

OpenGLとNDK1.6

日本Androidの会 関西支部

宮川 彬央

本日のお題

- OpenGL 概要
- NDK概要
- サンプルアプリのカスタマイズ . . .

OpenGL 概要

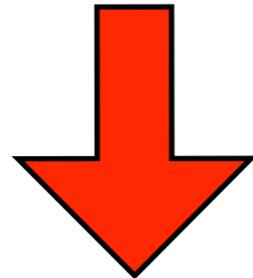
- 2D、3DグラフィックスAPI
- クロスプラットフォーム
- AndroidではOpenGL ES 1.0, 1.1の一部が対応

NDK 概要

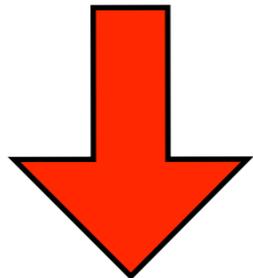
- Native Development Kitの略
- CやC++で実装したモジュールを作成できるツールキット
- NDKのみでアプリケーションを作ることはいできない
- 2009年9月にNDK1.6 r1が公開

アプリケーションとネイティブコード

アプリケーション(Java)

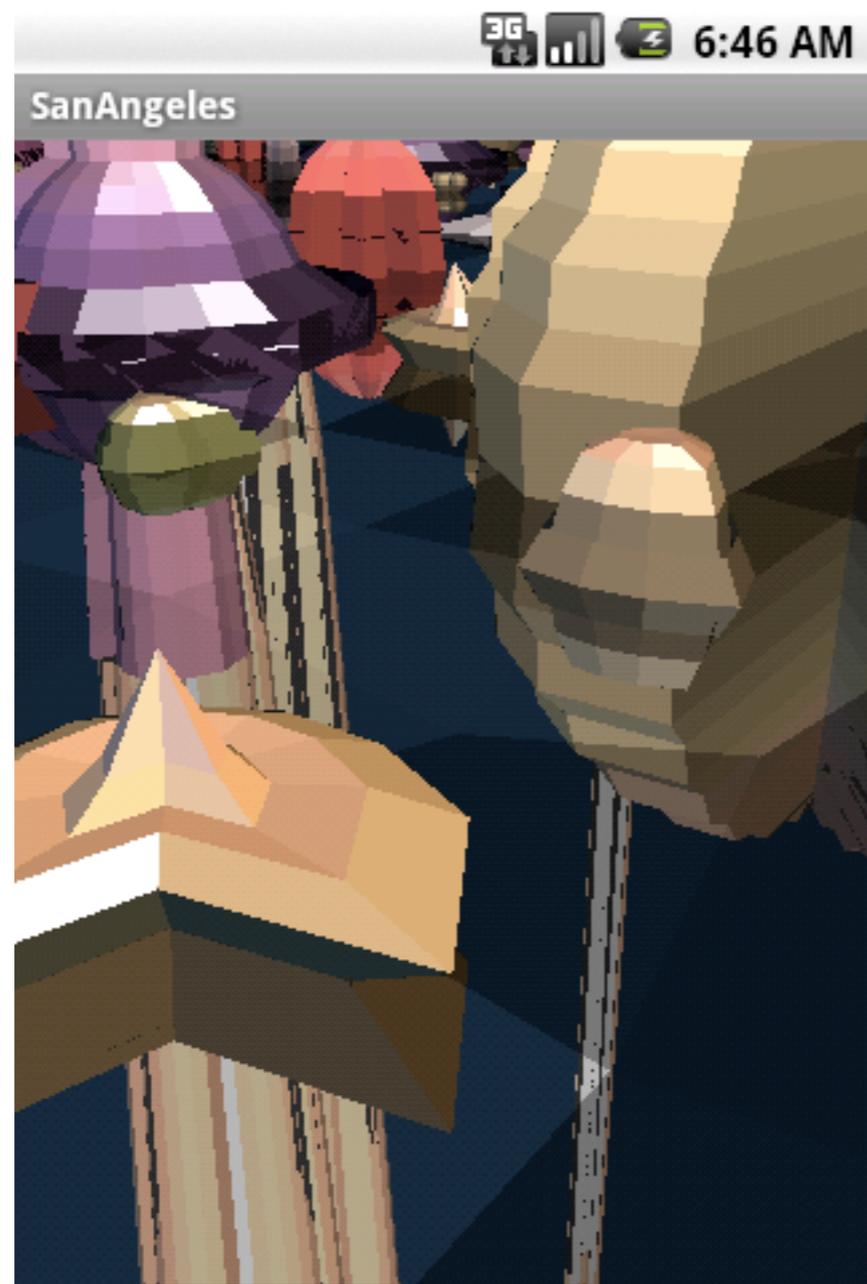


JNI(Java Native Interface)



C, C++ネイティブコード

サンプルアプリ



SanAngeles

- 頂点数 21642個
- ポリゴン数(三角形)
7214枚
- めちゃくちゃ遅い . . .

サンプルアプリの構造

- DemoActivity.java

ネイティブコードの呼び出し

```
static {  
    System.loadLibrary("sanangeles");  
