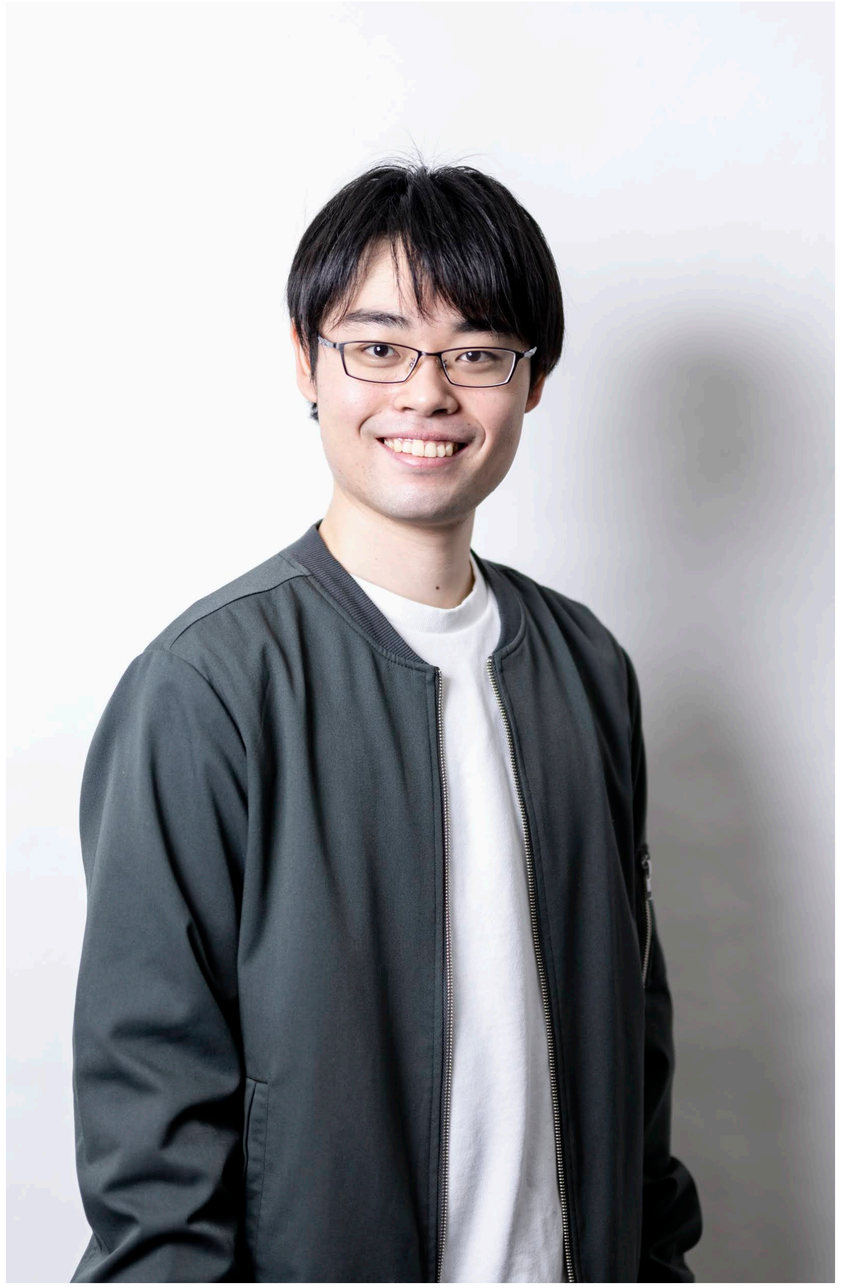
CyberAgent

WindowInsets 2022

feat. Jetpack Compose

Mori Atsushi 2022-08-18 #ca_aab



森 篤史

2019年度 新卒入社

Androidアプリエンジニア

Next Experts (Android)

Jetpack Composeゼミ代表

2019年度 未踏スーパークリエータ

@at_sushi_at

Mori-Atsushi

CYBER

OPENREC.tv

Jetpack Composeゼミ

- CAゼミ制度の1つ
  - 特定のテーマに沿って技術者が自主的に集まり、

    技術力の向上を目指して活動

- Jetpack Composeゼミの活動（メンバー7人）
  - 隔週の勉強会で最新情報キャッチアップ
  - プロダクション導入の情報共有
  - OSSの公開 / 技術記事の執筆

