日本Androidの会

Android SDK WG第３回セッション

センサーで遊ぼう

2009/01/31

日本Androidの会

江川　崇

自己紹介

- 株式会社　豆蔵　所属
- 日本Androidの会　幹事
- Android SDK WGリーダー

突然ですが

席替えします

ごぞんじですか？

ごぞんじですか？

- SDKでセンサーの値がとれます

ごぞんじですか？

- SDKでセンサーの値がとれます
- センサーは、端末の目です

ごぞんじですか？

- SDKでセンサーの値がとれます
- センサーは、端末の目です
- センサーは、端末と外界を繋ぐ絆です

ごぞんじですか？

- SDKでセンサーの値がとれます
- センサーは、端末の目です
- センサーは、端末と外界を繋ぐ絆です
- センサーを知ると、端末と仲良くなれます

ごぞんじですか？

- SDKでセンサーの値がとれます
- センサーは、端末の目です
- センサーは、端末と外界を繋ぐ絆です
- センサーを知ると、端末と仲良くなれます
- 端末と仲良くなると、Android開発が楽しくなります

でも、エミュレータじゃねぇ‥

でも、エミュレータじゃねぇ‥

- なので、実機で遊びましょう

でも、エミュレータじゃねぇ‥

- なので、実機で遊びましょう
- 端末を提供してくれた人
  ありがとうございます m(__)m

加速度と傾度で遊びます

配ったアプリを動かしてみよう

何か値が出てますね

- [x] 上が加速度
- [x] 下が傾き

ですよ

みんなでやってみよう

- 端末を振っているときに、背景の色を変えてください
- 振るのを止めたら、背景色を元（黒）に戻してください
  - layout.setBackgroundColor(Color.任意の色);
- 端末の傾きに応じて、画面に文字を表示してください
  - orientationというTextViewを使ってください

縦

横

水平

センサーの基礎は

配った資料にあります

「Android Sensor 江川」

でググっても出てきます

ローパス（ハイカット）フィルター

- 低振動数成分のみ残すフィルター

- かんたんなローパスフィルターの例
  - 重力の影響だけが残る＝＞傾きがわかる

“前回のローパスフィルター処理した値の x %”
＋“生の（フィルター処理していない）値の 100 - x %”

ハイパス（ローカット）フィルター

- 高振動数成分のみ残すフィルター

- かんたんなハイパスフィルターの例
  - 重力の影響が取り除かれる
  - （瞬間的な）加速度がわかる

“生の（フィルター処理していない）値”
－“ローパスフィルタ処理済の値”

例

かたむき

```
currentOrientationValues[0]
                = values[0] * 0.1f + currentOrientationValues[0] * 0.9f;
currentOrientationValues[1]
                = values[1] * 0.1f + currentOrientationValues[1] * 0.9f;
currentOrientationValues[2]
                = values[2] * 0.1f + currentOrientationValues[2] * 0.9f;
```

かそくど

```
currentAccelerationValues[0] = values[0] - currentOrientationValues[0];
currentAccelerationValues[1] = values[1] - currentOrientationValues[1];
currentAccelerationValues[2] = values[2] - currentOrientationValues[2];
```

もっとヒント

```
private float[] currentOrientationValues = {0.0f, 0.0f, 0.0f};
private float[] currentAccelerationValues = {0.0f, 0.0f, 0.0f};
public void onSensorChanged(int sensor, float[] values) {
        switch(sensor) {
        case SensorManager.SENSOR_ACCELEROMETER:
                accelerometerValue.setText(convertFloatsToString(values));
                // 傾き（ハイカット）
                currentOrientationValues[0] = values[0] * 0.1f + currentOrientationValues[0] * (1.0f - 0.1f);
                currentOrientationValues[1] = values[1] * 0.1f + currentOrientationValues[1] * (1.0f - 0.1f);
                currentOrientationValues[2] = values[2] * 0.1f + currentOrientationValues[2] * (1.0f - 0.1f);
                // 加速度（ローカット）
                currentAccelerationValues[0] = values[0] - currentOrientationValues[0];
                currentAccelerationValues[1] = values[1] - currentOrientationValues[1];
                currentAccelerationValues[2] = values[2] - currentOrientationValues[2];
               filteredAccelerationValue.setText(convertFloatsToString(currentAccelerationValues));
               filteredOrientationValue.setText(convertFloatsToString(currentOrientationValues));
                // 振ってる？　絶対値（あるいは２乗の平方根）の合計がいくつ以上か？
                ・・・ここを実装！！
                // かたむきは？３つの絶対値（あるいは２乗の平方根）のうちどれがいちばんでかいか？
                ・・・ここを実装！！
                break;
        case SensorManager.SENSOR_ORIENTATION:
                orientationValue.setText(convertFloatsToString(values));
                break;
        default:
        }
}
```

蛇足

- コード中に

```
static DecimalFormat format;
static {
    format = new DecimalFormat();
    format.applyLocalizedPattern("#0.000");
}
```

- とありますが、これをもし

```
public void onSensorChanged(int sensor, float[] values) {
  switch(sensor) {
  case SensorManager.SENSOR_ACCELEROMETER:
   ・・・
```

- の中でやると遅すぎて死にます。

まとめ

- ほんきでやると、シロートには結構むずかしい
  - フーリエ変換とか座標系
  - 機種によって挙動やっぱり違うよね
- でも、まあ無理でもない
  - 3Dより楽、 物理が得意じゃなくても余裕
  - センサーを作るわけじゃない
    と仲良くなればそれでいい
  - 大抵の場合、色々と抜け道はある