

# **UI optimization for night**

Keishin Yokomaku @ Drivemode, Inc.  
potatotips #23

# @KeithYokoma



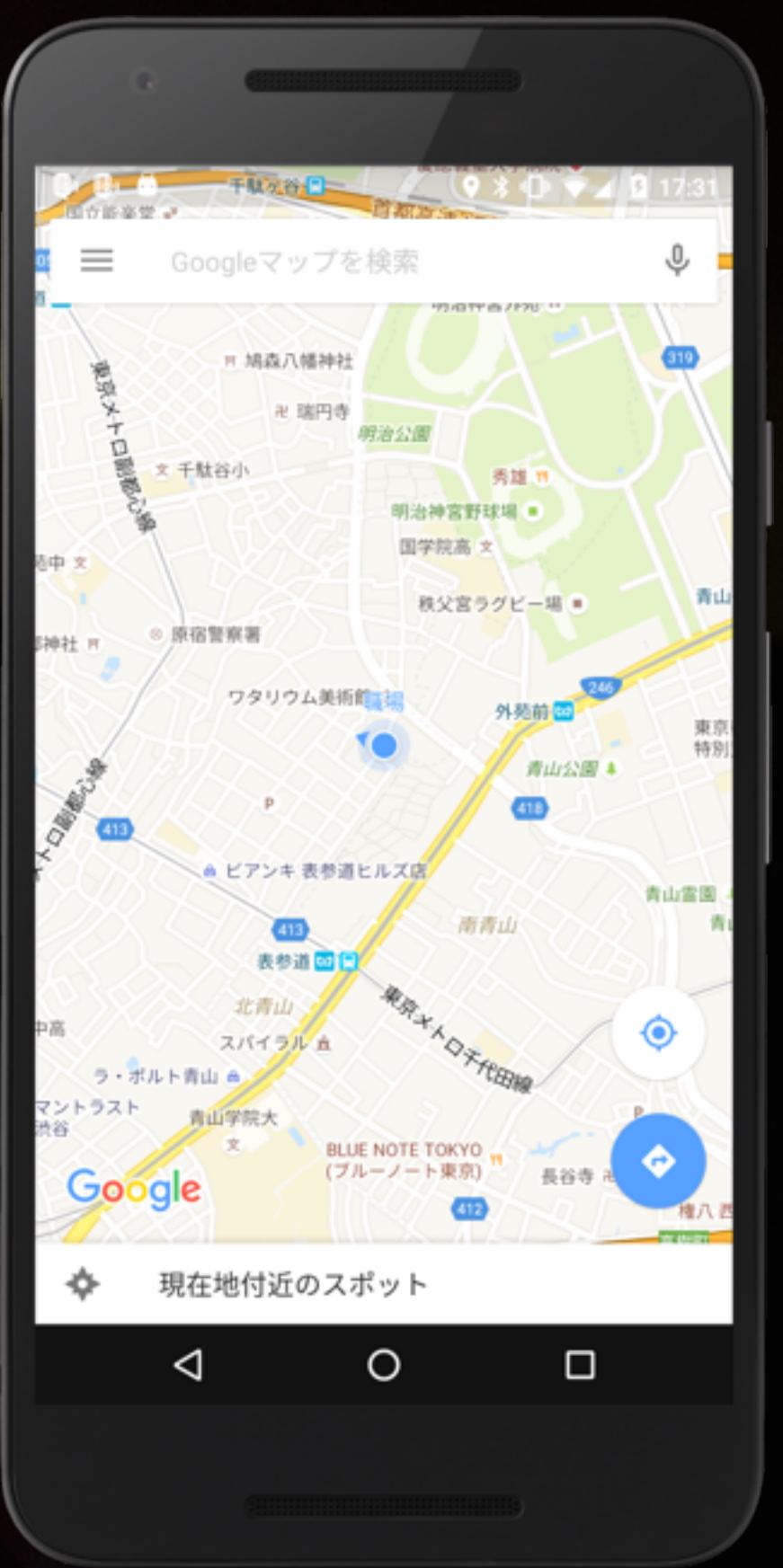
- Keishin Yokomaku at Drivemode, Inc.
- Work
  - Android apps      
  - Android Training and its publication
- Like
  - Bicycle, Photography, Tumblr and Motorsport
- Preferred Hashtag: #また闇の話してある.....