Katalog

A UI Catalog Library made with Jetpack Compose

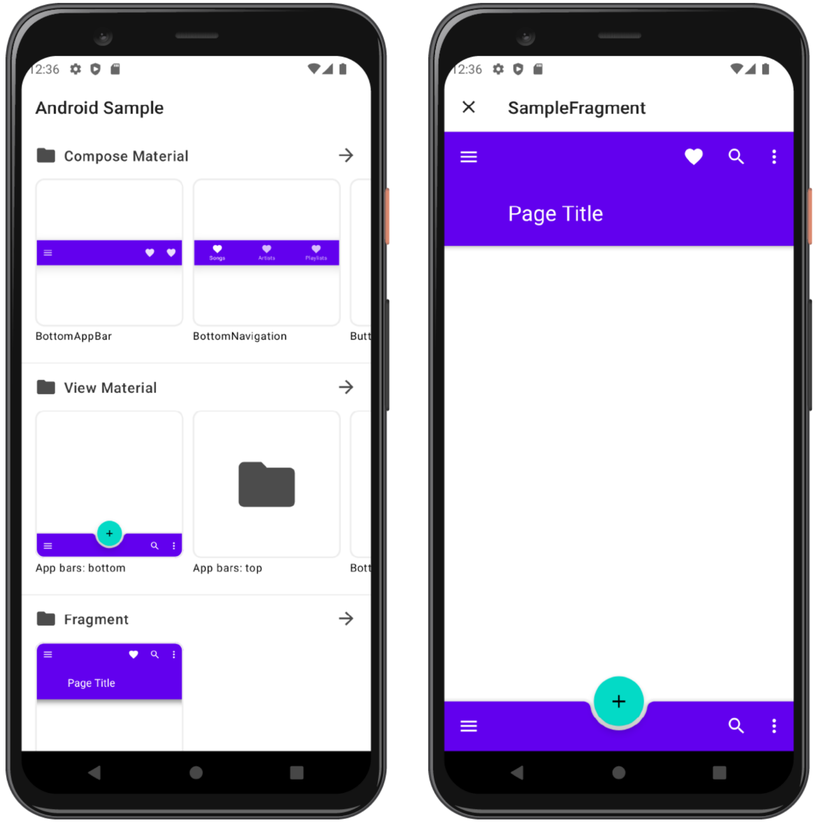
- UIコンポーネントやページを登録する

  デバッグ用アプリケーション

- DSLで記述できる

- Jetpack Composeの他、Android Viewや

  DataBinding / ViewBindingに対応



https://cyberagent-zemi.github.io/katalog/

OPENREC.tv

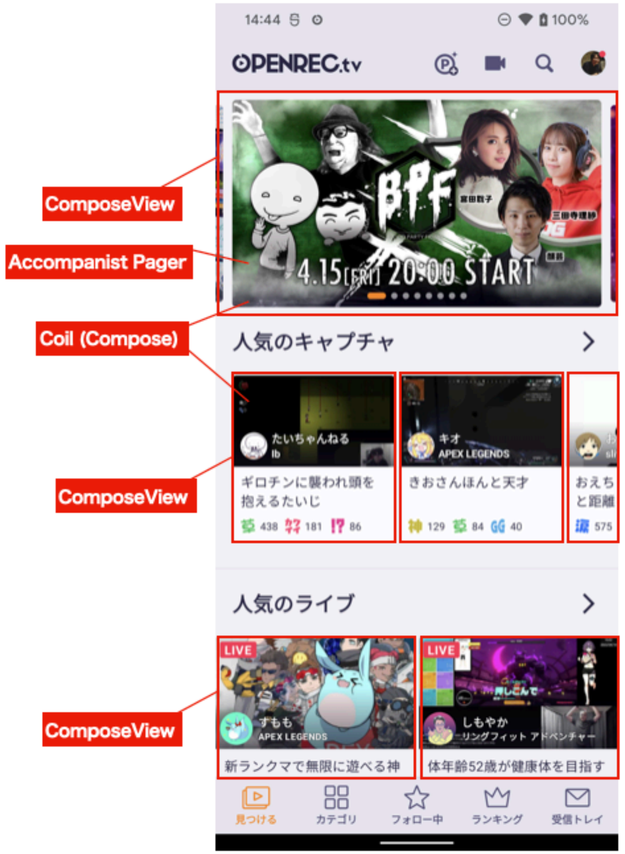
ゲーム実況やプレイ動画が楽しめるライブ動画プラットフォーム

- 高画質かつ業界最高水準の低遅延なライブ配信
- サブスク / PPV
- 様々なジャンルの配信（eスポーツ大会、声優、音楽ライブ 他）

OPENREC.tvとJetapck Compose

10画面以上を完全移行

コンポーネントを移行



新機能はComposeで

過去の登壇

https://ca-base-next.cyberagent.co.jp/2022/sessions/jetpack-compose-openrec/

Jetpack Compose 1.2

2022年7月 stable release

- LazyHorizontalGrid, LazyVerticalGridのstable化
- WindowInsets APIの追加
- Android ViewとComposeのnested scrollサポート
- Compose + RecyclerViewのパフォーマンス改善
- ダウンロードフォントのサポート（experimental）
- includeFontPaddingのサポート（experimental）

Jetpack Compose 1.2

2022年7月 stable release

- LazyHorizontalGrid, LazyVerticalGridのstable化
- **WindowInsets APIの追加**
- Android ViewとComposeのnested scrollサポート
- Compose + RecyclerViewのパフォーマンス改善
- ダウンロードフォントのサポート（experimental）
- includeFontPaddingのサポート（experimental）

◀ 今日はここの話

Contents

CyberAgent

1

WindowInsetsとは

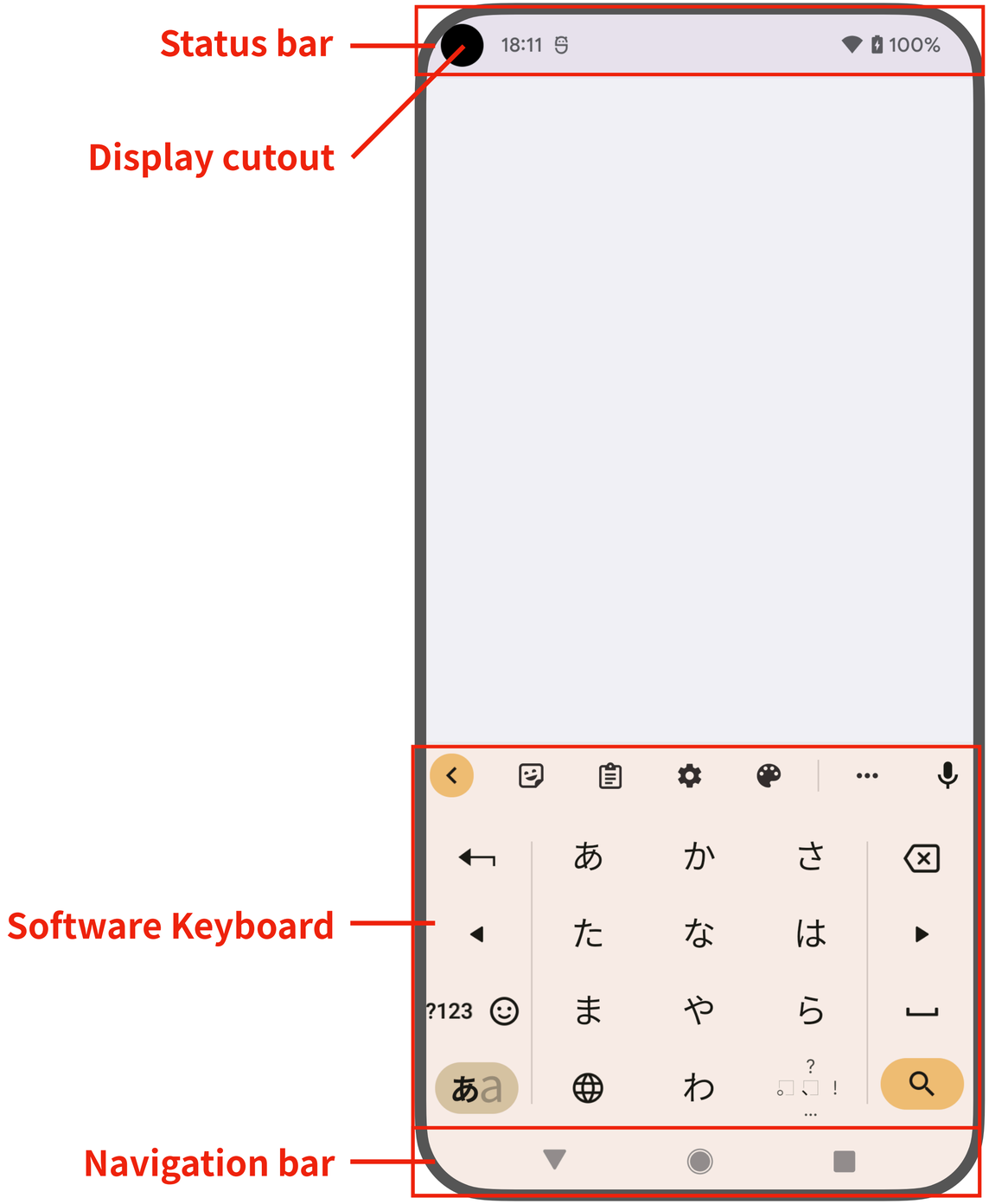
CyberAgent

WindowInsets

システムによって描画される領域

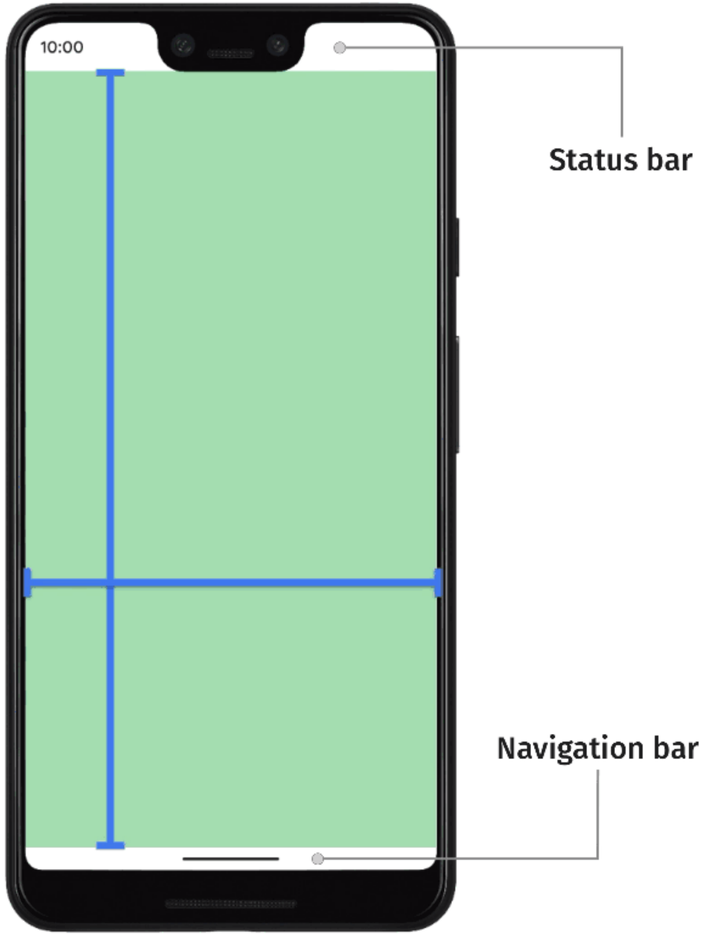
- Status bar
- Navigation bar
- Software keyboard(IME)
- Display cutout(ノッチ)

