

```
[1]MyTwokansuEng.java
```

```
/*
```

```
-----  
    2次関数のグラフの平行移動(英語版)  
    Android 4.1 (Jelly Bean)  
    Copyright (C) K. Niwa 2021. 10. 3  
-----
```

```
*/
```

```
package jp.kiyo.wuena.mytwokansueng;
```

```
import android.content.Context;
```

```
import android.graphics.Canvas;
```

```
import android.graphics.Color;
```

```
import android.graphics.Paint;
```

```
import android.graphics.Rect;
```

```
import android.graphics.RectF;
```

```
import android.util.AttributeSet;
```

```
import android.view.MotionEvent;
```

```
import android.view.View;
```

```
import android.graphics.*;
```

```
import android.util.AttributeSet;
```

```
import android.view.*;
```

```
//View クラスを継承した Twokan1 クラス
```

```
public class MyTwokansuEng 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;    //グラフ描写に利用
```

```
    int px,py,oldpx,oldpy; //グラフ描写に利用
```

```
    int fg;    //グラフ描写に利用
```

```
    double a=0,b=0;    //グラフ描写に利用
```

```
    static int n1,n2,n3;
```

```
    public MyTwokansuEng(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に読まれる)

```
    super(context, attrs, defStyle);  
}
```

public MyTwokansuEng(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に読まれる)

```
    super(context, attrs);  
}
```

public MyTwokansuEng(Context context) { //コンストラクタ (クラス名と同じメソッドで最初に読まれる)

```
    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()/2-360)+20, (getHeight()/2-600)+10, (getWidth()/2-360)+700, (getHeight()/2-600)+1190, paint);
```

```
    paint.setAlpha(10000);
```

```
    paint.setColor(Color.BLUE);
```

//額縁を付ける

```
for (int i=0;i<2;i++) {
```

```
    canvas.drawLine((getWidth()/2-360)+20+i, (getHeight()/2-600)+10+i, (getWidth()/2-360)+20+i, (getHeight()/2-600)+1190-i, paint);
```

```
    canvas.drawLine((getWidth()/2-360)+20+i, (getHeight()/2-600)+1190-i, (getWidth()/2-360)+700-i, (getHeight()/2-600)+1190-i, paint);
```

```
    canvas.drawLine((getWidth()/2-360)+700-i, (getHeight()/2-600)+1190-i, (getWidth()/2-
```

```

360)+700-i, (getHeight()/2-600)+10+i, paint);
        canvas.drawLine((getWidth()/2-360)+700-i, (getHeight()/2-600)+10+i, (getWidth()/2-
360)+20+i, (getHeight()/2-600)+10+i, paint);
    }

    //実験枠の描画
    paint.setColor(Color.BLACK);
    canvas.drawRect((getWidth()/2-360)+70, (getHeight()/2-600)+100, (getWidth()/2-
360)+650, (getHeight()/2-600)+680, paint);
    paint.setColor(Color.WHITE);
    canvas.drawRect((getWidth()/2-360)+71, (getHeight()/2-600)+101, (getWidth()/2-
360)+649, (getHeight()/2-600)+679, paint);

    //座標軸の描画
    paint.setColor(Color.BLACK);
    canvas.drawLine((getWidth()/2-360)+80, (getHeight()/2-600)+390, (getWidth()/2-
360)+640, (getHeight()/2-600)+390, paint);
    canvas.drawLine((getWidth()/2-360)+640, (getHeight()/2-600)+390, (getWidth()/2-
360)+635, (getHeight()/2-600)+385, paint);
    canvas.drawLine((getWidth()/2-360)+640, (getHeight()/2-600)+390, (getWidth()/2-
360)+635, (getHeight()/2-600)+395, paint);
    canvas.drawLine((getWidth()/2-360)+360, (getHeight()/2-600)+110, (getWidth()/2-
360)+360, (getHeight()/2-600)+670, paint);
    canvas.drawLine((getWidth()/2-360)+360, (getHeight()/2-600)+110, (getWidth()/2-
360)+355, (getHeight()/2-600)+115, paint);
    canvas.drawLine((getWidth()/2-360)+360, (getHeight()/2-600)+110, (getWidth()/2-
360)+365, (getHeight()/2-600)+115, paint);

    //x軸メモリの描写
    for (int xx=100;xx<630;xx=xx+20) {
        canvas.drawLine((getWidth()/2-360)+xx, (getHeight()/2-600)+387, (getWidth()/2-
360)+xx, (getHeight()/2-600)+393, paint);
    }

    //y軸メモリの描写
    for (int yy=130;yy<660;yy=yy+20) {

```

```

        canvas.drawLine((getWidth()/2-360)+357, (getHeight()/2-600)+yy, (getWidth()/2-
360)+363, (getHeight()/2-600)+yy, paint);
    }

    paint.setTextSize(20.0f);

