FractionView

Hands-on Android Custom View Workshop

by Chiu-Ki Chan

http://bit.ly/FractionView

Custom view

Custom

extends view

Custom *functionalities*

extends view

Constructors

XML attributes

Drawing

Measuring

Setters/Getters

Layout

Save state

Library projects

Event handlers

Constructors

XML attributes

Drawing

Measuring

Setters/Getters

Layout

Save state

Library projects

Event handlers

To learn more...

http://is.gd/AndroidCustomComp

Constructors

Constructors

```
public class FractionView extends View {
  public FractionView(Context context) {
    super(context);
  }

  public FractionView(Context context, AttributeSet attrs) {
    super(context, attrs);
  }

  public FractionView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
  }
}
```

Constructors

can also extend from children
e.g. TextView, LinearLayout

```
public class FractionView extends View {
  public FractionView(Context context) {
    super(context);
  }

  public FractionView(Context context, AttributeSet attrs) {
    super(context, attrs);
  }

  public FractionView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
  }
}
```

Constructors

```
public class FractionView extends View {
  public FractionView(Context context) {
    super(context);
  }

  public FractionView(Context context, AttributeSet attrs) {
    super(context, attrs);
  }

  public FractionView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
  }
}
```

from Java

Constructors

```
public class FractionView extends View {
  public FractionView(Context context) {
    super(context);
  }

  public FractionView(Context context, AttributeSet attrs) {
    super(context, attrs);
  }

  public FractionView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
  }
}
```

from XML

Constructors

```
public class FractionView extends View {
  public FractionView(Context context) {
    super(context);
  }

  public FractionView(Context context, AttributeSet attrs) {
    super(context, attrs);
  }

  public FractionView(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
  }
}
```

from XML with style

Custom functionality

```
public FractionView(Context context) {
  super(context);
  setBackgroundColor(Color.LTGRAY);
}

public FractionView(Context context, AttributeSet attrs) {
  super(context, attrs);
  setBackgroundColor(Color.LTGRAY);
}

public FractionView(Context context, AttributeSet attrs, int defStyle) {
  super(context, attrs, defStyle);
  setBackgroundColor(Color.LTGRAY);
}
```

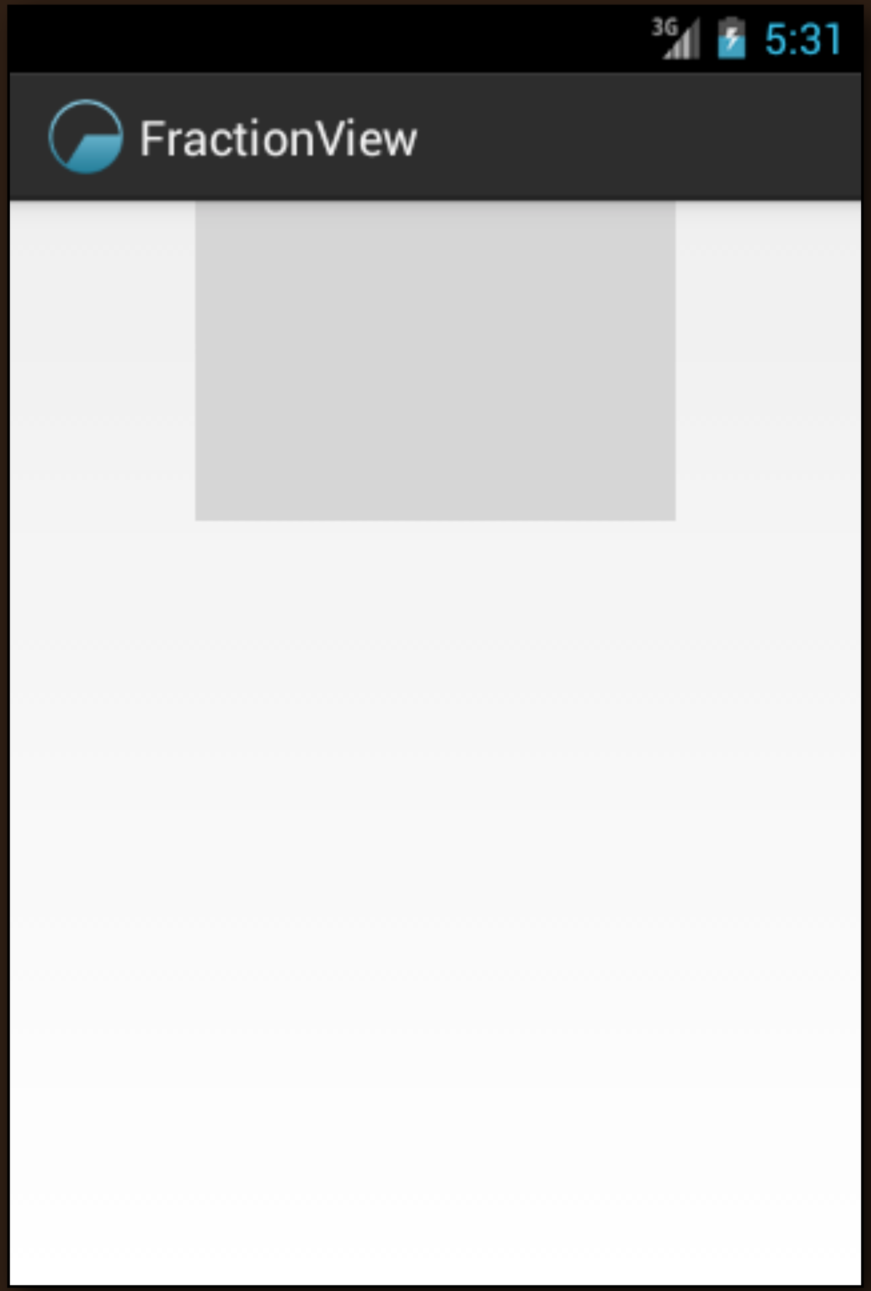
Activity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);
}
```

XML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent" >

  <com.sqisland.android.fraction_view.FractionView
    android:id="@+id/fraction"
    android:layout_width="180dp"
    android:layout_height="120dp" />

</LinearLayout>
```