}
```

```
/* 省略 */
```

```
public void onDrawFrame(GL10 gl) {  
    nativeRender();  
}
```

```
private static native void nativeInit();  
private static native void nativeResize(int w, int h);  
private static native void nativeRender();  
private static native void nativeDone();
```

サンプルアプリの構造

- app-android.c

ネイティブコードの実装

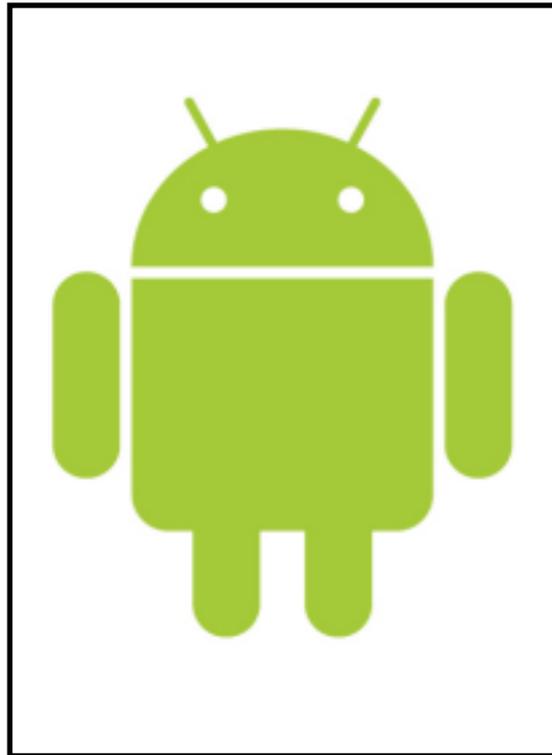
```
/* Call to initialize the graphics state */  
void  
Java_com_example_SanAngeles_DemoRenderer_nativeInit( JNIEnv* env )  
{
```

```
/* Call to render the next GL frame */  
void  
Java_com_example_SanAngeles_DemoRenderer_nativeRender( JNIEnv* env )  
{
```

