

Propose Mobile touch motion

개발자 대회를 통한 오픈소스 프로젝트 개발하기

Propose Mobile touch motion



오재웅

어떻게 오픈소스를 시작했나?

“평생 개발만 하며 살 수 있다”



오픈소스, 무엇이 힘들었나?

시간, 영어

시 간

개발기간은 무제한
오래동안 끝까지

영 어

한국어로 해도 되더라

왜 쓰지 않는 거죠?

- 초급개발자 : 너무 어려워
- 고급개발자 : 이미 잘하고 있는데 왜 사용 해야돼?
- 기업 : 문제 생기면 누가 책임져?

- 초급 개발자
쉽다고 말 할 때까지

- 고급 개발자
할 수 없는 것을 지원

- 기업

Non-invasive(비침투적) FrameWork

개발자 대회를 통해 얻은 것?

내가 무엇을 만들었는지 알게 되었다.





Android Mobile touch motion

어렵던 모션개발, 모두가 가능하도록

PROPOSE (청혼)

내가 잘하는 것으로 사랑하는 이에게 감동을 전해주고 싶습니다



Motion E-BOOK

사진을 터치 조작하여 추억을 재현할 수 있는 이야기책

→ 직접적인 인터랙션을 통한 흥미 부여



현실은...

<책장을 넘기는 모션 개발>



$$\begin{aligned} a \sin \theta + b \cos \theta &= \sqrt{a^2 + b^2} \sin (\theta + \alpha) \\ &= \sqrt{a^2 + b^2} \cos (\theta - \beta) \\ (\sin \alpha &= \cos \beta = \frac{b}{\sqrt{a^2 + b^2}}) \\ (\cos \alpha &= \sin \beta = \frac{a}{\sqrt{a^2 + b^2}}) \end{aligned}$$

(위) 모션개발에 필요한 수식 중 하나

모션 하나에 개발 소요시간 약 한달!

$$a \sin \theta + b \cos \theta = \sqrt{a^2 + b^2} \sin(\theta + \alpha)$$

$$= \sqrt{a^2 + b^2} \cos(\theta - \beta)$$

$$(\sin \alpha = \cos \beta = \frac{b}{\sqrt{a^2 + b^2}})$$

$$(\cos \alpha = \sin \beta = \frac{a}{\sqrt{a^2 + b^2}})$$

좌표 연산

터치분석

물리적 연산

개발이 너무 어려움

Etc

사용자 패턴
분석

Awesome UI

모션관련 Github를 주소를 모아놓은 사이트

그리하여, 모션 오픈소스로 눈을 돌렸으나



Open source Hell

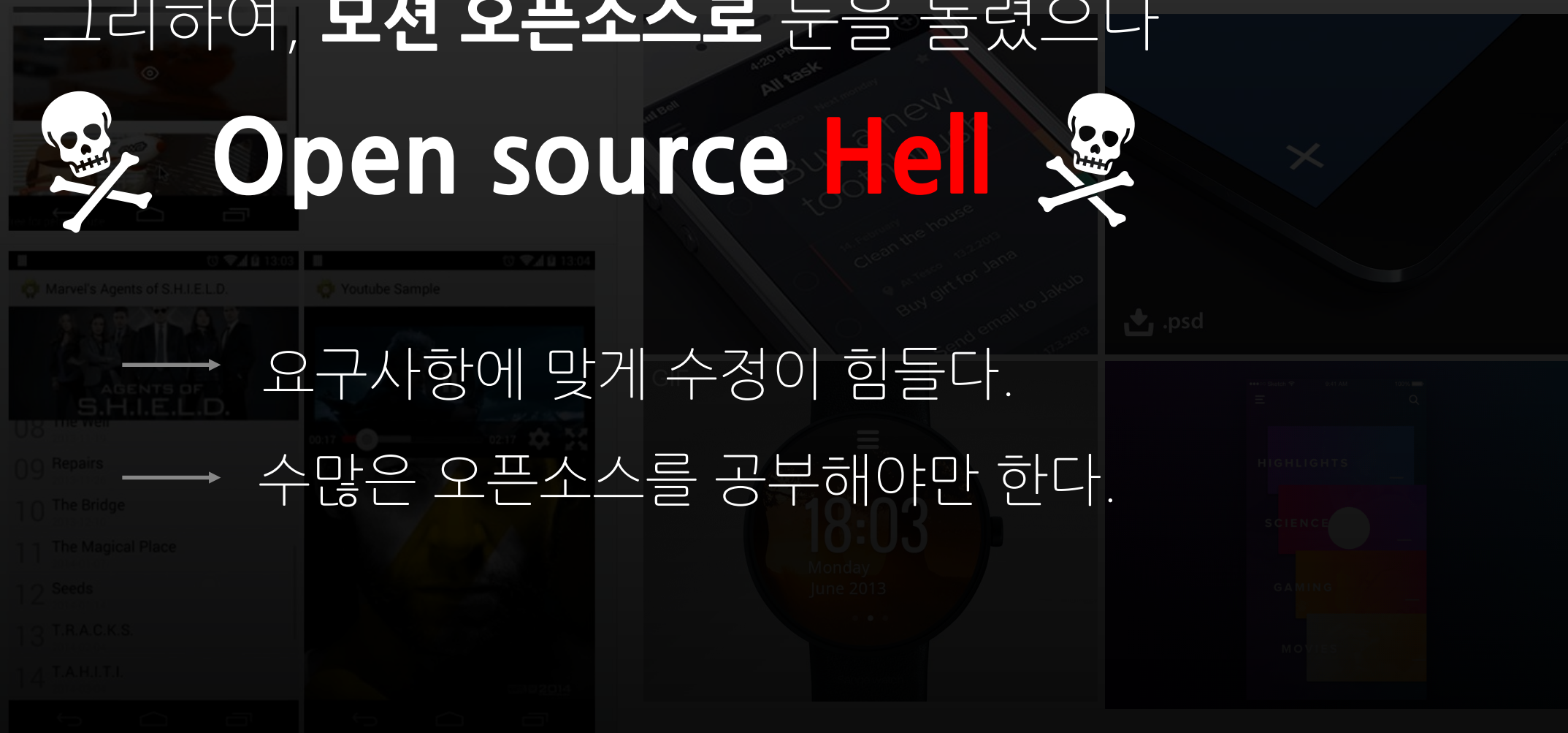


→ 요구사항에 맞게 수정이 힘들다.

→ 수많은 오픈소스를 공부해야만 한다.

