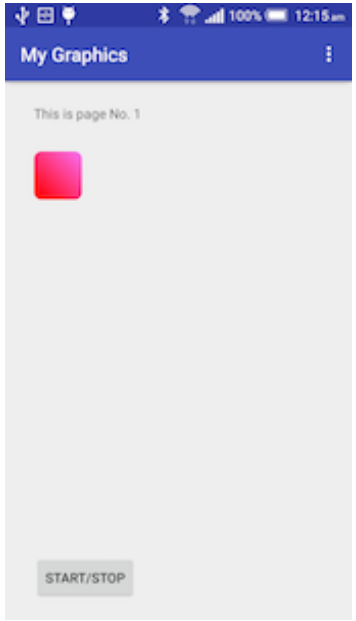
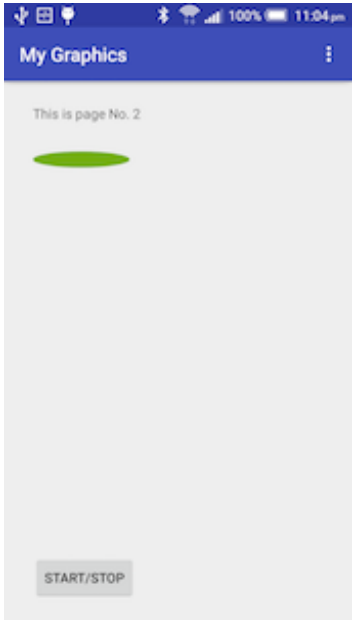
Graphics and Animation

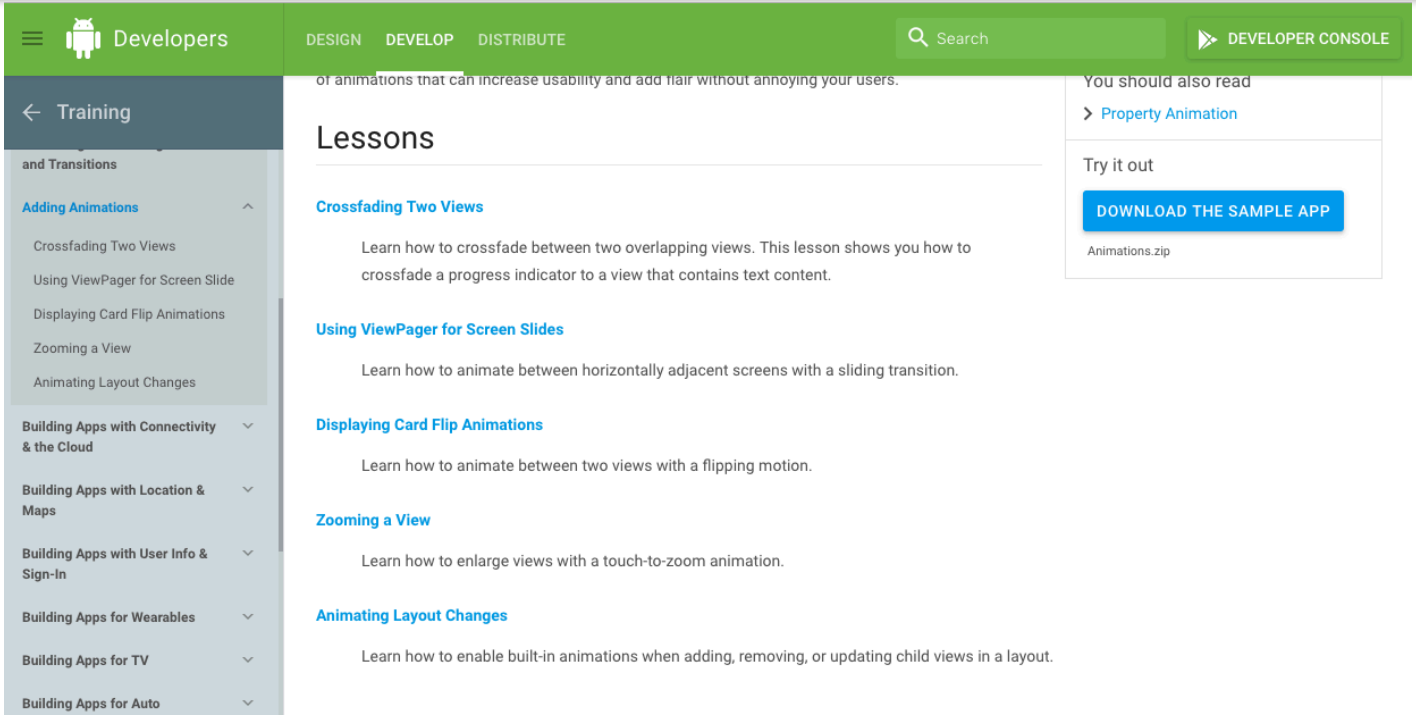
Lab 1

- ViewPager/ swipe gesture
- Drawable resources
- Extend View class





Android animations



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

“Using ViewPager for Screen Slides”