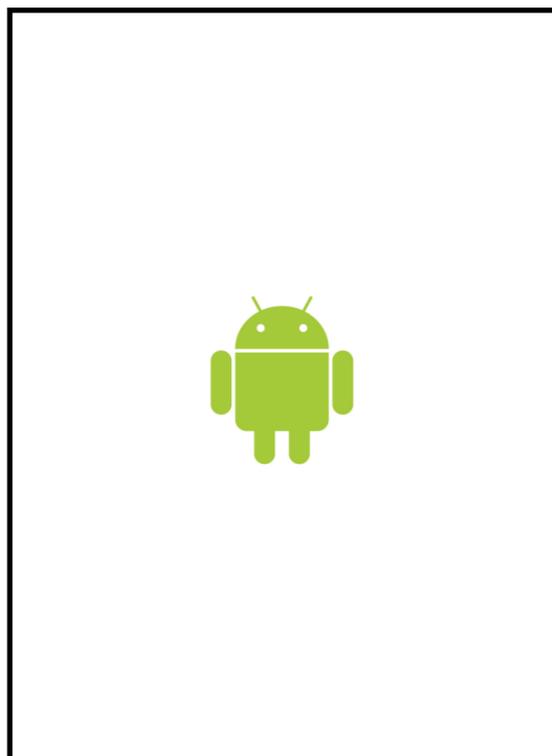
なぜこんなに遅いのか

- オブジェクトの数が多
- レンダリングに時間がかかる
- 手前のオブジェクトと奥のオブジェクトが同じポリゴン数

なぜこんなに遅いのか



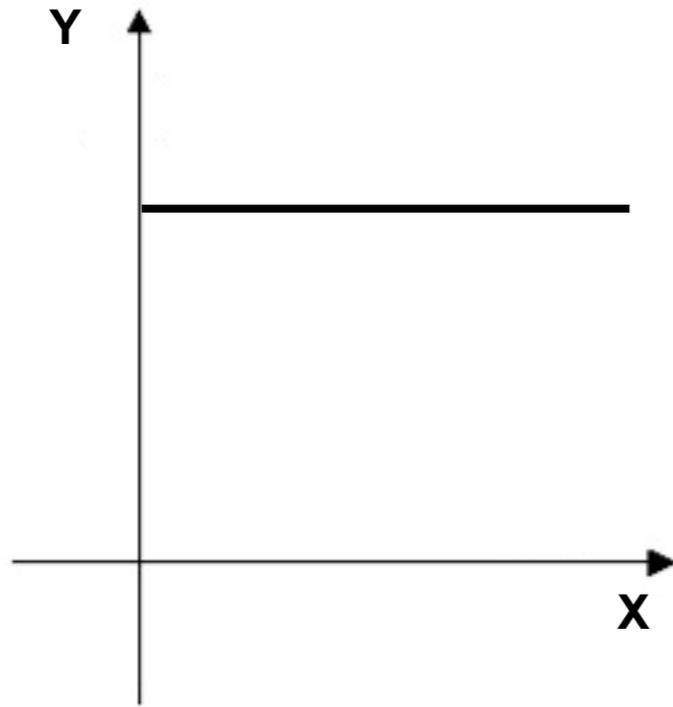
- カメラに近い(大きく表示)
- きれいに見せるには、ポリゴン数を増やす必要がある



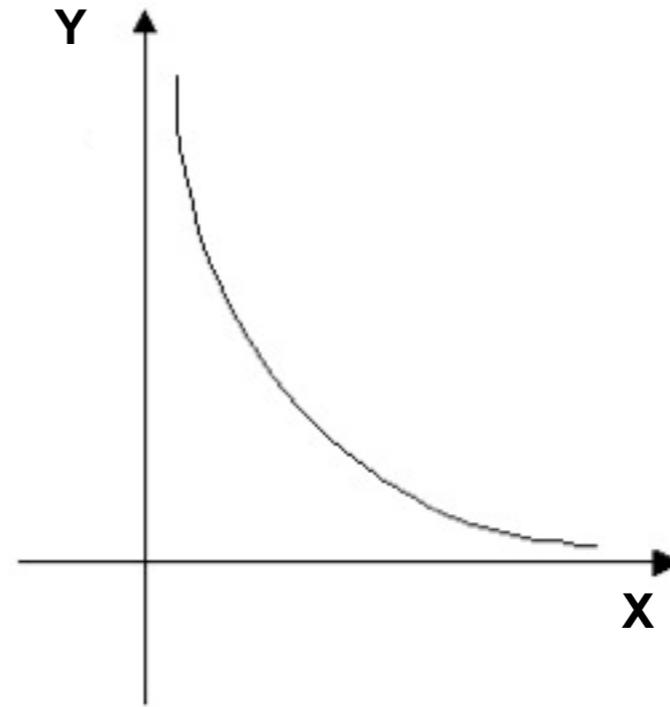
- カメラから遠い(小さく表示)
- 多少ポリゴン数が少なくても、荒がわからない
- ポリゴン数を減らして、速度UP