    //x 軸メモリの描写
    canvas.drawText("5", (getWidth()/2-360)+457, (getHeight()/2-600)+395+20, paint);
    canvas.drawText("10", (getWidth()/2-360)+554, (getHeight()/2-600)+395+20, paint);
    canvas.drawText("x", (getWidth()/2-360)+630-10, (getHeight()/2-600)+395+20, paint);
    canvas.drawText("0", (getWidth()/2-360)+340, (getHeight()/2-600)+395+20, paint);
    canvas.drawText("-5", (getWidth()/2-360)+247, (getHeight()/2-600)+395+20, paint);
    canvas.drawText("-10", (getWidth()/2-360)+144, (getHeight()/2-600)+395+20, paint);

    //y 軸メモリの描写
    canvas.drawText("5", (getWidth()/2-360)+340, (getHeight()/2-600)+300, paint);
    canvas.drawText("10", (getWidth()/2-360)+330, (getHeight()/2-600)+200, paint);
    canvas.drawText("y", (getWidth()/2-360)+340-10, (getHeight()/2-600)+115+20-10,
paint);
    canvas.drawText("-5", (getWidth()/2-360)+330, (getHeight()/2-600)+500, paint);
    canvas.drawText("-10", (getWidth()/2-360)+320, (getHeight()/2-600)+600, paint);

    //y=ax^2 のグラフの描画//
    paint.setTextSize(30.0f);
    paint.setColor(Color.RED);

    if(MainActivity.n1 < 0) {
        canvas.drawText("y=ax", (getWidth()/2-360)+450, (getHeight()/2-600)+630,
paint);
        canvas.drawText("2", (getWidth()/2-360)+450+100, (getHeight()/2-600)+630-15,
paint);
    }
    else if(MainActivity.n1 > 0) {
        canvas.drawText("y=ax", (getWidth()/2-360)+450, (getHeight()/2-600)+150,
paint);
        canvas.drawText("2", (getWidth()/2-360)+450+100, (getHeight()/2-600)+150-15,

```

```

paint);
    }

    fg=0;

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

        px=(int) (360+20*x);
        py=(int) (390-20*y);
        if (fg==0) {
            if((getHeight()/2-600)+py>(getHeight()/2-600)+110 && (getHeight()/2-
600)+py<(getHeight()/2-600)+670) {
                canvas.drawLine((int) (getWidth()/2-360)+px, (int) (getHeight()/2-
600)+py, (int) (getWidth()/2-360)+px, (int) (getHeight()/2-600)+py, paint);
            }
        }
        else {
            if((getHeight()/2-600)+py>(getHeight()/2-600)+110 && (getHeight()/2-
600)+py<(getHeight()/2-600)+670) {
                canvas.drawLine((int) (getWidth()/2-360)+oldpx, (int) (getHeight()/2-
600)+oldpy, (int) (getWidth()/2-360)+px, (int) (getHeight()/2-600)+py, paint);
            }
        }
        oldpx=px;oldpy=py;
        fg++;
    }

    //y=a(x-b)^2+c のグラフの描画
    paint.setColor(Color.BLUE);

    if(MainActivity.n1 > 0 && MainActivity.n2 >= 0) {
        canvas.drawText("y = a(x-b) + c", (getWidth()/2-360)+400, (getHeight()/2-
600)+630, paint);
        canvas.drawText("2", (getWidth()/2-360)+400+155, (getHeight()/2-600)+630-15,
paint);
    }
}

```

```

    }
    else if(MainActivity.n1 > 0 && MainActivity.n2 <= 0) {
        canvas.drawText("y=a(x-b) +c", (getWidth()/2-360)+100, (getHeight()/2-
600)+630, paint);
        canvas.drawText("2", (getWidth()/2-360)+100+155, (getHeight()/2-600)+630-15,
paint);
    }
    else if(MainActivity.n1 < 0 && MainActivity.n2 <= 0) {
        canvas.drawText("y=a(x-b) +c", (getWidth()/2-360)+100, (getHeight()/2-
600)+170, paint);
        canvas.drawText("2", (getWidth()/2-360)+100+155, (getHeight()/2-600)+170-15,
paint);
    }
    else if(MainActivity.n1 < 0 && MainActivity.n2 >= 0) {
        canvas.drawText("y=a(x-b) +c", (getWidth()/2-360)+400, (getHeight()/2-
600)+170, paint);
        canvas.drawText("2", (getWidth()/2-360)+400+155, (getHeight()/2-600)+170-15,
paint);
    }

    fg=0;