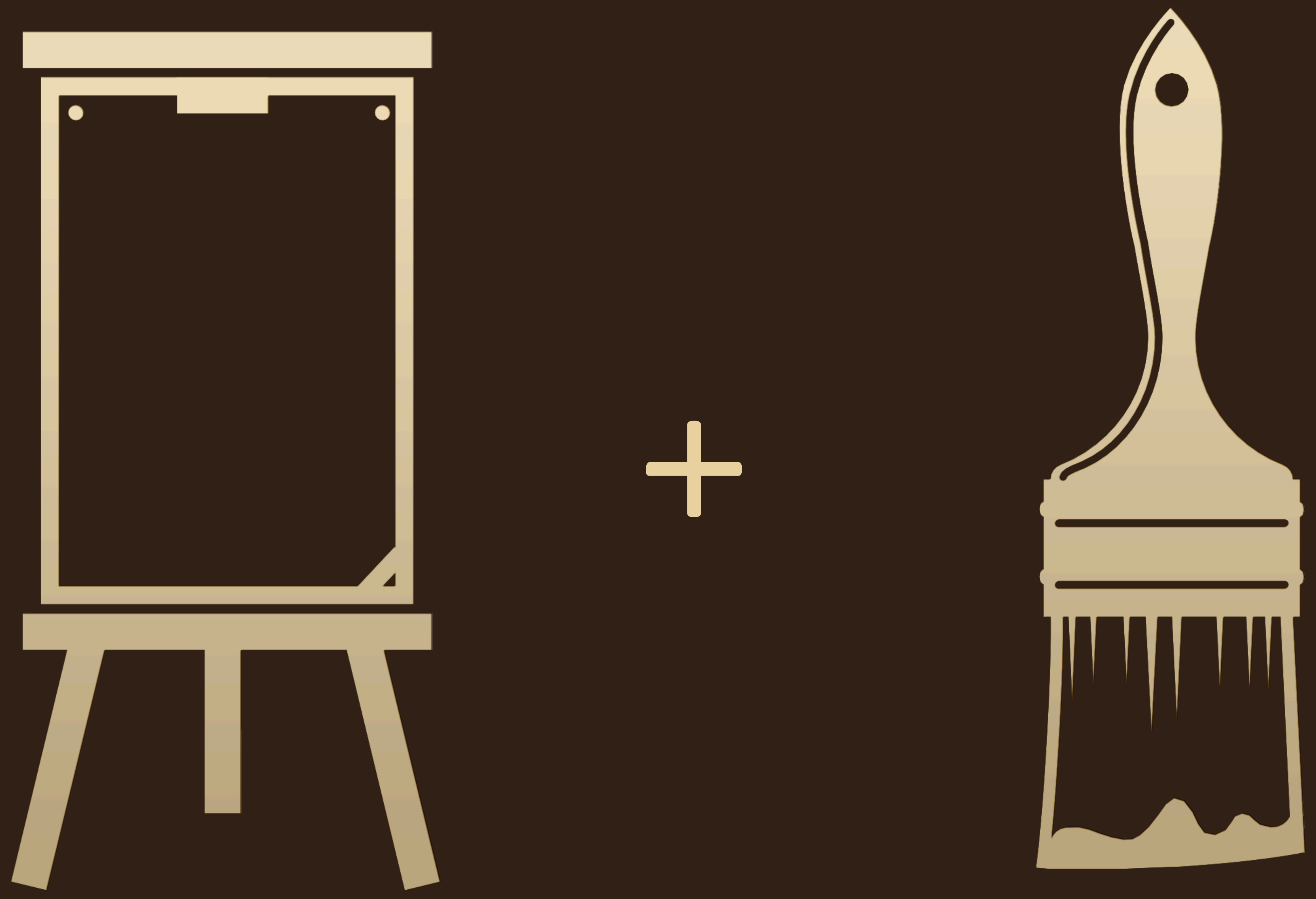
Full class name



3G
5:31
FractionView

Drawing

Drawing



Canvas

Draw

drawLine
drawRect
drawCircle
drawPath
drawText
drawBitmap

Transform

translate
rotate
scale
skew
save
restore

@chiuki

Icon credit: Okan Benn, from The Noun Project

Paint

Style

FILL

STROKE

FILL_AND_STROKE

Color

Stroke width

Cap

Join

Shader

MaskFilter

PathEffect

Xfermode

Icon credit: Yellow, from The Noun Project

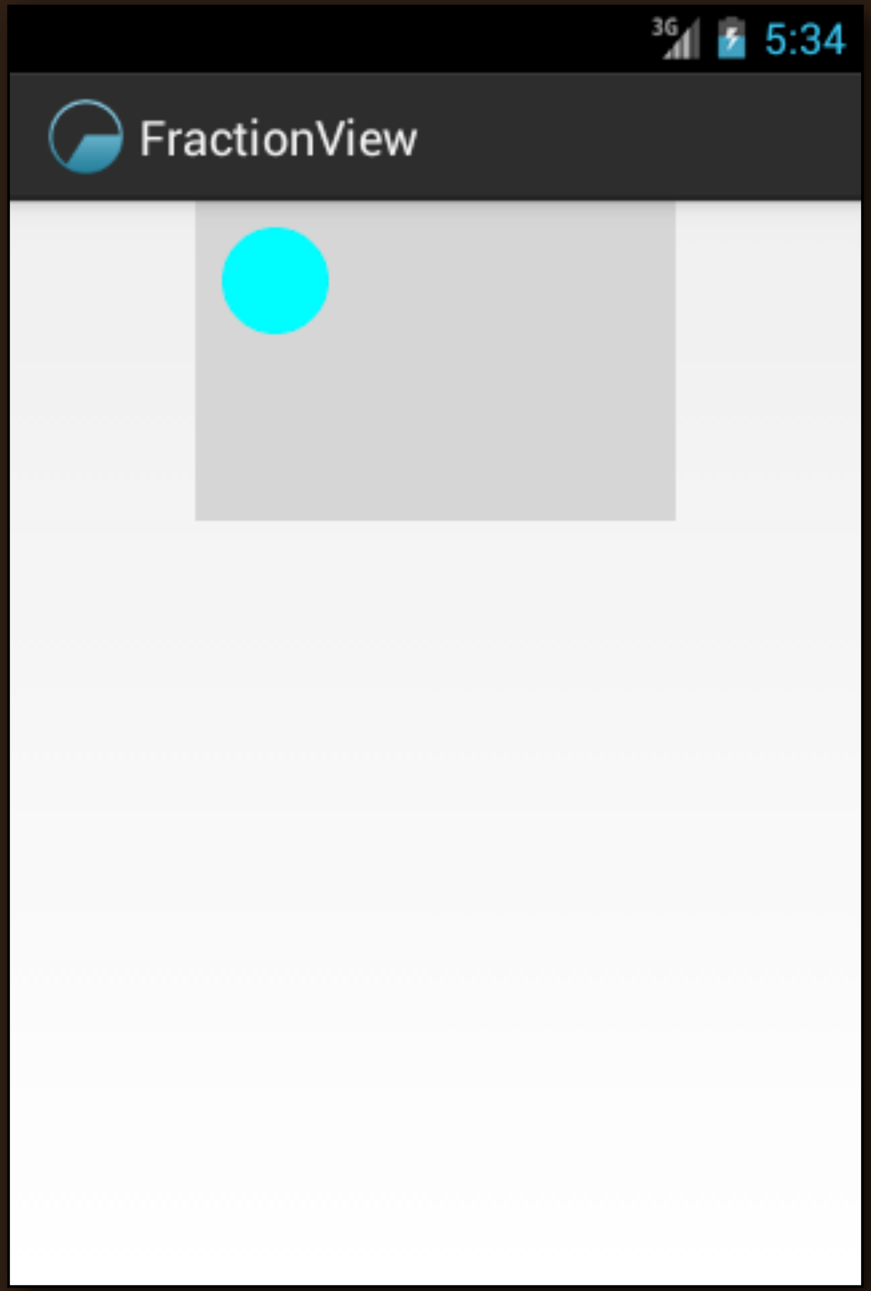
onDraw

```
protected void onDraw(Canvas canvas) {
  int cx = 30;
  int cy = 30;
  int radius = 20;
  canvas.drawCircle(cx, cy, radius, mCirclePaint);
}
```

Initialized in constructor

No new objects onDraw

```
public FractionView(Context context, AttributeSet attrs) {
  super(context, attrs);
  init();
}

// Called from all constructors
private void init() {
  mCirclePaint = new Paint(Paint.ANTI_ALIAS_FLAG);
  mCirclePaint.setColor(Color.CYAN);
  mCirclePaint.setStyle(Paint.Style.FILL);
}
```