なぜこんなに遅いのか

(サンプルアプリ)



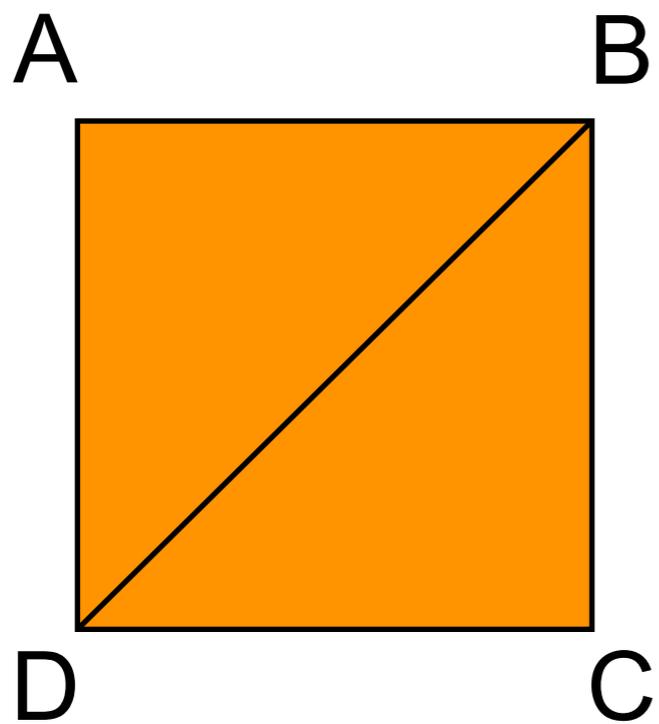
(理想)



- X: 視点からオブジェクトまでの距離
- Y: ポリゴン数

カスタマイズ その1

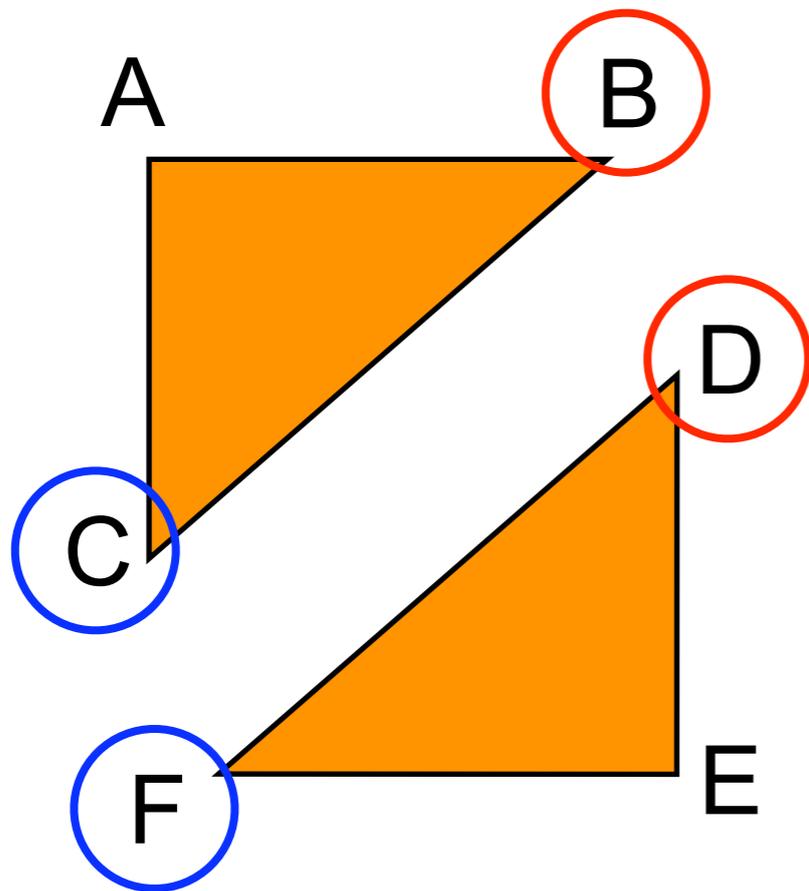
- 重複する頂点を除く
(例：四角形の場合)