重要なコンテンツが重ならないように
避ける必要がある



ジェスチャーナビゲーションとEdge-to-Edge対応

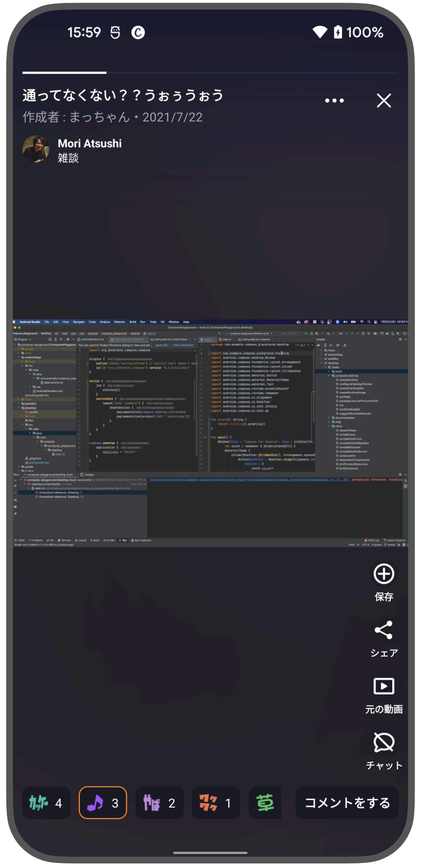
- Android 10からジェスチャーナビゲーションが追加された
- ナビゲーション バーの背後にもコンテンツを描画することが推奨されている
- アプリに対する没入感が増す



https://developer.android.com/training/gestures/edge-to-edge

OPENREC.tvでの対応

主要画面、新規画面で随時対応中



2

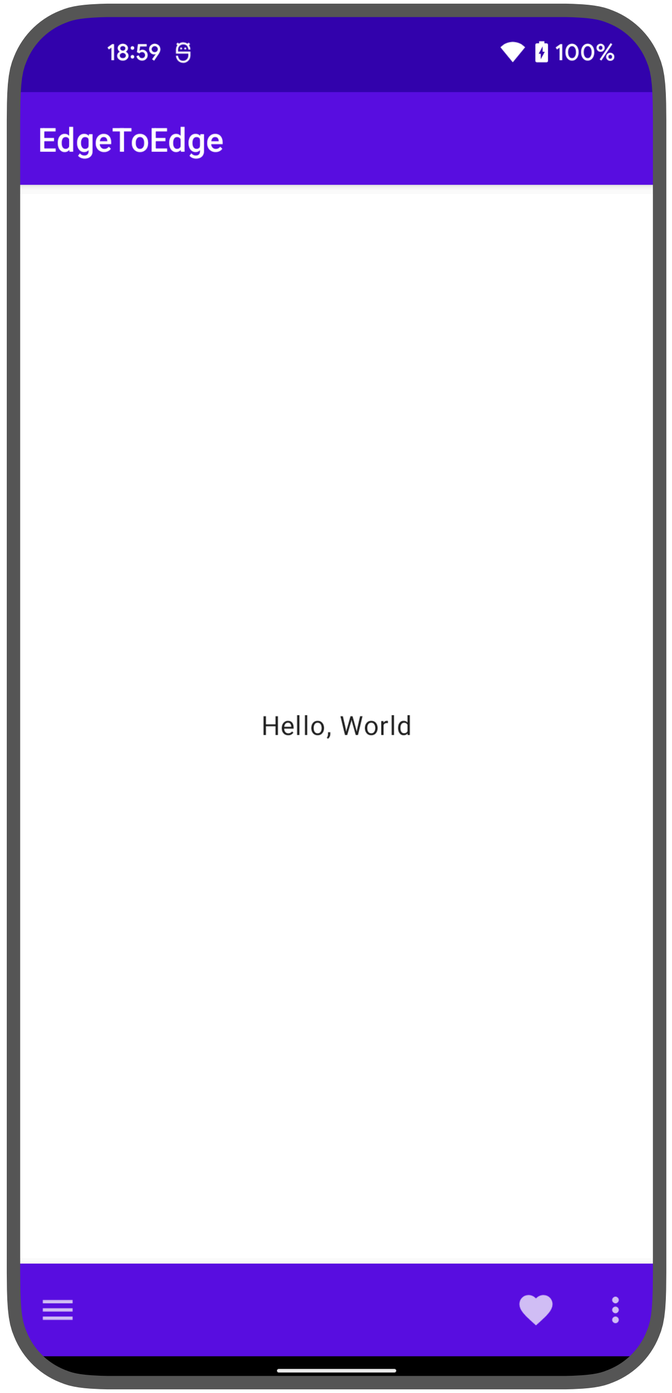
Edge-to-Edgeにチャレンジ

CyberAgent

```kotlin
class EdgeToEdgeActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContent {
            MaterialTheme {
                EdgeToEdge()
            }
        }
    }
}

@Composable
fun EdgeToEdge() {
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text(text = "EdgeToEdge") },
            )
        },
        bottomBar = {
            BottomAppBar {
                /* ... */
            }
        },
    ) {
        /* ... */
    }
}
```



描画領域を広げる

```
class EdgeToEdgeActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        WindowCompat.setDecorFitsSystemWindows(window, false)
        window.statusBarColor = Color.TRANSPARENT
        window.navigationBarColor = Color.TRANSPARENT

        setContent {
            MaterialTheme {
                EdgeToEdge()
            }
        }
    }
}
```

windowを広げる

navigation bar / status barを透明に

Android 10未満ではnavigation barの
自動着色が無いなど制約があるので、分岐が必要

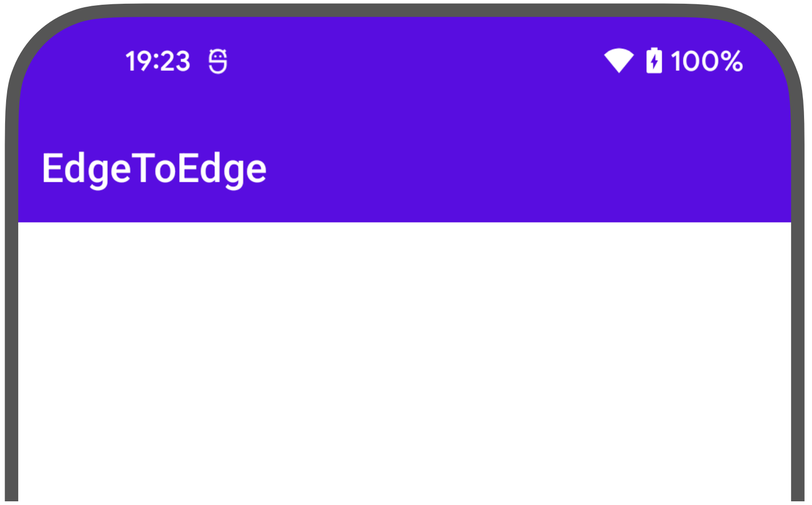
Status barを避ける

```
TopAppBar(
    modifier = Modifier
        .background(MaterialTheme.colors.primarySurface)
        .statusBarsPadding(),
    title = {
        Text(text = "EdgeToEdge")
    },
    elevation = 0.dp,
)
```