DraggablePanel

Apache
License
V2



Propose Mobile touch motion



결국 포기

Propose Mobile touch motion

모션개발

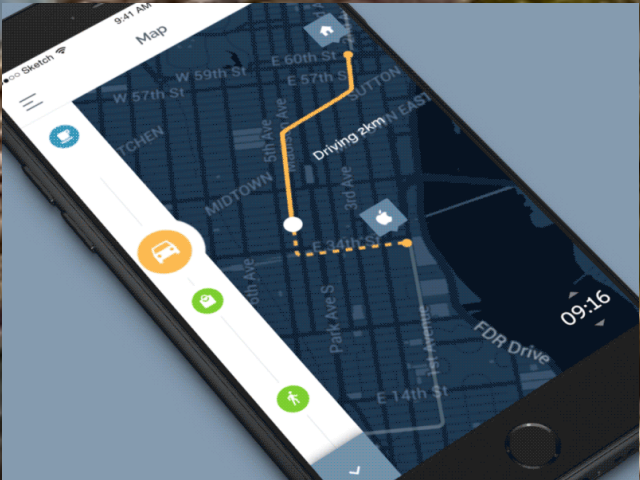
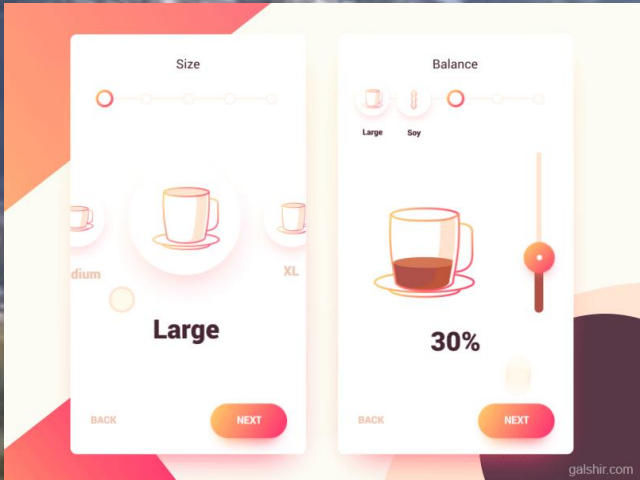


모션은 반드시 넘어야할 산!

App의 완성



개발자 Developer



PROPOSE

어렵던 모션개발, 모두가 가능하도록



PROPOSE 개발 목표



“ Easy, Adaptability, Portability, Safely ”

- 소프트웨어의 장인 Kevil Henny-

1

Easy

2

Adaptability

3

Portability

4

Safely

Easy

개발자라면 누구나 한번 하는 경험

“설치가 쉽다는데 잘 안돼 ㅜㅜ”



Easy



프로포즈는

빌드 파일에 선언만 하면 바로 사용 가능

```
compile 'com.markjmind.propose:propose:1.1.+'
```

Easy

개발자가 감당하기 어려운 모든 힘든 연산을 대신 처리해줍니다.

In this figure we see two similar triangles, both having parts of the projection line (green) as their hypotenuses. The catheti of the right triangle are x_1 and x_3 . It follows that

$$\frac{-y_1}{f} = \frac{x_1}{x_3} \text{ or } y_1 = -\frac{f x_1}{x_3}$$

A similar investigation, looking in the negative

$$\frac{-y_2}{f} = \frac{x_2}{x_3} \text{ or } y_2 = -\frac{f x_2}{x_3}$$

This can be summarized as

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = -\frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

which is an expression that describes the relationship of point P and its image coordinates (y_1, y_2) given by point Q in the image plane.

Rotated image and the virtual image plane

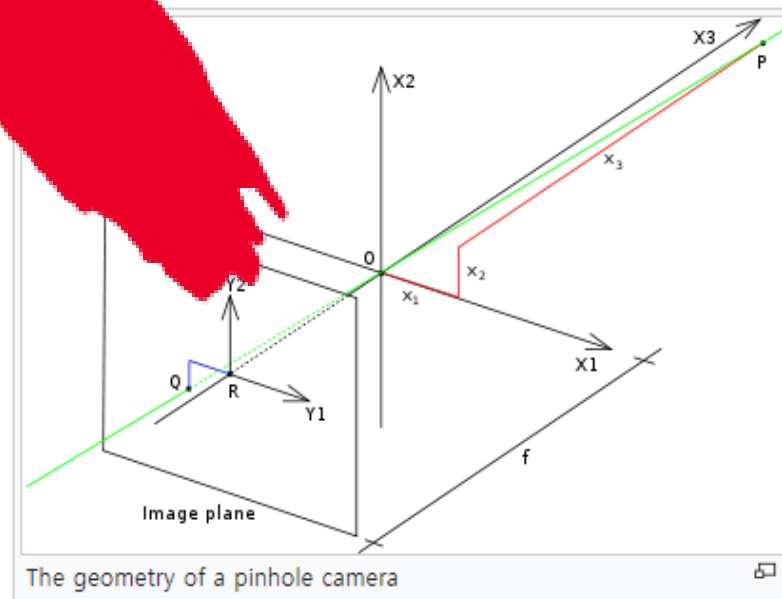
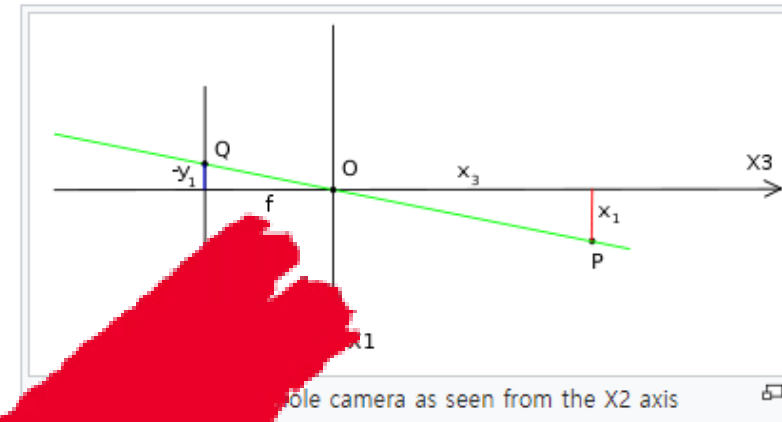
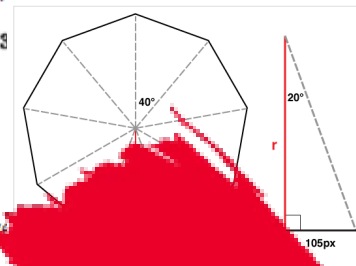
The mapping from 3D to 2D corresponds to how their distance to the focal point to produce an unrotated

- Rotate the coordinate system would solve the problem such an order that it
- Place the image plane (front) image plane with one.