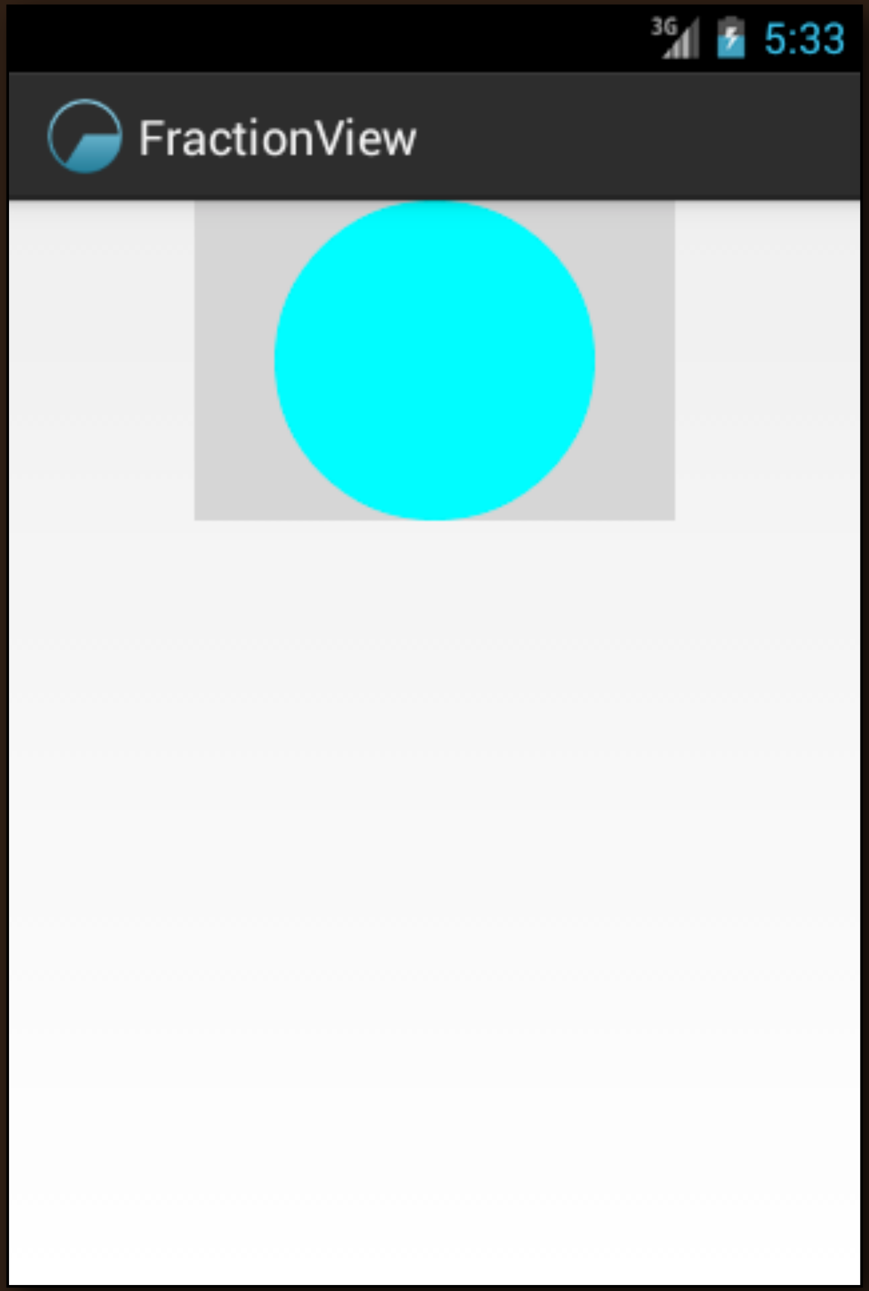
3G
5:34
FractionView

View size

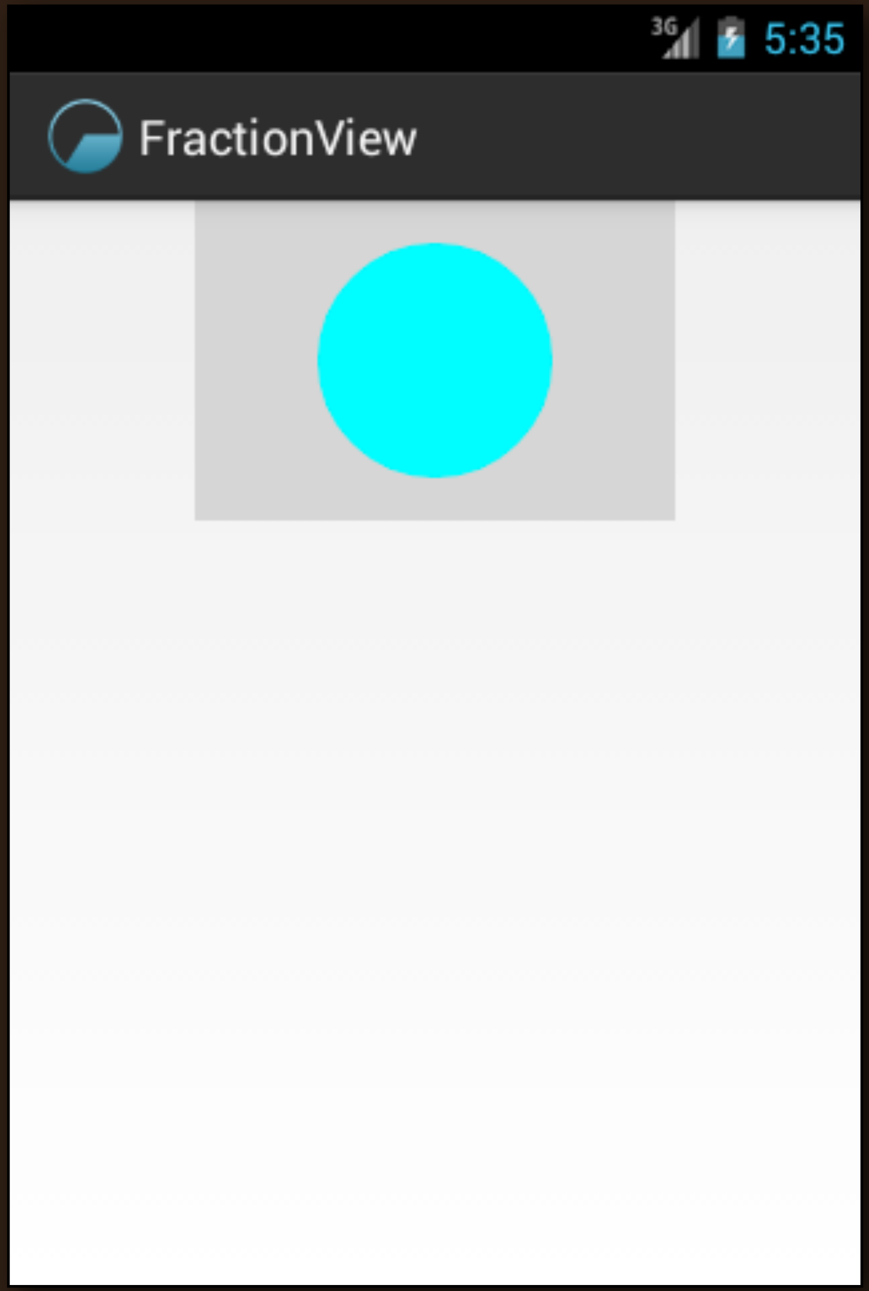
```
protected void onDraw(Canvas canvas) {
  int width = getWidth() - getPaddingLeft() - getPaddingRight();
  int height = getHeight() - getPaddingTop() - getPaddingBottom();
  int size = Math.min(width, height);
  int cx = width / 2 + getPaddingLeft();
  int cy = height / 2 + getPaddingTop();
  int radius = size / 2;
  canvas.drawCircle(cx, cy, radius, mCirclePaint);
}
```