    for (x=MainActivity.n2-4;x<=MainActivity.n2+4;x=x+0.01) {
        y=MainActivity.n1*(x-MainActivity.n2)*(x-MainActivity.n2)+MainActivity.n3;
        px=(int) (360+20*x);
        py=(int) (390-20*y);
        if (fg==0) {
            if((getHeight()/2-600)+py>(getHeight()/2-600)+110 && (getHeight()/2-
600)+py<(getHeight()/2-600)+670) {
                canvas.drawLine((int) (getWidth()/2-360)+px, (int) (getHeight()/2-
600)+py, (int) (getWidth()/2-360)+px, (int) (getHeight()/2-600)+py, paint);
            }
        }
        else {
            if((getHeight()/2-600)+py>(getHeight()/2-600)+110 && (getHeight()/2-
600)+py<(getHeight()/2-600)+670) {

```

```

        canvas.drawLine((int)(getWidth()/2-360)+oldpx, (int)(getHeight()/2-
600)+oldpy, (int)(getWidth()/2-360)+px, (int)(getHeight()/2-600)+py, paint);
    }
}
oldpx=px;oldpy=py;
fg++;
}

//実験枠の描画
//paint.setColor(Color.BLACK);
//canvas.drawLine((getWidth()/2-360)+90, (getHeight()/2-600)+100, (getWidth()/2-
360)+90, (getHeight()/2-600)+400, paint);
//canvas.drawLine((getWidth()/2-360)+90, (getHeight()/2-600)+100, (getWidth()/2-
360)+390, (getHeight()/2-600)+100, paint);

//表題の表示
paint.setColor(Color.BLUE);
paint.setTextSize(35.0f);
canvas.drawText("", (getWidth()/2-360)+60, (getHeight()/2-600)+65, paint);
canvas.drawText("【 The Graph of  $y = a(x - b)^2 + c$ 】", (getWidth()/2-360)+30+60-
40, (getHeight()/2-600)+65, paint);
//canvas.drawText("2", (getWidth()/2-360)+195+2-5+1+1+160, (getHeight()/2-600)+50,
paint);

//目標の提示
paint.setColor(Color.BLACK);
paint.setTextSize(30.0f);
canvas.drawText("■Let's observe that the graph of ", (getWidth()/2-360)+50,
(getHeight()/2-600)+440+330+20, paint);
//canvas.drawText("2", (getWidth()/2-240)+141-5, (getHeight()/2-343)+430, paint);
//canvas.drawText("2", (getWidth()/2-240)+331-13, (getHeight()/2-343)+430, paint);
canvas.drawText("  $y = a(x - b)^2 + c$  is a translation of the graph ", (getWidth()/2-
360)+50, (getHeight()/2-600)+470+330+20, paint);
canvas.drawText(" of  $y = ax^2$  by +b in the x-axis direction and ", (getWidth()/2-
360)+50, (getHeight()/2-600)+500+330+20, paint);
canvas.drawText(" +c in the y-axis direction.", (getWidth()/2-360)+50,

```

```
(getHeight()/2-600)+530+330+20, paint);
```

```
// a、b、c の値の表示
```

```
paint.setColor(Color.BLUE);
```

```
paint.setTextSize(45.0f);
```

```
canvas.drawText("a="+MainActivity.n1, (getWidth()/2-360)+120, (getHeight()/2-600)+90+600+40, paint);
```

```
canvas.drawText("b="+MainActivity.n2, (getWidth()/2-360)+210+90, (getHeight()/2-600)+90+600+40, paint);
```

```
canvas.drawText("c="+MainActivity.n3, (getWidth()/2-360)+300+180, (getHeight()/2-600)+90+600+40, paint);
```

```
//説明の表示
```

```
paint.setColor(Color.BLACK);
```

```
paint.setTextSize(30.0f);
```

```
canvas.drawText("Please touch the screen to move  $y = a x^2$ .", (getWidth()/2-360)+50, (getHeight()/2-600)+950, paint);
```

```
//canvas.drawText("2", (getWidth()/2-240)+231-13, (getHeight()/2-343)+535, paint);
```

```
canvas.drawText("Please touch the screen again to stop.", (getWidth()/2-360)+50, (getHeight()/2-600)+980, paint);
```

```
canvas.drawText("If you touch it further, the graph of  $y=ax^2$ ", (getWidth()/2-360)+50, (getHeight()/2-600)+1010, paint);
```

```
canvas.drawText("returns to its original position.", (getWidth()/2-360)+50, (getHeight()/2-600)+1035, paint);
```

```
canvas.drawText("When the screen goes dark, touch the title bar !", (getWidth()/2-360)+50, (getHeight()/2-600)+1065, paint);
```

```
paint.setColor(Color.BLUE);
```

```
paint.setTextSize(30.0f);
```

```
canvas.drawText("Copyright(C) Sohun 2021.10.3", (getWidth()/2-360)+150+10, (getHeight()/2-600)+1130, paint); //作者・作成年月の表示
```