In both cases, the result

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

This is the code for my `GLSurfaceView.Renderer`:



Propose Mobile touch motion

Easy

Code less

그러다 보니 할 일이 거의 없습니다.

```
if ((Build.VERSION.SDK_INT >= 21) && getBackground() == null) {
    setBackgroundResource(R.drawable.control_background_multi_material);
}

mUiThreadId = Thread.currentThread().getId();

a = context.obtainStyledAttributes(attrs, io.apptik.widget.mslider.R.styleable.MultiSlider,
    defStyle, styleRes);
mNoInvalidate = true;
int numThumbs = a.getInt(io.apptik.widget.mslider.R.styleable.MultiSlider_thumbNumber, 2);
initMultiSlider(numThumbs);

Drawable trackDrawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_android_track);
if (trackDrawable == null) {
    trackDrawable = ContextCompat.getDrawable(getContext(),
        R.drawable.multislider_track_material
    );
}

setTrackDrawable(getTintedDrawable(trackDrawable, a.getColor(io.apptik.widget.mslider.R.styleable.MultiSlider_
    mMinWidth = a.getDimensionPixelSize(R.styleable.MultiSlider_minWidth, mMinWidth);
mMaxWidth = a.getDimensionPixelSize(R.styleable.MultiSlider_maxWidth, mMaxWidth);
mMinHeight = a.getDimensionPixelSize(R.styleable.MultiSlider_minHeight, mMinHeight);
mMaxHeight = a.getDimensionPixelSize(R.styleable.MultiSlider_maxHeight, mMaxHeight);

setStep(a.getInt(io.apptik.widget.mslider.R.styleable.MultiSlider_scaleStep, mStep));
setStepsThumbsApart(a.getInt(io.apptik.widget.mslider.R.styleable
    .MultiSlider_stepsThumbsApart,
    mStepsThumbsApart));
setDrawThumbsApart(a.getBoolean(io.apptik.widget.mslider.R.styleable
    .MultiSlider_drawThumbsApart,
    mDrawThumbsApart));
setMax(a.getInt(io.apptik.widget.mslider.R.styleable.MultiSlider_scaleMax, mScaleMax), true);
setMin(a.getInt(io.apptik.widget.mslider.R.styleable.MultiSlider_scaleMin, mScaleMin), true);

mMirrorForRtl = a.getBoolean(io.apptik.widget.mslider.R.styleable.MultiSlider_mirrorForRTL,
    mMirrorForRtl);

// --> now place thumbs

defThumbDrawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_android_thumb);

if (defThumbDrawable == null) {
    if (Build.VERSION.SDK_INT >= 21) {
        defThumbDrawable = ContextCompat.getDrawable(getContext(), R.drawable.multislider_thumb_material_anim)
    } else {
        defThumbDrawable = ContextCompat.getDrawable(getContext(), R.drawable.multislider_thumb_material);
    }
}

defRangeDrawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_range);
if (defRangeDrawable == null) {
    defRangeDrawable = ContextCompat.getDrawable(getContext(),
        R.drawable.multislider_range_material
    );
}

Drawable range1Drawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_range1);
Drawable range2Drawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_range2);

defRangeColor = a.getColor(io.apptik.widget.mslider.R.styleable.MultiSlider_rangeColor, 0);
defThumbColor = a.getColor(io.apptik.widget.mslider.R.styleable.MultiSlider_thumbColor, 0);
setThumbDrawables(defThumbDrawable, defRangeDrawable, range1Drawable, range2Drawable); //

int thumbOffset = a.getDimensionPixelOffset(io.apptik.widget.mslider.R.styleable
    .MultiSlider_android_thumbOffset, defThumbDrawable.getIntrinsicWidth() / 2);
setThumbOffset(thumbOffset);
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
    ViewGroup move_lyt = (ViewGroup)findViewById(R.id.move_lyt);

    ObjectAnimator paperAnim = ObjectAnimator.ofFloat(move_lyt, View.ROTATION_Y, 0, 180);

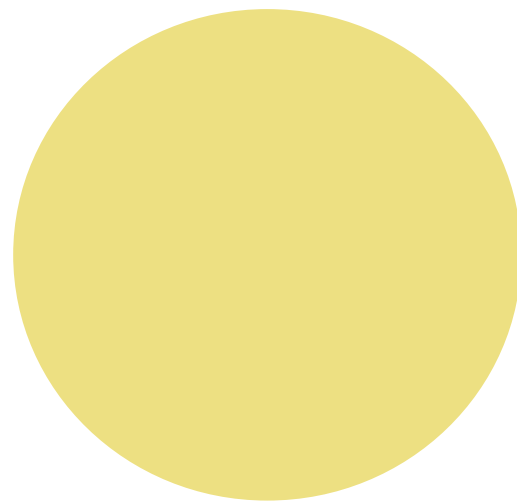
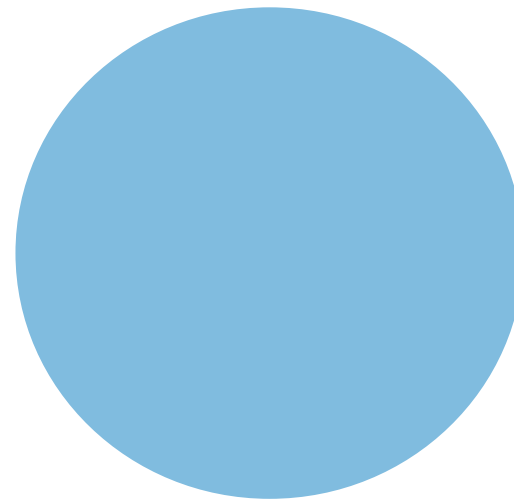
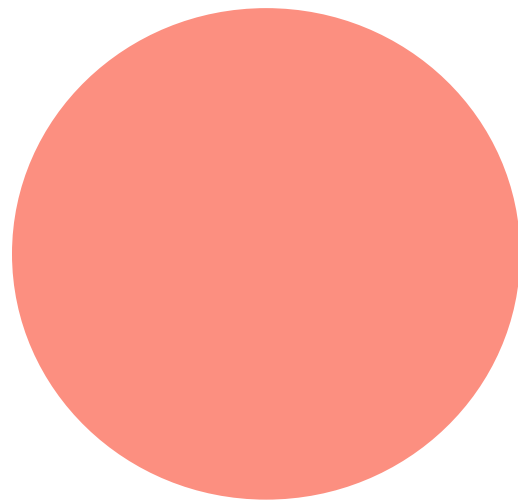
    Propose propose = new Propose(this);
    propose.motionRight.play(paperAnim);
    move_lyt.setOnTouchListener(propose);
}
```

<https://github.com/muabe/Samples/blob/master/paper/src/com/markmind/propose/sample/paper/MainActivity.java>

Adaptability(합성)

RGB 3가지 색으로 세상의 모든 색을 만들 수 있습니다.