Respect padding

No padding



With padding

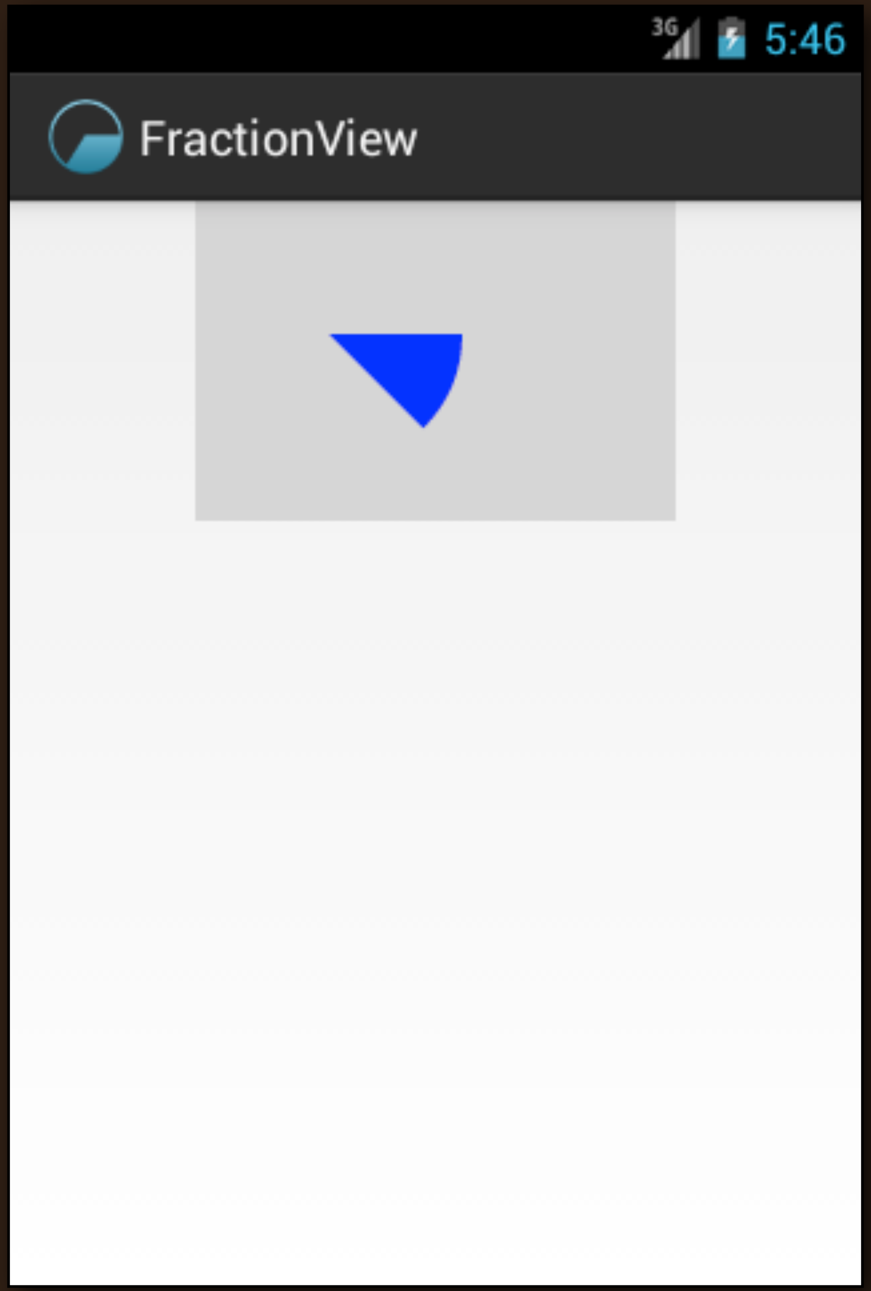


@chiuki

Draw fraction

Trying drawArc

```
protected void onDraw(Canvas canvas) {
  // canvas.drawCircle(cx, cy, radius, mCirclePaint);

  mSectorOval.top = 0;
  mSectorOval.left = 0;
  mSectorOval.bottom = 100;
  mSectorOval.right = 100;

  canvas.drawArc(mSectorOval, 0, 45, true, mSectorPaint);
}
```



@chiuki

drawArc

```java
protected void onDraw(Canvas canvas) {
  canvas.drawCircle(cx, cy, radius, mCirclePaint);

  mSectorOval.top = (height - size) / 2 + getPaddingTop();
  mSectorOval.left = (width - size) / 2 + getPaddingLeft();
  mSectorOval.bottom = mSectorOval.top + size;
  mSectorOval.right = mSectorOval.left + size;

  canvas.drawArc(mSectorOval, 0, 45, true, mSectorPaint);
}
```

Sweep Angle

```
protected void onDraw(Canvas canvas) {
  canvas.drawCircle(cx, cy, radius, mCirclePaint);

  mSectorOval.top = (height - size) / 2 + getPaddingTop();
  mSectorOval.left = (width - size) / 2 + getPaddingLeft();
  mSectorOval.bottom = mSectorOval.top + size;
  mSectorOval.right = mSectorOval.left + size;

  canvas.drawArc(mSectorOval, 0, getSweepAngle(), true, mSectorPaint);
}
```

