Custom Views
Стили и Темы

Лекция №6

Дроздов Артём

С чего начинается View?

С наследования!

```
public class ClockView extends View {

    public ClockView(Context context) {
        this(context, null);
    }

    public ClockView(Context context, AttributeSet attrs) {
        this(context, attrs, 0);
    }

    public ClockView(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        //some initialization
    }
}
```

Хьюстон, у нас размеры!

Покажем всем, насколько мы большие!

```
@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
   setMeasuredDimensions(WIDTH_IN_PX, HEIGHT_IN_PX);
}
```

Порисуем?

Просто добавь onDraw!

```
private final Paint paint = new Paint();

@Override
protected void onDraw(Canvas canvas) {
    paint.setColor(0xff00ff00);
    canvas.drawLine(0, 0, getWidth(), getHeight(),  paint);
}
```

Три простых шага

1. Наследуемся от View (или существующего наследника View)
2. Определяем свои размеры в onMeasure
3. Рисуемся в onDraw


ЕСЛИ ЭТО ТАК ПРОСТО
ТО ПОЧЕМУ ВСЕ ТАК НЕ ДЕЛАЮТ?
risovach.ru

Нельзя просто так взять, и получить параметры в конструкторе

Атрибуты

Перечисляем доступные атрибуты

```
<declare-styleable name="ClockView">
    <attr name="hour_radius" format="dimension"/>
    <attr name="hour_width" format="dimension"/>
</declare-styleable>
```

Атрибуты

Определяем параметры в xml-layout’е

```
 <ru.mail.park.customviews.ClockView
xmlns:app="http://schemas.android.com/apk/res-auto"
    app:hour_width="4dp"
    app:hour_radius="25dp"/>
```

Атрибуты

Достаем атрибуты в конструкторе

```
TypedArray typedArray = context.obtainStyledAttributes(attrs,
    R.styleable.ClockView, defStyleAttr, 0);

float radius =
    typedArray.getDimension(R.styleable.ClockView_hour_radius, 0);

float width =
    typedArray.getDimension(R.styleable.ClockView_hour_width, 0);

typedArray.recycle();
```

Нельзя просто так взять, и сказать свой размер

MeasureSpec

Сначала поймем, с чем имеем дело

```
@Override
protected void onMeasure(int widthMeasureSpec,
                         int heightMeasureSpec) {

    int wMode = MeasureSpec.getMode(widthMeasureSpec);
    int hMode = MeasureSpec.getMode(heightMeasureSpec);

    int width = MeasureSpec.getSize(widthMeasureSpec);
    int height = MeasureSpec.getSize(heightMeasureSpec);
    //...
}
```

MeasureSpec

Какие бывают MeasureSpec?

```
int mode = MeasureSpec.getMode(...);
int size = MeasureSpec.getSize(...);
```

mode == MeasureSpec.EXACTLY => size - точный размер, сколько мы должны занимать

mode == MeasureSpec.AT_MOST => size - максимальный размер, сколько мы можем занимать

mode == MeasureSpec.UNSPECIFIED => измеряй себя как хочешь

MeasureSpec

9 оттенков квадратного

```
if (wMode == MeasureSpec.EXACTLY) {
   if (hMode == MeasureSpec.EXACTLY) {
       setMeasuredDimension(width, height);
   } else if (hMode == MeasureSpec.AT_MOST) {
       setMeasuredDimension(width, Math.min(width, height));
   } else {
       setMeasuredDimension(width, width);
   }
} else if (wMode == MeasureSpec.AT_MOST) {
   if (hMode == MeasureSpec.EXACTLY) {
       setMeasuredDimension(Math.min(width, height), height);
   } else if (hMode == MeasureSpec.AT_MOST) {
      int minSize = Math.min(width, height);
       setMeasuredDimension(minSize, minSize);
   } else {
       setMeasuredDimension(width, width);
   }
} else {
   if (hMode == MeasureSpec.UNSPECIFIED) {
       setMeasuredDimension(getSuggestedMinimumWidth(), getSuggestedMinimumHeight());
   } else {
       setMeasuredDimension(height, height);
   }
}
```

