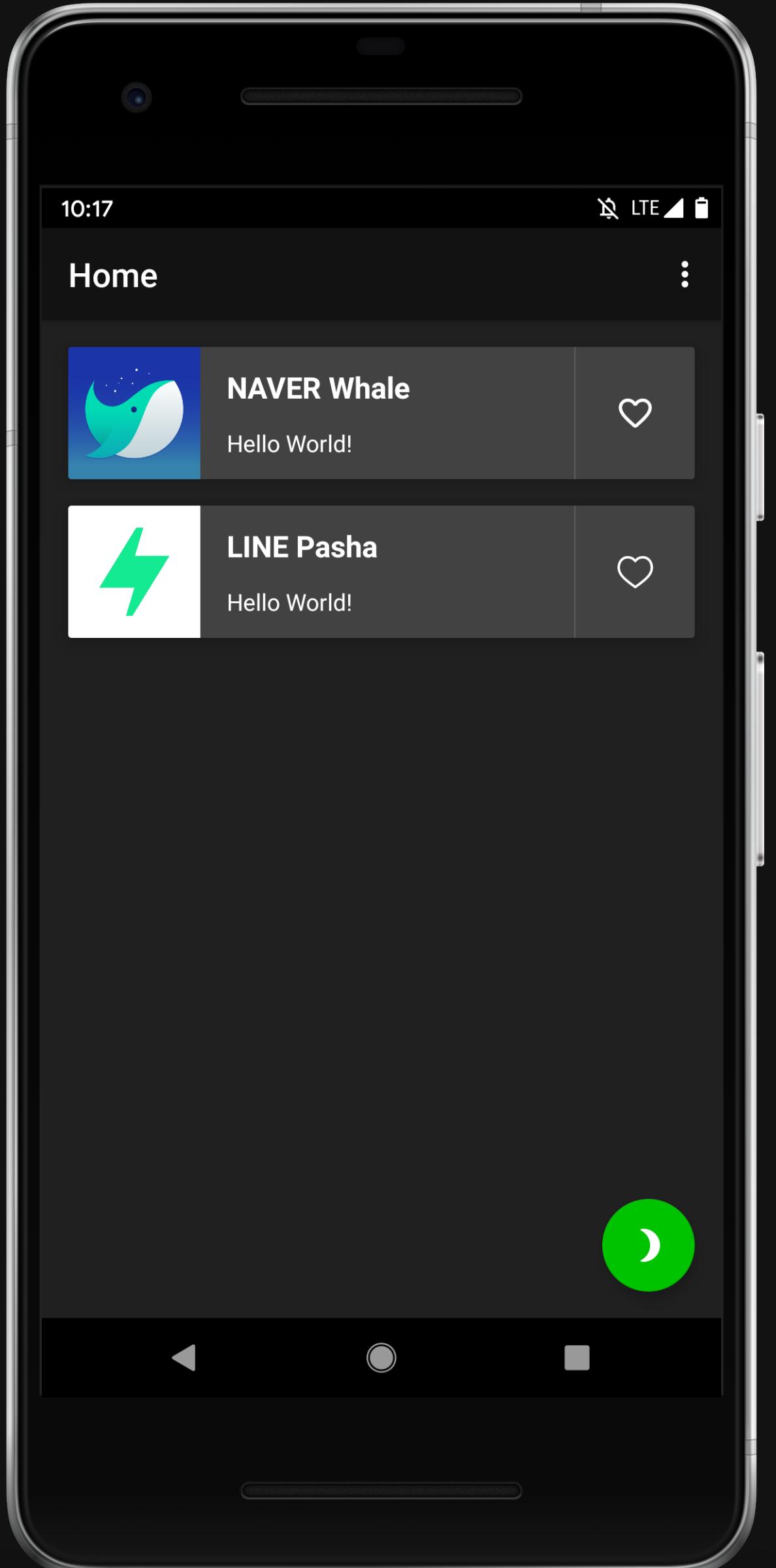
```
// For Kotlin  
view.isForceDarkAllowed = false
```

```
// For Java  
view.setForceDarkAllowed(false)
```





After
ForceDark



After
CodeLab

Android DarkTheme

CodeLab

CodeLab

⚠ 코드랩은 이 마크가 있는 페이지를 따라 진행됩니다.

문제



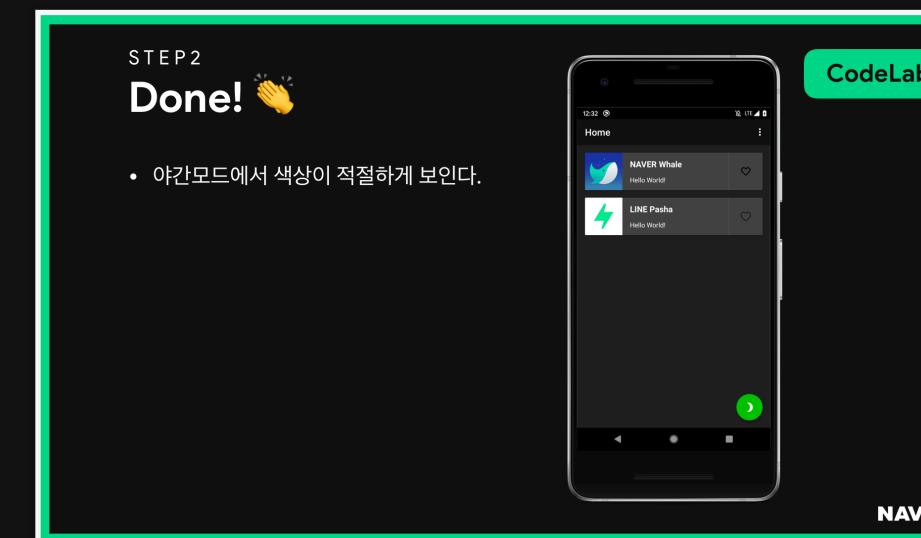
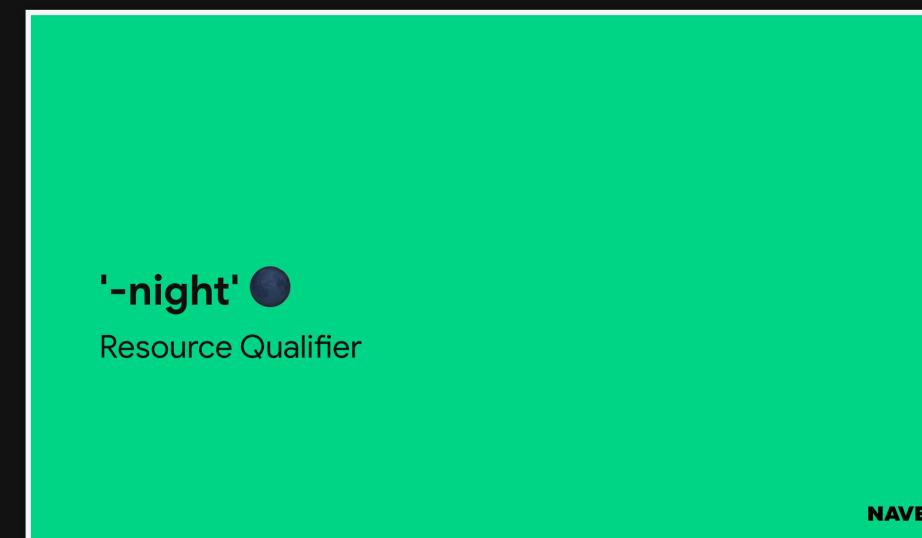
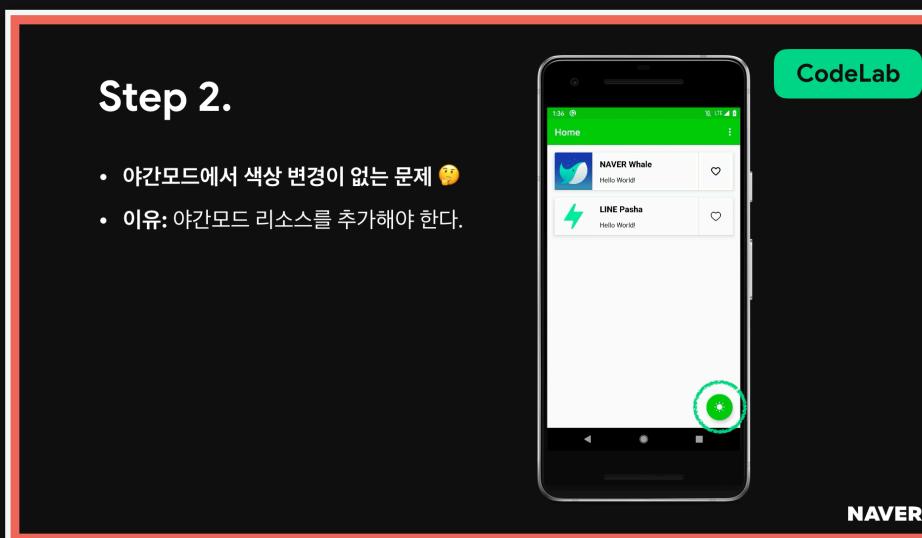
설명



적용



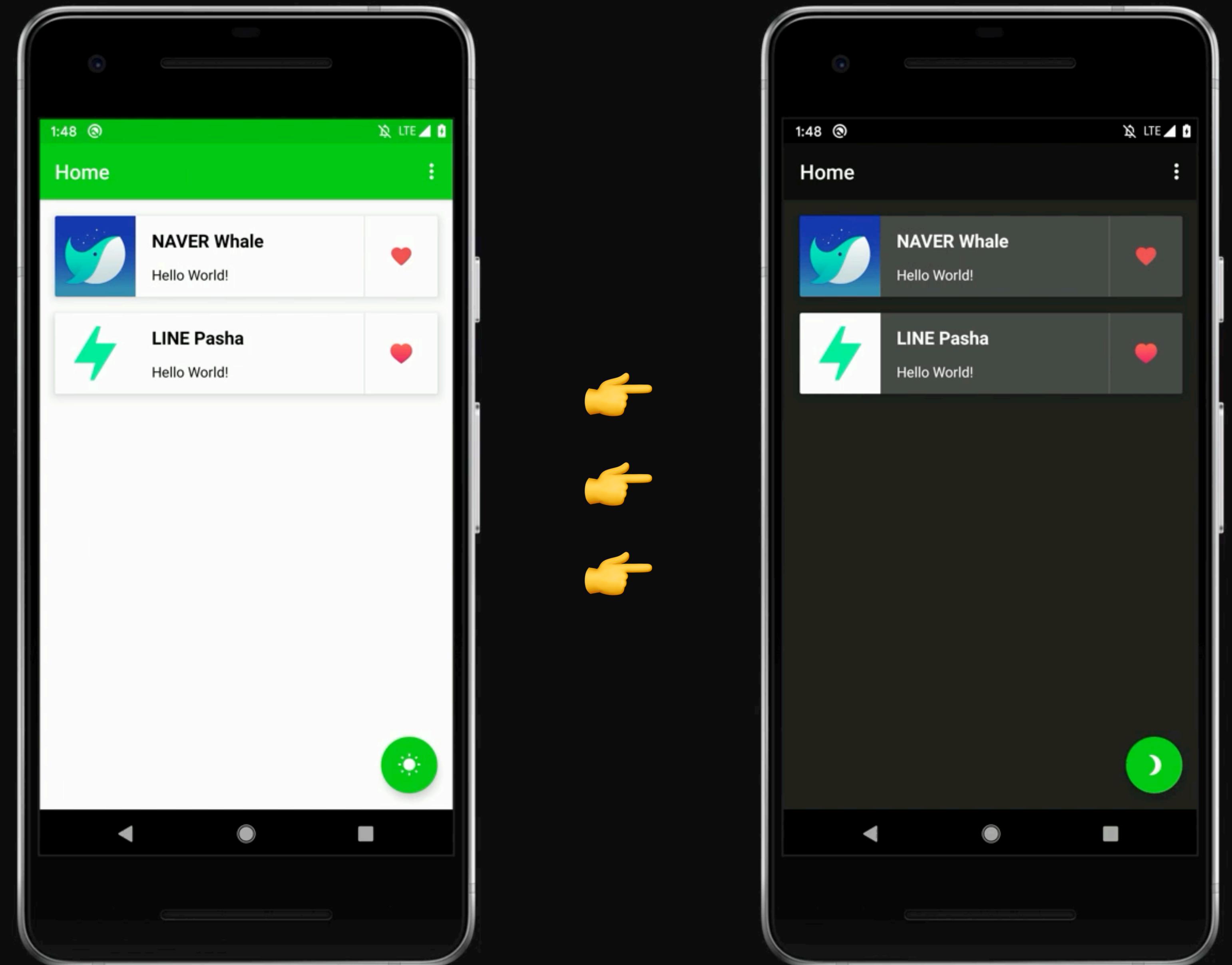
결과



준비

```
$ git clone https://github.com/fornewid/Android-DarkTheme-CodeLab
```

- Android Studio 3.4 이상
- API 21 이상 Emulator / 단말기 (STEP7은  필요)
- Kotlin
- DataBinding 



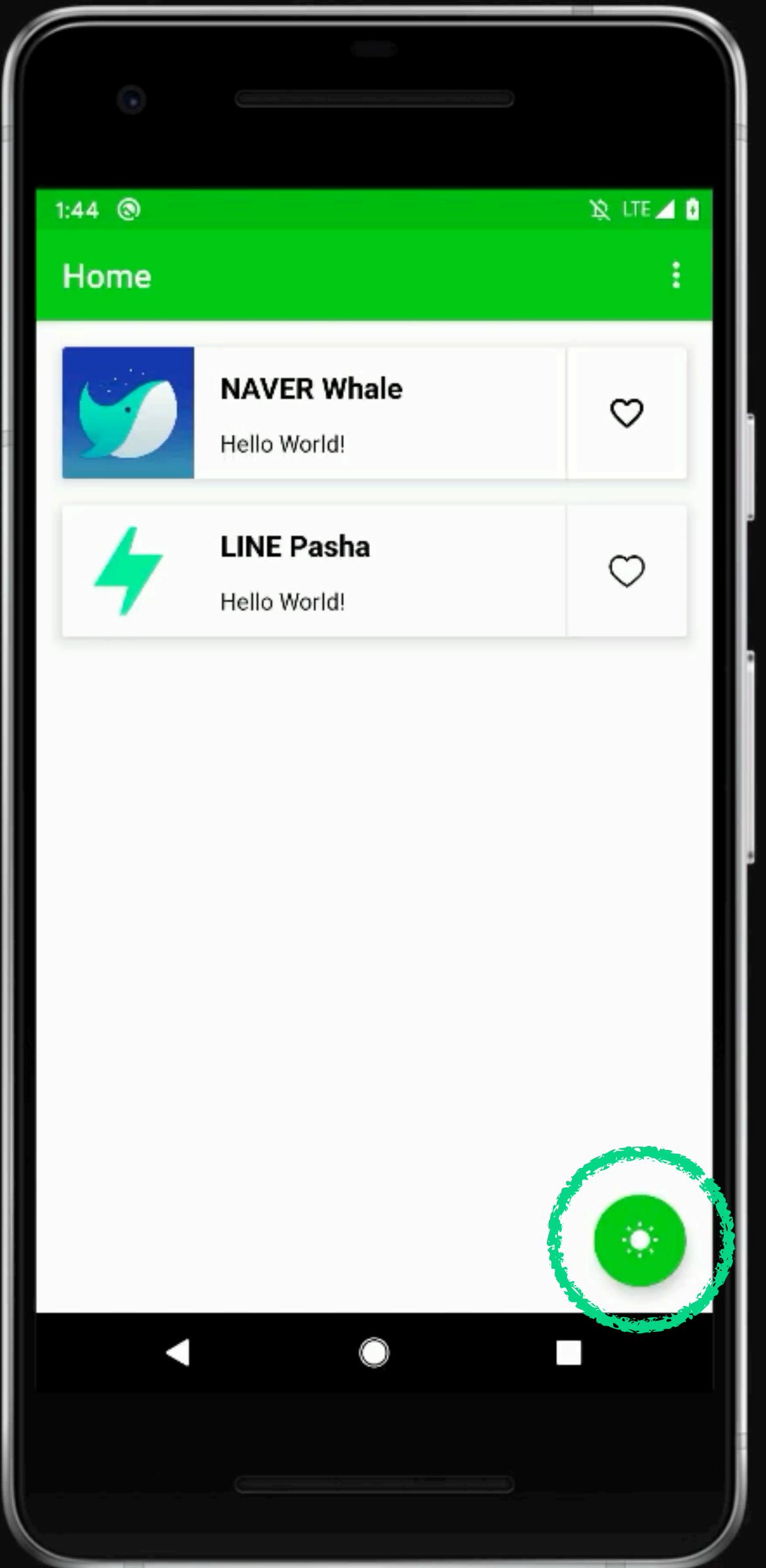
NAVER

코드랩 목표

Java/Kotlin 수정하지 않고,
다크테마 적용하기

Step 1.

- 야간/주간 모드가 변경되지 않는 문제 🤔
- AppCompatDelegate API 호출 시,
AppCompatActivity가 재시작되지 않는다.