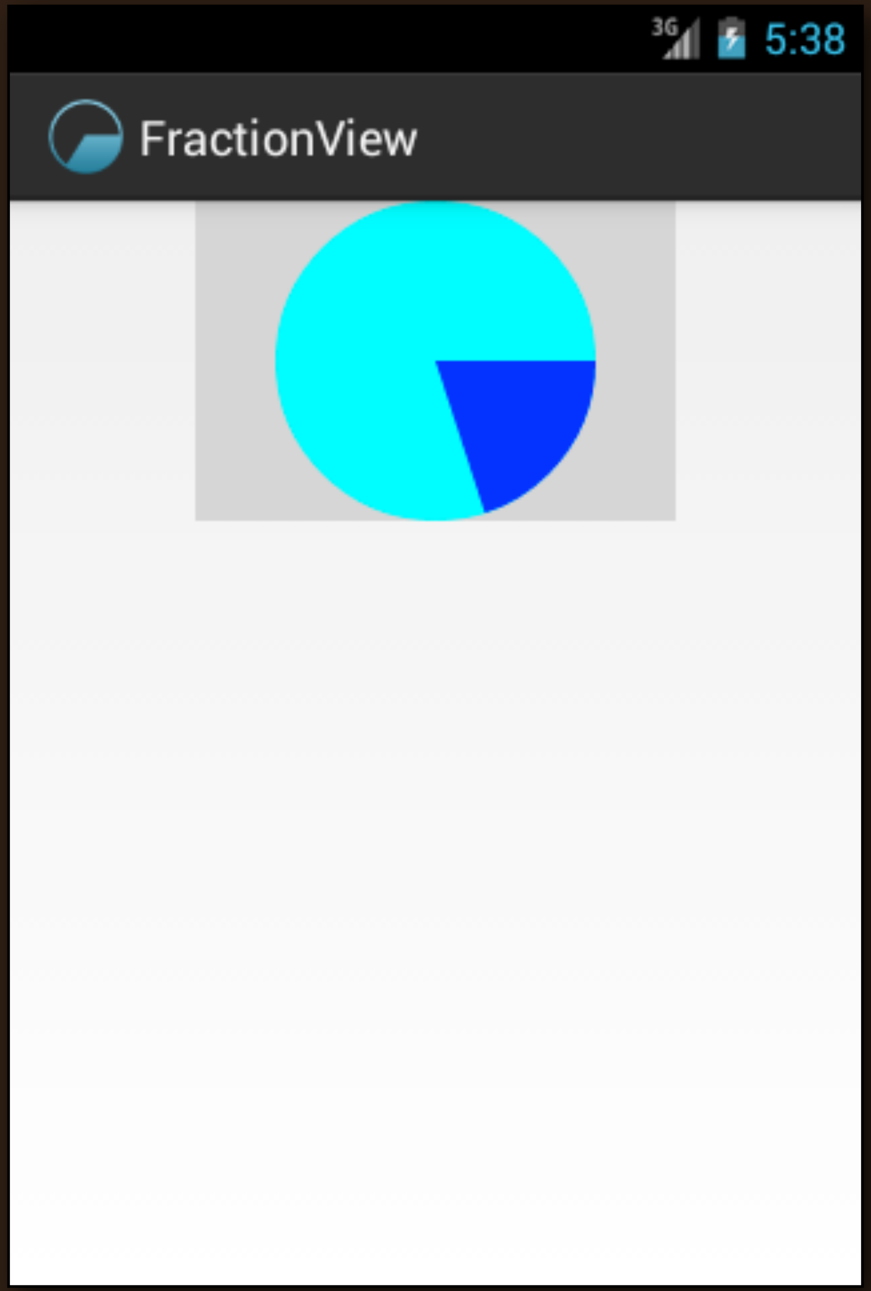
from numerator and denominator

Sweep Angle

```
protected void onDraw(Canvas canvas) {
  canvas.drawCircle(cx, cy, radius, mCirclePaint);

  mSectorOval.top = (height - size) / 2 + getPaddingTop();
  mSectorOval.left = (width - size) / 2 + getPaddingLeft();
  mSectorOval.bottom = mSectorOval.top + size;
  mSectorOval.right = mSectorOval.left + size;

  canvas.drawArc(mSectorOval, 0, getSweepAngle(), true, mSectorPaint);
}

private float getSweepAngle() {
  return mNumerator * 360f / mDenominator;
}
```

Default values

```
private int mNumerator = 1;
private int mDenominator = 5;
```


