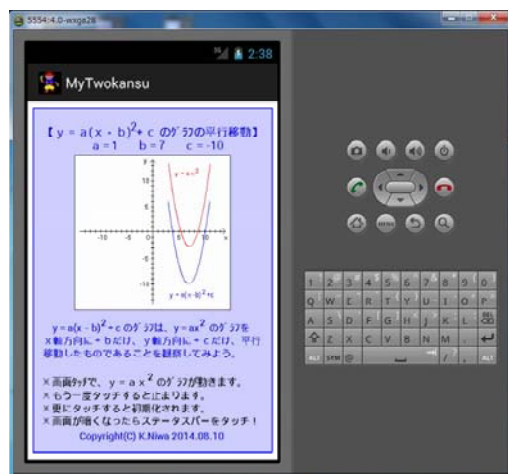


```
//-----
//
//      2次関数のグラフの平行移動
//       $y = a(x - b)^2 + c$  のグラフ
//      Ver3
//      Copyright (C) K.Niwa 2014. 08. 10
//-----
```



【エミュレータ画面例】

【スマートフォン画面例】
(Android 4.0)

【アプリの概要】

$y = a(x - b)^2 + c$ のグラフは、 $y = x^2$ のグラフを x 軸方向に $+b$ だけ、 y 軸方向に $+c$ だけ、平行移動したものであることを観察してみよう。また、この2つのグラフは形も大きさも同じで、位置だけが異なったものであることに注意して見てみよう。

【1】 Twokansu. java

```
package jp.seitoku.twokansu;

import android.content.Context; //ライブラリを記述
import android.graphics.*;
import android.util.AttributeSet;
import android.view.*;

//Viewクラスを継承したTwokansu1クラス
public class Twokansu extends View {

    int flag=0; //グラフの移動(1)、グラフの停止(2)、グラフの初期化(0) 初期化識別子
    double x,y; //グラフ描き用
    int px,py,oldpx,oldpy; //グラフ描き用
    int fg; //グラフ描き用
    double a=0,b=0; //グラフ描き用
    static int n1,n2,n3;

    public Twokansu(Context context, AttributeSet attrs, int defStyle) {
//コンストラクタ(クラス名と同じメソッドで最初記述)
        super(context, attrs, defStyle);
    }

    public Twokansu(Context context, AttributeSet attrs) { //コンストラクタ(クラス名と同じメソッドで最初記述)
        super(context, attrs);
    }
}
```

```

public Twokansu(Context context) { //インスタンス(クラス名)メソッドで最初書ける
    super(context);
}

//onDrawメソッド-----
@Override
protected void onDraw(Canvas canvas) {

    super.onDraw(canvas);
    canvas.drawColor(Color.WHITE);
    Paint paint = new Paint();
    paint.setColor(Color.BLUE);
    paint.setAlpha(50);

    canvas.drawRect((getWidth()/2-240)+10,(getHeight()/2-343)+10,(getWidth()/2-240)+470,(getHeight()
/2-343)+675,paint);
    paint.setAlpha(10000);
    paint.setColor(Color.BLUE);

    //額縁付与
    for (int i=0;i<2;i++) {

        canvas.drawLine((getWidth()/2-240)+10+i,(getHeight()/2-343)+10+i,(getWidth()/2-240)+10+i,(getHei
ght()/2-343)+675-i,paint);

        canvas.drawLine((getWidth()/2-240)+10+i,(getHeight()/2-343)+675-i,(getWidth()/2-240)+470-i,(getH
eight()/2-343)+675-i,paint);

        canvas.drawLine((getWidth()/2-240)+470-i,(getHeight()/2-343)+675-i,(getWidth()/2-240)+470-i,(get
Height()/2-343)+10+i,paint);

        canvas.drawLine((getWidth()/2-240)+470-i,(getHeight()/2-343)+10+i,(getWidth()/2-240)+10+i,(getHe
ight()/2-343)+10+i,paint);
    }

    //裏紙の描画
    paint.setColor(Color.BLACK);
    canvas.drawRect((getWidth()/2-240)+90,
(getHeight()/2-343)+100,(getWidth()/2-240)+390,(getHeight()/2-343)+400,paint);
    paint.setColor(Color.WHITE);
    canvas.drawRect((getWidth()/2-240)+91,
(getHeight()/2-343)+101,(getWidth()/2-240)+389,(getHeight()/2-343)+399,paint);

    //座標の描画
    paint.setColor(Color.BLACK);
    canvas.drawLine((getWidth()/2-240)+100,
(getHeight()/2-343)+250,(getWidth()/2-240)+380,(getHeight()/2-343)+250,paint);
    canvas.drawLine((getWidth()/2-240)+380,
(getHeight()/2-343)+250,(getWidth()/2-240)+375,(getHeight()/2-343)+245,paint);
    canvas.drawLine((getWidth()/2-240)+380,
(getHeight()/2-343)+250,(getWidth()/2-240)+375,(getHeight()/2-343)+255,paint);
    canvas.drawLine((getWidth()/2-240)+240,
(getHeight()/2-343)+110,(getWidth()/2-240)+240,(getHeight()/2-343)+390,paint);
    canvas.drawLine((getWidth()/2-240)+240,
(getHeight()/2-343)+110,(getWidth()/2-240)+245,(getHeight()/2-343)+115,paint);
    canvas.drawLine((getWidth()/2-240)+240,
(getHeight()/2-343)+110,(getWidth()/2-240)+235,(getHeight()/2-343)+115,paint);

    //x軸の描画
    for (int xx=110;xx<380;xx=xx+10) {
        canvas.drawLine((getWidth()/2-240)+xx,
(getHeight()/2-343)+247,(getWidth()/2-240)+xx,(getHeight()/2-343)+253,paint);
    }

    //y軸の描画
    for (int yy=120;yy<390;yy=yy+10) {
        canvas.drawLine((getWidth()/2-240)+237,