Нельзя просто так взять, и нарисоваться

Invalidate

onDraw птица гордая (и дорогая), пока не пнешь - не полетит

```
@Override
protected void onDraw(Canvas canvas) {
    state.update();

    drawClockArrow(canvas, hour);
    drawClockArrow(canvas, minute);
    drawClockArrow(canvas, second);

    invalidate();
}
```

Invalidate

onDraw птица гордая (и дорогая), пока не пнешь - не полетит

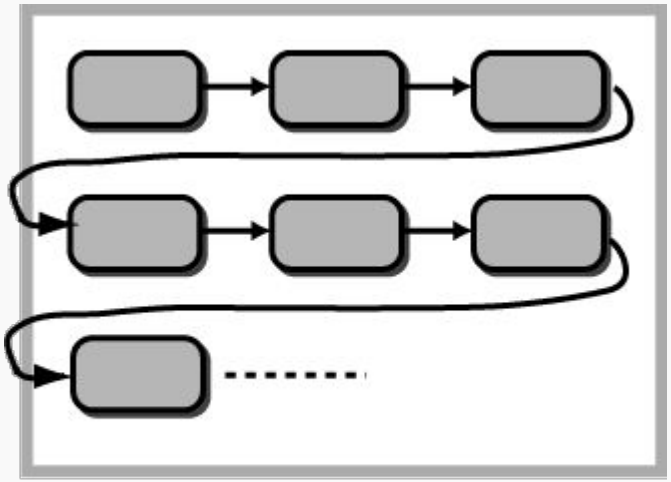
```
@Override
protected void onDraw(Canvas canvas) {
    state.update();

    drawClockArrow(canvas, hour);
    drawClockArrow(canvas, minute);
    drawClockArrow(canvas, second);

    postInvalidateDelayed(100);
}
```

Конец!
Парам-парам-пам!

Спасибо за внимание!

Custom Layout

Layout



Components Added by FlowLayout

Layout

XML

```
<ru.mail.park.customviews.FlowLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Text 1"/>

</ru.mail.park.customviews.FlowLayout>
```

https://developer.android.com/guide/topics/ui/declaring-layout.html

Layout

Обратная сторона onMeasure

```
@Override
protected void onMeasure(int widthMeasureSpec,  int heightMeasureSpec) {

    for (int i = 0; i < getChildCount(); i++) {
        View currentView = getChildAt(i);
        if (currentView.getVisibility() ==  GONE) {
            continue;
        }

        currentView.measure(childWidthMeasureSpec, childHeightMeasureSpec);
    }
    setMeasuredDimension(measuredWidth, measuredHeight);
}
```