ViewPager

public class ViewPager

extends ViewGroup

java.lang.Object

↳ android.view.View

↳ android.view.ViewGroup

↳ android.support.v4.view.ViewPager

Layout manager that allows the user to flip left and right through pages of data. You supply an implementation of a PagerAdapter to generate the pages that the view shows.

https://developer.android.com/reference/android/support/v4/view/ViewPager.html

```
<android.support.v4.view.ViewPager xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/viewPager"
android:layout_width="match_parent"
android:layout_marginTop="50dp"
android:layout_height="match_parent"
/>
```

PagerAdapter

```
viewPager = (ViewPager) findViewById(R.id.viewPager);
pageFragmentPagerAdapter = new PageFragmentPagerAdapter(getSupportFragmentManager());
viewPager.setAdapter(pageFragmentPagerAdapter);
```

PagerAdapter

public abstract class PagerAdapter
extends Object

java.lang.Object
↳ android.support.v4.view.PagerAdapter

Known Direct Subclasses

FragmentPagerAdapter, FragmentStatePagerAdapter

https://developer.android.com/reference/android/support/v4/view/PagerAdapter.html

PagerAdapter

```
private class PageFragmentPagerAdapter extends FragmentStatePagerAdapter {

    public PageFragmentPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        return PageFragment.create(position);
    }

    @Override
    public int getCount() {
        return 2;
    }
}
```

PagerAdapter

Difference between FragmentPagerAdapter and FragmentStatePagerAdapter

What is the difference between `FragmentPagerAdapter` and `FragmentStatePagerAdapter`?

188

About `FragmentPagerAdapter` Google's guide says:

> This version of the pager is best for use when there are a handful of typically more static

asked 3 years ago

viewed 50897 times

active 1 year ago

- ViewPager associates each page with a key Object instead of working with Views directly. This key is used to track and uniquely identify a given page independent of its position in the adapter.
- Fragments will be created as soon as ViewPager becomes visible. This means that some ordinary call-backs for Fragment such as onPause() etc. will never be called, use OnPageChangeListener

http://stackoverflow.com/questions/18747975/difference-between-fragmentpageradapter-and-fragmentstatepageradapter

PagerFragment

```
public class PageFragment extends Fragment {

    public static final String ARG_PAGE = "ARG_PAGE";
    private int pageNumber;

    public PageFragment() {
    }

    public static PageFragment create(int pageNumber) {
        PageFragment pageFragment = new PageFragment();
        Bundle bundle = new Bundle();
        bundle.putInt(ARG_PAGE, pageNumber);
        pageFragment.setArguments(bundle);
        return pageFragment;
    }

    @Override
    public void onCreate(Bundle b) {
        super.onCreate(b);
        pageNumber = getArguments().getInt(ARG_PAGE);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_page, container, false);
        TextView textView = (TextView) v.findViewById(R.id.title);
        textView.setText("This is page No. " + Integer.toString(pageNumber + 1));
        return v;
    }
}
```

Static factory methods

```
public static PageFragment create(int pageNumber) {
    PageFragment pageFragment = new PageFragment();
    Bundle bundle = new Bundle();
    bundle.putInt(ARG_PAGE, pageNumber);
    pageFragment.setArguments(bundle);
    return pageFragment;
}
```

"Static factory method is simply a static method that returns an instance of a class." – Effective Java, Joshua Bloch

http://stackoverflow.com/questions/929021/what-are-static-factory-methods

Drawable resources

Shape Drawable

This is a generic shape defined in XML.

FILE LOCATION:

`res/drawable/filename.xml`

The filename is used as the resource ID.

COMPILED RESOURCE DATATYPE:

Resource pointer to a `GradientDrawable`.