背景色つける

Status barの高さ確保

影が間に入ってしまうので消す



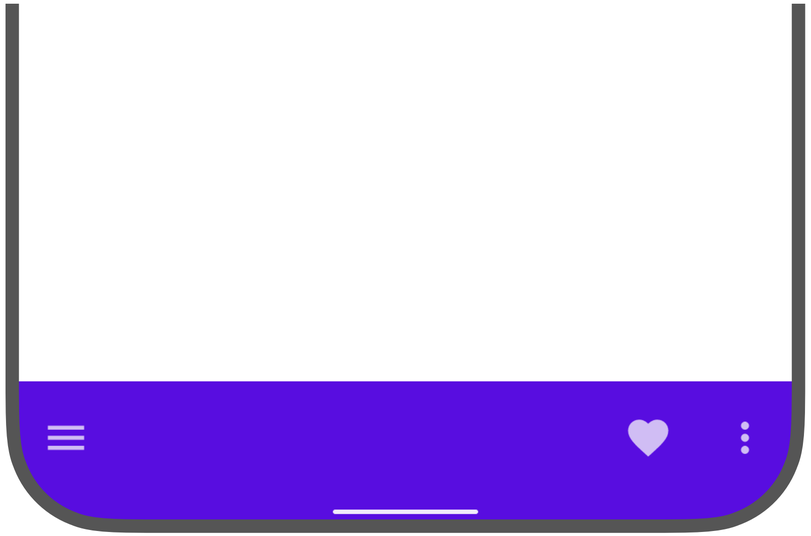
Navigation barを避ける

```
BottomAppBar(
    modifier = Modifier
        .background(MaterialTheme.colors.primarySurface)
        .navigationBarsPadding(),
    elevation = 0.dp,
) {
    /* ... */
}
```

背景色つける

Navigation barの高さ確保

影が間に入ってしまうので消す



Android View版との違い

Jetpack Compose

```
Box(
    modifier = Modifier.navigationBarsPadding(),
)
```

Insetsに変更があれば自動で反映される

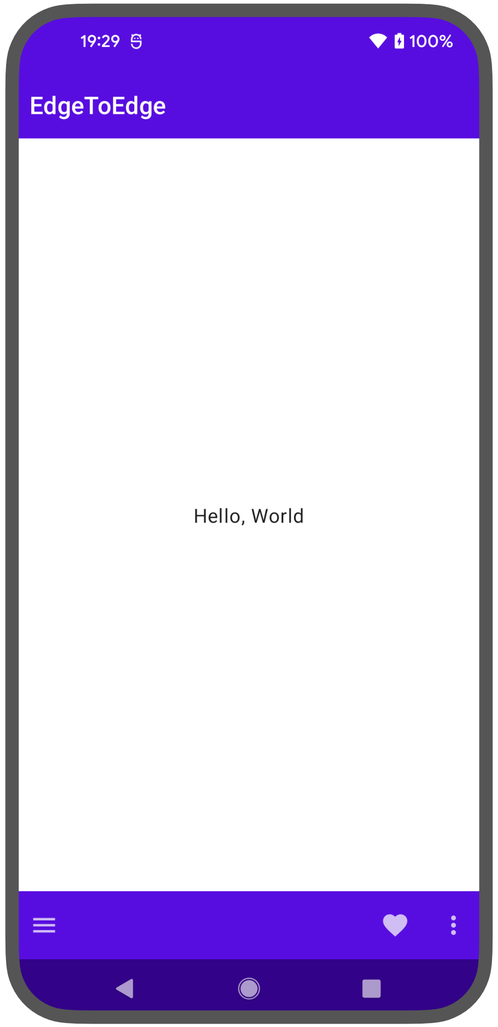
Android View + Compat

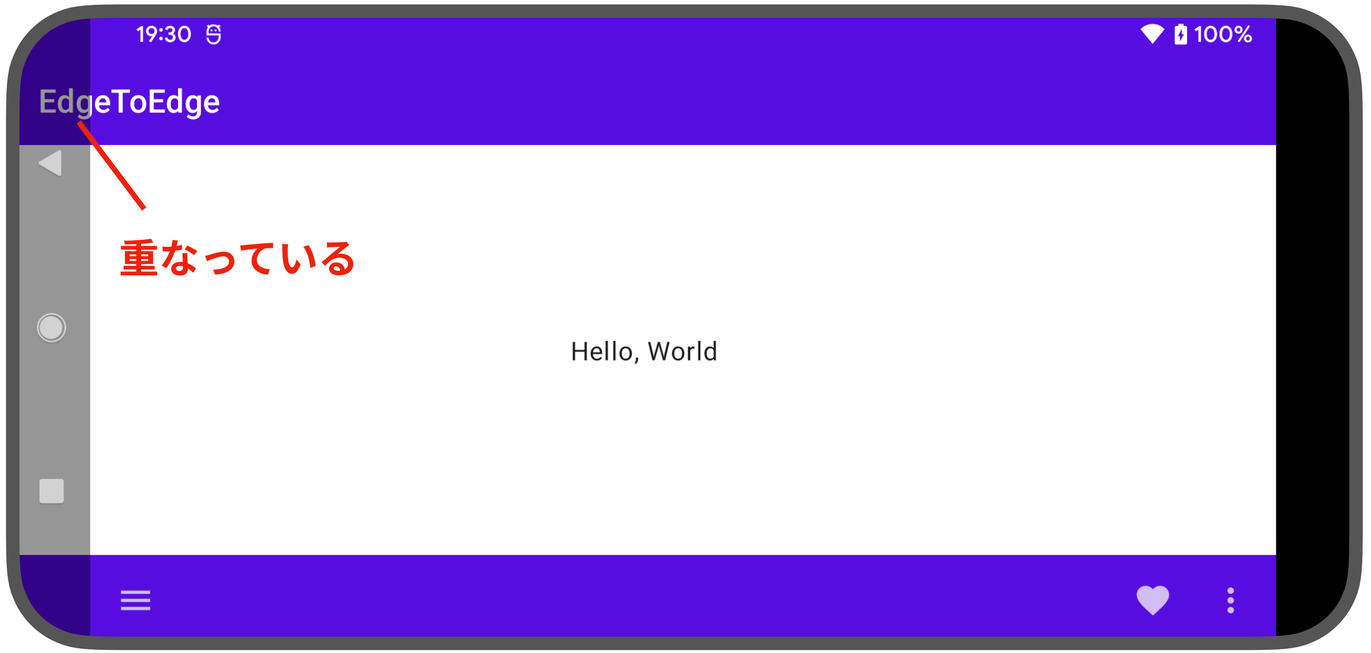
```
ViewCompat.setOnApplyWindowInsetsListener(view) { view, insets ->
    val systemInsets = insets.getInsets(
        WindowInsetsCompat.Type.statusBars(),
    )

    view.setPadding(
        systemInsets.left,
        systemInsets.top,
        systemInsets.right,
        systemInsets.bottom,
    )

    insets
}
```

