

Android アプリレポート

「トレーニングカウンター」

理工情報生命学術院システム

情報工学研究群（博士前期課程）情報理工学位プログラム

202420606 小松崎摩耶

目次

1. アプリ開発の背景	3
2. アプリの設計.....	3
3. 実装	5
4. 動作例	14
5. 得られた効果	15

1. アプリ開発の背景

スクワット補助のアプリとしてスクワットの回数を数えるアプリ「トレーニングカウンター」を作成した。背景としては大学生の運動不足がある。大学生になると運動する機会がめっきり少なくなり、パソコンに向かって座り続ける日々が続いてしまっている人も多いだろう。これではいかんと何か運動を始めようというとき、スクワットを思いつく人はかなりいるはずだ。スクワットについて調べてみると、スクワットで効果を得るには正しい姿勢で行わなければならないとわかる。調べた結果をもとに正しい姿勢を心掛けてスクワットを試みるが、本当に正しい姿勢になっているのか自分で確認することは難しい。そこで正しいスクワットを補助するアプリを作成することにした。具体的には正しい姿勢でスクワットできた回数を数えることで、スクワットを補助することとした。正しいスクワットをするにはいくつかのポイントがあるが、すべてのポイントを抑えられるよう補助するのは難しかったため、今回は「太ももと地面が平行になるまで腰を落とす」に着目した。ただ太ももと地面が平行になるまでというのは初心者には難しいので、どの程度まで腰を落とせば1回としてカウントするのかをユーザ自身で設定可能にすることで、自分のレベルに合わせてアプリを利用できるようにした。

2. アプリの設計

「トレーニングカウンター」はスクワットの回数をカウントするアプリである。スマホを画面が内側になるよう外腿につけて利用することを想定しており、直立した状態からスクワットしたときのスマホの傾き具合（画面の法線を回転軸とする）をカウント判定に使用する。直立した状態でスマホは0度であり、太ももと地面が平行になった状態で90度である。よってしゃがんだ状態における角度を大きく設定するほどきついトレーニングとなる。また直立状態の角度も設定することもでき、20度など設定するとしゃがんだ後完全に直立するまで戻る必要がなく、多少易しいトレーニングになる。

アプリはメイン画面、カウント画面、設定画面からなり、メイン画面をハブにしてボタンによってページ遷移できる。

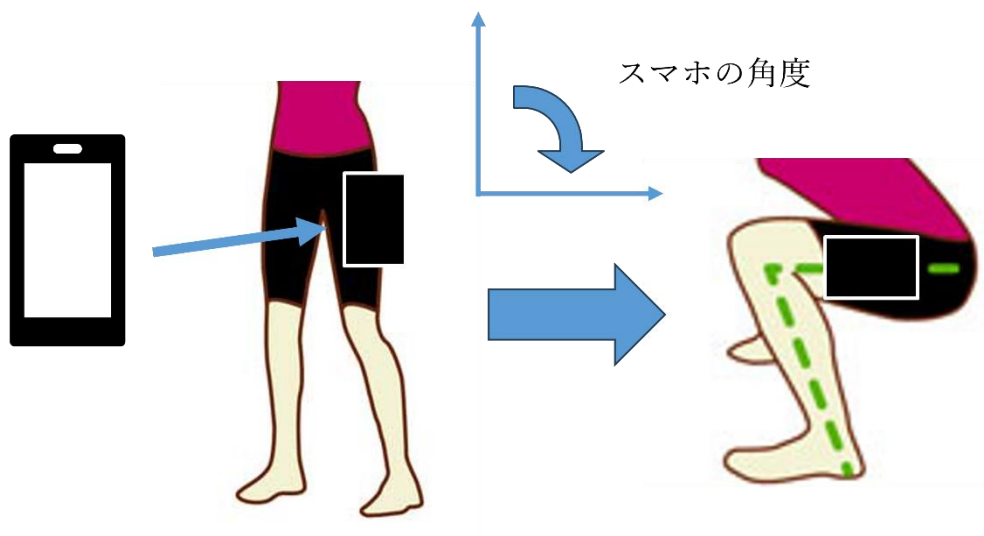


図 1 アプリの使い方とカウントの仕組み

- メイン画面

アプリ立ち上げ時に開かれる画面。Start ボタンを押すとカウント画面に遷移し、スクワットのカウントを始める。Setting ボタンを押すと設定画面に遷移し、各種設定ができる。Setting ボタンの下には現在の設定情報が表示されている。