```

```

(getHeight()/2-343)+yy,(getWidth()/2-240)+243,(getHeight()/2-343)+yy, paint);
    }

    //x軸の描画
    canvas.drawText("5", (getWidth()/2-240)+287, (getHeight()/2-343)+265, paint);
    canvas.drawText("10", (getWidth()/2-240)+334, (getHeight()/2-343)+265, paint);
    canvas.drawText("x", (getWidth()/2-240)+375, (getHeight()/2-343)+265, paint);
    canvas.drawText("0", (getWidth()/2-240)+227, (getHeight()/2-343)+265, paint);
    canvas.drawText("-5", (getWidth()/2-240)+187, (getHeight()/2-343)+265, paint);
    canvas.drawText("-10", (getWidth()/2-240)+134, (getHeight()/2-343)+265, paint);

    //y軸の描画
    canvas.drawText("5", (getWidth()/2-240)+225, (getHeight()/2-343)+205, paint);
    canvas.drawText("10", (getWidth()/2-240)+220, (getHeight()/2-343)+155, paint);
    canvas.drawText("y", (getWidth()/2-240)+220, (getHeight()/2-343)+115, paint);
    canvas.drawText("-5", (getWidth()/2-240)+220, (getHeight()/2-343)+305, paint);
    canvas.drawText("-10", (getWidth()/2-240)+215, (getHeight()/2-343)+355, paint);

    //y=ax^2 の方の描画/
    paint.setColor(Color.RED);

    if(TwokansuActivity.n1 < 0){
        canvas.drawText("y=ax ", (getWidth()/2-240)+280, (getHeight()/2-343)+350,
paint);
        canvas.drawText("2", (getWidth()/2-240)+322, (getHeight()/2-343)+345,
paint);
    }
    else if(TwokansuActivity.n1 > 0){
        canvas.drawText("y=ax ", (getWidth()/2-240)+280, (getHeight()/2-343)+140,
paint);
        canvas.drawText("2", (getWidth()/2-240)+322, (getHeight()/2-343)+135,
paint);
    }

    fg=0;

    for (x=-4+a;x<=4+a;x=x+0.01) {
        y=TwokansuActivity.n1*(x-a)*(x-a)+b;

        px=(int)(240+10*x);
        py=(int)(250-10*y);
        if (fg==0) {
            if ((getHeight()/2-343)+py>(getHeight()/2-343)+110 &&
(getHeight()/2-343)+py<(getHeight()/2-343)+390){
                canvas.drawLine((int)(getWidth()/2-240)+px,
(int)(getHeight()/2-343)+py,(int)(getWidth()/2-240)+px,(int)(getHeight()/2-343)+py,paint);
            }
            else {
                if ((getHeight()/2-343)+py>(getHeight()/2-343)+110 &&
(getHeight()/2-343)+py<(getHeight()/2-343)+390){
                    canvas.drawLine((int)(getWidth()/2-240)+oldpx,
(int)(getHeight()/2-343)+oldpy,(int)(getWidth()/2-240)+px,(int)(getHeight()/2-343)+py,paint);
                }
            }
            oldpx=px;oldpy=py;
            fg++;
        }
    }

    //y=a(x-b)^2+c の方の描画
    paint.setColor(Color.BLUE);

    if(TwokansuActivity.n1 > 0 && TwokansuActivity.n2 > 0){
        canvas.drawText("y=a(x-b)      +c", (getWidth()/2-240)+270,
(getHeight()/2-343)+375, paint);
        canvas.drawText("2", (getWidth()/2-240)+335, (getHeight()/2-343)+370,
paint);
    }
}