3ボタン＋横画面に注意！


19:29
100%
EdgeToEdge
Hello, World


19:30
100%
EdgeToEdge
重なっている
Hello, World

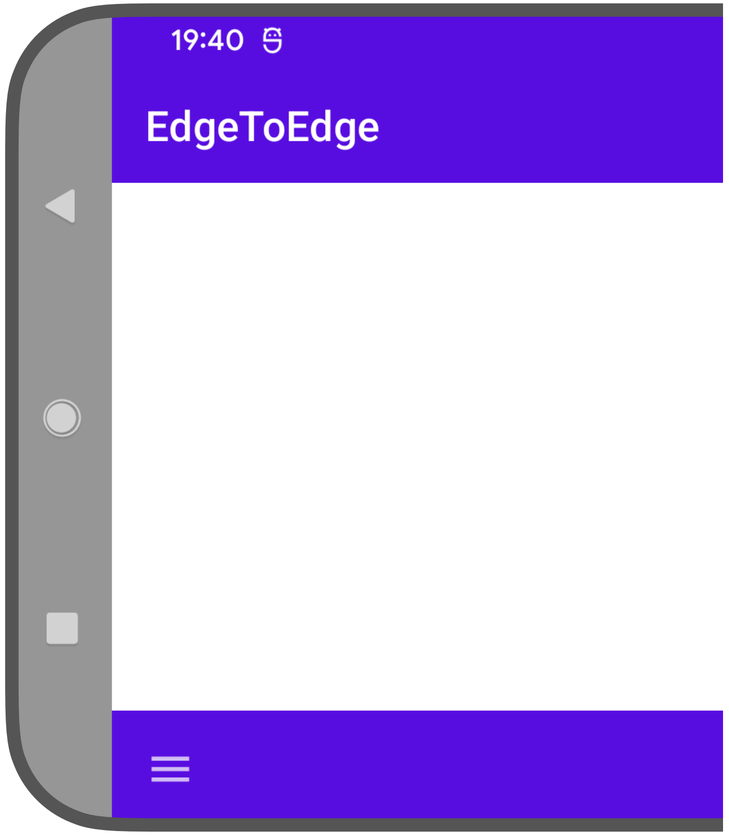
systemBar = statusBar + navigationBar

```
@Composable
fun EdgeToEdge() {
    val insets = WindowInsets.systemBars
    Scaffold(
        modifier = Modifier.windowInsetsPadding(
            insets.only(WindowInsetsSides.Horizontal),
        ),
        topBar = {
            TopAppBar(
                modifier = Modifier
                    .background(MaterialTheme.colors.primarySurface)
                    .windowInsetsPadding(
                        insets.only(WindowInsetsSides.Top),
                    ),
                /* ... */
            )
        },
        bottomBar = {
            BottomAppBar(
                modifier = Modifier
                    .background(MaterialTheme.colors.primarySurface)
                    .windowInsetsPadding(
                        insets.only(WindowInsetsSides.Bottom),
                    ),
                /* ... */
```

左右のpaddingを外側でつける

上のpaddingだけ

下のpaddingだけ



個別につけるとキレイ

```
@Composable
fun EdgeToEdge() {
    val insets = WindowInsets.systemBars
    Scaffold(
        topBar = {
            TopAppBar(
                modifier = Modifier
                    .background(MaterialTheme.colors.primarySurface)
                    .windowInsetsPadding(
                        insets.only(
                            WindowInsetsSides.Top
                                + WindowInsetsSides.Horizontal
                        ),
                    ),
                /* ... */
```

外側のpaddingは削除

上と左右にpaddingをつける

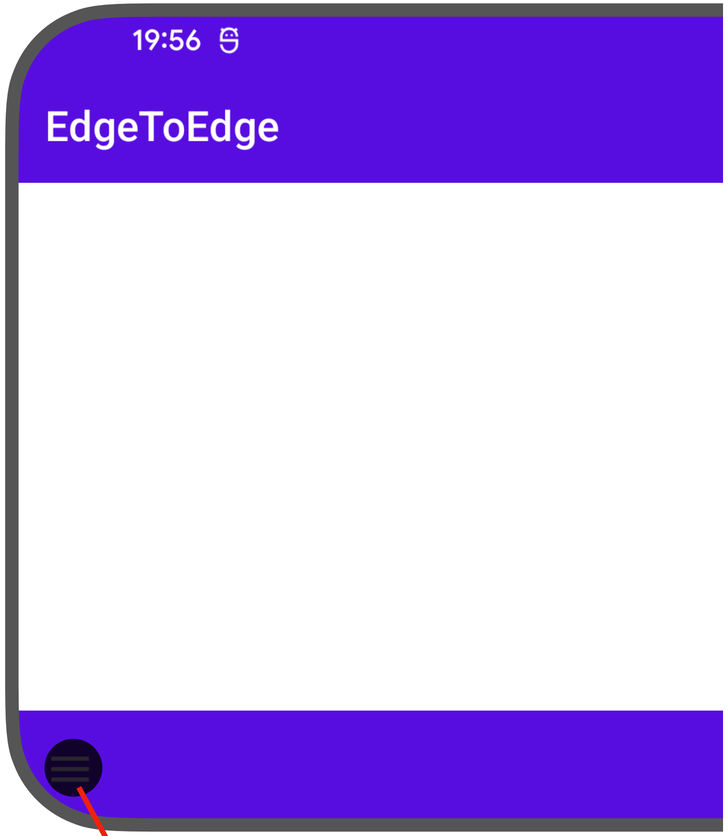
Display cutoutにも注意！

```
class EdgeToEdgeActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.P) {
            window.attributes.layoutInDisplayCutoutMode =
                LAYOUT_IN_DISPLAY_CUTOUT_MODE_SHORT_EDGES
        }
        /* ... */
    }
}
```

Android 9以降

Display Cutout領域にも描画する



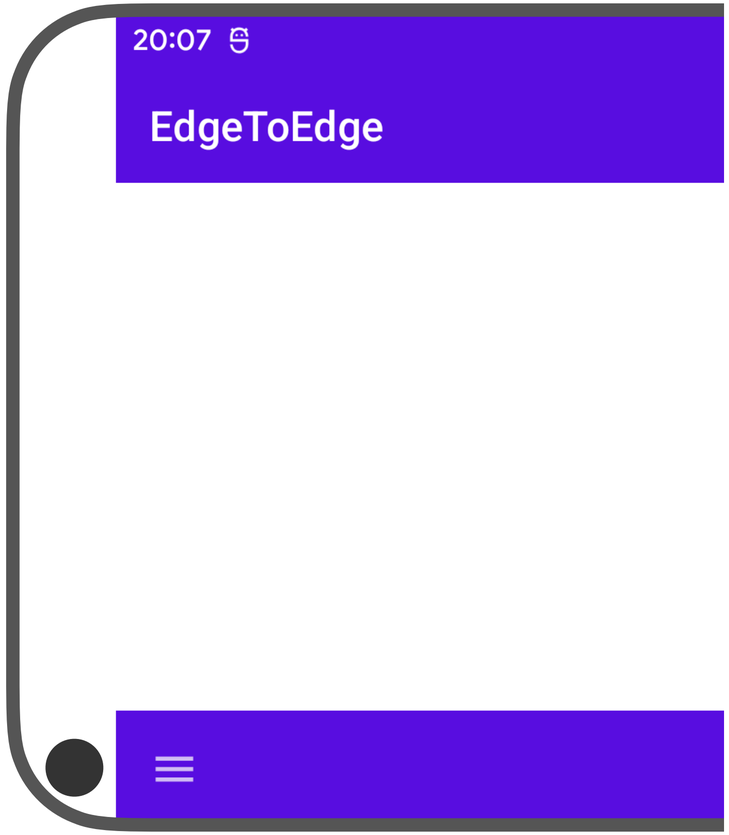
カメラと重なってしまう

Display cutoutを避ける

safeDrawing = systemBars + displayCutout + ime

imeは含みたくない場合

```
@Composable
fun EdgeToEdge() {
    val insets = WindowInsets.safeDrawing
    // or
    // val insets = WindowInsets.systemBars
    //     .union(WindowInsets.displayCutout)
    Scaffold(
        modifier = Modifier.windowInsetsPadding(
            insets.only(WindowInsetsSides.Horizontal),
        ),
        topBar = {
            /* ... */
```