AppCompat 1.1.0



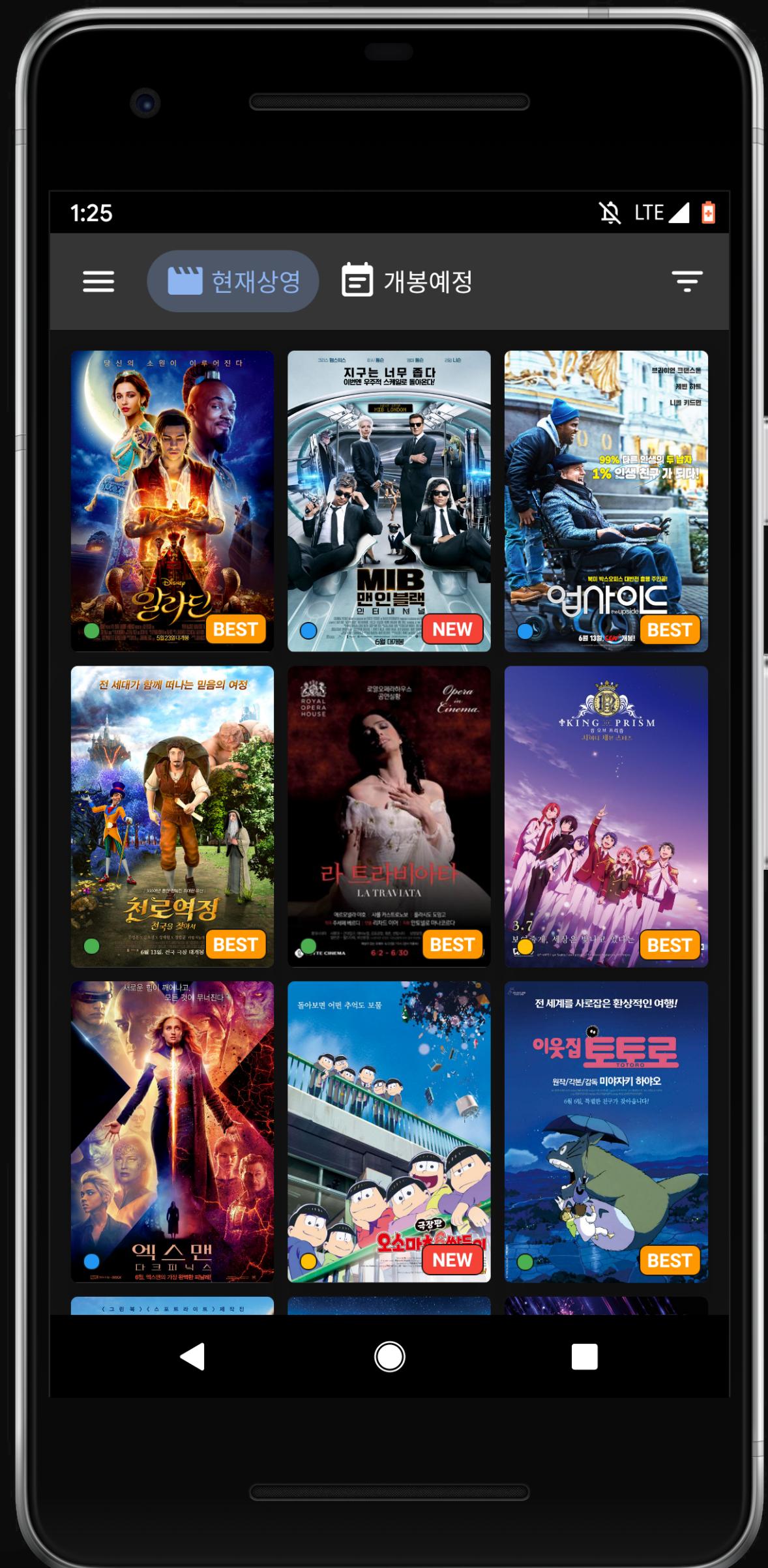
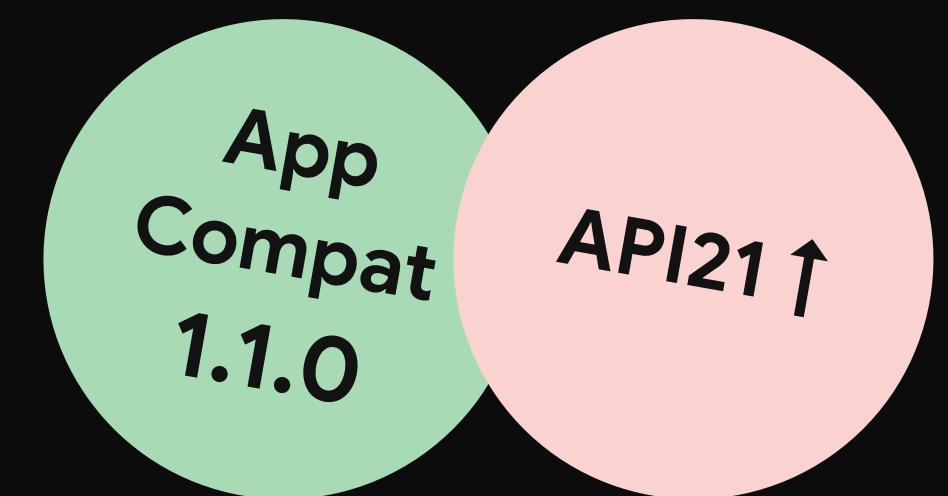
Recreate Activities on Night Mode Changes

AppCompat: Night Mode

- Android 4.0 (API14) 이상 지원
- AppCompat 1.1.0에서 동작 개선
 - Night Mode가 변경되면 AppCompatActivity가 **자동으로** 재시작된다.

NIGHT MODE Modes

- **MODE_NIGHT_YES**
항상 어둡게 표시
- **MODE_NIGHT_NO**
항상 밝게 표시
- **MODE_NIGHT_FOLLOW_SYSTEM**
시스템 설정에 따라 변경
- **MODE_NIGHT_AUTO_BATTERY**
절전모드에서 어둡게 표시



NAVER

NIGHT MODE

Set night mode

- 야간모드를 설정하는 방법이 2가지 있다.

```
// AppCompatDelegate.java
public abstract void setLocalNightMode(@NightMode int mode);
public static void setDefaultNightMode(@NightMode int mode) { ... }
```

- 참고: 값이 in-memory에 저장되므로, 별도로 값을 저장/복구 해야 한다.

NIGHT MODE

Get night mode

- Configuration 정보로 야간모드 적용 여부를 확인할 수 있다.

```
fun isDarkTheme(config: Configuration): Boolean {  
    return config.uiMode and // and는 &와 동일  
        Configuration.UI_MODE_NIGHT_MASK ==  
        Configuration.UI_MODE_NIGHT_YES  
}
```

NIGHT MODE

Manual Night Mode

- Activity를 재시작하지 않고, 수동으로 DarkTheme를 적용할 수도 있다.

```
<activity android:name=".MainActivity"
    android:configChanges="uiMode" />

class MainActivity : AppCompatActivity() {
    override fun onConfigurationChanged(newConfig: Configuration) {
        super.onConfigurationChanged(newConfig)
        val isDarkTheme: Boolean = isDarkTheme(newConfig)
        ... UI 업데이트 ...
    }
}
```

STEP 1

CodeLab

Use AppCompat 1.1.0 (1/1)

- start/build.gradle

```
//TODO: STEP1 - Use AppCompat 1.1.0  
implementation 'androidx.appcompat:appcompat:1.1.0-rc01'
```

STEP 1

Done! 

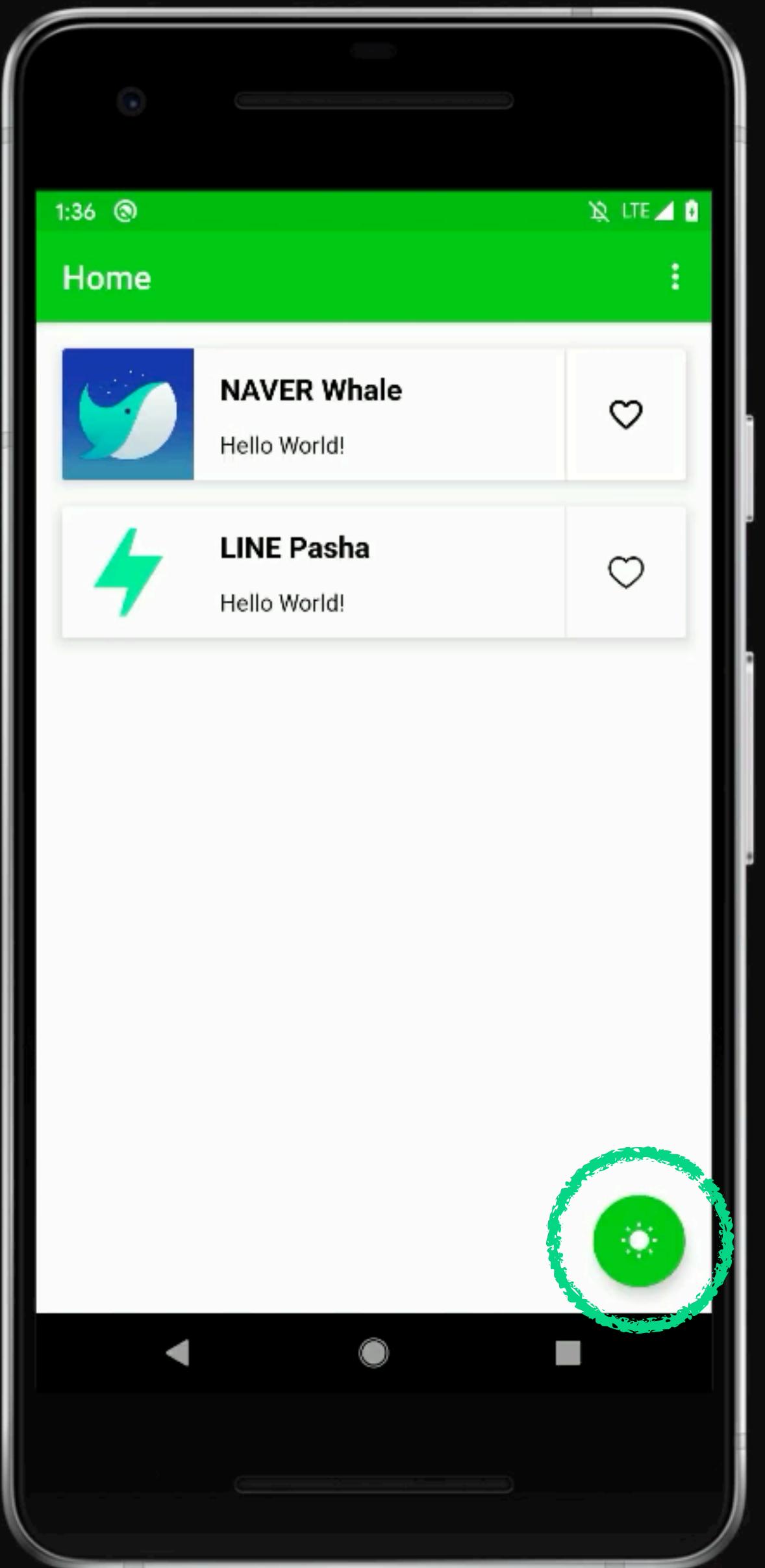
CodeLab

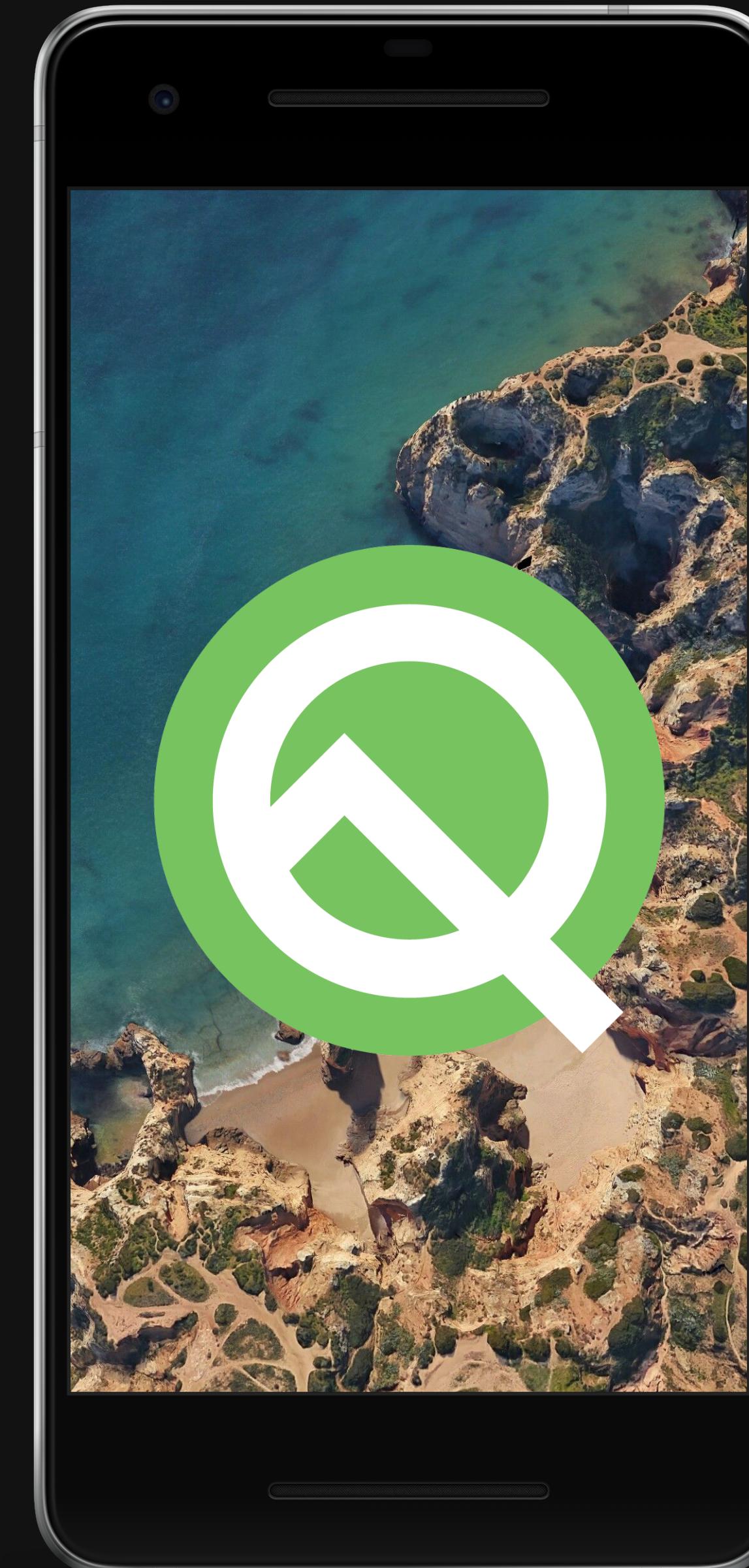
- DarkTheme.kt (*Pre-defined*)

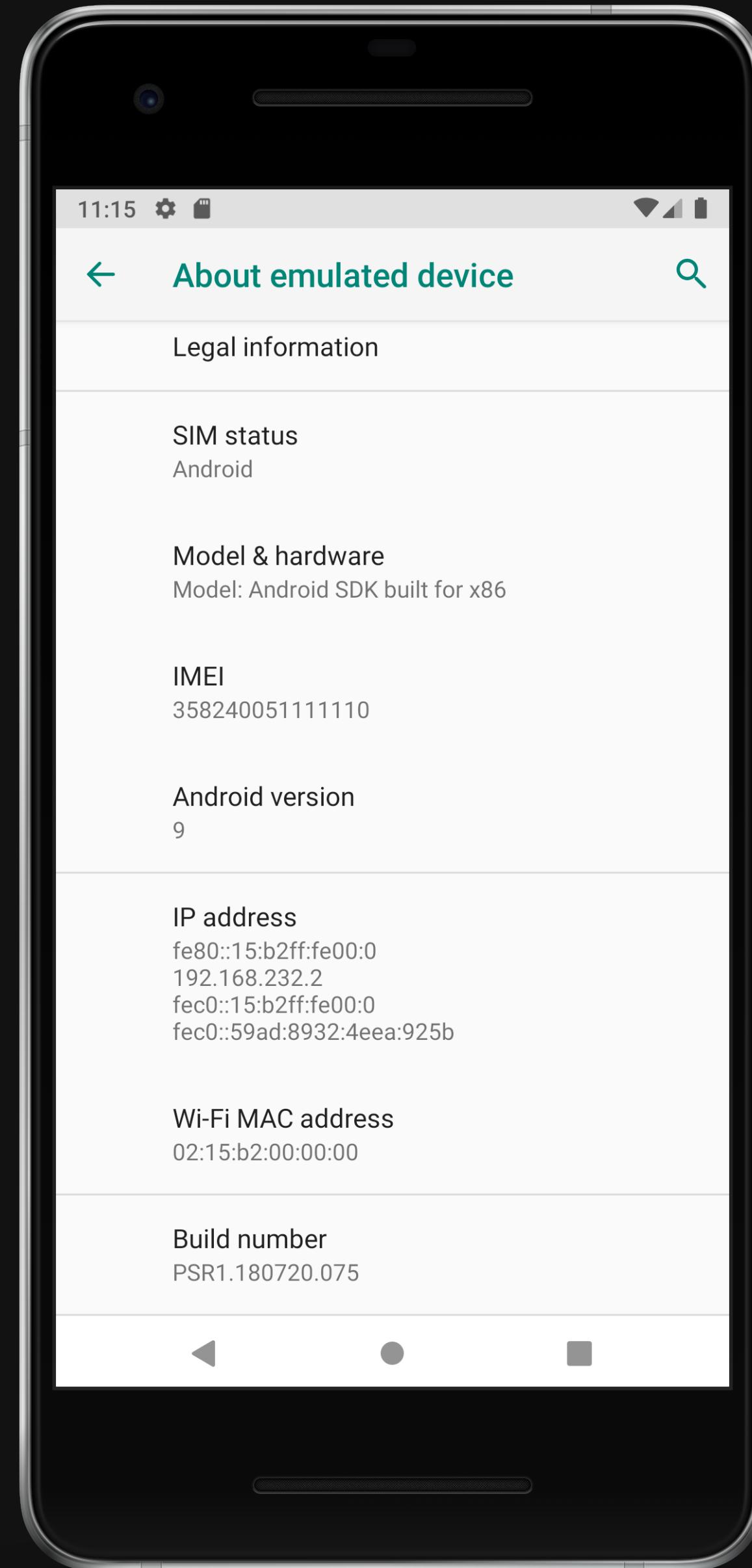
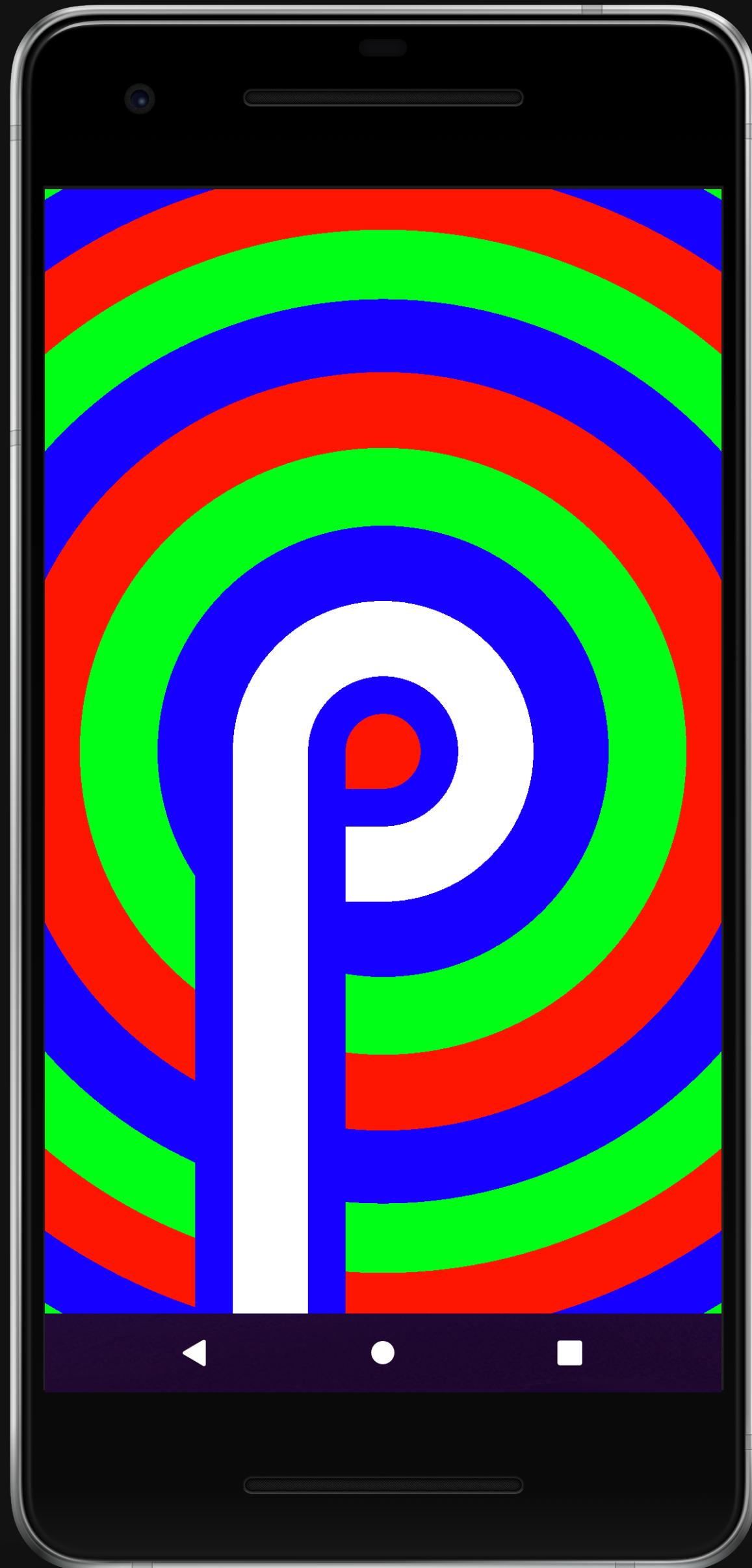
```
fun apply(enabled: Boolean = false) {  
    val nightMode = if (enabled) {  
        AppCompatDelegate.MODE_NIGHT_YES  
    } else {  
        AppCompatDelegate.MODE_NIGHT_NO  
    }  
    AppCompatDelegate.setDefaultNightMode(nightMode)  
}
```