RESOURCE REFERENCE:

In Java: `R.drawable.filename`

In XML: `@[package:]drawable/filename`

https://developer.android.com/guide/topics/resources/drawable-resource.html#Shape

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape=["rectangle" | "oval" | "line" | "ring"] >
    <corners
        android:radius="integer"
        android:topLeftRadius="integer"
        android:topRightRadius="integer"
        android:bottomLeftRadius="integer"
        android:bottomRightRadius="integer" />
    <gradient
        android:angle="integer"
        android:centerX="float"
        android:centerY="float"
        android:centerColor="integer"
        android:endColor="color"
        android:gradientRadius="integer"
        android:startColor="color"
        android:type=["linear" | "radial" | "sweep"]
        android:useLevel=["true" | "false"] />
    <padding
        android:left="integer"
        android:top="integer"
        android:right="integer"
        android:bottom="integer" />
    <size
        android:width="integer"
        android:height="integer" />
    <solid
        android:color="color" />
    <stroke
        android:width="integer"
        android:color="color"
        android:dashWidth="integer"
        android:dashGap="integer" />
</shape>
```

Drawable resources

```
Drawable drawable = ContextCompat.getDrawable(getContext(), R.drawable.gradient_box);
ImageView imageView = (ImageView) v.findViewById(R.id.body);
imageView.setImageDrawable(drawable);
```

A) drawables *with* theme attributes

```
ContextCompat.getDrawable(getActivity(), R.drawable.name);
```

You'll obtain a styled Drawable as your Activity theme instructs. This is probably what you need.

B) drawables *without* theme attributes

```
ResourcesCompat.getDrawable(getResources(), R.drawable.name, null);
```

You'll get your unstyled drawable the old way.

Please note: `ResourcesCompat.getDrawable()` is **not** deprecated!

http://stackoverflow.com/questions/29041027/android-getresources-getdrawable-deprecated-api-22

Extend View class

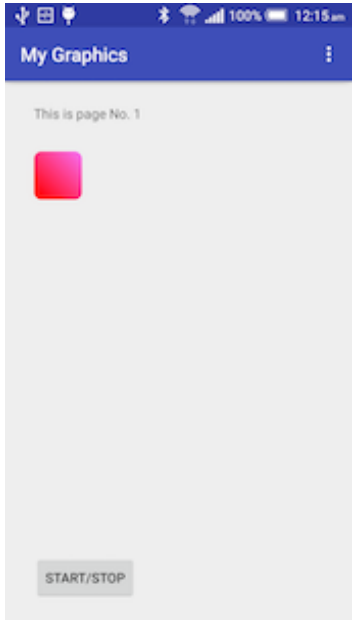
```
public class CustomDrawableView extends View {

    private ShapeDrawable mDrawable;

    public CustomDrawableView(Context context) {
        super(context);

        int x = 10;
        int y = 10;
        int width = 300;
        int height = 50;

        mDrawable = new ShapeDrawable(new OvalShape());
        mDrawable.getPaint().setColor(0xff74AC23);
        mDrawable.setBounds(x, y, x + width, y + height);
    }

    protected void onDraw(Canvas canvas) {
        mDrawable.draw(canvas);
    }
}
```

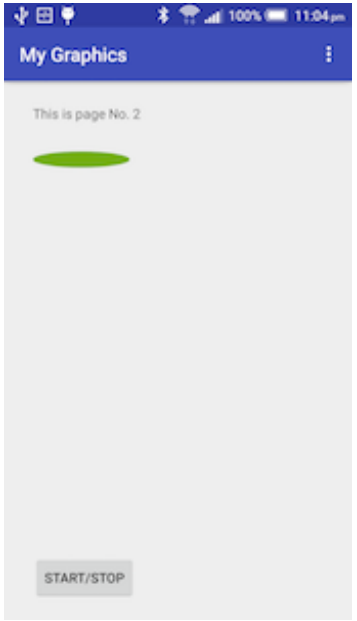
Add view programmatically