3

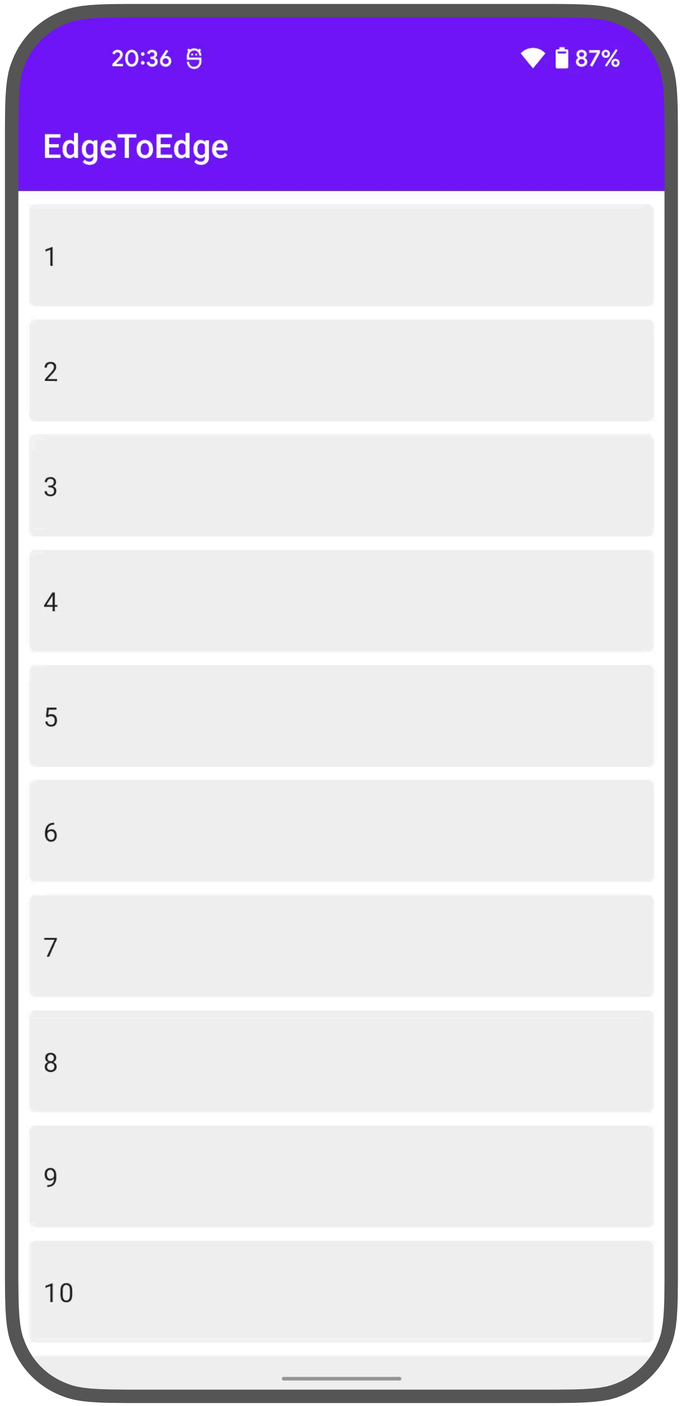
スクロールに対応する

CyberAgent

スクロールに対応する

最後までスクロールできるように
verticalScrollの後にpaddingを入れる

```
Column(
    modifier = Modifier
        .verticalScroll(rememberScrollState())
        .padding(8.dp)
        .windowInsetsPadding(
            WindowInsets.safeDrawing.only(WindowInsetsSides.Bottom),
        ),
    verticalArrangement = Arrangement.spacedBy(8.dp),
) {
    /* ... */
}
```

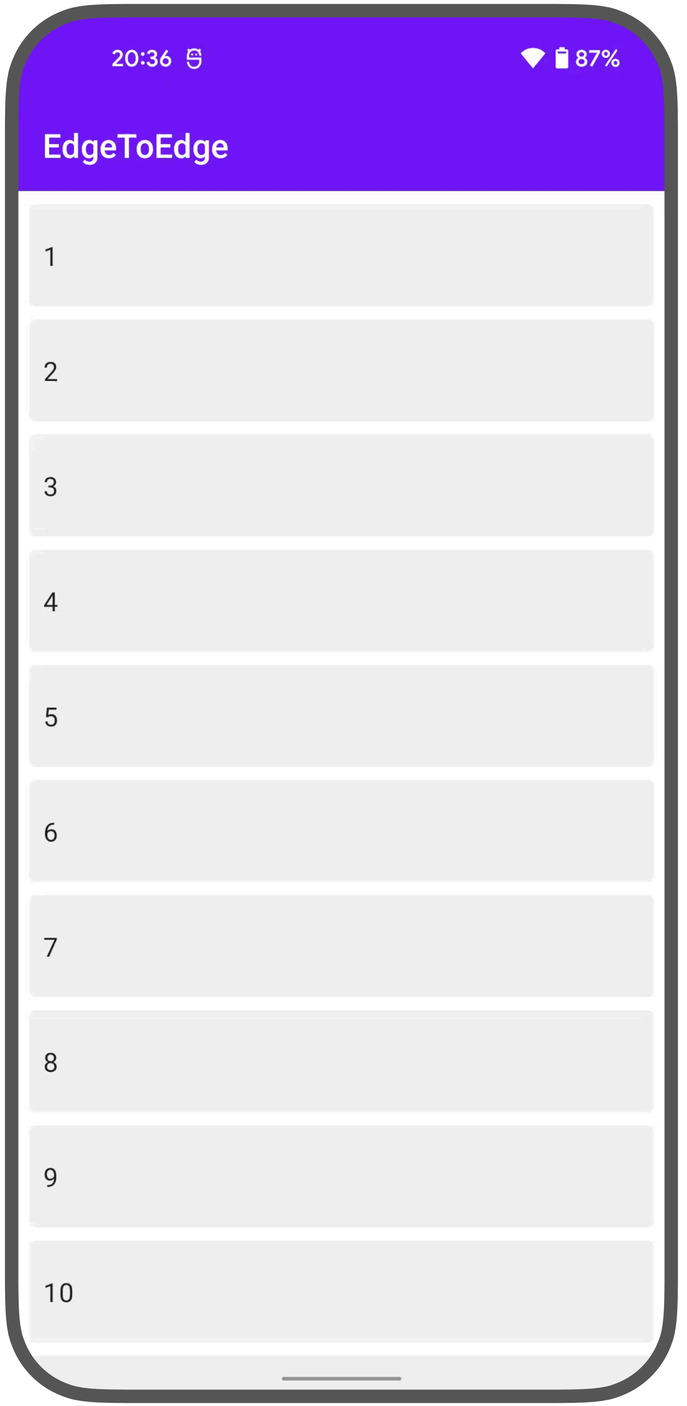


LazyColumnの場合

Bottomのサイズを取得

```
val bottomInset = with(LocalDensity.current) {
    WindowInsets.safeDrawing.getBottom(this).toDp()
}
LazyColumn(
    modifier = Modifier.fillMaxSize(),
    contentPadding = PaddingValues(
        top = 8.dp,
        bottom = 8.dp + bottomInset,
        start = 8.dp,
        end = 8.dp,
    ),
    verticalArrangement = Arrangement.spacedBy(8.dp),
) {
    /* ... */
}
```