3G
5:38
FractionView

Setter

Setter

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mFractionView = (FractionView) findViewById(R.id.fraction);
    mFractionView.setFraction(1, 3);
}
```

setFraction

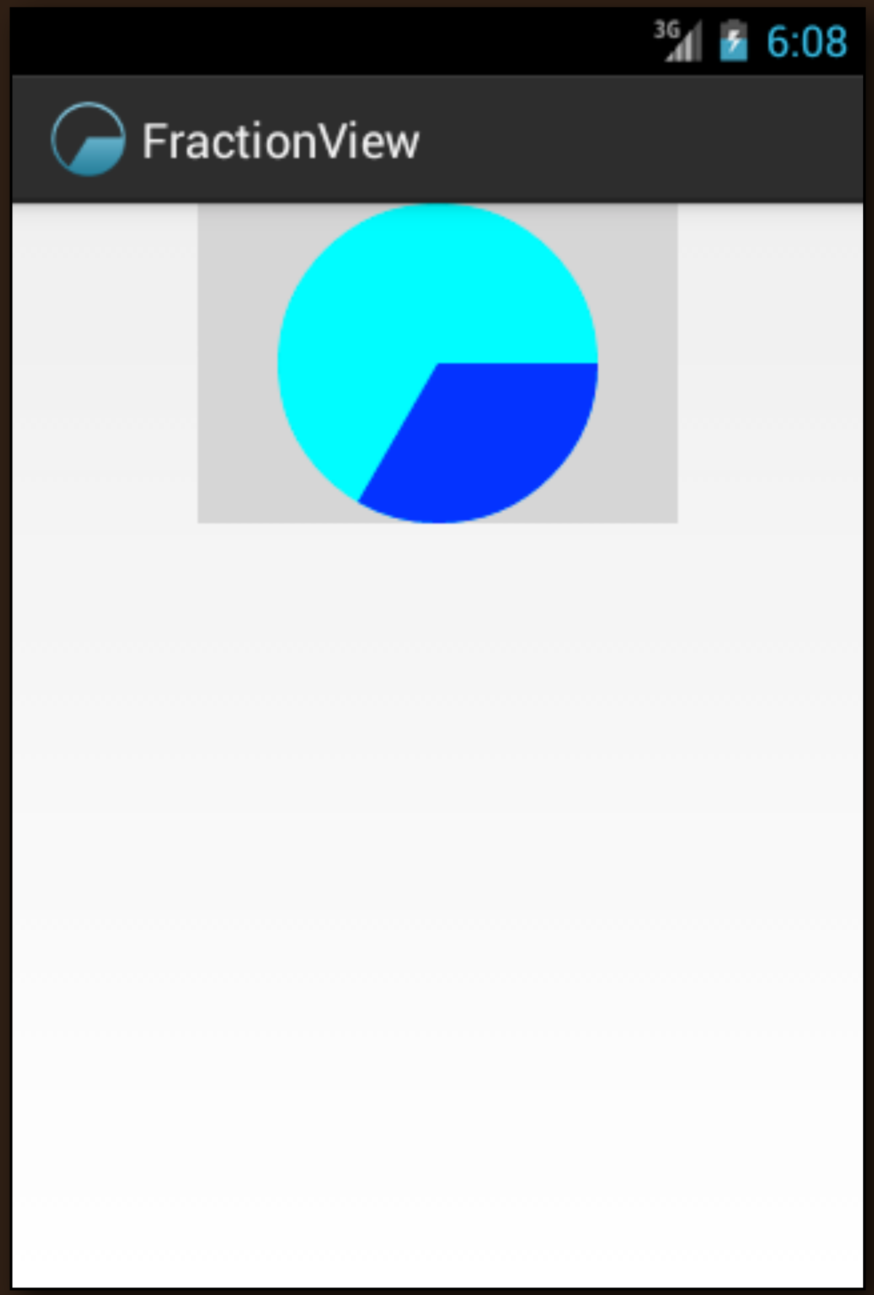
```
public void setFraction(int numerator, int denominator) {
    mNumerator = numerator;
    mDenominator = denominator;

    invalidate();
  }
```

Request redraw

Data integrity

```
public void setFraction(int numerator, int denominator) {
  if (numerator < 0) return;
  if (denominator <= 0) return;
  if (numerator > denominator) return;

  mNumerator = numerator;
  mDenominator = denominator;

  invalidate();
}
```

Prevent invalid state


3G
6:08
FractionView

Touch

Task:

Increment on touch

onTouchEvent

```
public boolean onTouchEvent(MotionEvent event) {
  if (event.getAction() != MotionEvent.ACTION_UP) {
    return true;
  }

  // Increment the numerator, cycling back to 0 when we have filled the
  // whole circle.
  int numerator = mNumerator + 1;
  if (numerator > mDenominator) {
    numerator = 0;
  }
  setFraction(numerator, mDenominator);

  return true;
}
```

Change view when finger is lifted

onTouchEvent

```
public boolean onTouchEvent(MotionEvent event) {
  if (event.getAction() != MotionEvent.ACTION_UP) {
    return true;
  }

  // Increment the numerator, cycling back to 0 when we have filled the
  // whole circle.
  int numerator = mNumerator + 1;
  if (numerator > mDenominator) {
    numerator = 0;
  }
  setFraction(numerator, mDenominator);

  return true;
}
```

Make sure we invalidate()

onTouchEvent

```
public boolean onTouchEvent(MotionEvent event) {
  if (event.getAction() != MotionEvent.ACTION_UP) {
    return true;
  }

  // Increment the numerator, cycling back to 0 when we have filled the
  // whole circle.
  int numerator = mNumerator + 1;
  if (numerator > mDenominator) {
    numerator = 0;
  }
  setFraction(numerator, mDenominator);

  return true;
}
```

Consume event

Extra credit:

Increment only if touch is inside circle

Save state

Rotate the screen

Windows: Ctrl + F12
Mac: Fn + Ctrl + F12

Save state

```
public Parcelable onSaveInstanceState() {
  Bundle bundle = new Bundle();
  bundle.putParcelable("superState", super.onSaveInstanceState());
  bundle.putInt("numerator", mNumerator);
  bundle.putInt("denominator", mDenominator);
  return bundle;
}
```

Restore state

```
public void onRestoreInstanceState(Parcelable state) {
  if (state instanceof Bundle) {
    Bundle bundle = (Bundle) state;
    mNumerator = bundle.getInt("numerator");
    mDenominator = bundle.getInt("denominator");
    super.onRestoreInstanceState(bundle.getParcelable("superState"));
  } else {
    super.onRestoreInstanceState(state);
  }
  setFraction(mNumerator, mDenominator);
}
```

invalidate()

Event listener

setOnChangeListener()

OnChangeListener