4頂点あれば、
三角形を2枚描画
して四角形を描画
できるはず。

カスタマイズ その1

- 重複する頂点を除く
三角形を2枚・・・



純粹に三角形を
2枚用意すると、
6頂点必要。

重複分のD、Fが余分

BとDの座標は等しい
CとFの座標は等しい

カスタマイズ その1

- 重複する頂点を除く

地面以外の各建物(?)に対して、
重複する頂点を削除(約1500)。

結果 . . . 50ms~100ms改善

ほとんど変わらない orz

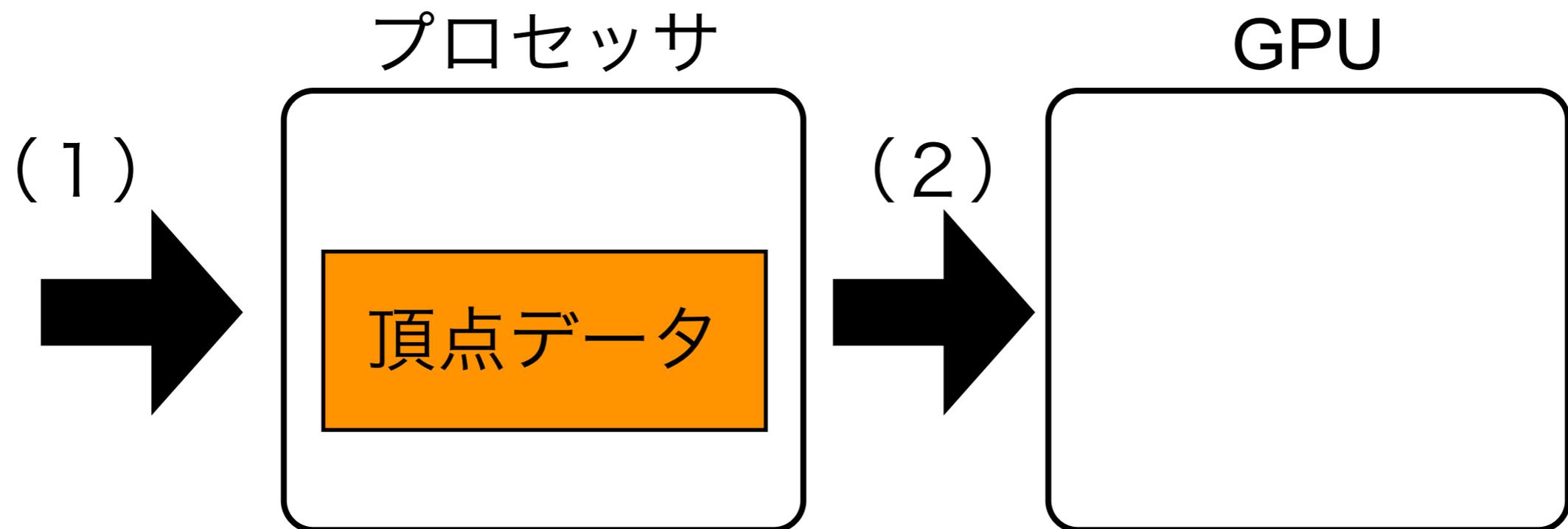
カスタマイズ その2

- VBOを使う
 - Vertex Buffer Object
(頂点バッファオブジェクト)
 - GPUのメモリ内にバッファを確保し、頂点データを格納
 - プロセッサからGPUへの転送コストが減る