```

```

else if(TwokansuActivity.n1 > 0 && TwokansuActivity.n2 < 0){
    canvas.drawText("y = a(x-b) + c", (getWidth()/2-240)+120,
(getHeight()/2-343)+375, paint);
    canvas.drawText("2", (getWidth()/2-240)+185, (getHeight()/2-343)+370,
paint);
}
else if(TwokansuActivity.n1 < 0 && TwokansuActivity.n2 < 0){
    canvas.drawText("y = a(x-b) + c", (getWidth()/2-240)+120,
(getHeight()/2-343)+135, paint);
    canvas.drawText("2", (getWidth()/2-240)+185, (getHeight()/2-343)+130,
paint);
}
else if(TwokansuActivity.n1 < 0 && TwokansuActivity.n2 > 0){
    canvas.drawText("y = a(x-b) + c", (getWidth()/2-240)+270,
(getHeight()/2-343)+135, paint);
    canvas.drawText("2", (getWidth()/2-240)+335, (getHeight()/2-343)+130,
paint);
}
}
fg=0;

for (x=TwokansuActivity.n2-4;x<=TwokansuActivity.n2+4;x=x+0.01) {
y=TwokansuActivity.n1*(x-TwokansuActivity.n2)*(x-TwokansuActivity.n2)+TwokansuActivity.n3;
px=(int)(240+10*x);
py=(int)(250-10*y);
if (fg==0) {
    if((getHeight()/2-343)+py>(getHeight()/2-343)+110 &&
(getHeight()/2-343)+py<(getHeight()/2-343)+390){
        canvas.drawLine((int)(getWidth()/2-240)+px,
(int)(getHeight()/2-343)+py,(int)(getWidth()/2-240)+px,(int)(getHeight()/2-343)+py,paint);
    }
}
else {
    if((getHeight()/2-343)+py>(getHeight()/2-343)+110 &&
(getHeight()/2-343)+py<(getHeight()/2-343)+390){
        canvas.drawLine((int)(getWidth()/2-240)+oldpx,
(int)(getHeight()/2-343)+oldpy,(int)(getWidth()/2-240)+px,(int)(getHeight()/2-343)+py,paint);
    }
}
oldpx=px;oldpy=py;
fg++;
}

// 真線描画
paint.setColor(Color.BLACK);
canvas.drawLine((getWidth()/2-240)+90,
(getHeight()/2-343)+100,(getWidth()/2-240)+90,(getHeight()/2-343)+400,paint);
canvas.drawLine((getWidth()/2-240)+90,
(getHeight()/2-343)+100,(getWidth()/2-240)+390,(getHeight()/2-343)+100,paint);

// 表の表示
paint.setColor(Color.BLUE);
paint.setTextSize(22.0f);
canvas.drawText("", (getWidth()/2-240)+60, (getHeight()/2-343)+65, paint);
canvas.drawText("【y = a(x-b) + c のグラフの平行移動】", (getWidth()/2-240)+30,
(getHeight()/2-343)+65, paint);
canvas.drawText("2", (getWidth()/2-240)+195+2, (getHeight()/2-343)+55, paint);

// 目標提示
paint.setColor(Color.BLUE);
paint.setTextSize(19.0f);
canvas.drawText("y = a(x-b) + c のグラフは y = ax のグラフを", (getWidth()/2-240)+48,
(getHeight()/2-343)+440, paint);
canvas.drawText("2", (getWidth()/2-240)+141, (getHeight()/2-343)+430, paint);
canvas.drawText("2", (getWidth()/2-240)+331, (getHeight()/2-343)+430, paint);
canvas.drawText("x軸方向にcだけ、y軸方向に+cだけ、平行", (getWidth()/2-240)+30,
(getHeight()/2-343)+465, paint);