A모션 + B모션 = C모션



Adaptability

motion1.play(anim1).with(anim2).with(anim3).next(anim3)

propose.add(motion1).add(motion2)

Code less

```
if ((Build.VERSION.SDK_INT >= 21) && getBackground() == null) {
    setBackgroundResource(R.drawable.control_background_multi_material);
}

mUiThreadId = Thread.currentThread().getId();

a = context.obtainStyledAttributes(attrs, io.apptik.widget.mslider.R.styleable.MultiSlider,
    defStyle, styleRes);
mNoInvalidate = true;
int numThumbs = a.getInt(io.apptik.widget.mslider.R.styleable.MultiSlider_thumbNumber, 2);
initMultiSlider(numThumbs);

Drawable trackDrawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_android_track);
if (trackDrawable == null) {
    trackDrawable = ContextCompat.getDrawable(getContext(),
        R.drawable.multislider_track_material
    );
}

setTrackDrawable(getTintedDrawable(trackDrawable, a.getColor(io.apptik.widget.mslider.R.styleable.MultiSlider_
    mMinWidth = a.getDimensionPixelSize(R.styleable.MultiSlider_minWidth, mMinWidth);
mMaxWidth = a.getDimensionPixelSize(R.styleable.MultiSlider_maxWidth, mMaxWidth);
mMinHeight = a.getDimensionPixelSize(R.styleable.MultiSlider_minHeight, mMinHeight);
mMaxHeight = a.getDimensionPixelSize(R.styleable.MultiSlider_maxHeight, mMaxHeight);

setStep(a.getInt(io.apptik.widget.mslider.R.styleable.MultiSlider_scaleStep, mStep));
setStepsThumbsApart(a.getInt(io.apptik.widget.mslider.R.styleable
    .MultiSlider_stepsThumbsApart,
    mStepsThumbsApart));
setDrawThumbsApart(a.getBoolean(io.apptik.widget.mslider.R.styleable
    .MultiSlider_drawThumbsApart,
    mDrawThumbsApart));
setMax(a.getInt(io.apptik.widget.mslider.R.styleable.MultiSlider_scaleMax, mScaleMax), true);
setMin(a.getInt(io.apptik.widget.mslider.R.styleable.MultiSlider_scaleMin, mScaleMin), true);

mMirrorForRtl = a.getBoolean(io.apptik.widget.mslider.R.styleable.MultiSlider_mirrorForRTL,
    mMirrorForRtl);

// --> now place thumbs

defThumbDrawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_android_thumb);

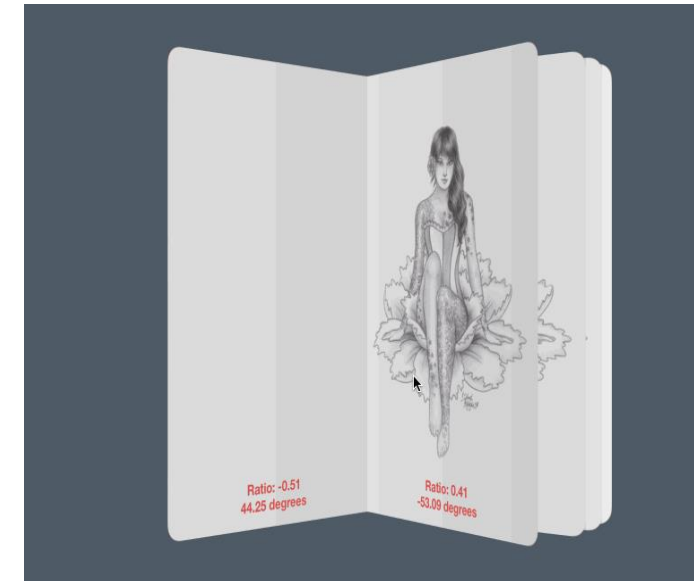
if (defThumbDrawable == null) {
    if (Build.VERSION.SDK_INT >= 21) {
        defThumbDrawable = ContextCompat.getDrawable(getContext(), R.drawable.multislider_thumb_material_anim)
    } else {
        defThumbDrawable = ContextCompat.getDrawable(getContext(), R.drawable.multislider_thumb_material);
    }
}

defRangeDrawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_range);
if (defRangeDrawable == null) {
    defRangeDrawable = ContextCompat.getDrawable(getContext(),
        R.drawable.multislider_range_material
    );
}

Drawable range1Drawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_range1);
Drawable range2Drawable = a.getDrawable(io.apptik.widget.mslider.R.styleable
    .MultiSlider_range2);

defRangeColor = a.getColor(io.apptik.widget.mslider.R.styleable.MultiSlider_rangeColor, 0);
defThumbColor = a.getColor(io.apptik.widget.mslider.R.styleable.MultiSlider_thumbColor, 0);
setThumbDrawables(defThumbDrawable, defRangeDrawable, range1Drawable, range2Drawable); //

int thumbOffset = a.getDimensionPixelOffset(io.apptik.widget.mslider.R.styleable
    .MultiSlider_android_thumbOffset, defThumbDrawable.getIntrinsicWidth() / 2);
setThumbOffset(thumbOffset);
```



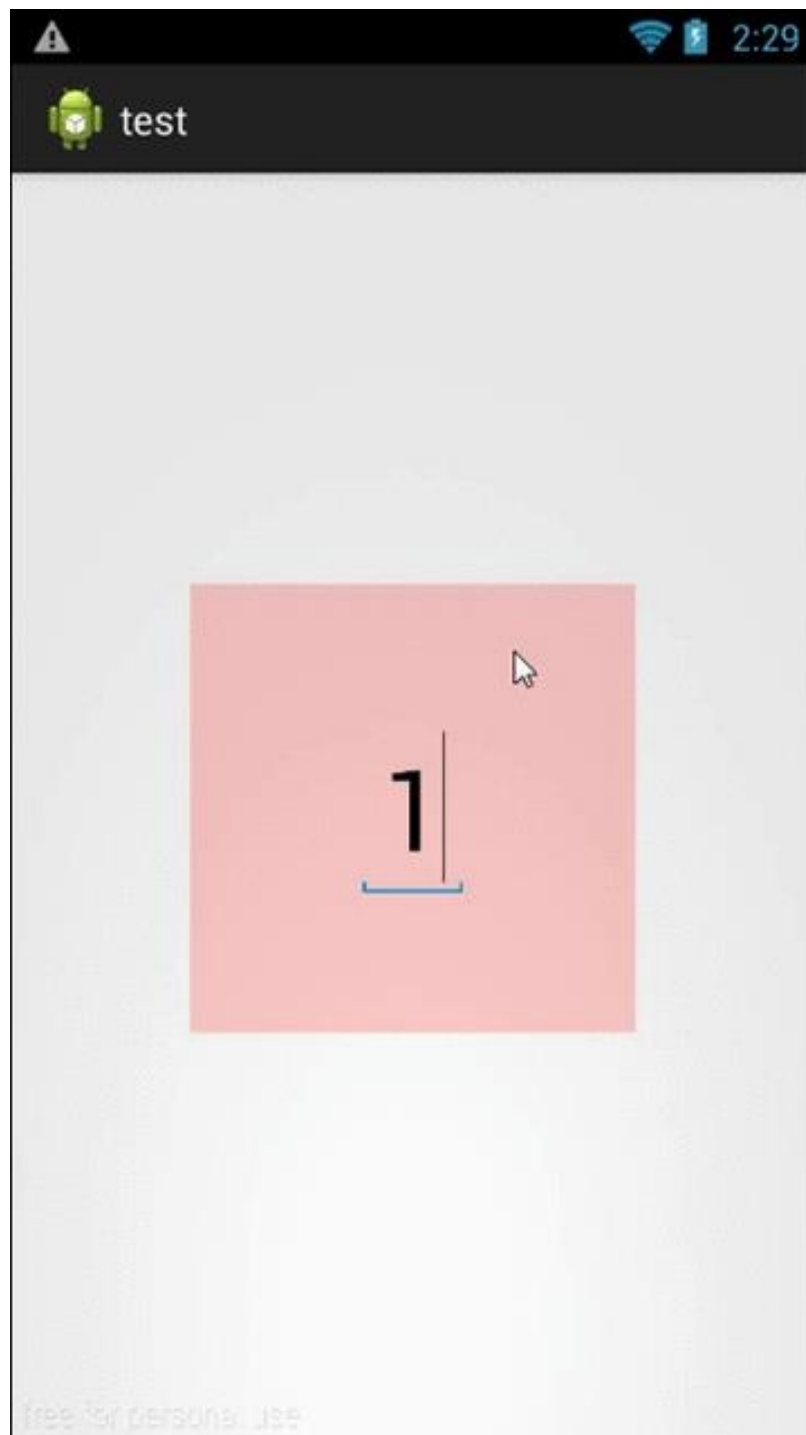
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
    ViewGroup move_lyt = (ViewGroup) findViewById(R.id.move_lyt);