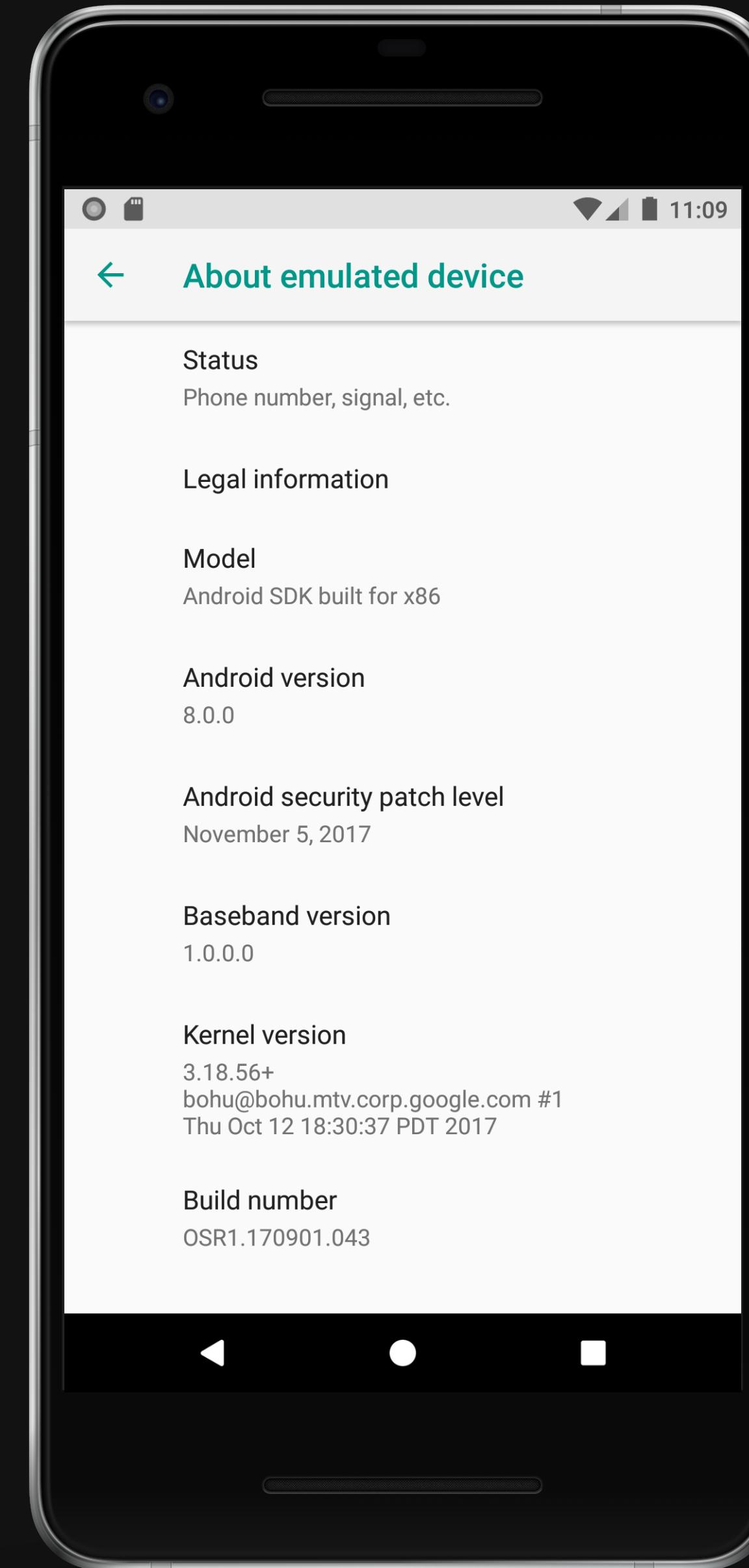
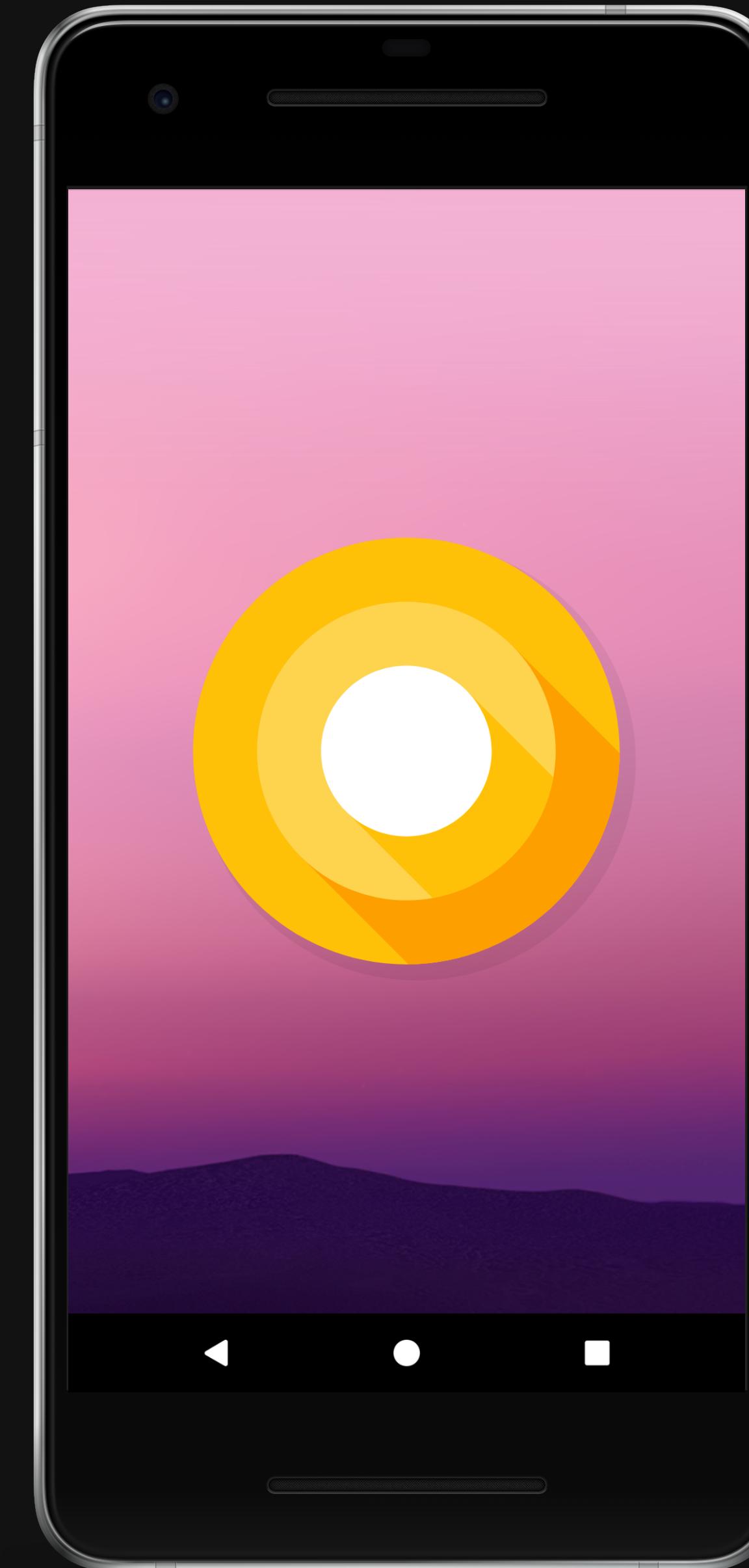
Step 2.

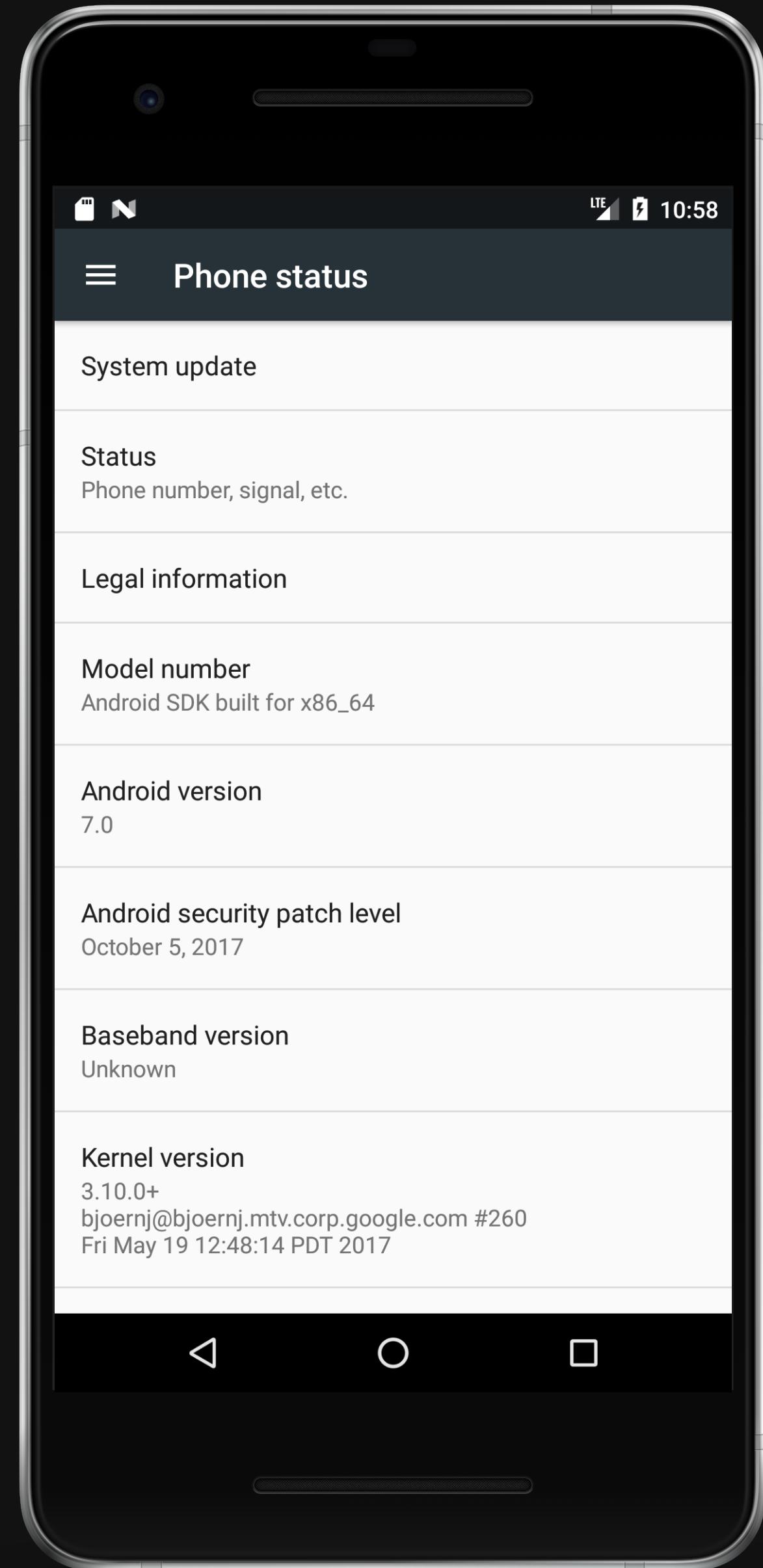
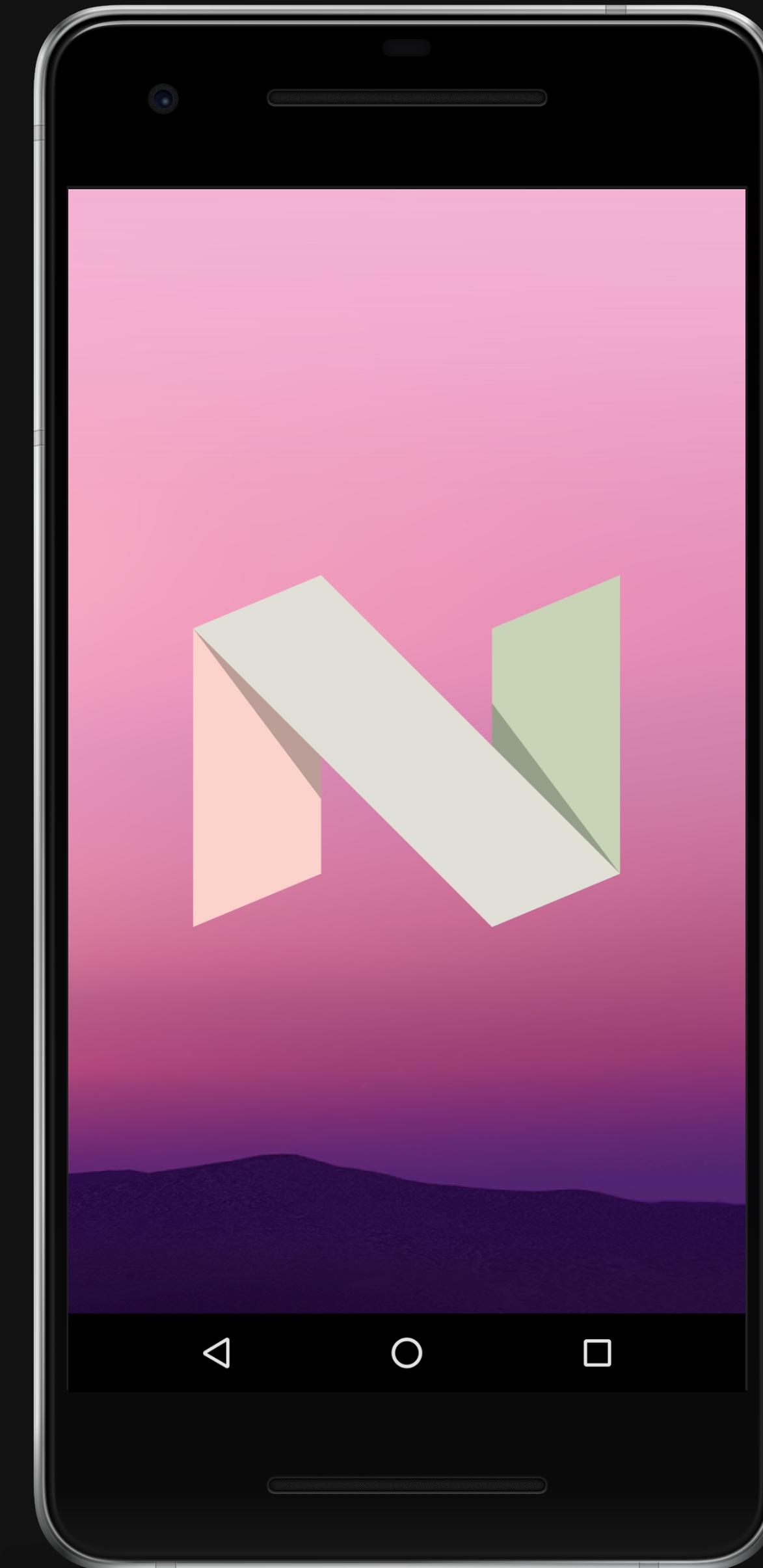
- 야간모드에서 색상 변경이 없는 문제 🤔
- 야간모드 리소스를 추가해야 한다.

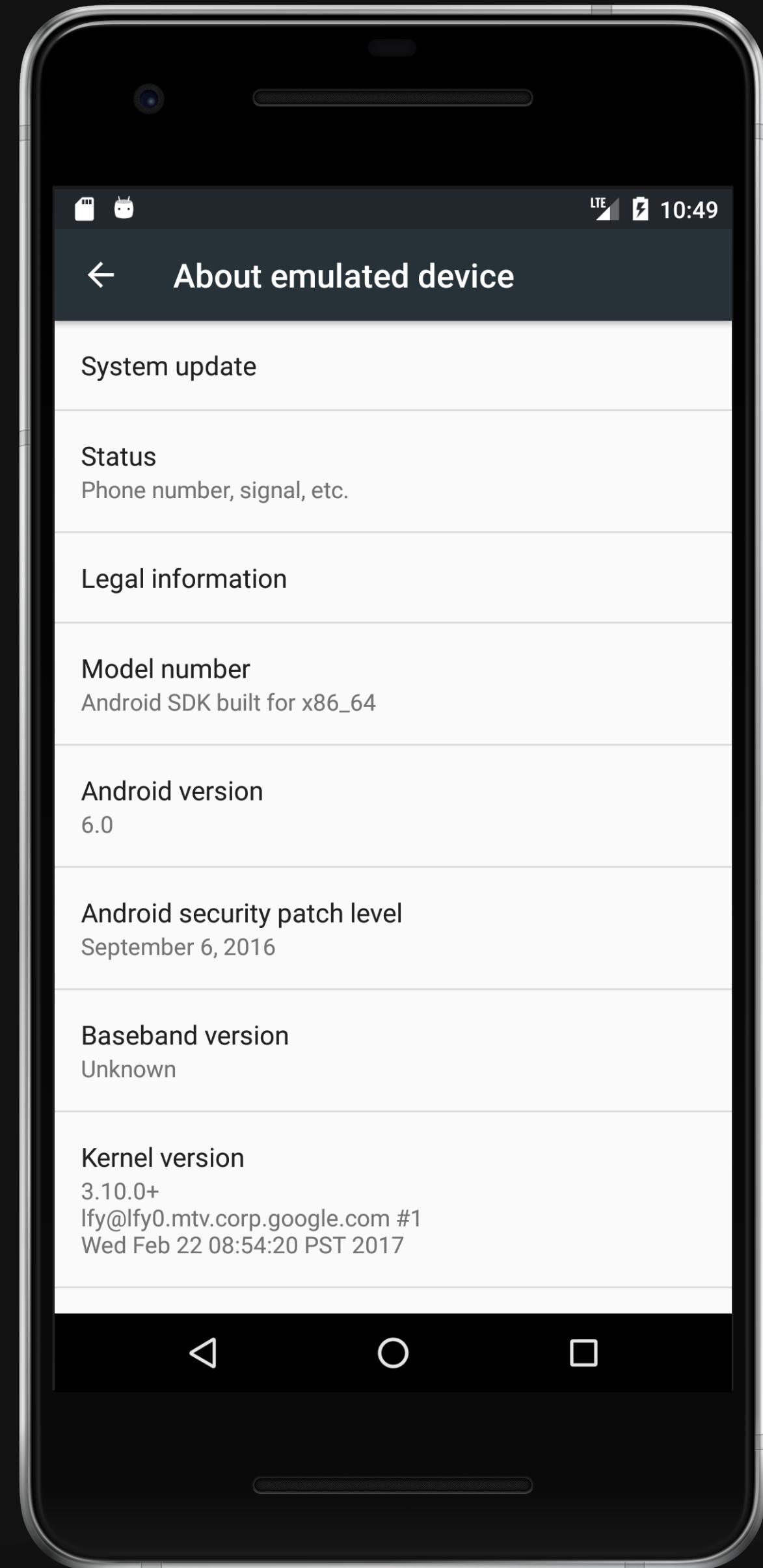
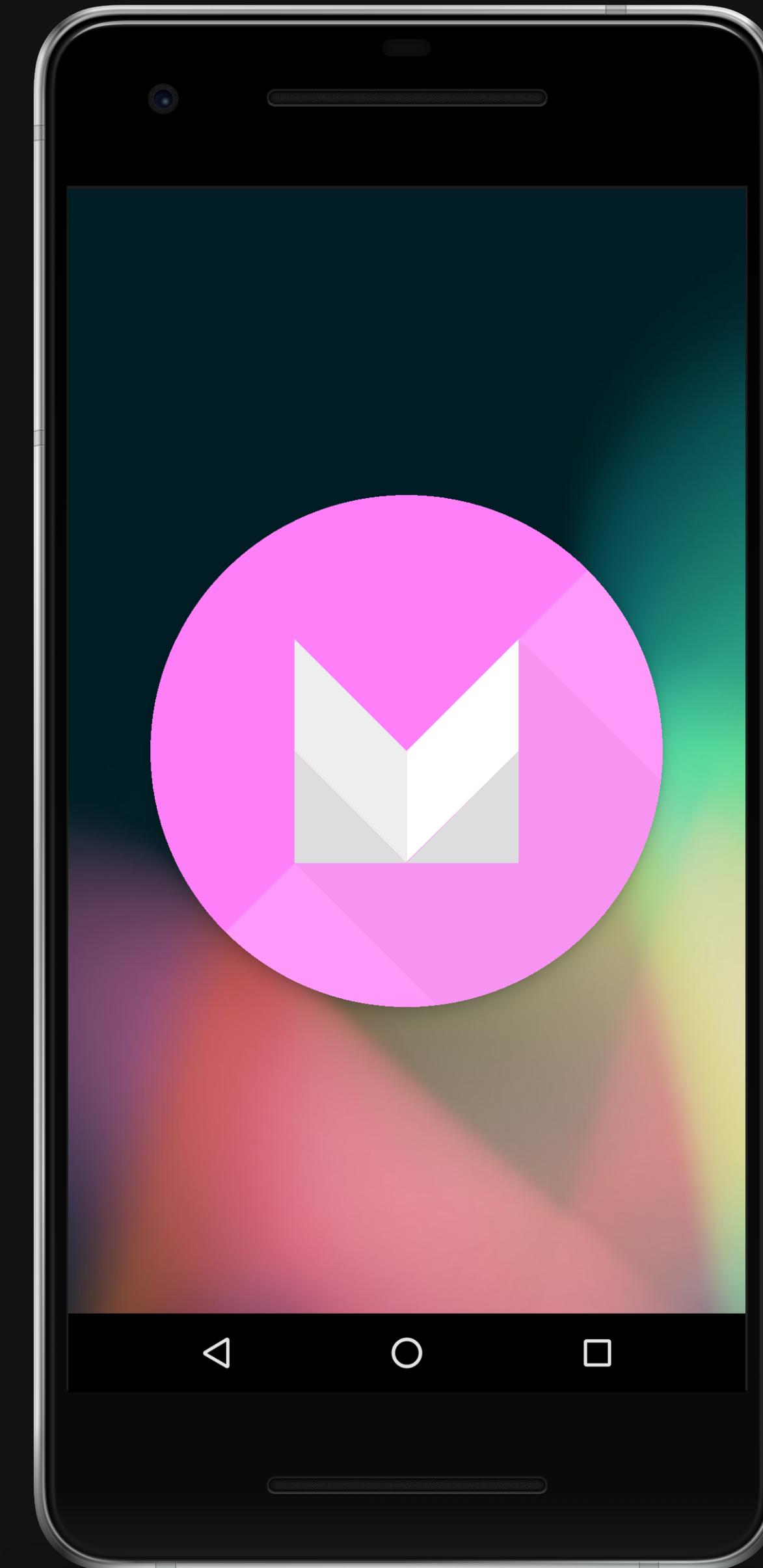