https://developer.android.com/guide/topics/ui/declaring-layout.html

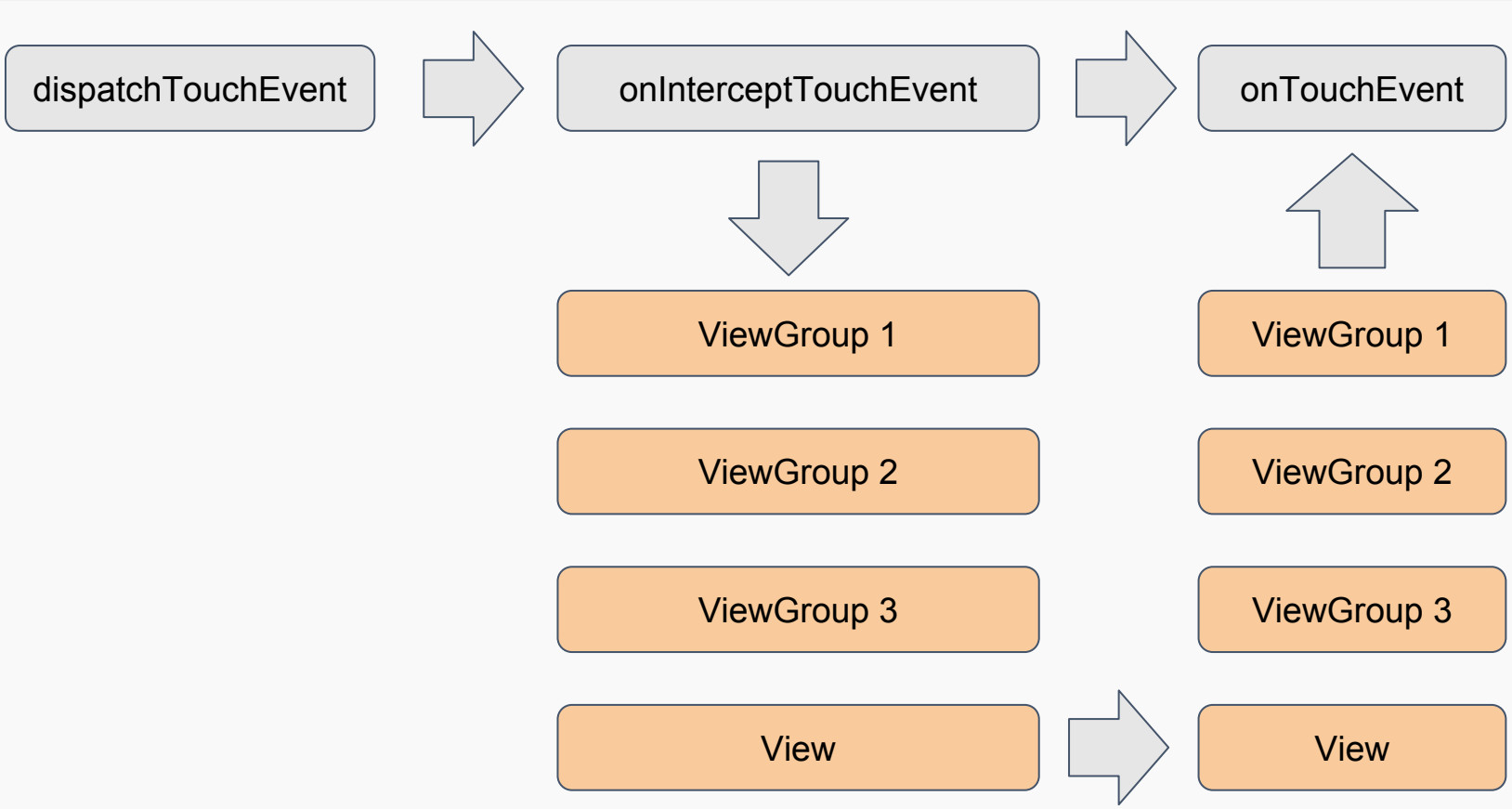
“

Лэйаут написать - не поле перейти

Артем Андреевич Дроздов

Обработка касаний

Распространение событий



https://developer.android.com/training/gestures/index.html

onInterceptTouchEvent


```
@Override
public boolean onInterceptTouchEvent(MotionEvent ev) {
    final int action = MotionEventCompat.getActionMasked(ev);

    if (action == MotionEvent.ACTION_CANCEL || action == MotionEvent.ACTION_UP) {
        isScrolling = false;
        return false;
    }

    switch (action) {
        case MotionEvent.ACTION_DOWN:
            startX = prevX = ev.getX();
            break;
        case MotionEvent.ACTION_MOVE: {
            if (isScrolling) {
                return true;
            }

            final float xDiff = Math.abs(ev.getX() - startX);

            if (xDiff > touchSlop) {
                isScrolling = true;
                return true;
            }
            break;
        }
    }

    return false;
}
```

https://developer.android.com/training/gestures/index.html

onTouchEvent


```
@Override
public boolean onTouchEvent(MotionEvent ev) {
    switch (MotionEventCompat.getActionMasked(ev)) {
        case MotionEvent.ACTION_MOVE:
            scrollBy((int) (prevX - ev.getX()), 0);
            prevX = ev.getX();
            break;
        case MotionEvent.ACTION_CANCEL:
        case MotionEvent.ACTION_UP:
            isScrolling = false;
            break;
    }
    return true;
}
```

https://developer.android.com/training/gestures/index.html

Конец!
Парам-парам-пам!

Спасибо за внимание!