

Android でも Haptics 入門

potatotips #84
@Kaito-Dogi



自己紹介

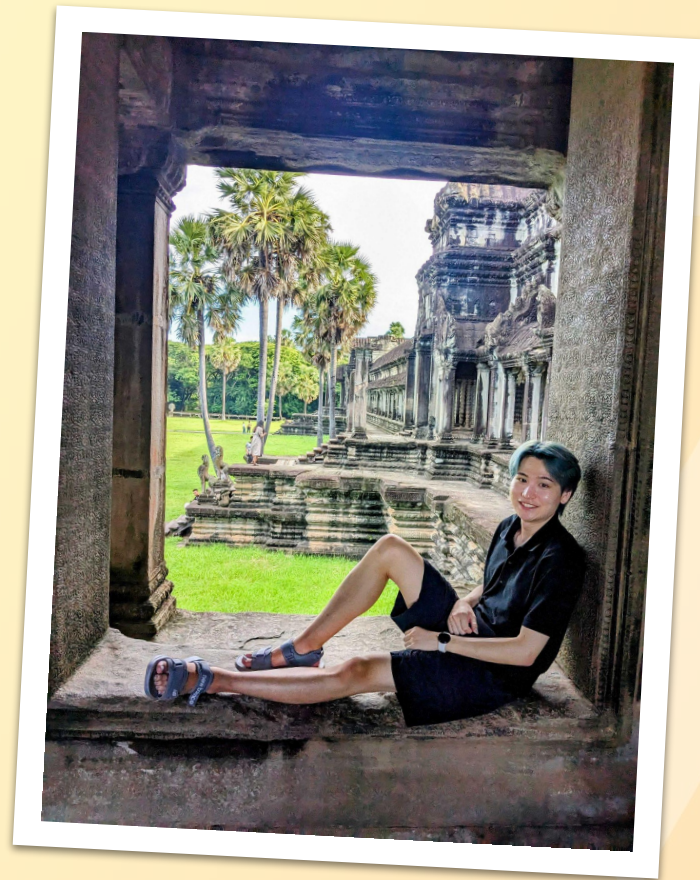
- どぎー
- 株式会社ゆめみ
- Android エンジニア
- カンボジア行ってきました🇰🇲



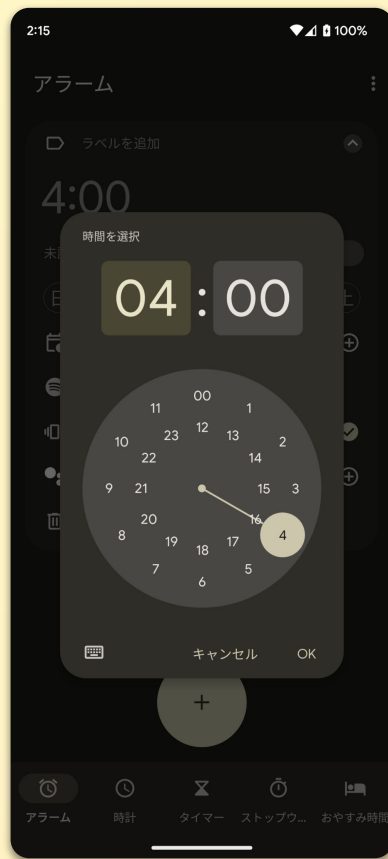
@Kaito_Dogi



@Kaito-Dogi



“Haptics”
聞いたことがありますか？



“振動”

Haptics とは

- ❑ Haptics is everything you feel through the sense of touch.
> 触覚を通して感じるものすべて
- ❑ Android apps can give users a richer experience with subtlety and depth.
> 繊細さと奥行きのあるリッチなユーザー体験

引用：<https://developer.android.com/develop/ui/views/haptics>

視覚、聴覚、そして“触覚”

Android でどう実装するか

Android での実装方法

HapticFeedback Constants

- ❑ Haptics の意味・ジェスチャーを重視
- ❑ UI イベントへの応答
- ❑ View が必要

VibrationEffect

- ❑ Haptics の強さを重視
- ❑ View なしで使える
- ❑ カスタムパターンを定義できる

引用：<https://youtu.be/00jRoEFnpk8?si=AVNTj6BDWhllJibp>

HapticFeedbackConstants

```
class HapticTouchListener : View.OnTouchListener {  
    override fun onTouch(view: View, event: MotionEvent) : Boolean {  
        when (event.actionMasked) {  
            MotionEvent.ACTION_DOWN ->  
                view.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)  
            MotionEvent.ACTION_UP ->  
                view.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY_RELEASE)  
        }  
        return true  
    }  
}
```