```
public interface OnChangeListener {
  public void onChange(int numerator, int denominator);
}

private OnChangeListener mListener = null;

public void setOnChangeListener(OnChangeListener listener) {
  mListener = listener;
}
```

setFraction

```
public void setFraction(int numerator, int denominator) {
  // Sanity check

  // Set value

  invalidate();

  if (mListener != null) {
    mListener.onChange(numerator, denominator);
  }
}
```

Call setFraction() internally to invalidate and notify listeners

Task:

Show fraction as text

Task:

Show fraction as text

Step 1: Add TextView to activity_main.xml

Task:

Show fraction as text

Step 2: Find TextView in MainActivity.java

Task:

Show fraction as text

Step 3: Set listener

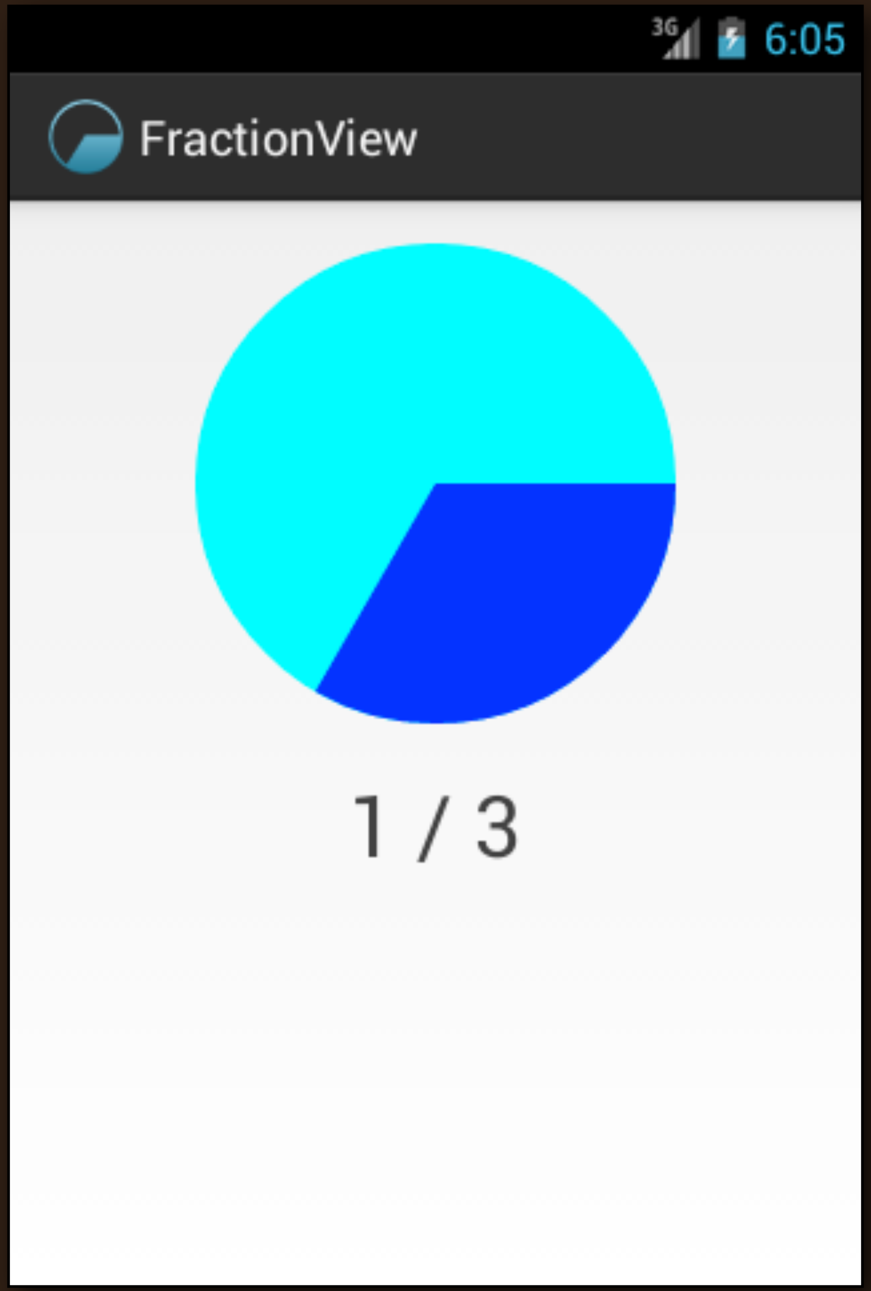
Set listener

```
mFractionView.setOnChangeListener(new FractionView.OnChangeListener() {
  public void onChange(int numerator, int denominator) {
    String fraction = getString(R.string.fraction, numerator, denominator);
    mTextView.setText(fraction);
  }
});
```

Formatted string

```
mFractionView.setOnChangeListener(new FractionView.OnChangeListener() {
  public void onChange(int numerator, int denominator) {
    String fraction = getString(R.string.fraction, numerator, denominator);
    mTextView.setText(fraction);
  }
});
```

```
<string name="fraction">%1$d / %2$d</string>
```


3G
6:05
FractionView
1 / 3

Summary

Constructors
Drawing
Setter
Touch
Save state
Event listener

Thank you!

Source code and slides:
http://bit.ly/FractionView

To learn more:
http://is.gd/AndroidCustomComp

Stay in touch:
http://eepurl.com/lR5uD
http://blog.sqisland.com
http://twitter.com/chiuki