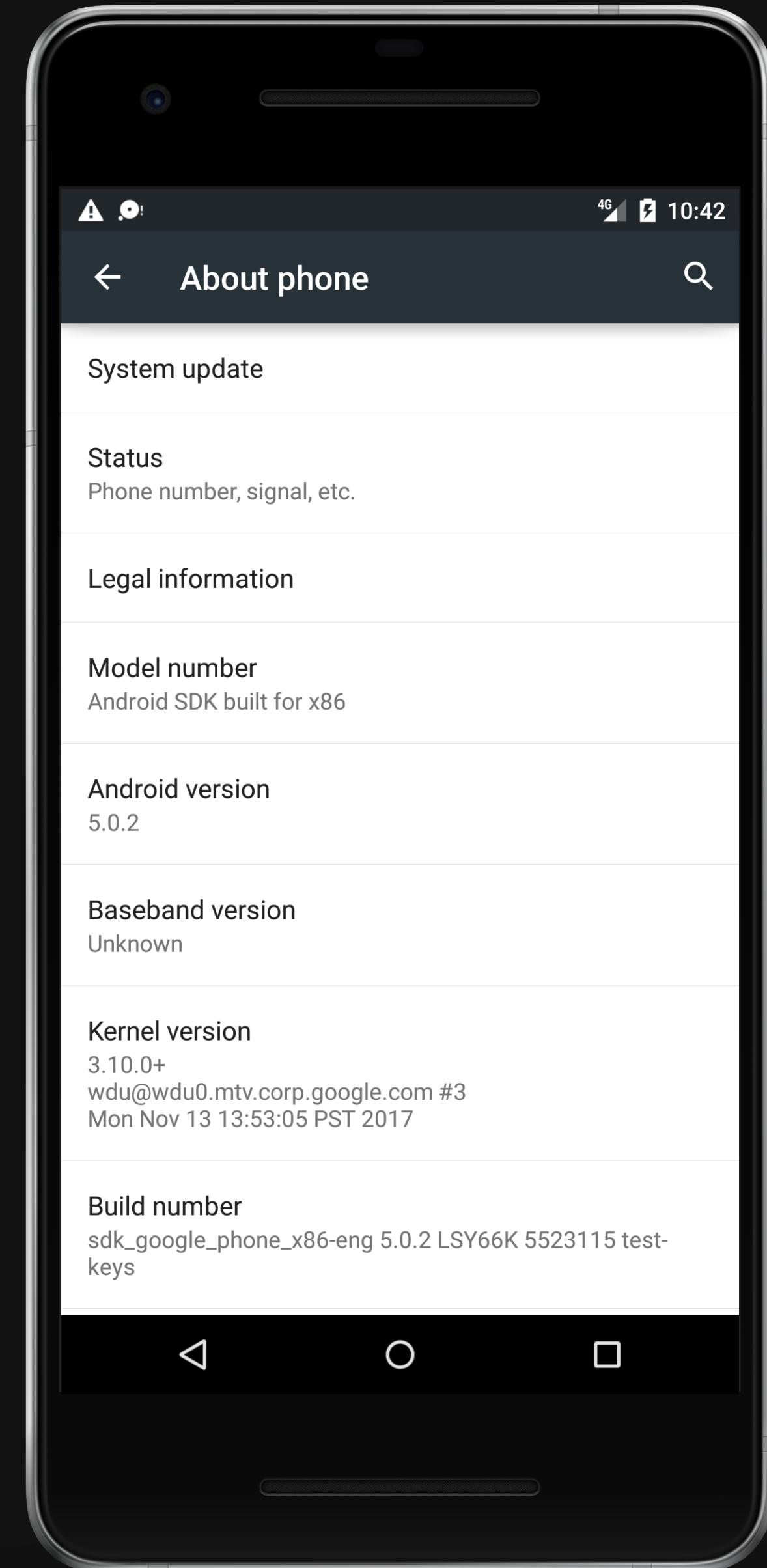














Dark Theme

- L OS 이전에는 기본 테마가 Dark Theme였다.

Dark Theme:

```
<style name="Theme.DeviceDefault" />
```

Light Theme:

```
<style name="Theme.DeviceDefault.Light" />
```

Dark Theme

- AppCompat에서도 Dark Theme를 제공한다.

Dark Theme:

```
<style name="Theme.AppCompat"/>
```

Light Theme:

```
<style name="Theme.AppCompat.Light"/>
```

'-night' ●

Resource Qualifier

'-night' Resource Qualifier

- Night Mode에서는 **-night** resource qualifier가 활성화된다.
- '-night' 폴더에 리소스를 추가해야 한다.
 - 예) values-**night**, raw-**night** 등등

```
<!-- values/colors.xml -->
<color name="colorCustom">#000000</color>

<!-- values-night/colors.xml -->
<color name="colorCustom">#ffffff</color>
```

Link: <https://developer.android.com/guide/topics/resources/providing-resources#AlternativeResources>

'-night' Resource Qualifier

- ⚠️ Resource Qualifier 우선순위를 주의해야 한다.

res/drawable-hdpi/ic_example.png

res/drawable-night-hdpi/ic_example.png

res/drawable-land/ic_example.png

res/drawable-land-night/ic_example.png

Declare '-night' resources (1/3)

- res/values-**night**/styles.xml

```
<!-- TODO: STEP2 - Declare '-night' resources -->
<style name="AppTheme" parent="Theme.AppCompat.NoActionBar">
    ...
</style>

<style name="AppTheme.Splash"> ... </style>

<style name="AppTheme.Whale" parent="Theme.AppCompat" />
```

Declare '-night' resources (2/3)

- res/values-**night**/colors.xml

```
<!-- TODO: STEP2 - Declare '-night' resources -->
<color name="colorPrimary">#121212</color>
<color name="colorPrimaryDark">#000000</color>
<color name="colorAccent">#00c300</color>
<color name="colorBackground">#202020</color>
...
<color name="colorOnError">#202020</color>
```

STEP 2

CodeLab

Declare '-night' resources (3/3)

- Splash 로고 이름 변경

Before:

res/drawable-night/ic_logo_naver_night.png

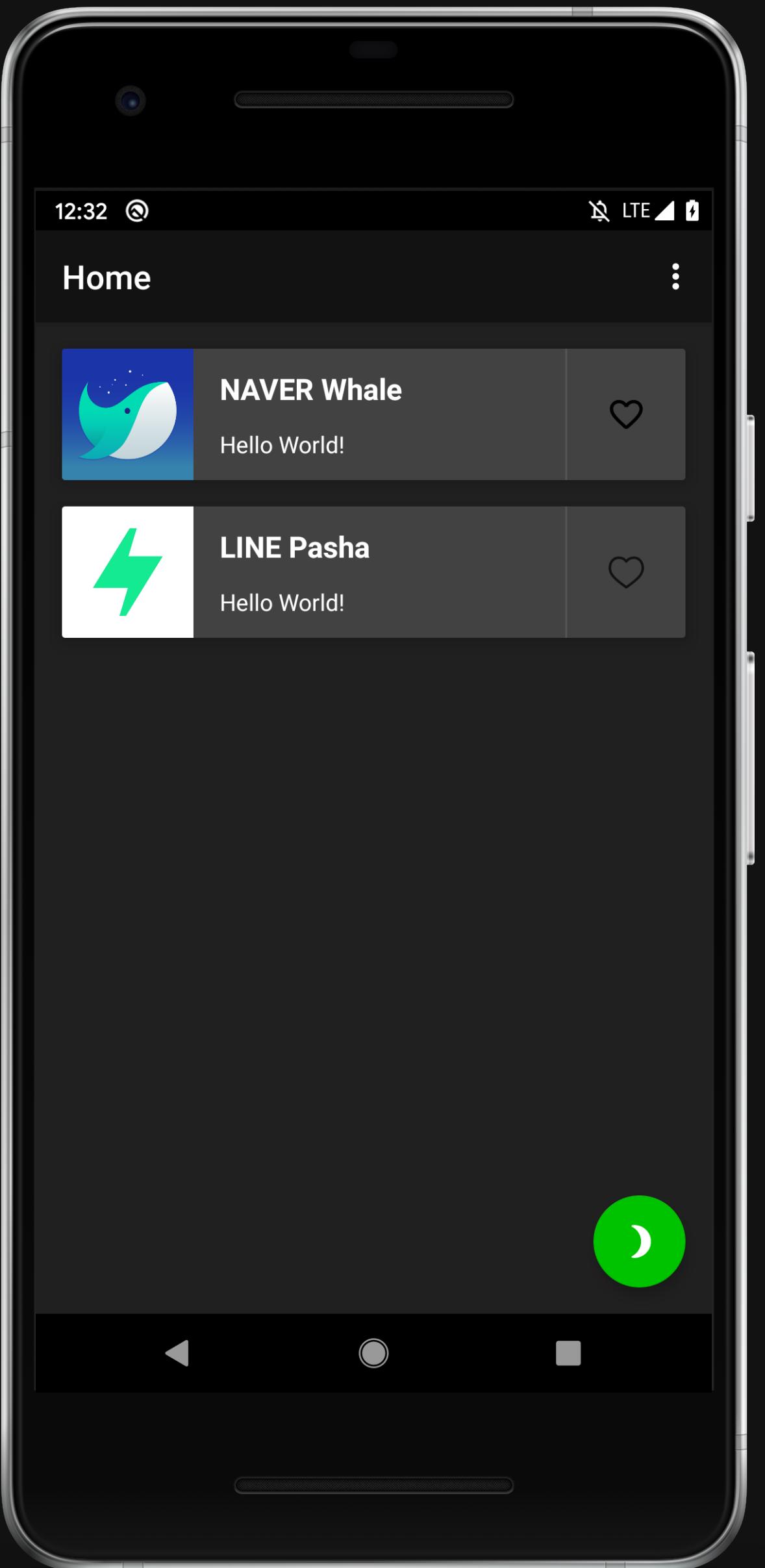
After:

res/drawable-night/ic_logo_naver.png

STEP 2

Done! 🎉

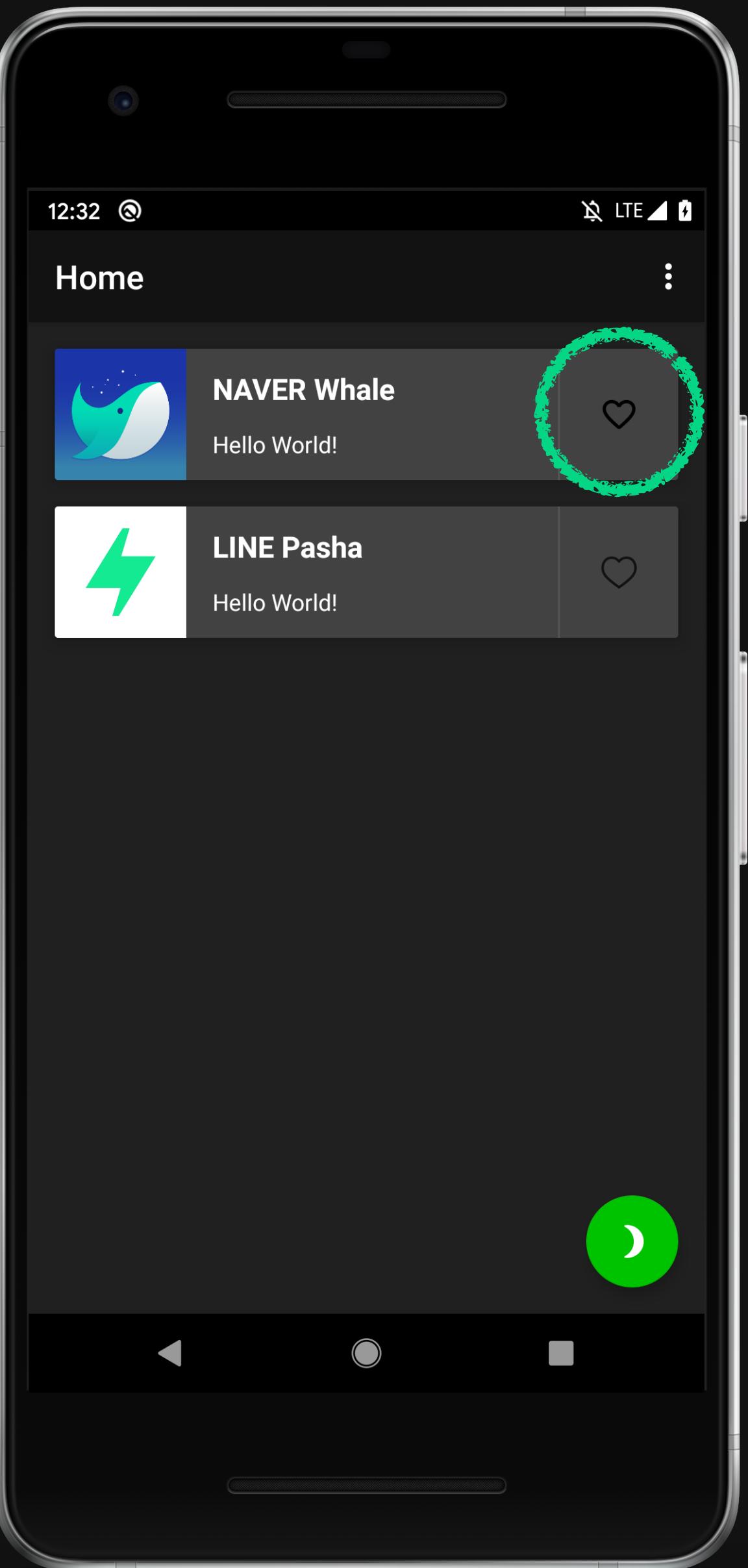
- 야간모드에서 색상이 적절하게 보인다.



CodeLab

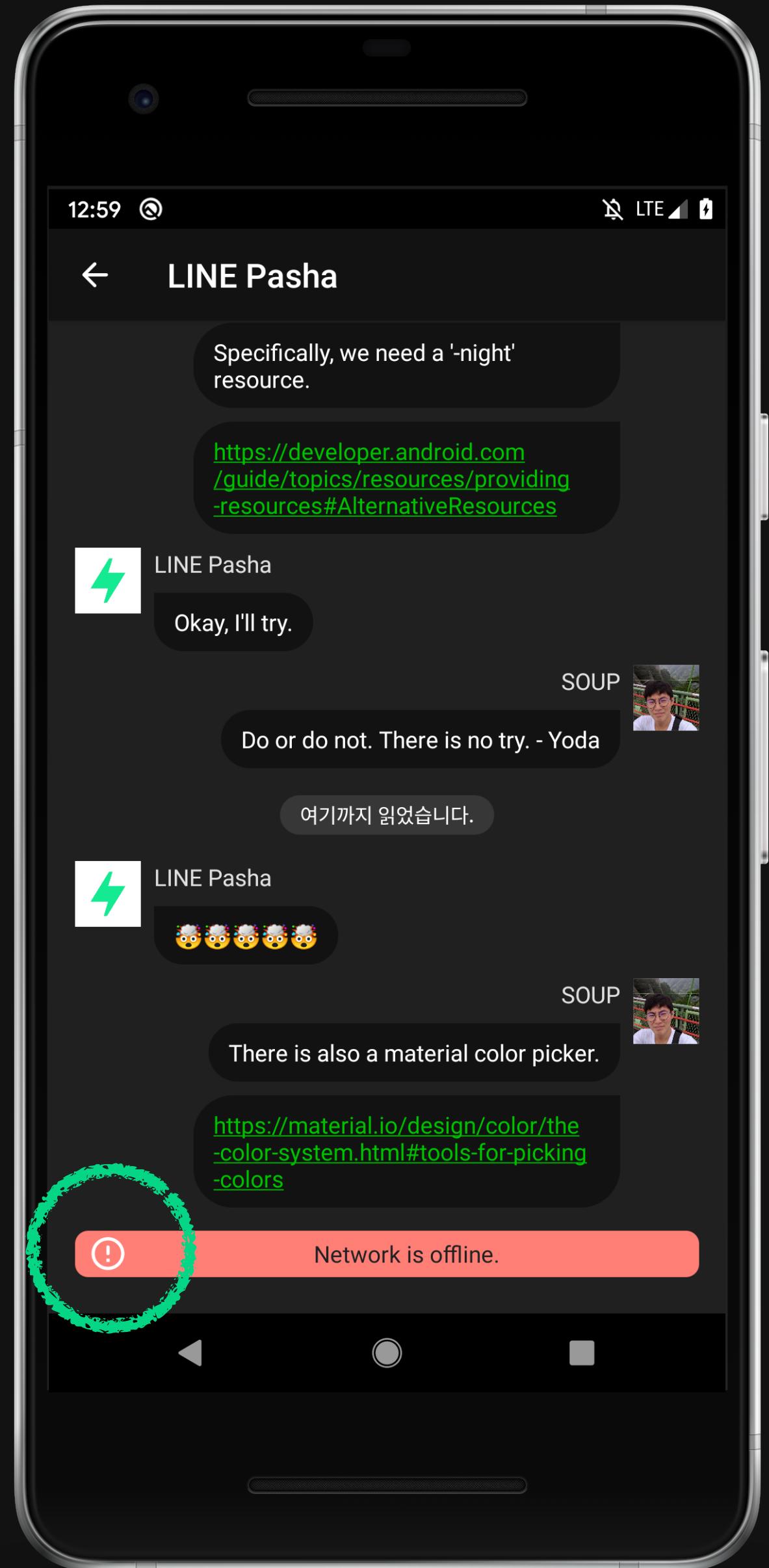
Step 3.