```
ImageView imageView = (ImageView) v.findViewById(R.id.body);
imageView.setVisibility(View.GONE);
RelativeLayout relativeLayout = (RelativeLayout) v.findViewById(R.id.container);
CustomDrawableView customDrawableView = new CustomDrawableView(getContext());
RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(ViewGroup.LayoutParams.WRAP_CONTENT,
params.setMargins(80, 80, 0, 0);
params.addRule(RelativeLayout.BELOW, textView.getId());
relativeLayout.addView(customDrawableView, params);
```

Lab 2

- Property animations
- View animation





Animations

- Property Animation
- View Animation
- Drawable Animation

https://developer.android.com/guide/topics/graphics/overview.html

Animation Resources

An animation resource can define one of two types of animations:

Property Animation

Creates an animation by modifying an object's property values over a set period of time with an `Animator`.

View Animation

There are two types of animations that you can do with the view animation framework:

- Tween animation: Creates an animation by performing a series of transformations on a single image with an `Animation`
- Frame animation: or creates an animation by showing a sequence of images in order with an `AnimationDrawable`.

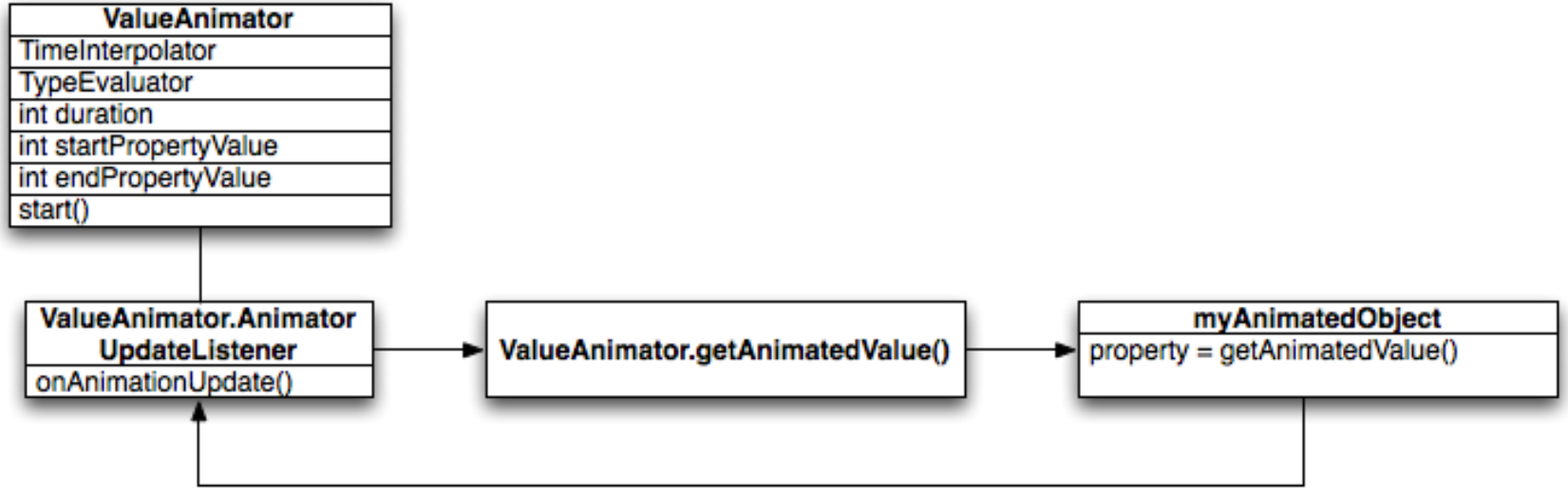
https://developer.android.com/guide/topics/resources/animation-resource.html

Property animation

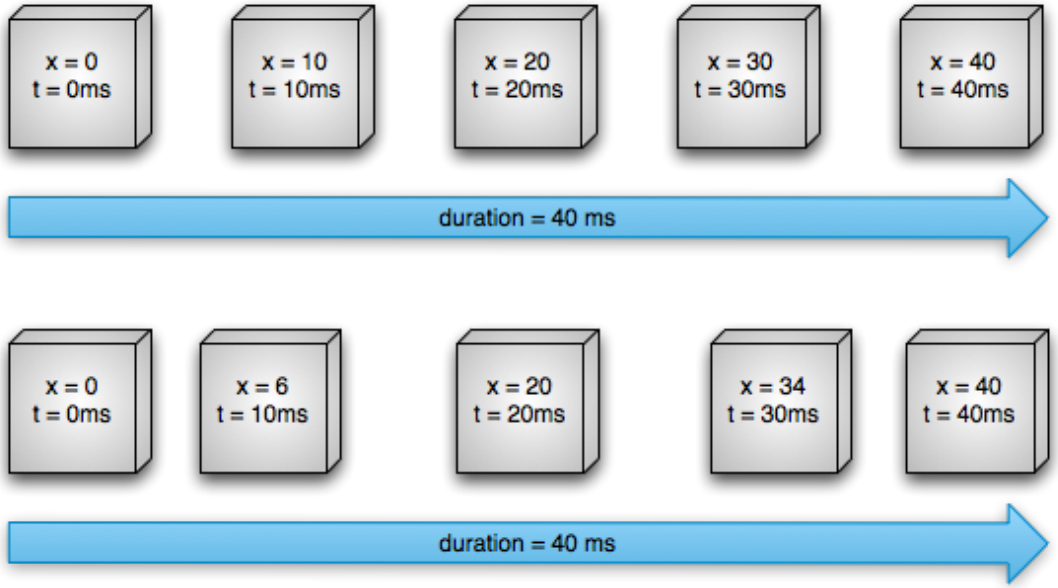
- Duration
- Time interpolation
- Repeat count and behavior
- Animator sets
- Frame refresh delay

https://developer.android.com/guide/topics/graphics/prop-animation.html

How animations are calculated?



Interpolators



Interpolators

`AccelerateDecelerateInterpolator`	An interpolator whose rate of change starts and ends slowly but accelerates through the middle.
`AccelerateInterpolator`	An interpolator whose rate of change starts out slowly and then accelerates.
`AnticipateInterpolator`	An interpolator whose change starts backward then flings forward.
`AnticipateOvershootInterpolator`	An interpolator whose change starts backward, flings forward and overshoots the target value, then finally goes back to the final value.
`BounceInterpolator`	An interpolator whose change bounces at the end.
`CycleInterpolator`	An interpolator whose animation repeats for a specified number of cycles.
`DecelerateInterpolator`	An interpolator whose rate of change starts out quickly and and then decelerates.
`LinearInterpolator`	An interpolator whose rate of change is constant.
`OvershootInterpolator`	An interpolator whose change flings forward and overshoots the last value then comes back.
`TimeInterpolator`	An interface that allows you to implement your own interpolator.

Evaluators

- IntEvaluator
- FloatEvaluator
- ArgbEvaluator
- TypeEvaluator