contentPaddingに指定



4

ソフトウェア
キーボードを避ける

CyberAgent

キーボードを避ける

adjustResizeにしないと二重にpaddingがつく

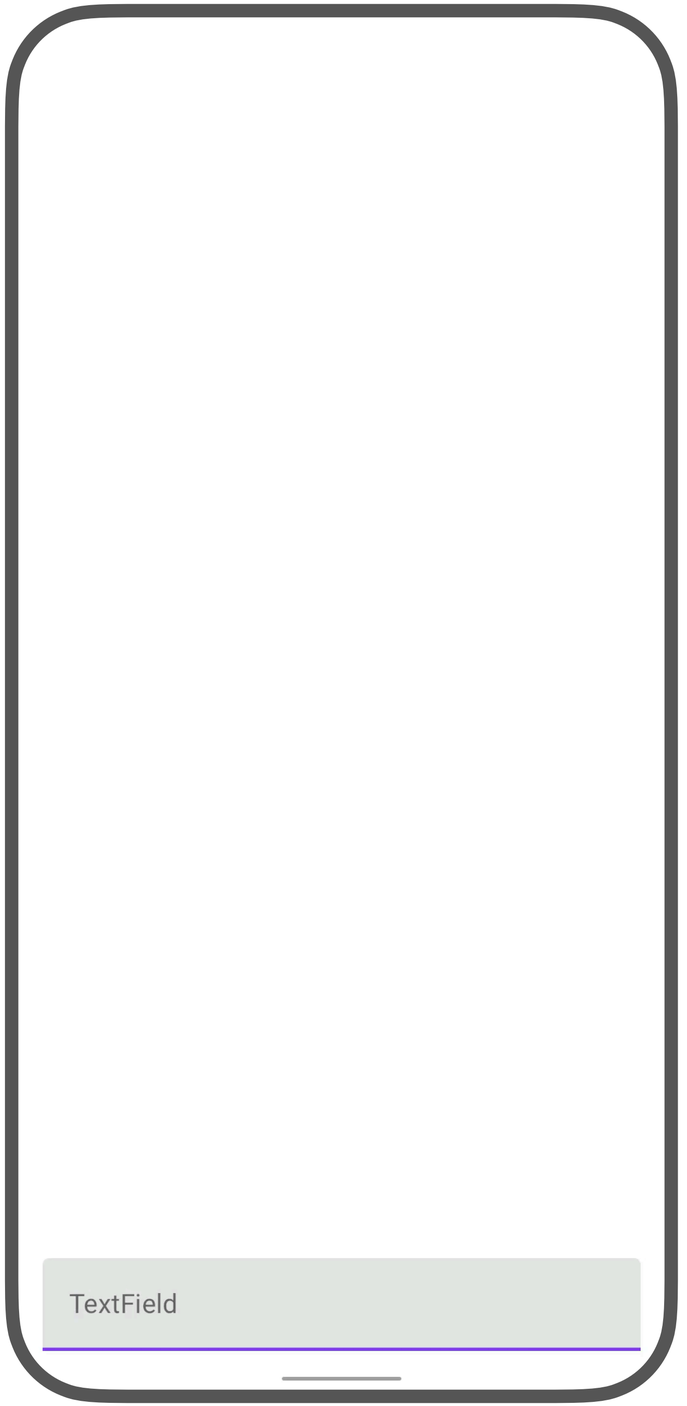
```
<activity
    android:name=".EdgeToEdgeActivity"
    android:windowSoftInputMode="adjustResize"
    android:theme="@style/Theme.MaterialComponents.DayNight.NoActionBar"
    android:exported="false" />
```

---

imePaddingはキーボード非表示のとき0になる

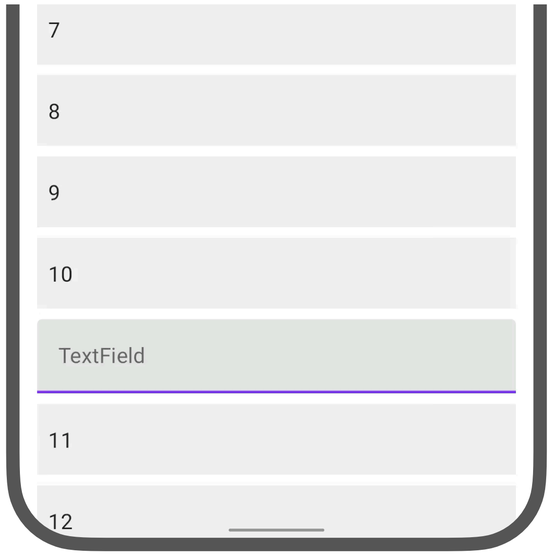
```
@Composable
fun EdgeToEdge() {
    var text by remember { mutableStateOf("") }
    Box(
        modifier = Modifier
            .fillMaxSize()
            .safeDrawingPadding()
            .padding(horizontal = 16.dp, vertical = 8.dp),
    ) {
        TextField(
            modifier = Modifier
                .align(Alignment.BottomCenter)
                .fillMaxWidth(),
            value = text,
            onValueChange = { text = it },
            placeholder = { Text(text = "TextField") },
        )
    }
}
```