**No darkness today** 😂

**Using phone  
in the darkness...**





A close-up of a man's face, appearing shocked or distressed. He has his hands pressed against his eyes, fingers spread. His mouth is slightly open, showing his tongue. He is wearing a dark suit jacket over a white shirt and a yellow striped tie.

**“My eyes!!!!”**

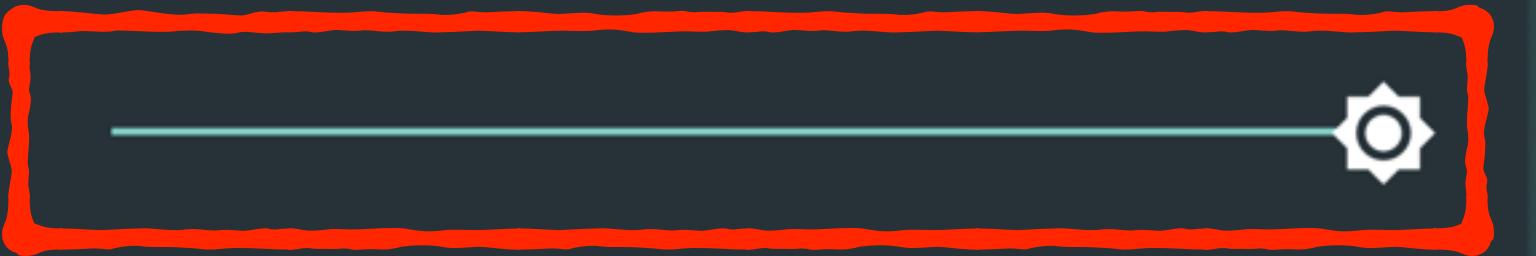
**“Bright display is dazzling eyes”**

100%



11:24

11月20日金曜日



Drivemode WiFi 5GHz ▾

Bluetooth ▾



NTT DOCOMO



機内モード



通知を非表示



自動回転



ライト



現在地



アクセスポイント



USBデバッグが接続されました  
タップしてUSBデバッグを無効化



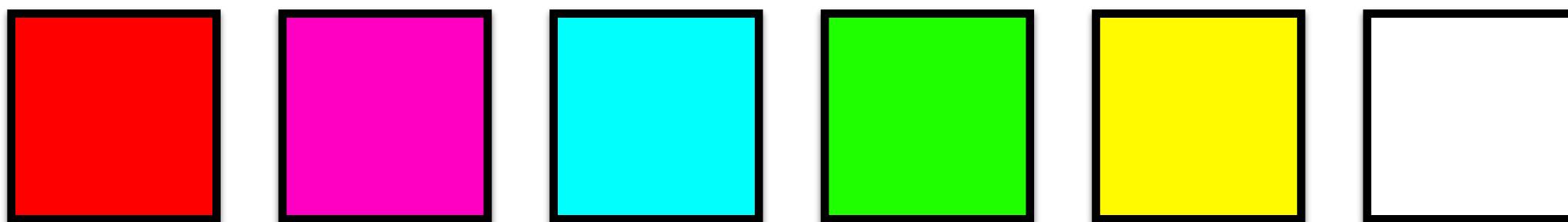
**“Is it enough to use  
auto adjustment of display brightness?”**

**“No”**

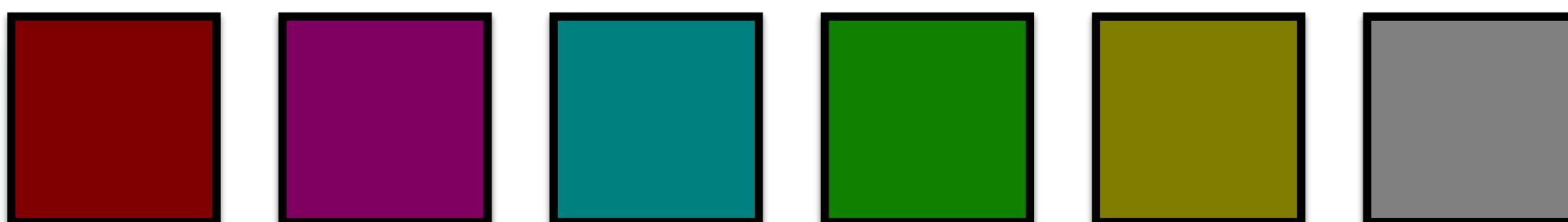
# Brightness of colours

---

- Colour properties of brightness
  - HSL (Hue, Saturation, **Lightness**; a.k.a HSB or HSV)
- Bright colours (Brightness = 100%)