カスタマイズ その2

- VBOを使う

頂点データの処理(VBO未使用)



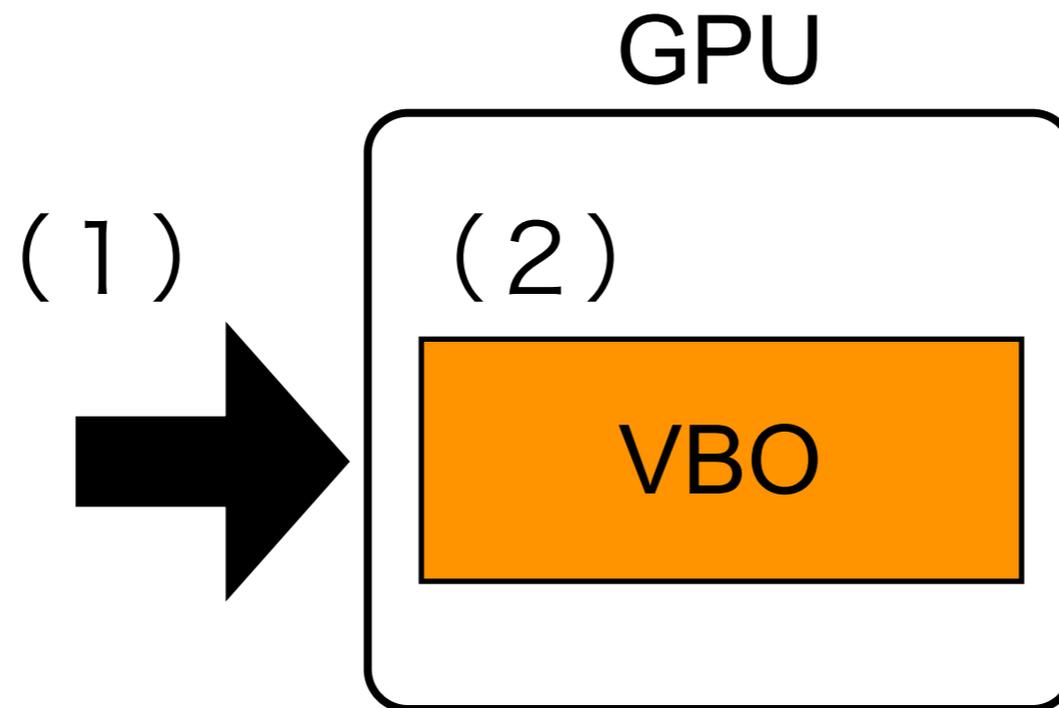
(1) 頂点データをプロセッサメモリに作成

(2) 描画コマンドの度に頂点データを転送

カスタマイズ その2

- VBOを使う

頂点データの処理(VBO使用)



- (1) 頂点データをGPUメモリ内のVBOに転送
- (2) 描画コマンドは、VBOの頂点データを用いて描画を行う

カスタマイズ その2

- VBOを使う

頂点配列、カラー配列、法線配列を
全部VBOに放り込んでみましたが・・・

結果・・・ほぼ変化なし

コーディングが誤ってるかも知れません。

カスタマイズまとめ

- 頂点の重複を除くことは有効
- VBOは要調査
- 必要最低限の機能のみ有効にする
- 表示するポリゴン数を減らす

補足

OpenGL ES概要

【京都GTUG 2009年8月】

Android Hackathon in Kyoto 事前勉強会
その2用資料

<http://sites.google.com/a/kyoto-gtug.org/home/docs>

ありがとうございました！

不明な点や質問等あれば、

【日本Androidの会 関西支部 宮川彬央】

miyagawa@android-group.jp

までどうぞ！

※自分もわからない場合は、

日本Androidの会のMLに流すかもしれません...