- カウント画面

スクワットの回数をカウントする画面。上部に設定された内容に従ってカウントされたスクワットの回数が表示される。その下に位置する Reset ボタンを押すとカウントがリセットされる。また Finish ボタンを押すとカウント終了となりメイン画面に戻る。

- 設定画面

各種設定を行う画面。Position ではラジオボタンによってスマホを取り付ける足を選択できる。Degree ではカウント判定時に使用するスマホの傾きを任意に設定できる。OK ボタンを押すと設定を反映させメイン画面に戻る。Cancel ボタンを押すと設定を反映させないでメイン画面に戻る。

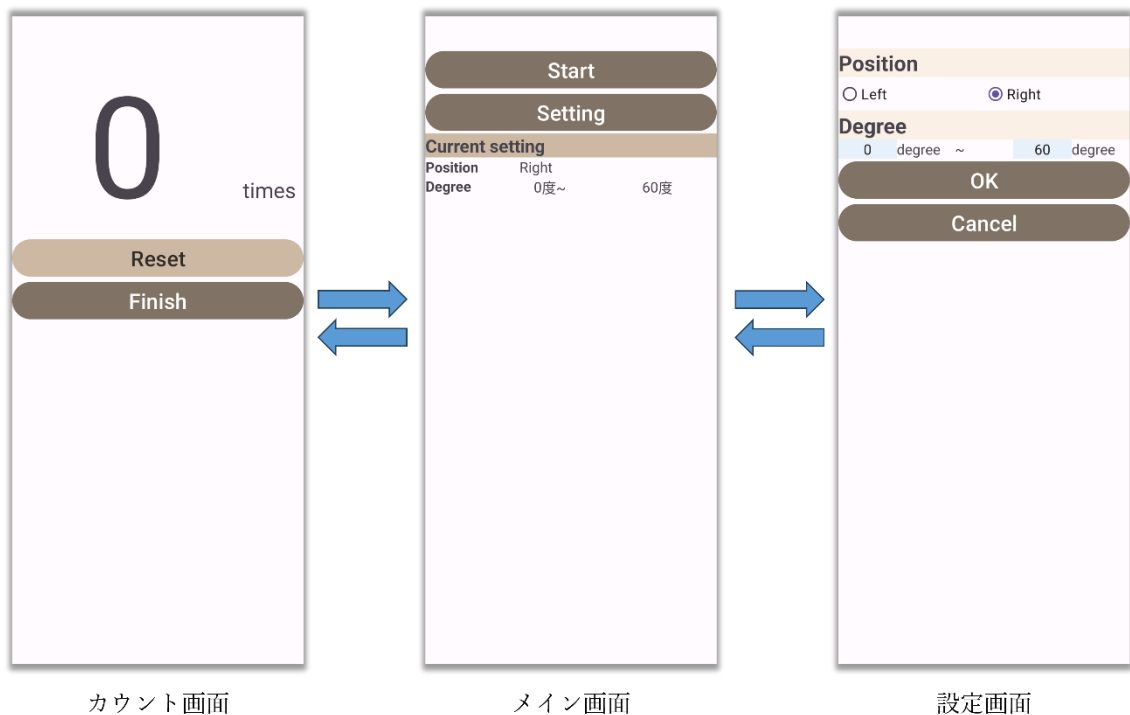


図 2 アプリ画面

3. 実装

メイン画面、カウント画面、設定画面それぞれで java ファイルを作成した。

- メイン画面 (MainActivity.java)