```

```

        canvas.drawText("  移動したものであることを観察してみよう。", (getWidth()/2-240)+30,
(getHeight()/2-343)+490, paint);

//a,b,cの値表示
paint.setColor(Color.BLUE);
paint.setTextSize(21.0f);
    canvas.drawText("a="+TwokansuActivity.n1, (getWidth()/2-240)+120,
(getHeight()/2-343)+90, paint);
    canvas.drawText("b="+TwokansuActivity.n2, (getWidth()/2-240)+210,
(getHeight()/2-343)+90, paint);
    canvas.drawText("c="+TwokansuActivity.n3, (getWidth()/2-240)+300,
(getHeight()/2-343)+90, paint);

//説明表示
paint.setColor(Color.BLACK);
paint.setTextSize(19.0f);
    canvas.drawText("※ 画面タッチで、y=ax のグラフが動きます。", (getWidth()/2-240)+30,
(getHeight()/2-343)+545, paint);
    canvas.drawText("2", (getWidth()/2-240)+231, (getHeight()/2-343)+535, paint);
    canvas.drawText("※ もう一度タッチすると止まります。", (getWidth()/2-240)+30,
(getHeight()/2-343)+570, paint);
    canvas.drawText("※ 更にタッチすると初期化されます。", (getWidth()/2-240)+30,
(getHeight()/2-343)+595, paint);
    canvas.drawText("※ 画面が暗なったらステータスバーをタッチ!", (getWidth()/2-240)+30,
(getHeight()/2-343)+620, paint);
    paint.setColor(Color.BLUE);
    paint.setTextSize(19.0f);
    canvas.drawText("Copyright(C) K.Niwa 2014.08.10", (getWidth()/2-240)+100,
(getHeight()/2-343)+650, paint);
//著作権表示

//調節のとき、グラフが平行移動
if (flag==1) {
    if (TwokansuActivity.n2<=0 && TwokansuActivity.n3<=0){
        if (a>=TwokansuActivity.n2) {
            a=a-0.1;
        }
        else if (b>=TwokansuActivity.n3) {
            b=b-0.1;
        }
    }
    else if (TwokansuActivity.n2>=0 && TwokansuActivity.n3>=0){
        if (a<=TwokansuActivity.n2) {
            a=a+0.1;
        }
        else if (b<=TwokansuActivity.n3) {
            b=b+0.1;
        }
    }
    else if (TwokansuActivity.n2>=0 && TwokansuActivity.n3<=0){
        if (a<=TwokansuActivity.n2) {
            a=a+0.1;
        }
        else if (b>=TwokansuActivity.n3) {
            b=b-0.1;
        }
    }
    else if (TwokansuActivity.n2<=0 && TwokansuActivity.n3>=0){
        if (a>=TwokansuActivity.n2) {
            a=a-0.1;
        }
        else if (b<=TwokansuActivity.n3) {
            b=b+0.1;
        }
    }
}

invalidate(); //再描画(clear & goto onDraw) して、この行戻てる。
//invalidate()は onDrawメソッドの中は記述できる。
//invalidate()は 繰返処理に用いる。