    ObjectAnimator paperAnim = ObjectAnimator.ofFloat(move_lyt, View.ROTATION_Y, 0, 180);

    Propose propose = new Propose(this);
    propose.motionRight.play(paperAnim);
    move_lyt.setOnTouchListener(propose);
}
```

<https://github.com/muabe/Samples/blob/master/paper/src/com/markjmind/propose/sample/paper/MainActivity.java>

Propose Mobile touch motion



jaggi772 <notifications@github.com>

9월 6일 ☆

muabe/Propose, 나, Mention에게

@muabe i have tried to make animate like <https://www.youtube.com/watch?v=X0x4wgDQV20> shown video but no success. I need this type of animation. Plz help.

Plantaer <notifications@github.com>

7월 24일 ☆

muabe/Propose, Subscribed에게

I'm really excited about this library. More immersive apps is what i want to achieve. Could you please share with Cube Out Transform (https://www.youtube.com/watch?v=K_5Y16Fz0Ps) project or something similar?

1. I don't understand how split one object (like circular ImageView on video) between left drawer and different Layout. How is it attached?
2. In Story Book on video (hadn't found sample code), how to transform only a half of Layout, View (whatever it is)? I mean page of the book.

Thank you in advance for your help.

josityjanga <notifications@github.com> 수신거부

16. 1. 25. ☆

JaeWoongOh/And.에게

Hi, the library works really nice I have downloaded the storybook example and would like to know, how the 3D effect, when touching the book, can be disabled? I don't want it to be moved in y-direction

가능성에 대한 해외 개발자들의 빗발치는 문의

현실적으로 불가능

In this figure we see two similar triangles, both having (green) as their hypotenuses. The catheti of the left triangle are x_1 and x_3 . Since it follows that

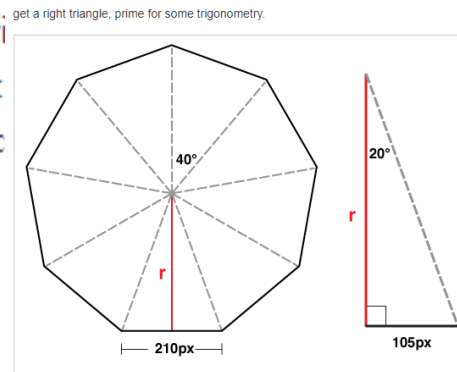
$$\frac{-y_1}{f} = \frac{x_1}{x_3} \text{ or } y_1 = -\frac{f x_1}{x_3}$$

A similar investigation, looking in the negative direction,

$$\frac{-y_2}{f} = \frac{x_2}{x_3} \text{ or } y_2 = -\frac{f x_2}{x_3}$$

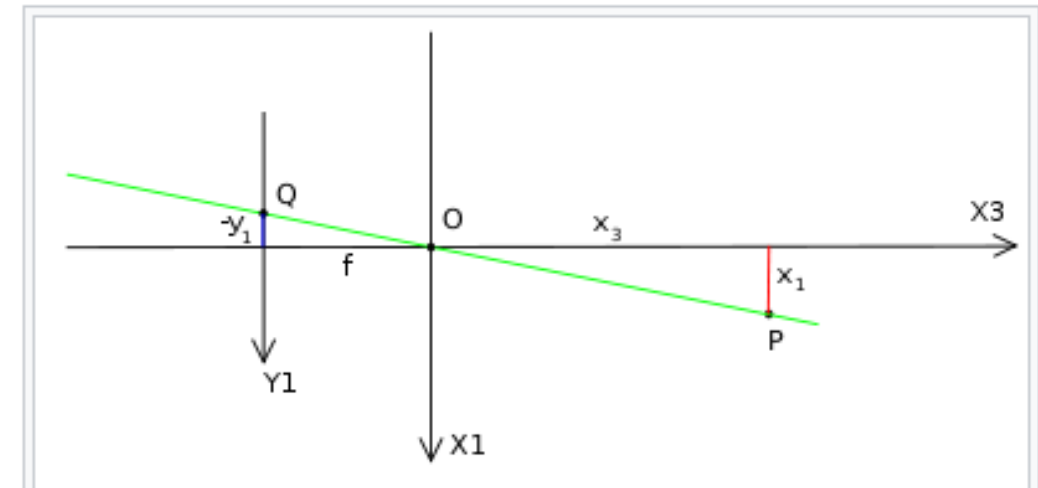
This can be summarized as

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = -\frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



We can determine the length of r in this diagram with a basic tangent equation.

$$\begin{aligned} \tan &= \frac{\text{opposite}}{\text{adjacent}} \\ \tan(20^\circ) &= \frac{105\text{px}}{r} \\ r &= \frac{105\text{px}}{\tan(20^\circ)} \\ r &= 288\text{px} \end{aligned}$$