Android 11以降は
アニメーションがつく



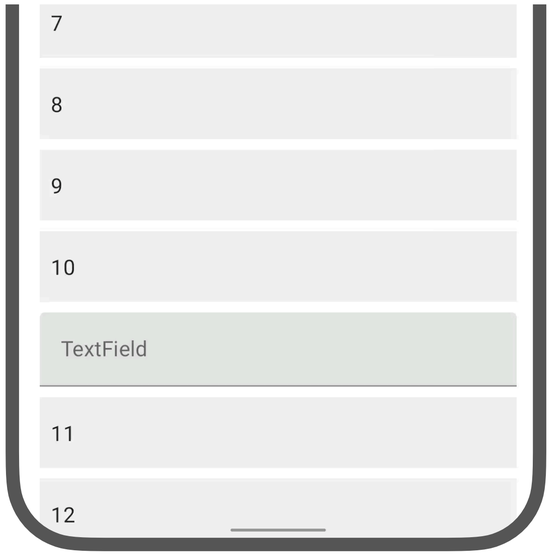
スクロール内のTextFieldに注意

Column + Modifier.verticalScroll



キーボードに隠れてしまう
Compose 1.3 alpha02で修正済み

LazyColumn



キーボードが閉じてしまう
IssueTracker：#179203700

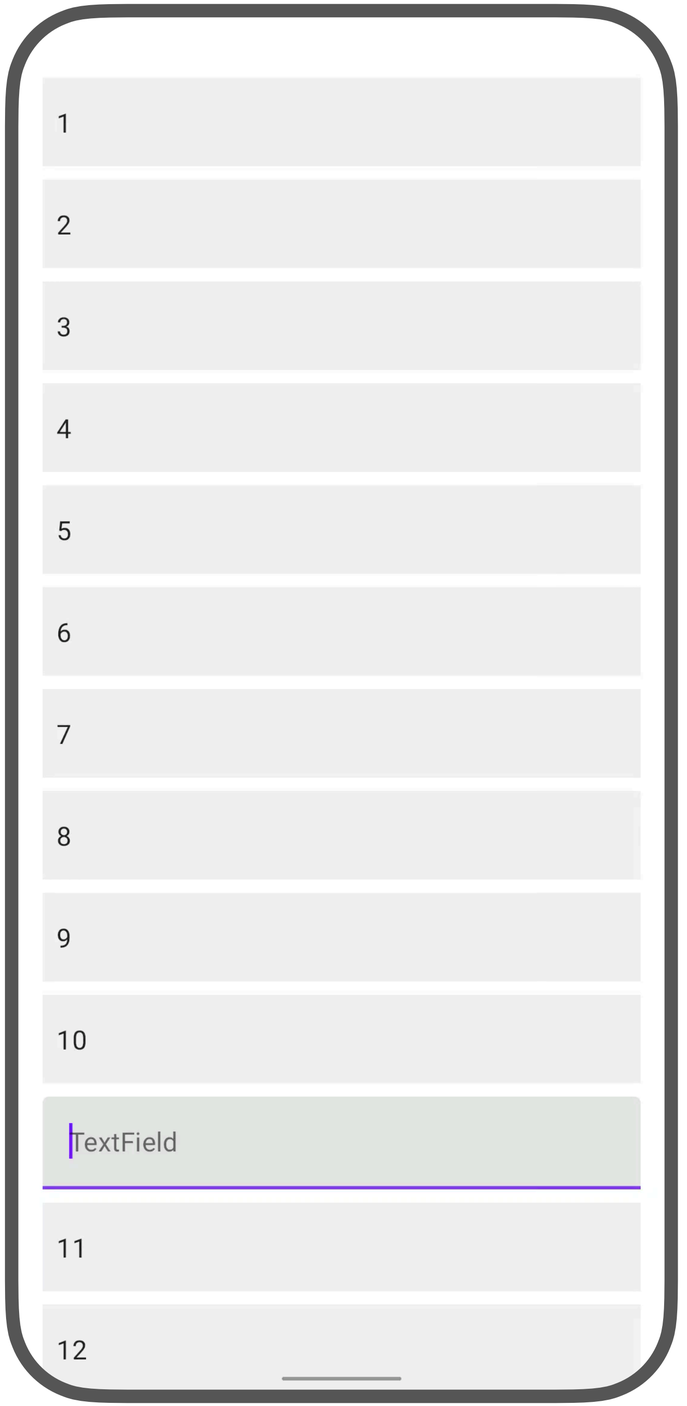
ime Insetsは外側につける

Compose 1.3 alpha 02以降

```
@Composable
fun EdgeToEdge() {
    var text by remember { mutableStateOf("") }
    Column(
        modifier = Modifier
            .fillMaxSize()
            .windowInsetsPadding(WindowInsets.ime)
            .verticalScroll(rememberScrollState())
            .windowInsetsPadding(WindowInsets.safeDrawing)
            .padding(
                horizontal = 16.dp,
                vertical = 8.dp
            ),
        verticalArrangement = Arrangement.spacedBy(8.dp),
    ) {
        /* ... */
```

ime insetsはマージンとして設定

ime分はconsumeされている



Modifier.imeNestedScroll

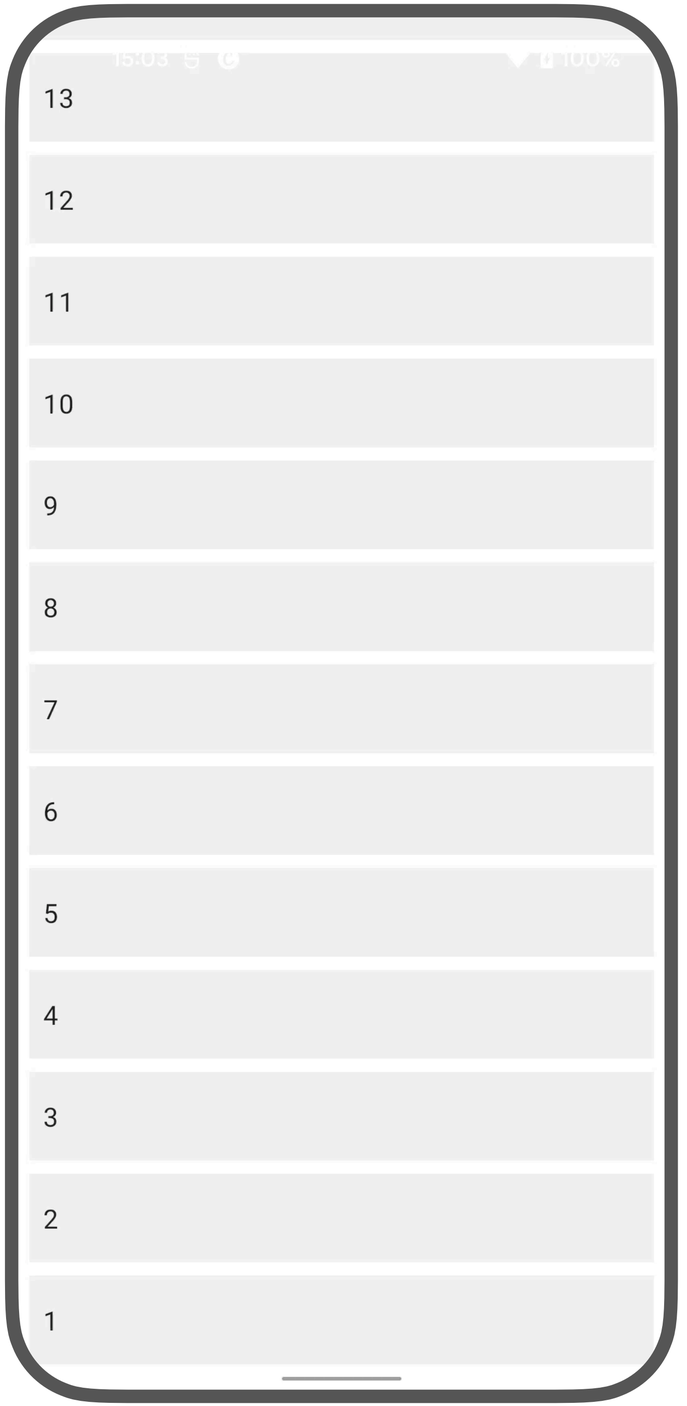
スクロールとキーボードを連動する
(android 11以降 / experimental)

```
@OptIn(ExperimentalLayoutApi::class)
@Composable
fun EdgeToEdge() {
    LazyColumn(
        modifier = Modifier
            .fillMaxSize()
            .imePadding()
            .imeNestedScroll(),
        verticalArrangement = Arrangement.spacedBy(8.dp),
        reverseLayout = true,
        contentPadding = PaddingValues(
            vertical = 16.dp,
            horizontal = 8.dp,
        ),
    ) {
        /* ... */
    }
}
```

OptInが必要

Nested scrollを指定

逆順にする(必須)



まとめ

- Jetpack Composeを使うと
  AndroidViewより比較的簡単に
  WindowInsetsを扱える
- 横画面やdisplay cutoutに注意
- ソフトウェアキーボードの
  操作も行える

余談：DialogやMaterial 3でWindowInsets対応が
進んでいるみたいです🎉