引用：<https://developer.android.com/develop/ui/views/haptics/haptic-feedback>

HapticFeedbackConstants

```
class HapticTouchListener : View.OnTouchListener {  
    override fun onTouch(view: View, event: MotionEvent) : Boolean {  
        when (event.actionMasked) {  
            MotionEvent.ACTION_DOWN ->  
                view.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)  
            MotionEvent.ACTION_UP ->  
                view.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY_RELEASE)  
        }  
        return true  
    }  
}
```

押す・離すのペアで
物理的なボタンを模倣

引用：<https://developer.android.com/develop/ui/views/haptics/haptic-feedback>

HapticFeedbackConstants

```
class HapticTouchListener : View.OnTouchListener {  
    override fun onTouch(view: View, event: MotionEvent) : Boolean {  
        when (event.actionMasked) {  
            MotionEvent.ACTION_DOWN ->  
                view.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)  
            MotionEvent.ACTION_UP ->  
                view.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY_RELEASE)  
        }  
        return true  
    }  
}
```

View に定義されている

引用 : <https://developer.android.com/develop/ui/views/haptics/haptic-feedback>

VibrationEffect

1. createPredefined

- 事前定義されたパターンを使用する

2. createWaveform

- カスタムパターンを定義する

3. startComposition

- 事前定義されたパターンを合成して、よりリッチなカスタムパターンを定義する

VibrationEffect#createPredefined

```
val vibrator = context.getSystemService(Vibrator::class.java)

vibrator.vibrate(VibrationEffect.createPredefined(VibrationEffect.EFFECT_CLICK))
```

引用：<https://developer.android.com/develop/ui/views/haptics/haptic-feedback>

VibrationEffect#createWaveform

```
val vibrator = context.getSystemService(Vibrator::class.java)

val timings: LongArray = longArrayOf(50, 50, 100, 50, 50)
val amplitudes: IntArray = intArrayOf(64, 128, 255, 128, 64)
val repeatIndex = -1

vibrator.vibrate(
    VibrationEffect.createWaveform(timings, amplitudes, repeatIndex),
)
```

引用：<https://developer.android.com/develop/ui/views/haptics/custom-haptic-effects>

VibrationEffect#startComposition

```
val vibrator = context.getSystemService(Vibrator::class.java)

val scale = 0.8f
val delayMs = 100

vibrator.vibrate(
    VibrationEffect.startComposition().addPrimitive(
        VibrationEffect.Composition.PRIMITIVE_SLOW_RISE,
    ).addPrimitive(
        VibrationEffect.Composition.PRIMITIVE_CLICK, scale, delayMs,
    ).compose(),
)
```

引用：<https://developer.android.com/develop/ui/views/haptics/custom-haptic-effects>

Android でどう取り入れるか

Material Design における原則

1. システムパターンに従う
 - カスタムパターンは必要最低限
2. 全体的なユーザー体験を考える
 - そのインタラクション・コンテキスト・環境に合うパターンを選ぶ
 - 単独でも、オーディオ・ビジュアルと組み合わせて使用してもよい
3. Haptics はシンプルな情報を伝えるもの
4. ユーザーを驚かせない
 - 不快な触覚を避ける
 - パターンを予測どおりに使用する

引用：<https://m2.material.io/design/platform-guidance/android-haptics.html#usage>

OS・デバイスの制限を受ける



サンプルコード



参考

- ❑ Implement haptics on Android
<https://developer.android.com/develop/ui/views/haptics>
- ❑ Advanced Haptics: The when, what, and how of new haptic APIs (Android Dev Summit '19)
<https://youtu.be/00jRoEFnpk8?si=AVNTj6BDWhlJibp>
- ❑ Add haptic feedback to events
<https://developer.android.com/develop/ui/views/haptics/haptic-feedback>
- ❑ Create custom haptic effects
<https://developer.android.com/develop/ui/views/haptics/custom-haptic-effects>
- ❑ Android haptics
<https://m2.material.io/design/platform-guidance/android-haptics.html>
- ❑ UX foundation for haptic framework
<https://source.android.com/docs/core/interaction/haptics/haptics-ux-foundation>
- ❑ Haptic Samples
<https://github.com/android/platform-samples/tree/main/samples/user-interface/haptics>
- ❑ android / platform / frameworks / base / refs/heads/main / . / core / java / android / os / VibrationEffect.java
<https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/java/android/os/VibrationEffect.java>
- ❑ CoreHaptics入門 by Satsuki Hashiba
<https://fortee.jp/iosdc-japan-2023/proposal/b630fa79-2c28-41ec-a6b2-6954b62f1858>

ありがとうございました 🙌