The geometry of a pinhole camera as seen from the X2 axis

which is an expression that describes the relation between the 3D coordinates (x_1, x_2, x_3) of point P and its image coordinates (y_1, y_2) given by point Q in the image plane.

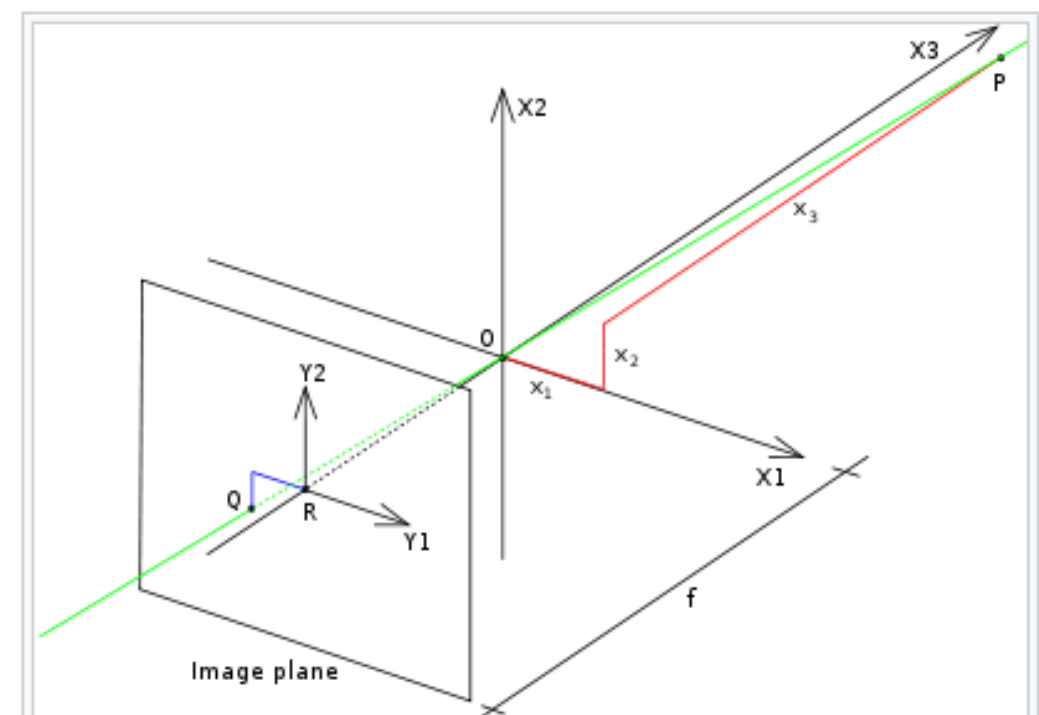
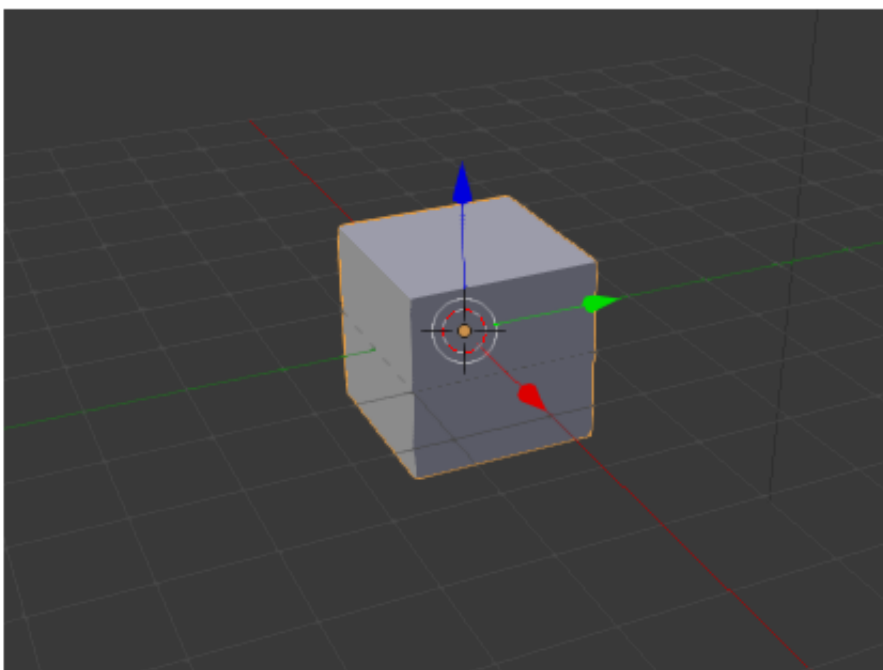
Rotated image and the virtual image plane [\[edit\]](#)

The mapping from 3D to 2D is a projection. This corresponds to how their distance to the focal point produces an unrotated image.

- Rotate the coordinate system such that the image plane is parallel to the X1-X2 plane.
- Place the image plane at a distance f from the origin.

In both cases, the result

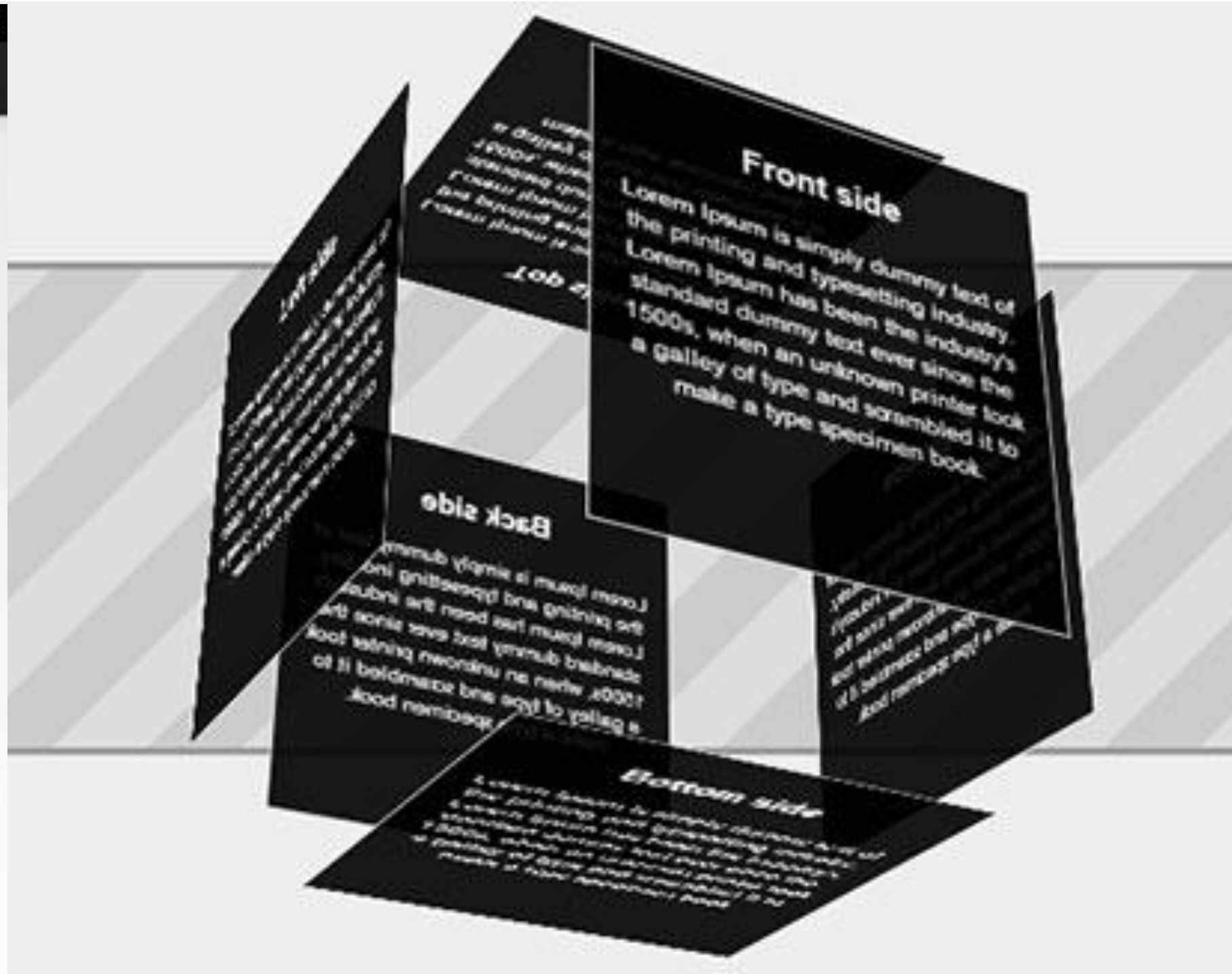
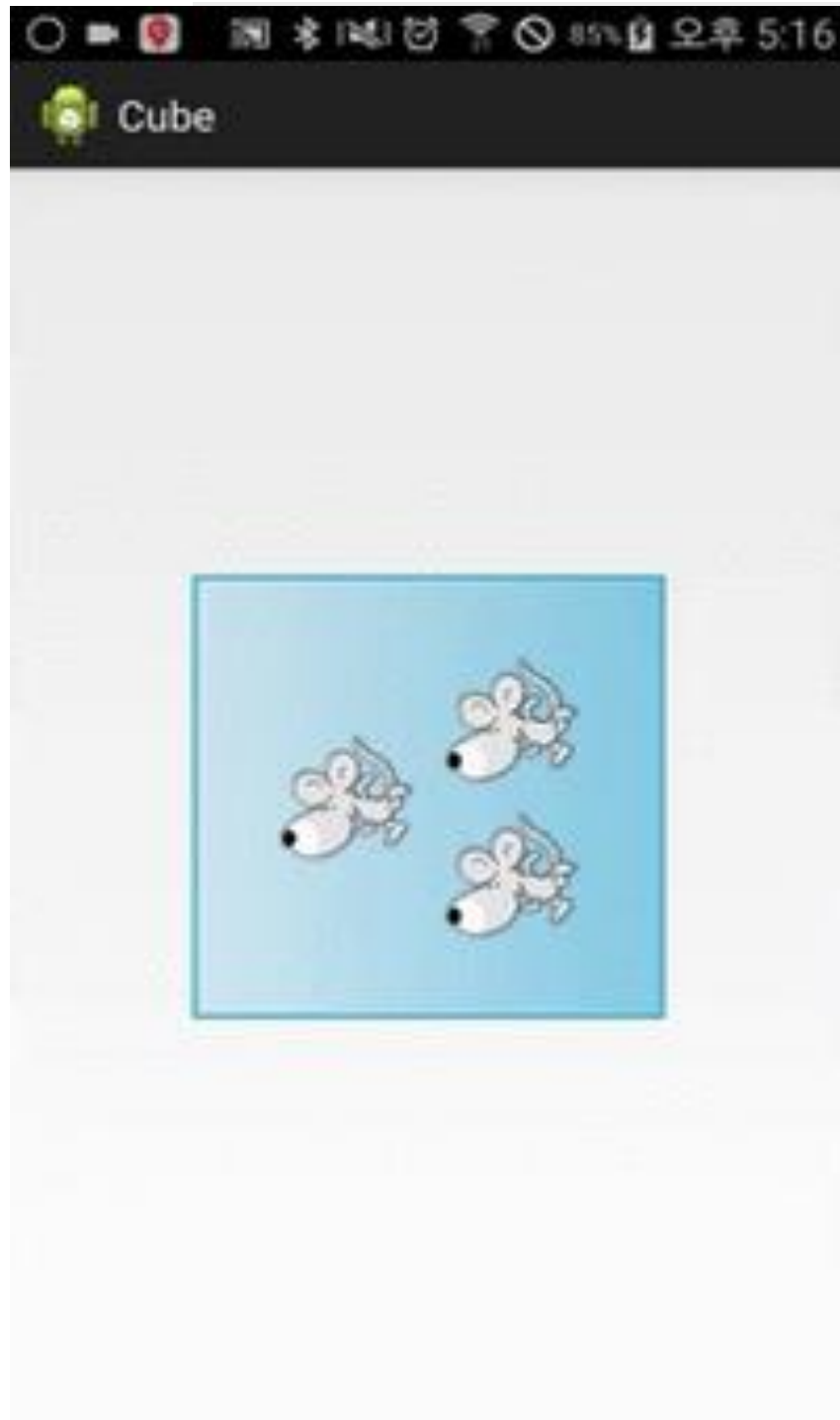
$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = -\frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



The geometry of a pinhole camera

This is the code for my `GLSurfaceView.Renderer`:

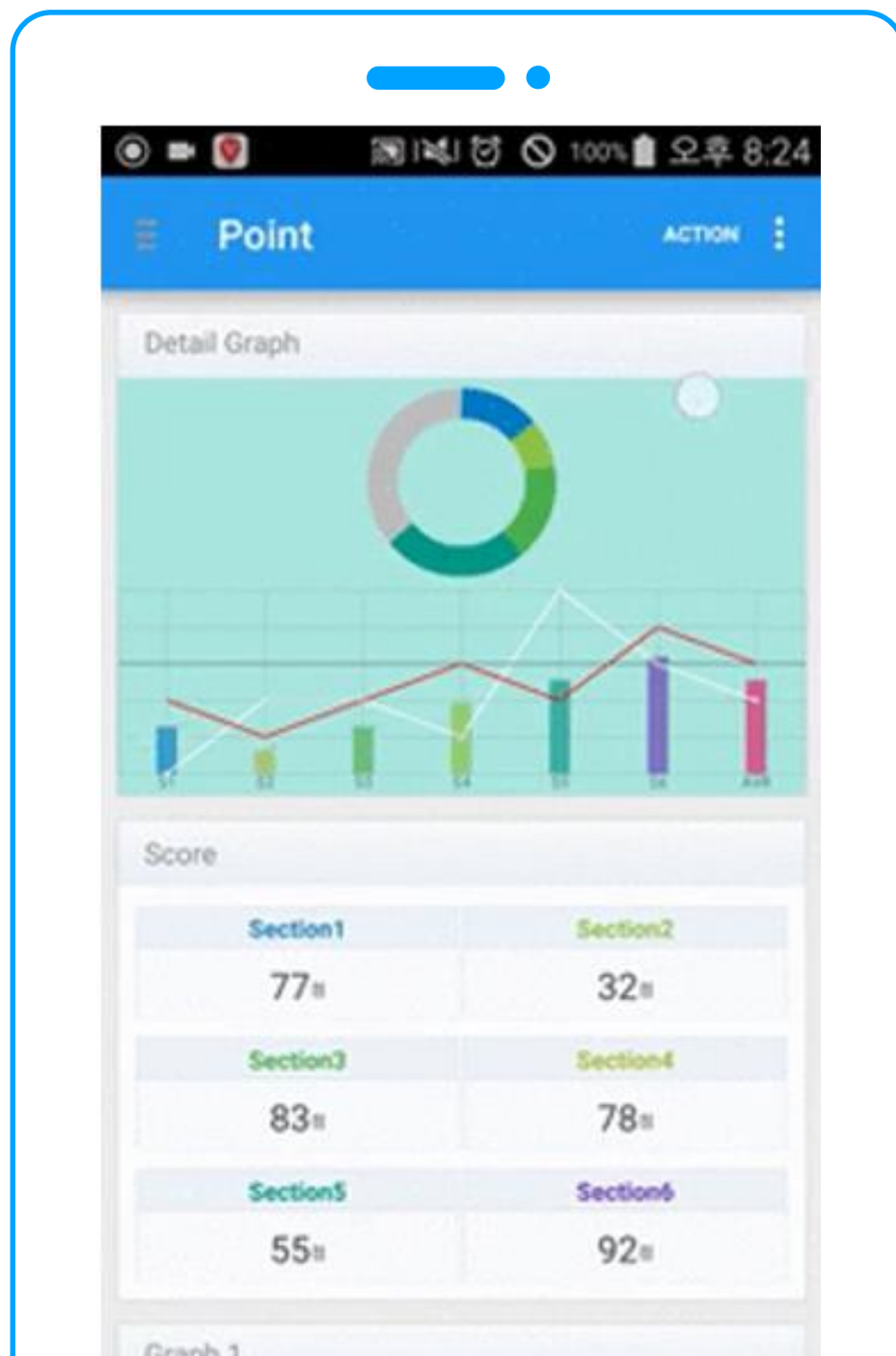
Propose Mobile touch motion



https://github.com/muabe/Samples/tree/master/propose_cube

Portability

Non-invasive(비침투적) Framework로 설계



원소스코드를 침해하지 않고
도입할 수 있는 Framework

Portability

완료된 프로젝트에서도 추가만으로 모션이 부여됩니다.