値の受け取り

getStringExtra によってメイン画面に遷移する前の画面から position (スマホを左右の足のどちらに着けるか)、fromDegree (直立した状態でのスマホの角度)、toDegree (しゃがんだ状態でもスマホの角度) の値を受け取る。

```
public class MainActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        ///
        Intent intent = getIntent();
        position = intent.getStringExtra("POSITION");
        if (position == null){
            position = "Right";
        }
    }
}
```

```

}
fromDegree = intent.getIntExtra("FROM_DEGREE", 0);
toDegree = intent.getIntExtra("TO_DEGREE", 60);
///
}
}

```

設定された値の表示

textView に対し setText を用いて各値をセットし画面に表示する。

```

public class MainActivity extends AppCompatActivity {
    ///
    private String position = "Right";
    private int fromDegree = 0;
    private int toDegree = 60;
    ///
    protected void onCreate(Bundle savedInstanceState) {
        ///
        TextView textView_position = findViewById(R.id.textView_position);
        TextView textView_fromDegree = findViewById(R.id.textView_fromDegree);
        TextView textView_toDegree = findViewById(R.id.textView_toDegree);

        textView_position.setText(position);
        textView_fromDegree.setText(String.valueOf(fromDegree));
        textView_toDegree.setText(String.valueOf(toDegree));
        ///
    }
}

```

ページ遷移

ボタンによってページ遷移を行う。Start ボタン (countActivitySwitchButton) を押すと、putExtra で設定値を渡すと同時にカウント画面 (CountActivity) に遷移する。Setting ボタン (settingActivitySwitchButton) を押すと、putExtra で設定値を渡すと同時に設定画面 (SettingActivity) に遷移する。

```

public class MainActivity extends AppCompatActivity {
protected void onCreate(Bundle savedInstanceState) {
    ///

    Button countActivitySwitchButton =
(Button) findViewById(R.id.button_start);

    countActivitySwitchButton.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View arg0) {
            Intent intent_count = new Intent(getApplicationContext(),
CountActivity.class);

            intent_count.putExtra("POSITION", position);
            intent_count.putExtra("FROM_DEGREE", fromDegree);
            intent_count.putExtra("TO_DEGREE", toDegree);
            startActivity(intent_count);
        }
    });

    Button settingActivitySwitchButton =
(Button) findViewById(R.id.button_setting);

    settingActivitySwitchButton.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View arg0) {
            Intent intent_setting = new
Intent(getApplicationContext(), SettingActivity.class);

            intent_setting.putExtra("POSITION", position);
            intent_setting.putExtra("FROM_DEGREE", fromDegree);
            intent_setting.putExtra("TO_DEGREE", toDegree);
            settingsActivityResultLauncher.launch(intent_setting);
        }
    });
}
    ///
}
}

```

- カウント画面 (CountActivity.java)

値の受け取り

getStringExtra で遷移元 (メイン画面) から値を受けとる。

```
public class CountActivity extends AppCompatActivity implements
SensorEventListener {
    ///
    private String position;
    private int fromDegree;
    private int toDegree;
    ///
    protected void onCreate(Bundle savedInstanceState) {
        ///
        Intent intent = getIntent();
        position = intent.getStringExtra("POSITION");
        fromDegree = intent.getIntExtra("FROM_DEGREE", 0);
        toDegree = intent.getIntExtra("TO_DEGREE", 60);
        ///
    }
}
```