- 야간모드 아이콘의 시인성이 떨어지는 문제 🤔



Step 3.

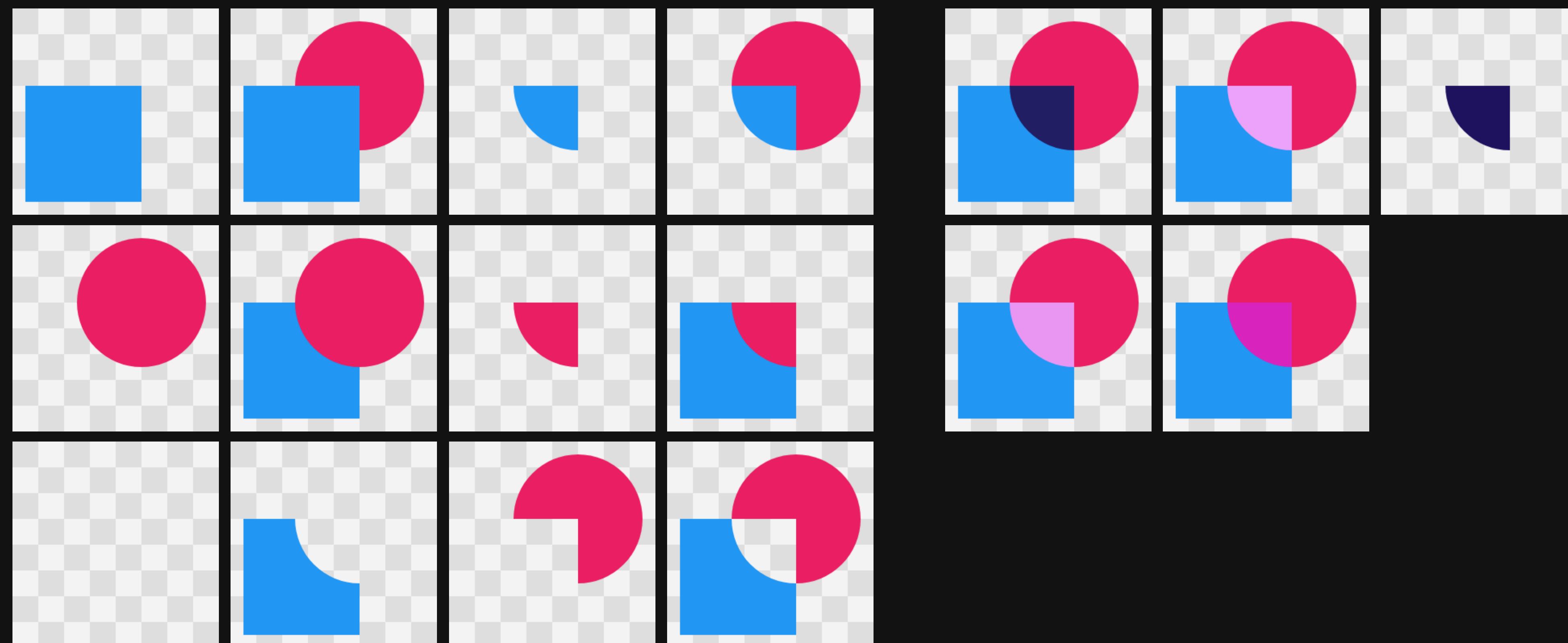
- 야간모드 아이콘의 시인성이 떨어지는 문제 🤔
- Drawable 색상이 고정되어 있다.



Tint 

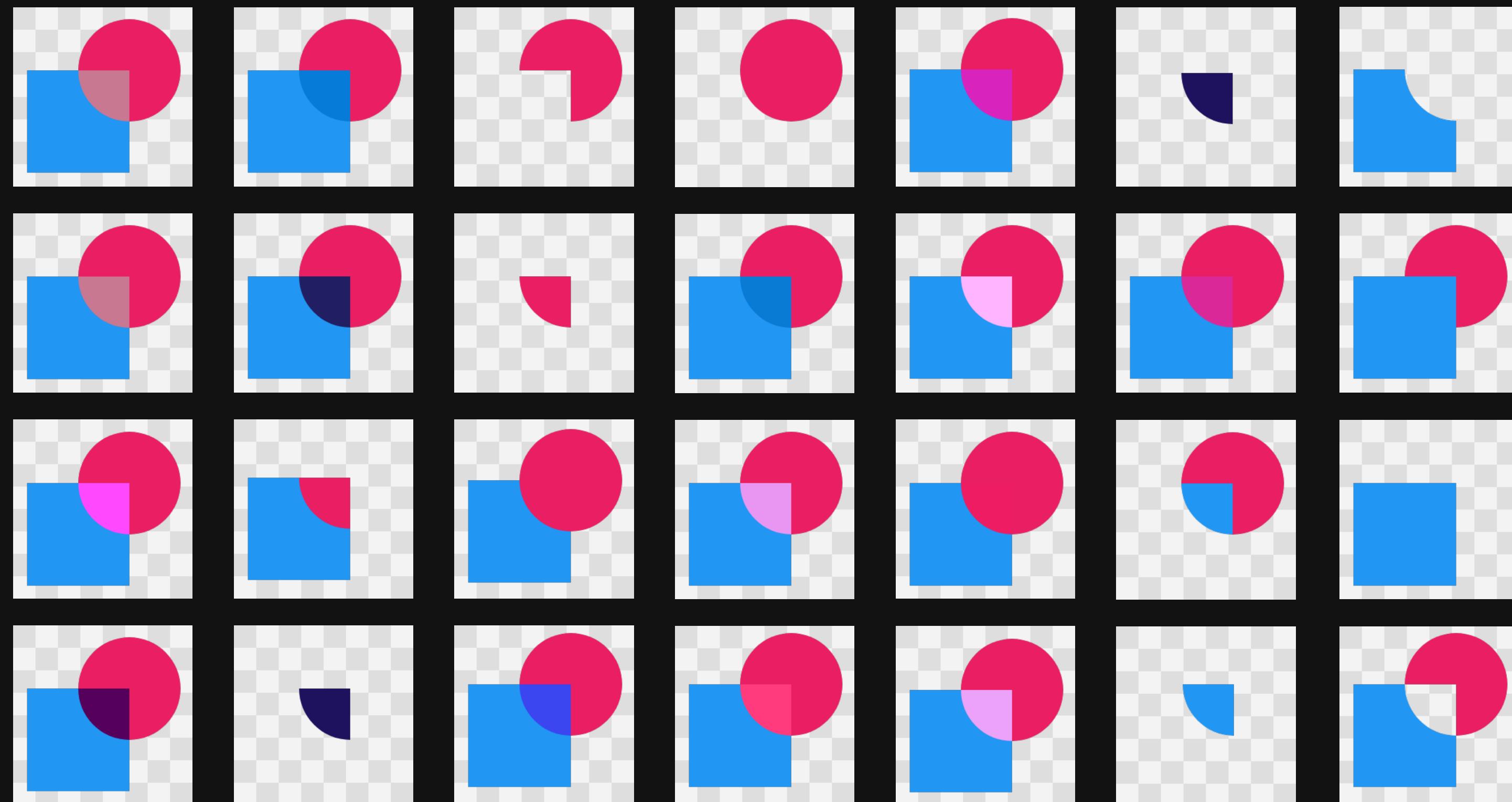
View, VectorDrawable, etc...

PorterDuff.Mode



Link: <https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html>

BlendMode Q



아마도

ForceDark??

Link: <https://developer.android.com/reference/android/graphics/BlendMode>

Tint

- 앞서 본 Mode를 이용하여 **Drawable**에 색조를 입힐 수 있다. (기본: **SRC_IN**)
- **ColorStateList**로 **Stateful Tint**도 가능하다.
- API 21 부터 추가되었지만, Compat API가 제공된다.
 - **DrawableCompat**
 - **VectorDrawableCompat**

Tinting Drawables

- Drawable XML에서 `android:tint` 속성을 사용할 수 있다.

```
<!-- res/drawable/shape.xml -->
<shape android:tint="@color/btn_background">
    <solid android:color="@color/white" />
</shape>
```

```
<!-- res/drawable/vector.xml -->
<vector android:tint="@color/btn_background">
    <path android:fillColor="@color/white" />
</vector>
```

Tinting Views

- View도 Tint API가 제공된다.

```
<ImageView android:tint="@color/btn_icon" />
<View android:backgroundTint="@color/btn_background" />
```

```
imageView.setImageTintList(ColorStateList)
view.setBackgroundTintList(ColorStateList)
```

```
ImageViewCompat.setImageTintList(view, ColorStateList)
ViewCompat.setBackgroundTintList(view, ColorStateList)
```

STEP 3

CodeLab

Tinting icons (1/2)

- res/drawable/ic_round_favorite_off.xml

```
<!-- TODO: STEP3 - Tinting icons -->
<vector android:width="24dp"
        android:height="24dp"
        android:tint="@color/colorIcon">
```

STEP 3

CodeLab

Tinting icons (2/2)

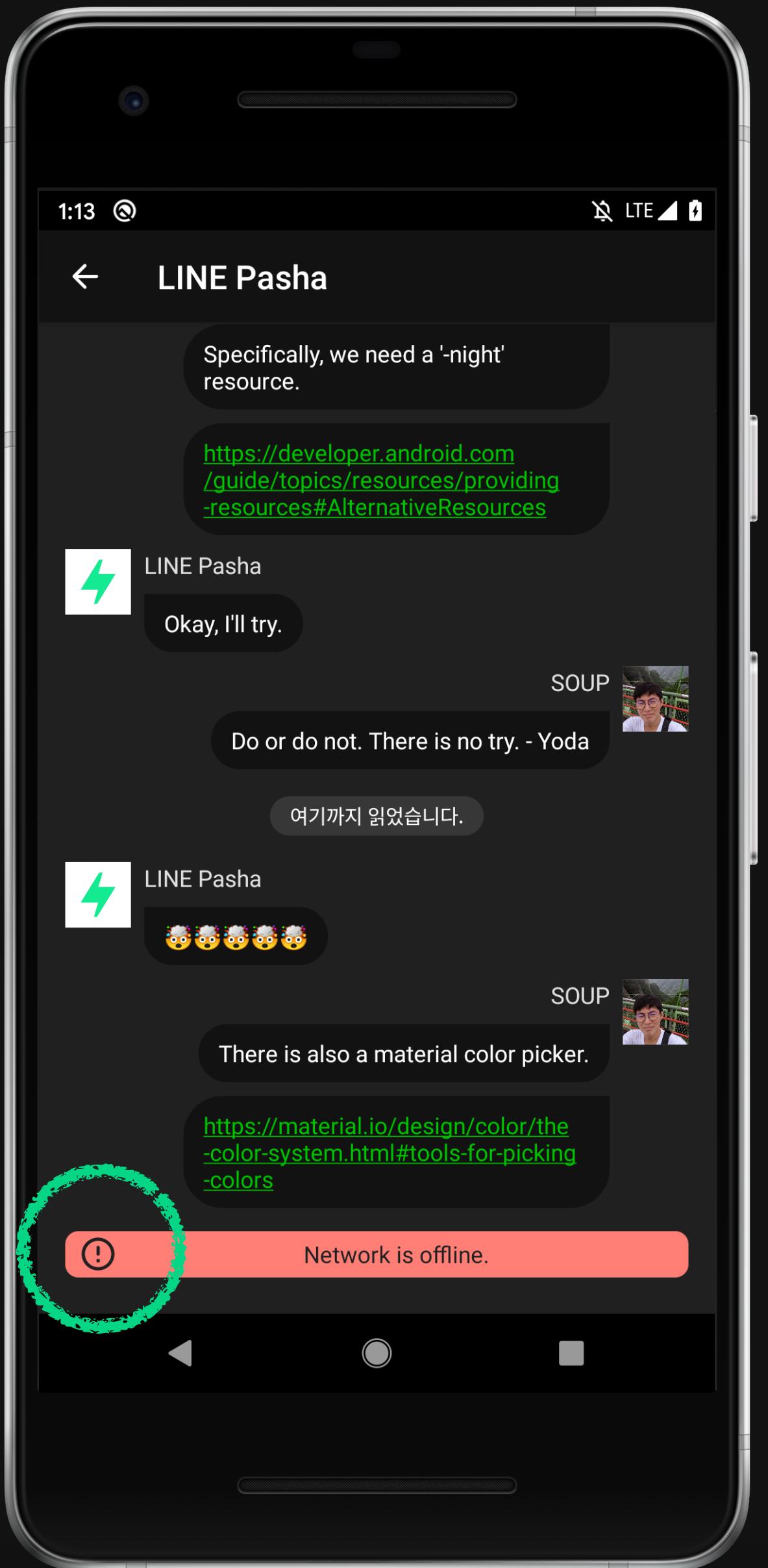
- res/drawable/ic_round_error_outline.xml

```
<!-- TODO: STEP3 - Tinting icons -->
<vector android:width="24dp"
        android:height="24dp"
        android:tint="@color/colorOnError">
```

STEP 3

Done! 🎉

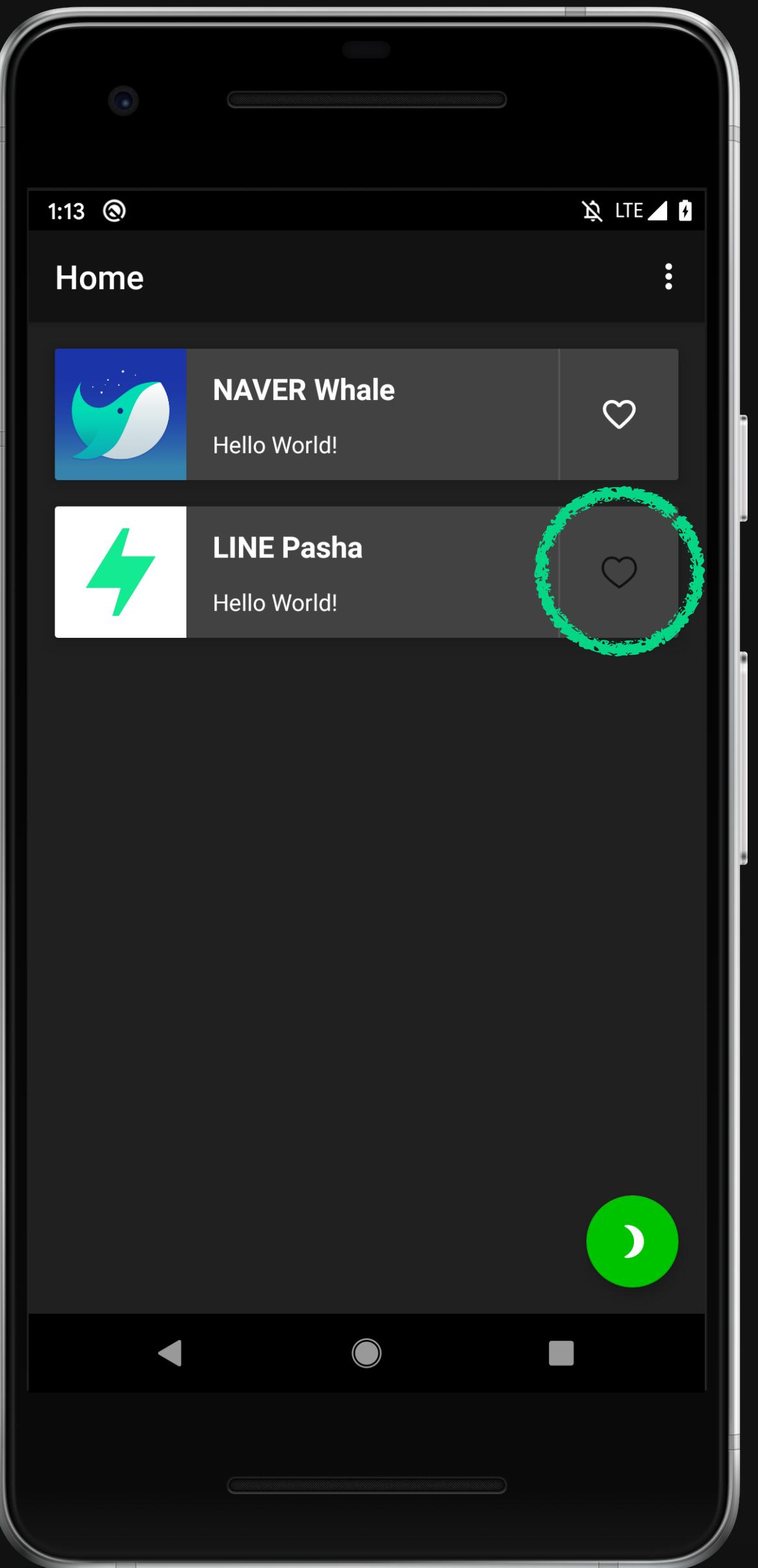
- Drawable 색상이 적절하게 보인다.



CodeLab

Step 4.