```
private int optThumbValue(Thumb thumb, int value) {
    if (thumb == null || thumb.getThumb() == null) return value;
    int currIdx = mThumbs.indexOf(thumb);
    if (mThumbs.size() > currIdx + 1 && value > mThumbs.get(currIdx + 1).getValue() - mStepsThumbsApart * mStep) {
        value = mThumbs.get(currIdx + 1).getValue() - mStepsThumbsApart * mStep;
    }
    if (currIdx > 0 && value < mThumbs.get(currIdx - 1).getValue() + mStepsThumbsApart * mStep) {
        value = mThumbs.get(currIdx - 1).getValue() + mStepsThumbsApart * mStep;
    }
    if ((value - mScaleMin) % mStep != 0) {
        value += mStep - ((value - mScaleMin) % mStep);
    }
    if (value < thumb.getMin()) {
        value = thumb.getMin();
    }
    if (value > thumb.getMax()) {
        value = thumb.getMax();
    }
    return value;
}

private synchronized void setThumbValue(Thumb thumb, int value, boolean fromUser) {
    if (thumb == null || thumb.getThumb() == null) return;
    value = optThumbValue(thumb, value);
    if (value != thumb.getValue()) {
        thumb.value = value;
    }
    if (hasOnThumbValueChangeListener()) {
        mOnThumbValueChangeListener.onValueChanged(this, thumb, mThumbs.indexOf(thumb), thumb
            .getValue());
    }
    updateThumb(thumb, getWidth(), getHeight());
}

private synchronized void setThumbValue(int thumb, int value, boolean fromUser) {
    setThumbValue(mThumbs.get(thumb), value, fromUser);
}

private void updateTrackBounds(int w, int h) {
    w -= getPaddingRight() + getPaddingLeft();
    h -= getPaddingTop() + getPaddingBottom();
    int right = w;
    int bottom = h;
    int top = 0;
    int left = 0;
    if (mTrack != null) {
        mTrack.setBounds(0, 0, right, bottom);
    }
}

Propose propose = new Propose(this);
propose.motionRight.play(rightMove, (int)(end)).with(shake);
propose.motionDown.play(rightDown, (int)(end));
propose.motionRight.enableTabUp(false);
propose.motionDown.enableSingleTabUp(false).enableFling(false).enableTabUp(false);
moving_lyt.setOnTouchListener(propose);
}
```

Safely

아무리 좋은 Framework도 검증되지 않으면 사용할지 못할 겁니다
Propose 이미 많은 레퍼런스들로 인해 안정성이 검증된 Framework입니다.

LG전자
전력관리 iot 앱



- 차트관련 모션적용
- 검수 통과

SK 헬스커넥트
헬스온 앱



- 헬스온 2.0 일부적용
- 헬스온 3.0
전체 모션 적용

순천시 나들이
관광앱

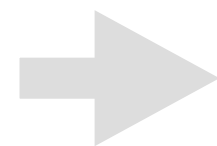


- 모션 전체 적용

전남 여성
안심 귀가 앱



- 모션 전체 적용



안정성 검증

향후 계획

Propose
2.0 개발

모션 추가

프로퍼티 타입
(xml, 어노테이션)



Rowandjj commented on 7 Jul • edited ▾



ok. i'll try that.thank you:)

BTW, when will you finish propose2? @muabe



ashish soni 7개월 전

Thanks, and would you tell me when you are uploading propose2 tutorial?

답글 · 👍 💬 ❤️



Plantaer commented on 24 Jul



빠른 응답에 감사드립니다. 나는 Propose2.0에 대한 정보를 메일 (native1311@gmail.com)을 통해 얻고 싶다.
Google이이를 전적으로 번역하기를 바랍니다. 행운을 빕니다.



Propose

감사합니다.