角度センサー

スクワットのカウントに利用する角度センサーを加速度センサーと磁気センサーによって実装。

```
public class CountActivity extends AppCompatActivity implements
SensorEventListener {
    ///
    private float[] mAccelerationValue = new float[3];
    private float[] mGeoMagneticValue = new float[3];
    private float[] mOrientationValue = new float[3];
    private float[] mInRotationMatrix = new float[9];
    private float[] mOutRotationMatrix = new float[9];
    private float[] mInclinationMatrix = new float[9];
    ///
}
```



```

protected void onCreate(Bundle savedInstanceState) {
    ///
    SensorManager sensorManager = (SensorManager)
    getSystemService(SENSOR_SERVICE);
    Sensor accelerationSensor =
    sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    Sensor magneticSensor =
    sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    sensorManager.registerListener(this, accelerationSensor,
    SensorManager.SENSOR_DELAY_UI);
    sensorManager.registerListener(this, magneticSensor,
    SensorManager.SENSOR_DELAY_UI);
    mRollText = (TextView) findViewById(R.id.text_count);
    ///
}

@Override
public void onSensorChanged(SensorEvent sensorEvent) {
    switch(sensorEvent.sensor.getType()) {
        case Sensor.TYPE_MAGNETIC_FIELD: mGeoMagneticValue =

        sensorEvent.values.clone(); break;
        case Sensor.TYPE_ACCELEROMETER: mAccelerationValue =

        sensorEvent.values.clone(); break;
    }
    SensorManager.getRotationMatrix(mInRotationMatrix,
    mInclinationMatrix, mAccelerationValue, mGeoMagneticValue);
    SensorManager.remapCoordinateSystem(mInRotationMatrix,
    SensorManager.AXIS_X, SensorManager.AXIS_Z, mOutRotationMatrix);
    SensorManager.getOrientation(mOutRotationMatrix,
    mOrientationValue);
    ///
}
}

```

カウント

取得したスマホの角度 (roll) に従ってスクワットのカウントを行う。roll が toDegree よりも大きくなった場合にカウント数 (count) を 1 足す。また立った時に fromDegree に達しないままスクワットを続けようとする場合カウントしないようにするため、それを確認するフラッグ (countFlag) を用意した。countFlag=0 は一度 fromDegree まで直立したことを示しており、この状態でスクワットをするとカウント可能となる。また、スマホをつける足によってカウントに利用するスマホの角度の正負が逆になるので処理を分けている。最後に setText を用いて TextView にカウント数をセットし、画面に表示する機能を実装している。

```
public class CountActivity extends AppCompatActivity implements
SensorEventListener {
    ///
    private TextView mRollText;
    private int count = 0;
    private int countFlag = 0;
    private double roll = 0;
    ///
    @Override
    public void onSensorChanged(SensorEvent sensorEvent){
        ///
        roll = Math.toDegrees((double)mOrientationValue[2]);
        if (Objects.equals(position, "Right")){
            if (countFlag==0 && roll>=toDegree){
                count = count + 1;
                countFlag = 1;
            } else if (countFlag==1 && roll<=fromDegree) {
                countFlag = 0;
            }
        }else {
            if (countFlag==0 && roll<=-toDegree){
                count = count + 1;
                countFlag = 1;
            } else if (countFlag==1 && roll>=-fromDegree) {
                countFlag = 0;
            }
        }
    }
}
```

```

    }

    }

    String rollText = String.valueOf(count);
    mRollText.setText(rollText);

}

}

```

リセット

ボタンによってカウントのリセットを行う。Reset ボタン (buttonReset) を押すとカウントが 0 に戻る。

```

public class CountActivity extends AppCompatActivity implements
SensorEventListener {
protected void onCreate(Bundle savedInstanceState) {
///
Button buttonReset = (Button) findViewById(R.id.button_reset);
buttonReset.setOnClickListener(new View.OnClickListener() {
    public void onClick(View arg0) {
        count = 0;
    }
});
///
}
}

```

ページ遷移

ボタンによってページ遷移を行う。Finish ボタン (buttonFinish) を押すとメイン画面 (MainActivity) に遷移する。

```

public class CountActivity extends AppCompatActivity implements
SensorEventListener {
protected void onCreate(Bundle savedInstanceState) {
///

```

```

Button buttonFinish = (Button) findViewById(R.id.button_finish);
buttonFinish.setOnClickListener(new View.OnClickListener() {
    public void onClick(View arg0) {
        Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
        startActivity(intent);
    }
});
}
}