```

//もちろん onTouchEventメソッドの中は記述できる。

```
}  
//認識したとき、グラフが停止する  
else if (flag==2) {  
    //何変を加えない  
}  
//認識したとき、グラフを初期化する(元の位置に戻す)  
else if (flag==0) {  
    a=0;    //初期化する  
    b=0;    //初期化する  
    invalidate(); //再描画する(clear & goto onDraw) して、この行戻ってくる。  
}  
}
```

```
}//protected void onDraw(Canvas canvas)
```

//画面タッチしたときのイベント処理-----

```
-----  
@Override  
public boolean onTouchEvent(MotionEvent event) {  
  
    flag=flag+1;    //flagに1を加える  
    flag=flag % 3;    //flagに1、2、0を代入する  
  
    invalidate(); //再描画する(clear & goto onDraw)  
    return false;  
}  
}
```

[2] main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:id="@+id/LinearLayout1"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" >  
  
    <jp.seitoku.twokansu.Twokansu  
        android:layout_height="match_parent"  
        android:layout_width="match_parent"  
        android:id="@+id/myview1">  
    </jp.seitoku.twokansu.Twokansu>  
  
</LinearLayout>
```

[3] TwokansuActivity.java

```
package jp.seitoku.twokansu;  
  
//ライブラリを読み込む  
import android.app.Activity;  
import android.os.Bundle;  
import android.text.Editable;  
import android.widget.LinearLayout;  
import android.view.*;  
import android.view.View.OnClickListener;  
import android.widget.*;  
  
//MyActivityクラスはActivityクラスを継承する
```

```

public class TwokansuActivity extends Activity {
    TextView text0; //TextView型宣言
        TextView text1; //TextView型宣言
        EditText edit1; //EditText型宣言
        TextView text2; //TextView型宣言
        EditText edit2; //EditText型宣言
        TextView text3; //TextView型宣言
        EditText edit3; //EditText型宣言
        Button button1; //Button型宣言
        static int n1; //y=a(x-b)^2+c のa の値
        static int n2; //y=a(x-b)^2+c のb の値
        static int n3; //y=a(x-b)^2+c のc の値
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LinearLayout layout = new LinearLayout(this);
        layout.setOrientation(LinearLayout.VERTICAL);
        text0 = new TextView(this);
        text0.setText("2次関数 y=a(x-b)^2+c にして");
        layout.addView(text0);
        text1 = new TextView(this);
        text1.setText("整数を入力! (ただし 3 ≤ a ≠ 0)");
        layout.addView(text1);
        edit1 = new EditText(this);
        layout.addView(edit1);
        text2 = new TextView(this);
        text2.setText("整数を入力! (ただし -7 ≤ ≤)");
        layout.addView(text2);
        edit2 = new EditText(this);
        layout.addView(edit2);
        text3 = new TextView(this);
        text3.setText("整数を入力! (ただし -10 ≤ ≤ 0)");
        layout.addView(text3);
        edit3 = new EditText(this);
        layout.addView(edit3);
        button1 = new Button(this);
        button1.setText("click");
        layout.addView(button1);
        button1.setOnClickListener(new MyBtnClickAdapter());
        setContentView(layout);
    }
    class MyBtnClickAdapter implements OnClickListener {
        public void onClick(View view) {
            Editable ed1 = edit1.getText();
            Editable ed2 = edit2.getText();
            Editable ed3 = edit3.getText();

```

```

try {
    n1 = Integer.parseInt(ed1.toString());
    if (n1 < -3 || n1 > 3) {
        n1 = 1;
    }
} catch (NumberFormatException e) {
    n1 = 0;
}

try {
    n2 = Integer.parseInt(ed2.toString());
    if (n2 < -7 || n2 > 7) {
        n2 = 0;
    }
} catch (NumberFormatException e) {
    n2 = 0;
}

try {
    n3 = Integer.parseInt(ed3.toString());
    if (n3 < -10 || n3 > 10) {
        n3 = 0;
    }
} catch (NumberFormatException e) {
    n3 = 0;
}

setContentView(R.layout.main); //resフォルダのlayoutフォルダのmain.xmlに移動
}
}
}

```