- Dark colours (Brightness = 50%)



# Theme

---

- Set various colours on various parts of the application with **Theme**.
- Separate themes between daytime and night.
- You can change your theme at runtime with
  - *Activity#setTheme()*
  - Note: Call *Activity#setTheme()* before calling *Activity#onCreate()*

A wide-angle aerial photograph of a city during sunset. The sky is filled with dramatic, orange and yellow clouds. In the foreground, there's a dense urban area with numerous buildings, some with illuminated windows. A prominent feature is a long, elevated highway or bridge that stretches across the middle ground. In the far distance, a tall skyscraper stands out against the horizon. The overall atmosphere is hazy and golden.

**“When should we change our theme?”**

# Changing theme

---

- Dark theme during nighttime(after **sunset** and before **sunrise**)
- Dark theme under dark places(using illuminance sensors)

# Sunrise equation

---

- Equation to calculate sunrise and sunset time
  - [https://en.wikipedia.org/wiki/Sunrise\\_equation](https://en.wikipedia.org/wiki/Sunrise_equation)
- The implementation is available on AOSP
  - TwilightCalculator.java (<http://bit.ly/1T2A0hF>)
  - TwilightManager.java (<http://bit.ly/1QEG8yr>)
  - Calculates sunrise/sunset from date(epoch), latitude and longitude

# Using TwilightCalculator

---

```
public class ThemeManager {  
    private final TwilightCalculator mCalculator;  
    private final LocationManager mLocationManager;  
  
    public void prepare() {  
        Location location = mLocationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
        if (location != null) {  
            mCalculator.calculateTwilight(System.currentTimeMillis(),  
                location.getLatitude(), location.getLongitude());  
        } else {  
            // request location on LocationManager and get updated location to calculate twilight  
        }  
    }  
  
    public boolean isInNight() {  
        long now = System.currentTimeMillis();  
        return now > mCalculator.mSunset || now < mCalculator.mSunrise;  
    }  
}
```

# Using TwilightCalculator

---

```
public class ThemeManager {  
    private final TwilightCalculator mCalculator;  
    private final LocationManager mLocationManager;  
  
    public void prepare() {  
        Location location = mLocationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
        if (location != null) {  
            mCalculator.calculateTwilight(System.currentTimeMillis(),  
                location.getLatitude(), location.getLongitude());  
        } else {  
            // request location on LocationManager and get updated location to calculate twilight  
        }  
    }  
  
    public boolean isInNight() {  
        long now = System.currentTimeMillis();  
        return now > mCalculator.mSunset || now < mCalculator.mSunrise;  
    }  
}
```

# Using TwilightCalculator

---

```
public class ThemeManager {  
    private final TwilightCalculator mCalculator;  
    private final LocationManager mLocationManager;  
  
    public void prepare() {  
        Location location = mLocationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
        if (location != null) {  
            mCalculator.calculateTwilight(System.currentTimeMillis(),  
                location.getLatitude(), location.getLongitude());  
        } else {  
            // request location on LocationManager and get updated location to calculate twilight  
        }  
    }  
  
    public boolean isInNight() {  
        long now = System.currentTimeMillis();  
        return now > mCalculator.mSunset || now < mCalculator.mSunrise;  
    }  
}
```

# Running the calculation

---

- Once per day on date changed
  - Intent.ACTION\_DATE\_CHANGED: has some bugs...
  - AlarmManager#setExact() or set(): seems fine
- When timezone is changed
- If you use TwilightManager...
  - It automatically calculate sunrise/sunset time so no further stuff needed

**“Why don’t you use `UiModeManager`? ”**

# UiModeManager

---

- Pros
  - Easily change ui mode with sensors and location
  - Be able to locate resources with qualifier for night
- Cons
  - Need to enable CarMode which brings huge Android Auto logo in front.
  - Steering wheel notification icon is always on the status bar.

# **UI optimization for night**

Keishin Yokomaku @ Drivemode, Inc.  
potatotips #23