```

- 設定画面

値の受け取り

getStringExtra によってメイン画面から値を受け取る。

```

public class SettingActivity extends AppCompatActivity {
    ///
    private String position;
    private String fromDegree;
    private String toDegree;
    ///
    protected void onCreate(Bundle savedInstanceState) {
        ///
        Intent intent = getIntent();
        position = intent.getStringExtra("POSITION");
        fromDegree = intent.getStringExtra("FROM_DEGREE");
        toDegree = intent.getStringExtra("TO_DEGREE");
        ///
    }
}

```

値の入力

position は RadioButton によって、fromDegree と toDegree は EditText によってユーザーからの入力を行う。

```

RadioGroup position_radioGroup = (RadioGroup)
findViewById(R.id.radioButton_position);
EditText fromDegree_set = (EditText)
findViewById(R.id.editText_fromDegree);
EditText toDegree_set = (EditText)
findViewById(R.id.editText_toDegree);

```

値の受け渡し（入力値の反映）

ユーザの入力情報を取り出し、遷移先（MainActivity）に値を渡す。

```

public class SettingActivity extends AppCompatActivity {
protected void onCreate(Bundle savedInstanceState) {
    ///
    Button buttonOk = (Button) findViewById(R.id.button_ok);
    buttonOk.setOnClickListener(new View.OnClickListener() {
        public void onClick(View arg0) {
            Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
            position_set =
findViewById(position_radioGroup.getCheckedRadioButtonId());
            position = position_set.getText().toString();
            fromDegree = fromDegree_set.getText().toString();
            toDegree = toDegree_set.getText().toString();

            intent.putExtra("POSITION", position);
            intent.putExtra("FROM_DEGREE", Integer.parseInt(fromDegree));
            intent.putExtra("TO_DEGREE", Integer.parseInt(toDegree));

            setResult(RESULT_OK, intent);
            startActivity(intent);
        }
    });
    ///
}
}

```

キャンセル

ボタンによってページ遷移を行う。入力された値ではなく設定画面に遷移するときにメイン画面から受け取った値をそのままメイン画面に渡す。

```
public class SettingActivity extends AppCompatActivity {
protected void onCreate(Bundle savedInstanceState) {
    ///
    Button buttonCancel = (Button) findViewById(R.id.button_cancel);
    buttonCancel.setOnClickListener(new View.OnClickListener() {
        public void onClick(View arg0) {
            Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
            intent.putExtra("POSITION", position);
            intent.putExtra("FROM_DEGREE", fromDegree);
            intent.putExtra("TO_DEGREE", toDegree);
            startActivity(intent);
        }
    });
}
}
```