- Lottie가 야간모드로 바뀌지 않는 문제 🤔
- Lottie Bug! 🐛





Lottie Issue

- Animation 설정할 때, 동일 리소스 ID는 업데이트되지 않는 이슈가 있다.

```
// LottieComposition.java
private static String rawResCacheKey(@RawRes int resId) {
    return "rawRes_" + resId;
}
```

Link: <https://github.com/airbnb/lottie-android/issues/1305>

STEP 4

CodeLab

Set 'Lottie' dynamically (1/2)

- res/layout/home_card_pasha.xml

```
<!--TODO: STEP4 - Set 'Lottie' animation dynamically -->
<import type="soup.codelab.darktheme.R" />

<com.airbnb.lottie.LottieAnimationView
    android:id="@+id/favoriteButton"
    ...
    app:lottie_rawResDay="@{R.raw.ic_lottie_favorite_day}"
    app:lottie_rawResNight="@{R.raw.ic_lottie_favorite_night}" />
```

Set 'Lottie' dynamically (2/2)

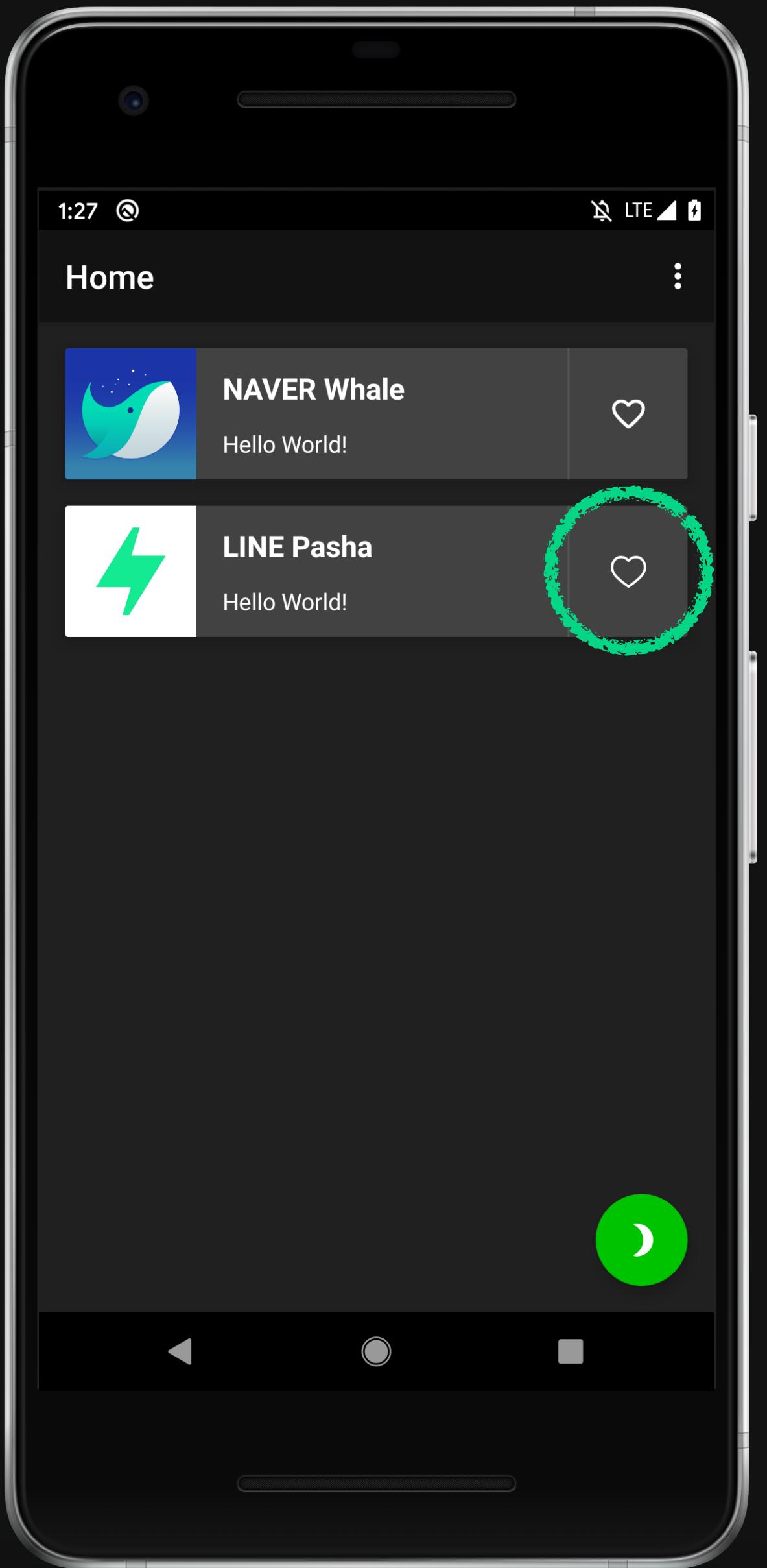
- LottieBindingAdapter.kt

```
@BindingAdapter("lottie_rawResDay", "lottie_rawResNight")
fun setAnimationAsync(view: LottieAnimationView,
                      @RawRes dayRawRes: Int,
                      @RawRes nightRawRes: Int) {
    if (DarkTheme.isEnabled(view.context)) {
        view.setAnimation(nightRawRes)
    } else {
        view.setAnimation(dayRawRes)
    }
}
```

STEP 4

Done! 🎉

- Lottie가 적절히 변경된다.



CodeLab

Step 5.

- 관리하는 Color가 많아지는 문제 😱
- Color Attribute를 이용하여 그룹 단위로 관리할 수 있다.

```
<!-- res/values/colors.xml -->
<color name="colorPrimary" >#FF00c300</color>
<color name="colorPrimaryDark" >#FF00b300</color>
<color name="colorAccent" >#FF00c300</color>
<color name="colorBackground" >#FFFFFF</color>
<color name="colorText" >#FF000000</color>
<color name="colorIcon" >#FF000000</color>
<color name="colorSurface" >#FEEEEE</color>
<color name="colorTagBackground" >#1A000000</color>
<color name="colorTagText" >#EE121212</color>
<color name="colorDivider" >#1F000000</color>
<color name="colorBorder" >#33000000</color>
<color name="colorName" >#dd000000</color>
<color name="colorError" >#FFee4b4b</color>
<color name="colorOnError" >#FFffffff</color>
```

ColorStateList

with '`android:alpha`' attribute

ColorStateList

- android:alpha attribute가 API 21 부터 추가되었습니다.

```
<selector>
    <item android:alpha="0.1"
          android:color="#ffffffff" />
</selector>
```

ColorStateList

- android:alpha attribute이 포함된 ColorStateList 리소스를 읽을 때,
API 23 이하 버전에서는 AppCompat API를 사용해야 합니다.

`AppCompatResources.getColorStateList(context, R.color.divider)`

STEP 5

CodeLab

Use color attribute (1/5)

- res/values/attrs.xml

```
<!-- TODO: STEP5 - Use color attribute -->
<attr name="colorOnBackground" format="color" />
```

STEP 5

CodeLab

Use color attribute (2/5)

- res/values/styles.xml

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
    ...
    <!-- TODO: STEP5 - Use color attribute -->
    <item name="colorOnBackground">@android:color/black</item>
</style>
```

STEP 5

CodeLab

Use color attribute (3/5)

- res/values-night/styles.xml

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
    ...
    <!-- TODO: STEP5 - Use color attribute -->
    <item name="colorOnBackground">@android:color/white</item>
</style>
```

STEP 5

CodeLab

Use color attribute (4/5)

- res/color/divider.xml

```
<selector>
    <!-- TODO: STEP5 - Use color attribute -->
    <item android:alpha="0.1"
          android:color="?colorOnBackground" />
</selector>
```

STEP 5

CodeLab

Use color attribute (5/5)

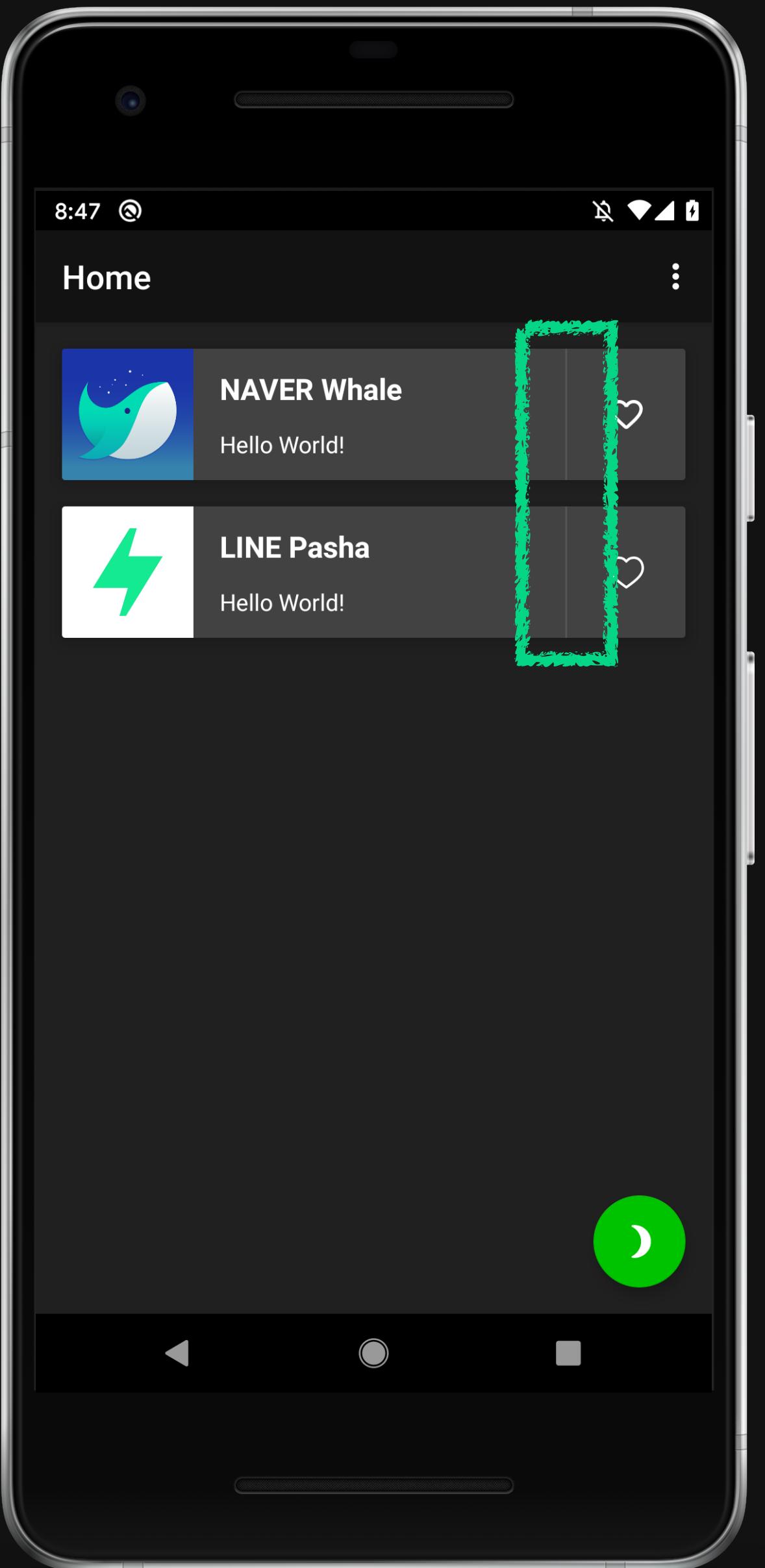
- res/layout/home_card_whale.xml
- res/layout/home_card_pasha.xml

```
<!-- TODO: STEP5 - Use color attribute -->
<ImageView
    android:id="@+id/divider"
    ...
    android:background="#ffffff"
    app:backgroundTint="@color/divider" />
```

STEP 5

Done! 🎉

- 동일한 색상으로 적용된다.



CodeLab

Color Attributes

in Material Design Components

Pre-Defined Color Attributes in MDC

Primary	Primary Variant	Secondary	Secondary Variant
1 #6200EE	2 #3700B3	3 #03DAC6	4 #018786

Primary	Primary Variant	Secondary
1 200 #BB86FC	2 700 #3700B3	3 200 #03DAC6

Background	Surface	Error
5 #FFFFFF	6 #FFFFFF	7 #B00020

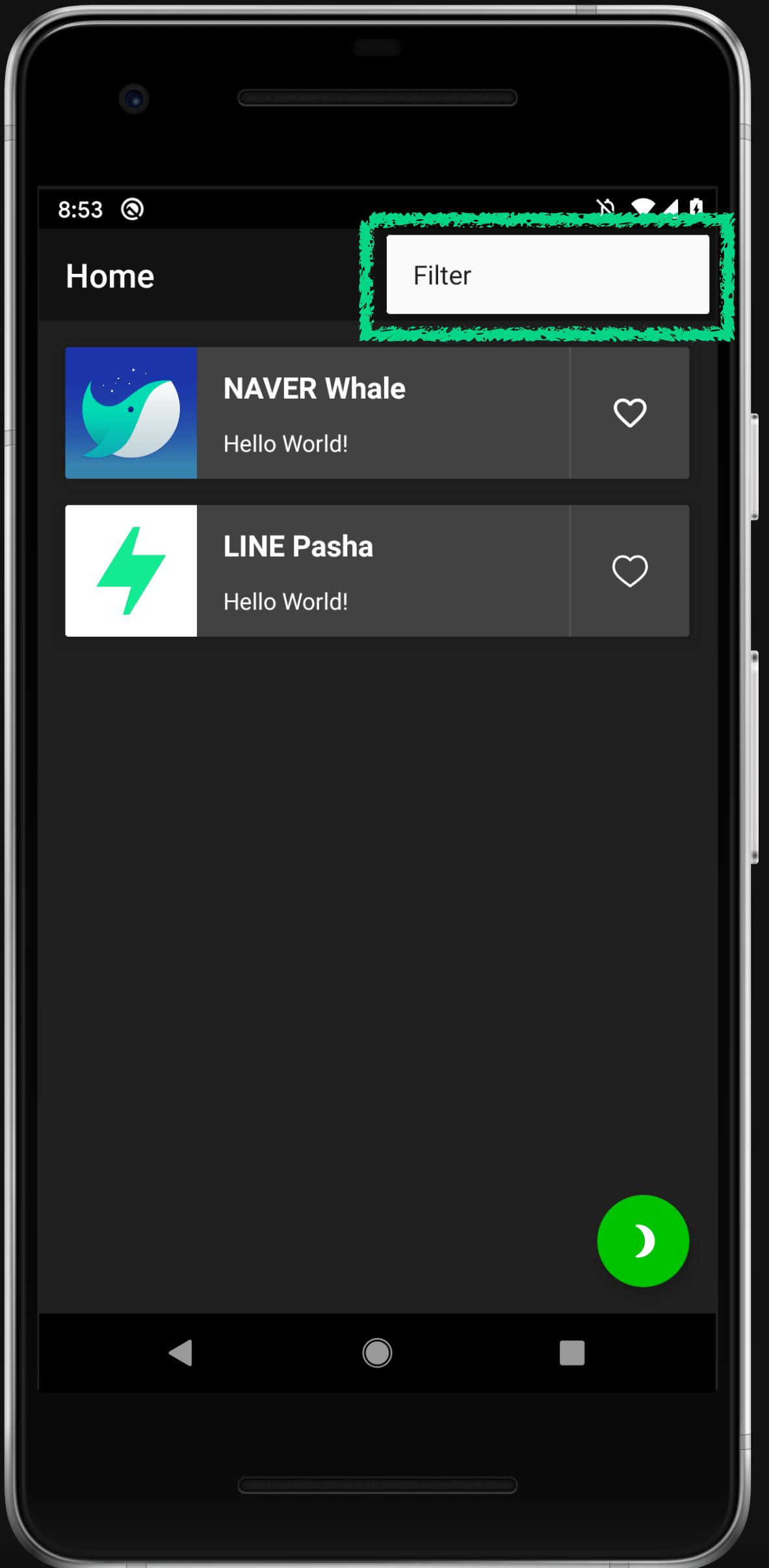
Background	Surface	Error
4 #121212	5 #121212	6 #CF6679

On Primary	On Secondary
8 #FFFFFF	9 #000000
On Background	On Surface

On Primary	On Secondary
7 #000000	8 #000000
On Background	On Surface

Step 6.

- AppTheme를 사용하지 않는 부분은 야간모드에서 밝게 보이는 문제 🤔
- android:theme 속성을 잘 확인해보자.



DayNight Themes



AppCompat, MDC

DayNight Theme

- AppCompat에서 DayNight theme를 제공한다.

values/theme.xml

```
<style name="Theme.AppCompat.DayNight"  
      parent="Theme.AppCompat.Light" />
```

values-night/theme.xml

```
<style name="Theme.AppCompat.DayNight"  
      parent="Theme.AppCompat" />
```

DayNight Theme

- AppCompat에선 ThemeOverlay도 DayNight theme를 제공한다.

values/theme.xml

```
<style name="ThemeOverlay.AppCompat.DayNight"  
      parent="ThemeOverlay.AppCompat.Light"/>
```

values-night/theme.xml

```
<style name="ThemeOverlay.AppCompat.DayNight"  
      parent="ThemeOverlay.AppCompat.Dark"/>
```

DayNight Theme

- MDC에도 DayNight theme를 제공한다.

values/theme.xml

```
<style name="Theme.MaterialComponents.DayNight"  
      parent="Theme.MaterialComponents.Light" />
```

values-night/theme.xml

```
<style name="Theme.MaterialComponents.DayNight"  
      parent="Theme.MaterialComponents" />
```

DayNight Theme

- 하지만 MDC에 ThemeOverlay는 DayNight theme를 지원하지 않는다.

values/theme.xml

```
<style name="ThemeOverlay.MaterialComponents.DayNight"  
      parent="ThemeOverlay.MaterialComponents.Light"/>
```

values-night/theme.xml

```
<style name="ThemeOverlay.MaterialComponents.DayNight"  
      parent="ThemeOverlay.MaterialComponents"/>
```

STEP 6

CodeLab

Use DayNight theme (1/1)

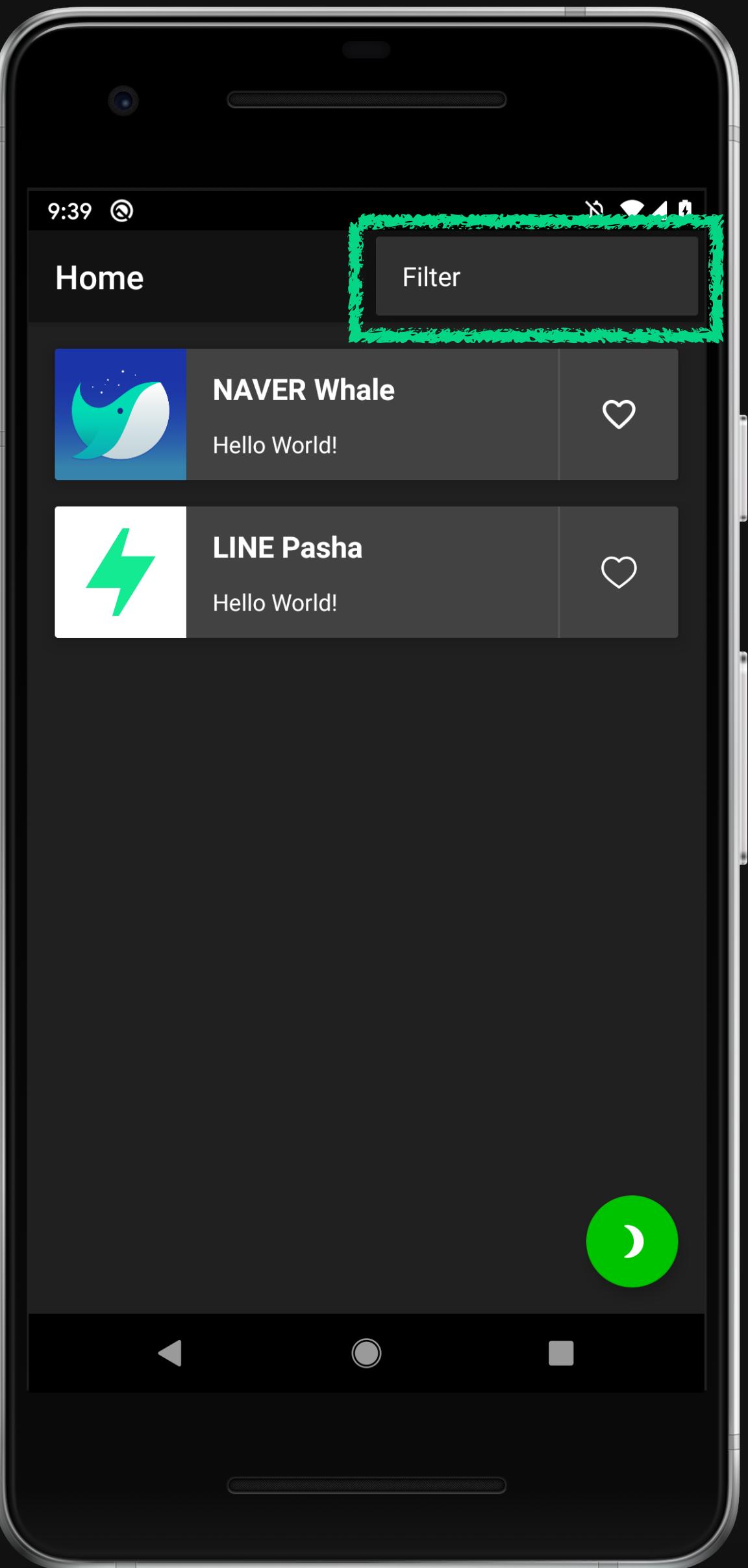
- res/layout/home_activity.xml

```
<!--TODO: STEP6 - Use DayNight theme -->
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    ...
    app:popupTheme="@style/ThemeOverlay.AppCompat.DayNight" />
```