```
//識別子が1のとき、グラフが平行移動する
```

```
if (flag==1) {
```

```
    if (MainActivity.n2<=0 && MainActivity.n3<=0) {
```

```
        if (a>=MainActivity.n2) {
```

```
            a=a-0.1;
```



```

    }
    else if (b>=MainActivity.n3) {
        b=b-0.1;
    }
}
else if (MainActivity.n2>=0 && MainActivity.n3>=0) {
    if (a<=MainActivity.n2) {
        a=a+0.1;
    }
    else if (b<=MainActivity.n3) {
        b=b+0.1;
    }
}
else if (MainActivity.n2>=0 && MainActivity.n3<=0) {
    if (a<=MainActivity.n2) {
        a=a+0.1;
    }
    else if (b>=MainActivity.n3) {
        b=b-0.1;
    }
}
else if (MainActivity.n2<=0 && MainActivity.n3>=0) {
    if (a>=MainActivity.n2) {
        a=a-0.1;
    }
    else if (b<=MainActivity.n3) {
        b=b+0.1;
    }
}
}

```

`invalidate()`; //再描画する (clear & goto onDraw) そして、この行へ戻ってくる。

//`invalidate()`は、`onDraw` メソッドの中にも記述できる。

//`invalidate()`は、繰り返し処理に利用できる。

//もちろん、`onTouchEvent` ソッドの中にも記述できる。

```

}

```

//識別子が2のとき、グラフが停止する

```

else if (flag==2) {
    //何も変化を加えない
}
//識別子が0のとき、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) そして、この行へ戻ってくる。
}

} //protected void onDraw(Canvas canvas)

//画面に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;

} //public boolean onTouchEvent(MotionEvent event)

} //public class MyTwokansu extends View

```

[2]activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

```

```
tools:context=".MainActivity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello World!"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<jp.kiyo.wuena.mytwokansueng.MyTwokansuEng
```

```
    android:id="@+id/myfview1"  
    android:layout_height="match_parent"  
    android:layout_width="match_parent"/>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

[3]MainActivity.java

```
/*
```

2次関数のグラフの平行移動(英語版)

Android 4.1 (Jelly Bean)

Copyright (C) K. Niwa 2021. 10. 3

```
*/
```

```
package jp.kiyo.wuena.mytwokansueng;
```

```
//ライブラリを読み込む
```

```
import android.app.Activity;  
import android.os.Bundle;  
import android.text.Editable;  
import android.text.InputType;  
import android.widget.LinearLayout;
```

```

import android.view.*;
import android.view.View.OnClickListener;
import android.widget.*;
import androidx.appcompat.app.AppCompatActivity;

import static android.text.InputType.TYPE_CLASS_NUMBER;
import static android.text.InputType.TYPE_NUMBER_FLAG_DECIMAL;
import static android.text.InputType.TYPE_NUMBER_VARIATION_NORMAL;

public class MainActivity extends AppCompatActivity {

    TextView text0;    //TextView 型で宣言する

    TextView text1;    //TextView 型で宣言する
    EditText edit1;    //EditText 型で宣言する

    TextView text2;    //TextView 型で宣言する
    EditText edit2;    //EditText 型で宣言する

    TextView text3;    //TextView 型で宣言する
    EditText edit3;    //EditText 型で宣言する

    Button button1;    //Button 型で宣言する

    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
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    */

```

```

@Override
public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    LinearLayout layout = new LinearLayout(this);
    layout.setOrientation(LinearLayout.VERTICAL);

    text0 = new TextView(this);
    text0.setText(" About quadratic function  $y=a(x-b)^2+c$ ");
    layout.addView(text0);

    text1 = new TextView(this);
    text1.setText(" Enter the integer a ! (However  $-3 \leq a \leq 3$   $a \neq 0$ )");
    layout.addView(text1);

    edit1 = new EditText(this);
    //edit1.setInputType(TYPE_NUMBER_FLAG_DECIMAL);
    layout.addView(edit1);

    text2 = new TextView(this);
    text2.setText(" Enter the integer b ! (However  $-7 \leq b \leq 7$ )");
    layout.addView(text2);

    edit2 = new EditText(this);
    layout.addView(edit2);

    text3 = new TextView(this);
    text3.setText(" Enter the integer c ! (However  $-10 \leq c \leq 10$ )");
    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);
} //public void onCreate(Bundle savedInstanceState)

```

```

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.activity_main); //res フォルダの layout フォルダの
main.xmlに移動する

} //public void onClick(View view)

} //class MyBtnClickAdapter implements OnClickListener

} //public class MainActivity extends AppCompatActivity
```