4. 動作例

アプリの動作例を図3に示す。

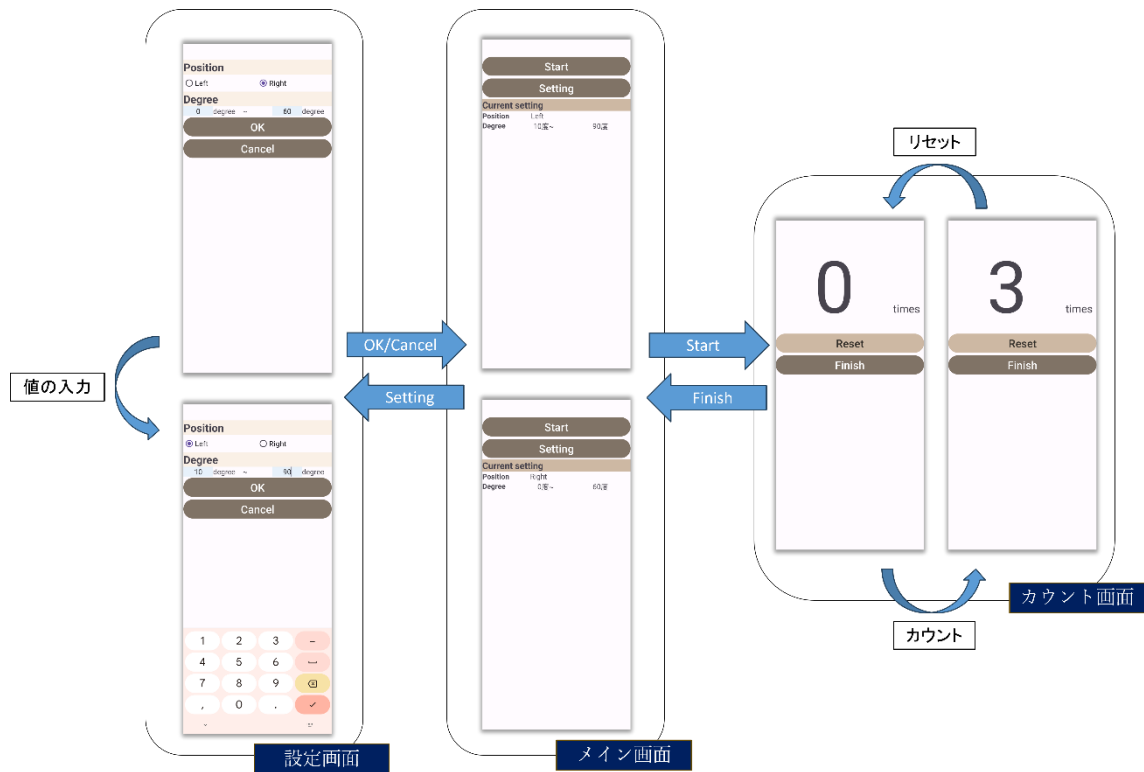


図 3 動作例

5. 得られた効果

追加したい機能がいくつもあったが時間が足りず実装できなかった。例えば、これは実装したものうまくいかなかった機能だが、カウントしたときにカウントができたことを知らせる効果音になる機能をつけたかった。コードを書いて、音源もフォルダに追加して、スマホにアプリをいれて動作させたのだが、他の機能は問題なく動作するにもかかわらず、効果音はならなかった。エラーが吐き出されなかったのでスマホ側に問題があったのだろうか。またもう一つ追加したかった機能はスクワット以外の運動のカウントだ。アプリ名を「トレーニングカウンター」としていたのは、腕立て伏せや懸垂なども同じようにカウントできたらと思ったからだ。しかしスマホをうまく腕に装着できずアプリの動作確認ができそうになかったため今回はあきらめた。運動時にスマホを足に着けるバンドのような道具も持っていなかったが、腿の側面にポケットがついているズボンを持っていたのでスクワットの回数はカウントすることができた。他にも回数を記録する機能や時間を計る機能などもつけたかったが時間が足りなくなってしまい最小限の機能しかつける

ことができなかった。難しかったのは Android で提供してくれる機能を把握して自分のアプリに組み込むことだ。結局講義で学んだ機能以外ではページ間の値の受け渡しくらいしか新たに使うことができなかった。Android Studio を使うのは初めてだったからというのは言い訳になってしまうが、やりたいことをやるのにどういったクラスを利用すればいいのかなどを把握しきれなかった。アプリ開発の経験が少なすぎるためだと思う。回数を重ねて慣れなければならないと感じた。

参考

- ・ 美 WISE. (2018). <全 20 種徹底解説>正しいスクワットのやり方と注意点. ダイエットなら美 WISE ! . [HTTPS://START-DIET.COM/%E6%AD%A3%E3%81%97%E3%81%84%E3%82%B9%E3%82%AF%E3%83%AF%E3%83%83%E3%83%88%E3%81%AE%E3%82%84%E3%82%8A%E6%96%B9/](https://start-diet.com/%E6%AD%A3%E3%81%97%E3%81%84%E3%82%B9%E3%82%AF%E3%83%AF%E3%83%83%E3%83%88%E3%81%AE%E3%82%84%E3%82%8A%E6%96%B9/) (2024/5/19)