STEP 6

Done! 🎉

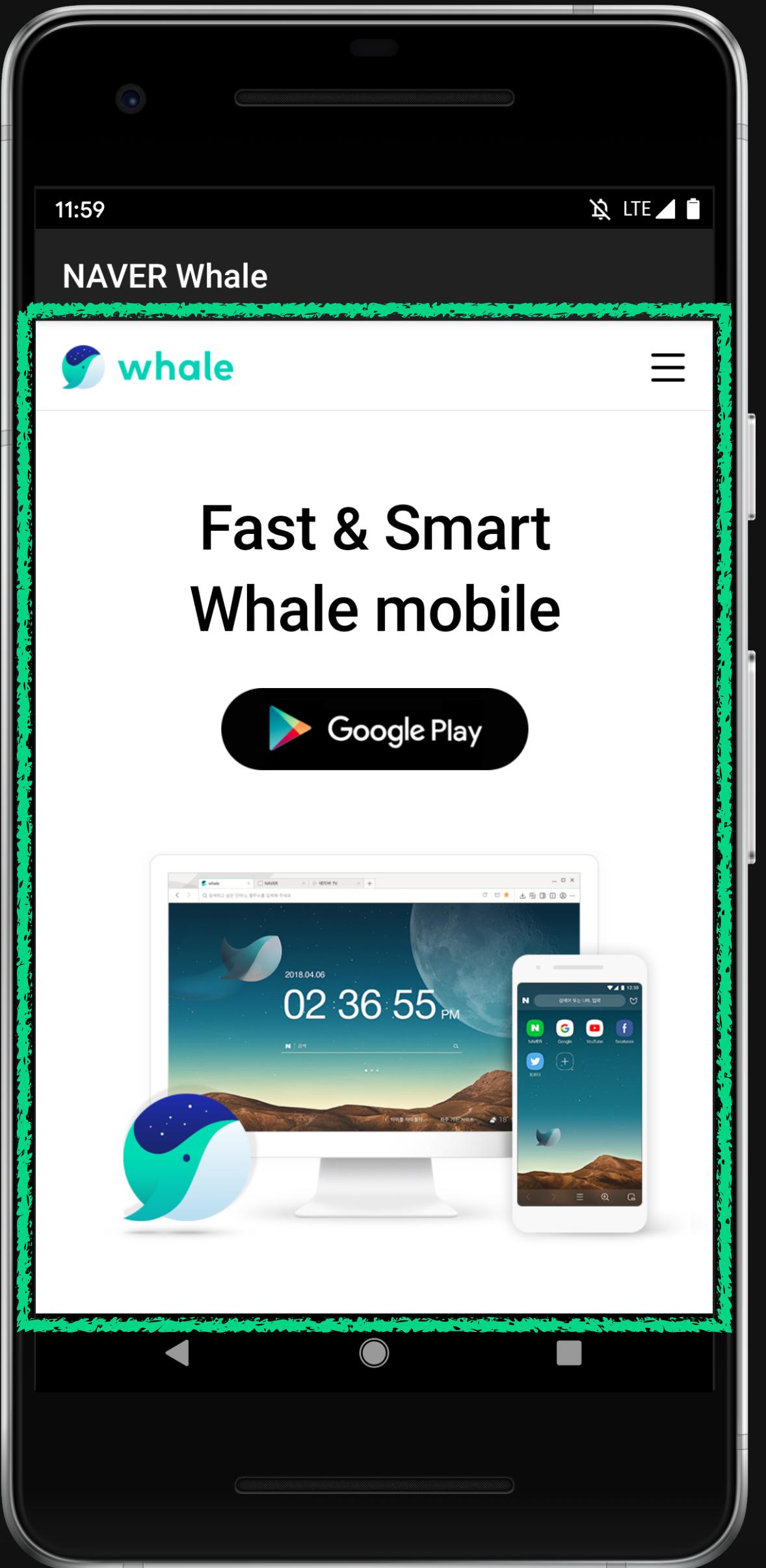
- 메뉴 색상이 적절하게 보인다.



CodeLab

Step 7. Optional

- WebView는 항상 밝게 보이는 문제 🤔
- ForceDark를 이용하여 어둡게 표시해보자.

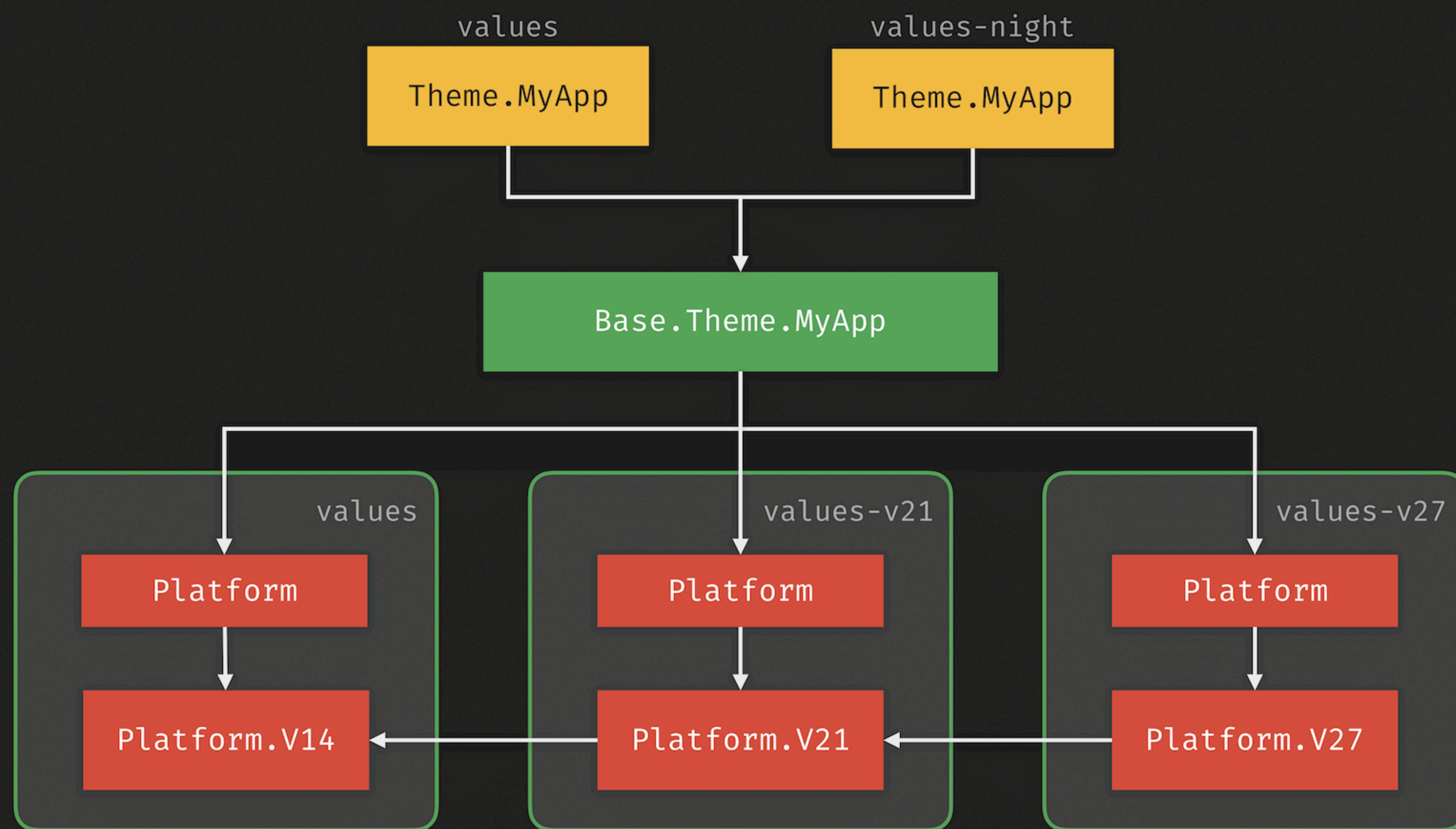


"Developing Themes with Style"

#3. Layer your themes for re-use



Link: <https://j.mp/themes-styles?slide=176>



Platform.Theme.MyApp

Use only to set API level specific attributes

```
values/themes.xml
<style name="Platform.V14.Theme.AppName"
      parent="Theme.MaterialComponents" />
<style name="Platform.Theme.AppName"
      parent="Platform.V14.Theme.AppName" />
```

```
values-v27/themes.xml
<style name="Platform.V27.Theme.AppName"
      parent="Platform.V14.Theme.AppName">
    <!-- Attributes specific to v27 -->
</style>
<style name="Platform.Theme.AppName"
      parent="Platform.V27.Theme.AppName" />
```

Platform.Theme.MyApp

Use only to set API level specific attributes

```
values/themes.xml
<style name="Platform.V14.Theme.AppName"
      parent="Theme.MaterialComponents" />
<style name="Platform.Theme.AppName"
      parent="Platform.V14.Theme.AppName" />
```

```
values-v27/themes.xml
<style name="Platform.V27.Theme.AppName"
      parent="Platform.V14.Theme.AppName">
    <!-- Attributes specific to v27 -->
</style>
<style name="Platform.Theme.AppName"
      parent="Platform.V27.Theme.AppName" />
```

Everything common across your themes

Usually have one of these

values/themes.xml

```
<style name="Base.Theme.AppName"  
      parent="Platform.Theme.AppName">
```

<!-- Things common across app -->

```
<item name="textAppearanceHeadline1">...</item>  
<item name="editTextStyle">...</item>
```

```
</style>
```

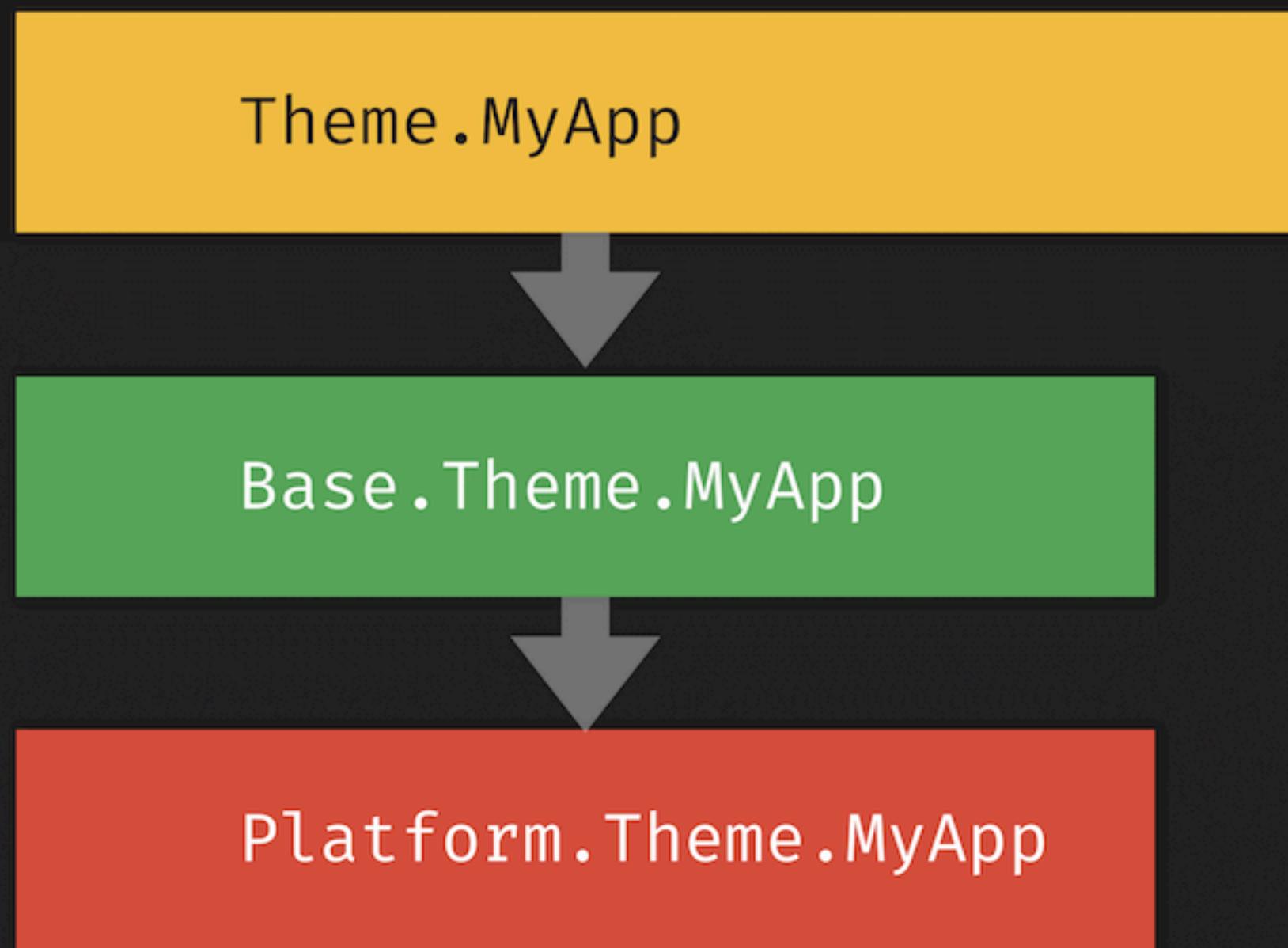
Base.Theme.MyApp



Platform.Theme.MyApp

Entry point

Can have multiple of these, but KISS



values/themes.xml

```
<style name="Theme.AppName"  
parent="Base.Theme.AppName" />
```

values-night/themes.xml

```
<style name="Theme.AppName"  
parent="Base.Theme.AppName">
```

<!-- Overrides for dark theme -->

```
</style>
```

Layer your themes (1/4)

- start/build.gradle

```
//TODO: STEP7 - Layer your themes
android {
    compileSdkVersion 29
    buildToolsVersion "29.0.0"
    defaultConfig {
        applicationId "soup.codelab.darktheme"
        minSdkVersion 21
        targetSdkVersion 29
    }
}
```

Layer your themes (2/4)

- res/values/themes.xml

```
<!-- TODO: STEP7 - Layer your themes -->
<style name="Platform.Theme.Whale"
      parent="Theme.AppCompat.DayNight" />

<style name="Base.Theme.Whale"
      parent="Platform.Theme.Whale" />
```

Layer your themes (3/4)

- res/values-v29/themes.xml

```
<!-- TODO: STEP7 - Layer your themes -->
<style name="Platform.Theme.Whale"
      parent="Theme.AppCompat.DayNight">
    <item name="android:isLightTheme">true</item>
    <item name="android:forceDarkAllowed">true</item>
</style>
```

STEP7. OPTIONAL

CodeLab

Layer your themes (4/4)

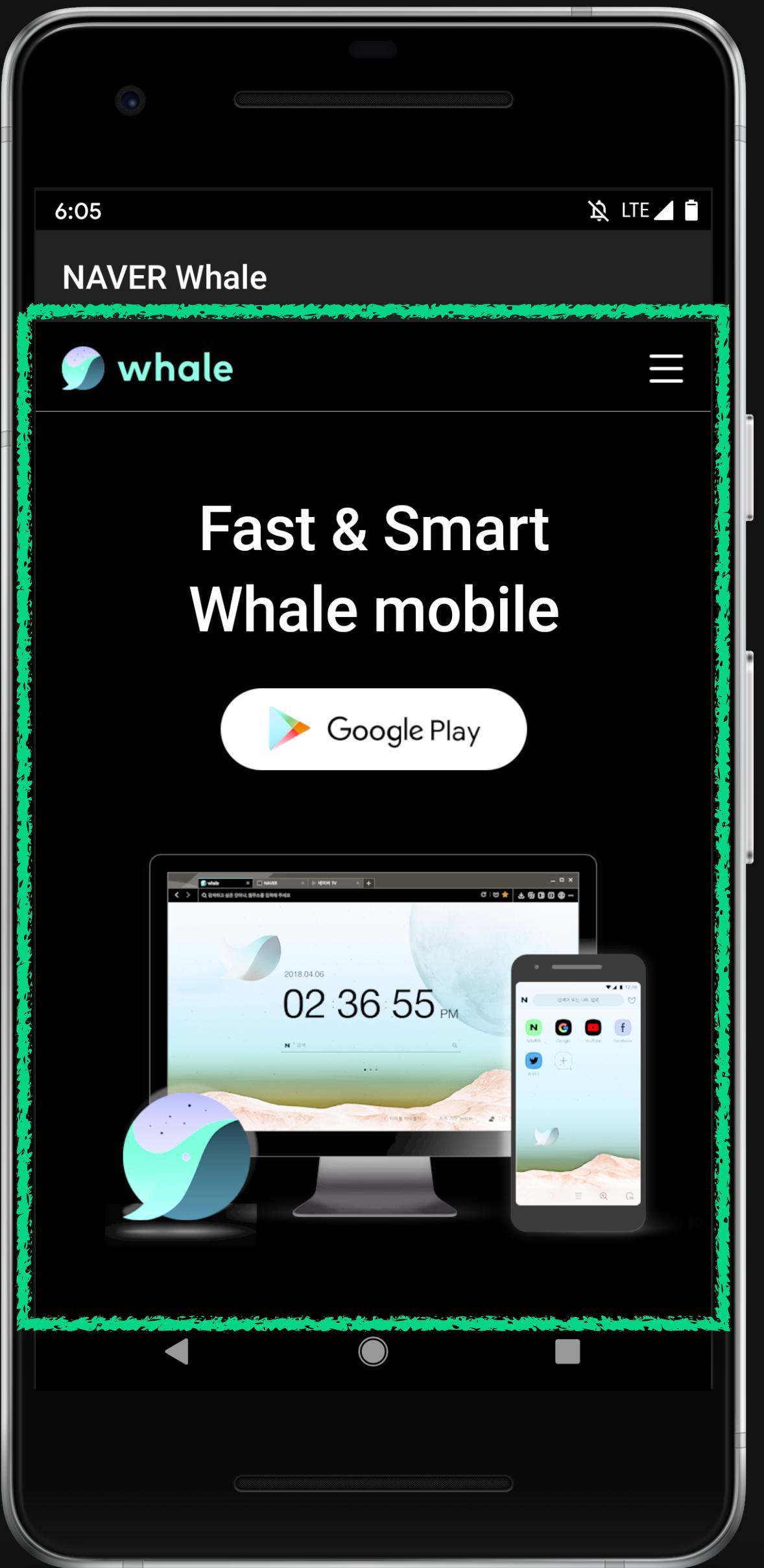
- res/values/styles.xml
- res/values-night/styles.xml

```
<!-- TODO: STEP7 - Layer your themes -->
<style name="AppTheme.Whale"
      parent="Base.Theme.Whale" />
```

STEP7. OPTIONAL

Done! 🎉

- Web Contents 색상이 반전되어 보인다.



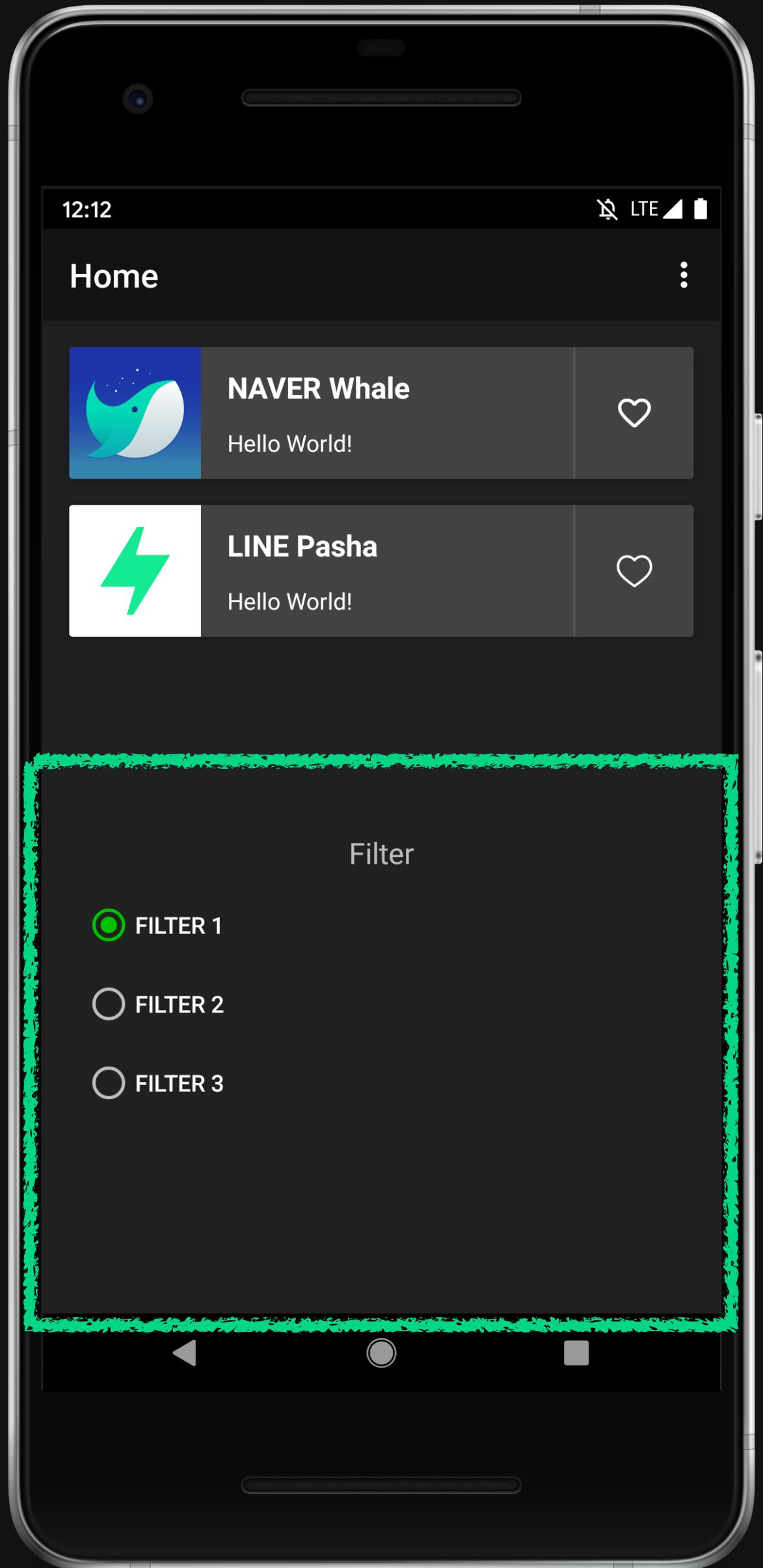
CodeLab

CodeLab

Step 8. Optional

- 다크테마에서 BottomSheet 시인성 문제 😱
- Elevation으로 인한 그림자가 잘 안보인다.

Black Bg + ***Black*** Shadow = ***Black***



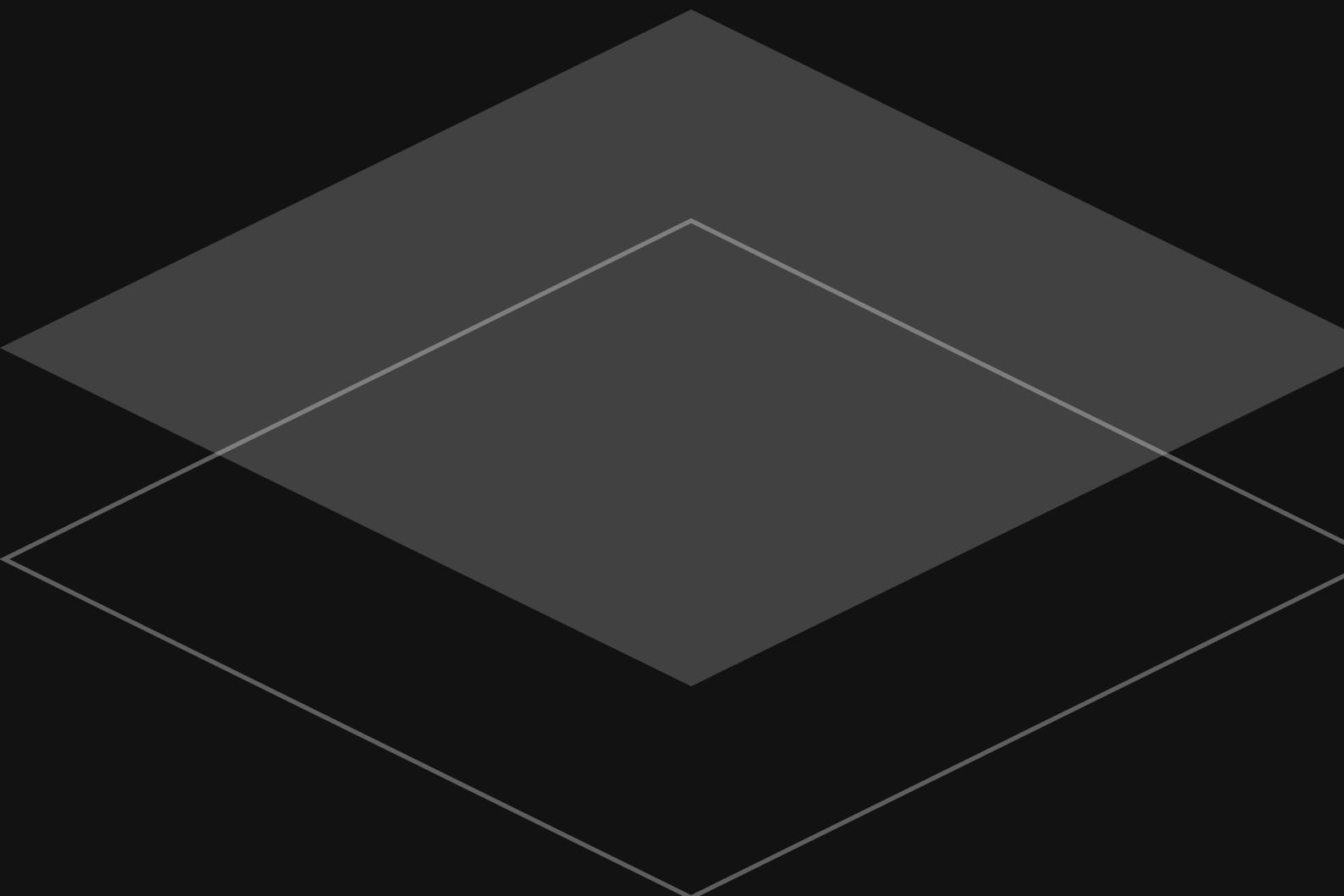
MDC 1.1.0 

Elevation Overlays for Dark Theme

Elevation

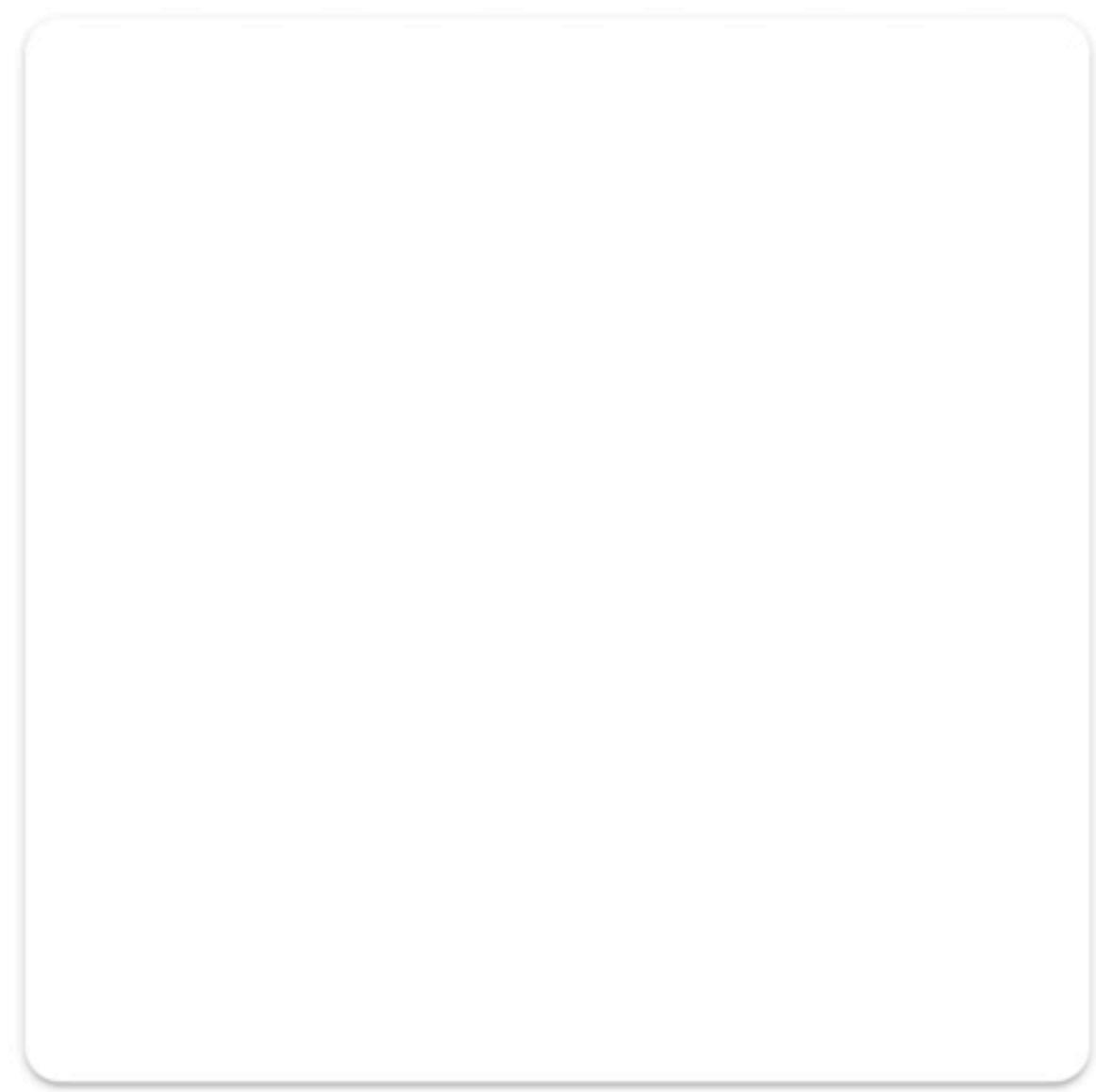
- Higher elevation, lighter surface
- 밝기는 반투명한 흰색 **overlay**로 조절된다.
 - Elevation (0dp ~ 24dp)
 - Overlay Transparency (0% ~ 16%)

Elevation Overlay



Surface

Shadow



24

20

16

12

8

4

1

Surface



ELEVATION

Elevation Overlays

- MDC Components는 아래 속성에 따라 Elevation Overlays가 적용된다.

```
<style name="Theme.MaterialComponents" parent="...">
    <item name="elevationOverlaysEnabled">true</item>
    <item name="elevationOverlaysColor">?attr/colorOnSurface</item>
</style>
```

- **MaterialShapeDrawable**, **ElevationOverlayProvider**를 이용하면 Custom View도 Elevation Overlays를 손쉽게 적용할 수 있다.
- 원치 않는 경우, **elevationOverlaysEnabled** 속성을 **false**로 변경하면 된다.

 ElevationOverlayProvider

- Elevation에 따라 Overlay 색상과 배경색을 적절히 섞어준다.

```
@ColorInt // layered color, if elevationOverlaysEnabled == true
public int layerOverlayIfNeeded(@ColorInt int bgColor, float elevation)
```

```
@ColorInt // layered color, always
public int layerOverlay(@ColorInt int bgColor, float elevation)
```

```
@ColorInt // layered color, of attr/colorSurface
public int getSurfaceColorWithOverlayIfNeeded(float elevation)
```

ELEVATION

C MaterialShapeDrawable

- MDC에서 배경에 사용되는 Drawable로,
내부적으로 `ElevationOverlayProvider`를 사용한다.

```
private val bgDrawable = MaterialShapeDrawable().apply {  
    initializeElevationOverlay(context)  
    fillColor = context.getColorAttr(R.attr.colorSurface)  
}
```

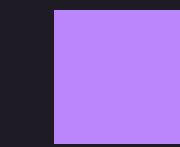
- 동적으로 Elevation 변경이 가능하다.

```
bgDrawable.setElevation(float elevation)
```

ELEVATION

Accessibility

- MDC Guideline: WCAG's AA 준수를 권장
 - 작은 텍스트 - 최소 4.5:1
 - 큰 텍스트 - 최소 3.0:1
- Text, BG 간 대비가 최소 15.8:1 는 되어야
24dp elevation에서 body text가 AA를 만족
- 24dp = 최대 elevation (MDC)



8% Primary

Dark Primary

#1F1B24

#121212

Link: <https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html>

STEP8. OPTIONAL

CodeLab

Get Lighter Surface (1/4)

- start/build.gradle

```
//TODO: STEP8 - Get Lighter Surface  
implementation 'com.google.android.material:material:1.1.0-alpha08'
```

STEP8. OPTIONAL

CodeLab

Get Lighter Surface (2/4)

- res/values/styles.xml

```
<!-- TODO: STEP8 - Get Lighter Surface -->
<style name="AppTheme.BottomSheet"
      parent="Theme.MaterialComponents.DayNight">
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="colorSurface">@color/colorBackground</item>
</style>
```

STEP8. OPTIONAL

CodeLab

Get Lighter Surface (3/4)

- MaterialViewBindingAdapter.kt

```
/* TODO: STEP8 - Get Lighter Surface */
@BindingAdapter("elevationOverlaysEnabled")
fun setElevationOverlaysEnabled(view: View, enabled: Boolean) {
    view.background = if (enabled) {
        MaterialShapeDrawable.createWithElevationOverlay(
            view.context, view.elevation)
    } else { ... }
}
```

STEP8. OPTIONAL

CodeLab

Get Lighter Surface (4/4)

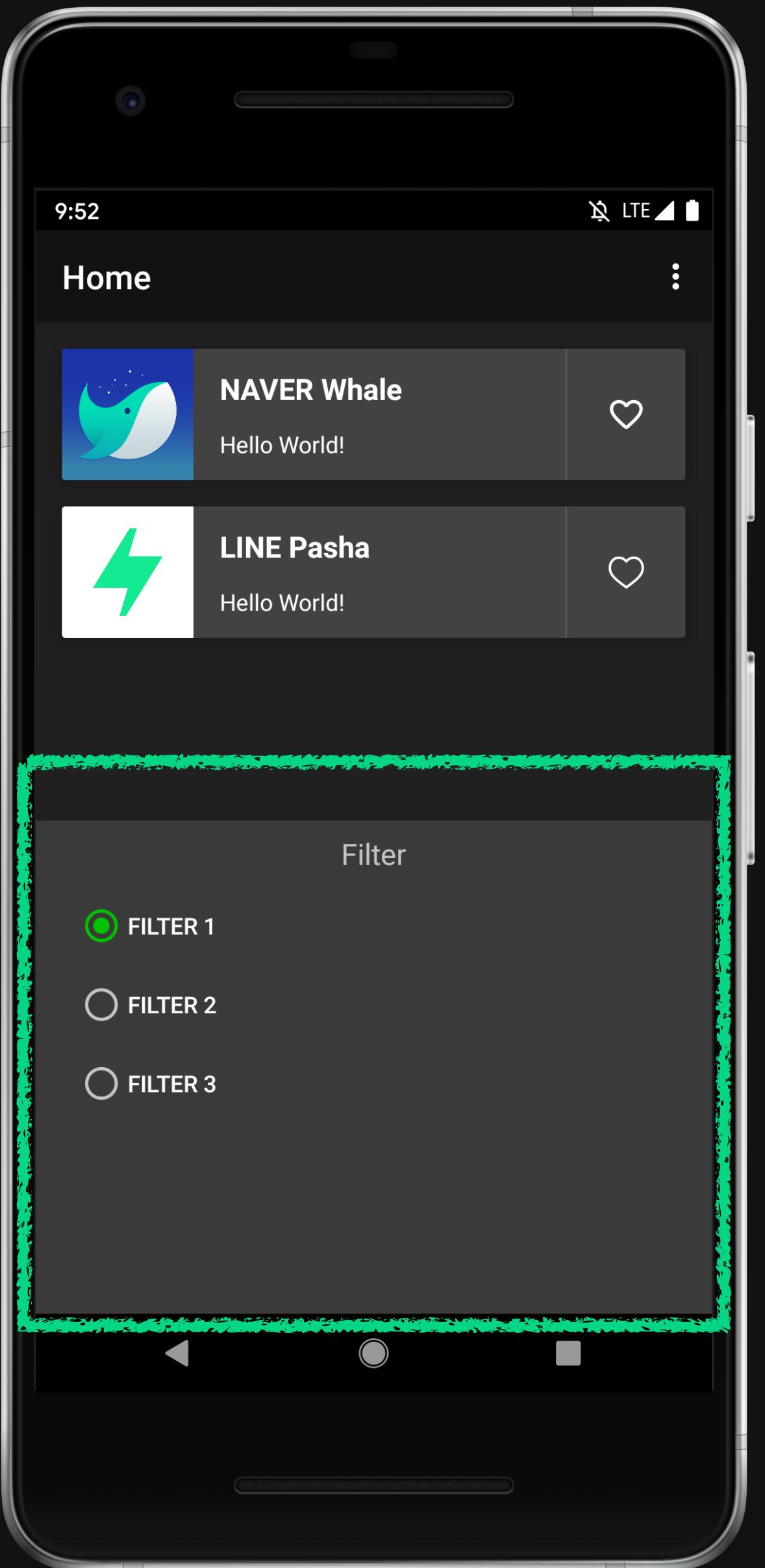
- res/layout/home_bottom_sheet.xml

```
<!-- TODO: STEP8 - Get Lighter Surface -->
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:elevation="8dp"
    android:theme="@style/AppTheme.BottomSheet"
    app:elevationOverlaysEnabled="@{true}">
```

STEP 8. OPTIONAL

Done! 🎉

- BottomSheet 표면이 밝게 보인다.



CodeLab

Others

Notification

- Notification은 자동 적용된다.
- Custom Notification은 아래 style을 사용할 것
 - **TextAppearance.Compat.Notification**
 - 단, 삼성 P OS는 적용 대상이 아니다.