```
public class FloatEvaluator implements TypeEvaluator {

    public Object evaluate(float fraction, Object startValue, Object endValue) {
        float startFloat = ((Number) startValue).floatValue();
        return startFloat + fraction * (((Number) endValue).floatValue() - startFloat);
    }
}
```

There is only one method to implement in the TypeEvaluator interface, the evaluate() method. This allows the animator that you are using to return an appropriate value for your animated property at the current point of the animation.

Lab example

```
<set>
    <objectAnimator
        android:duration="500"
        android:propertyName="x"
        android:repeatCount="infinite"
        android:repeatMode="reverse"
        android:valueTo="800"
        android:valueType="floatType" />
    <objectAnimator
        android:duration="500"
        android:propertyName="y"
        android:repeatCount="infinite"
        android:repeatMode="reverse"
        android:valueTo="300"
        android:valueType="floatType" />
</set>
<objectAnimator
    android:duration="500"
    android:propertyName="alpha"
    android:repeatCount="infinite"
    android:repeatMode="reverse"
    android:valueTo="0f" />
```

Lab example

```
set = (AnimatorSet) AnimatorInflater.loadAnimator(getContext(), R.animator.property_animator);
set.setTarget(imageView);

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if (!isAnimatorSetOn) {
            set.start();
            isAnimatorSetOn = true;
        } else {
            set.cancel();
            isAnimatorSetOn = false;
        }
    }
});
```

View animation

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <scale
        android:interpolator="@android:anim/accelerate_decelerate_interpolator"
        android:fromXScale="1.0"
        android:toXScale="1.4"
        android:fromYScale="1.0"
        android:toYScale="0.6"
        android:pivotX="50%"
        android:pivotY="50%"
        android:fillAfter="false"
        android:duration="700" />
```

```
        <set
            android:interpolator="@android:anim/accelerate_interpolator"
            android:startOffset="700">
            <scale
                android:fromXScale="1.4"
                android:toXScale="0.0"
                android:fromYScale="0.6"
                android:toYScale="0.0"
                android:pivotX="50%"
                android:pivotY="50%"
                android:duration="400" />
            <rotate
                android:fromDegrees="0"
                android:toDegrees="-45"
                android:toYScale="0.0"
                android:pivotX="50%"
                android:pivotY="50%"
                android:duration="400" />
    </set>
</set>
```

https://developer.android.com/guide/topics/resources/animation-resource.html#View

View animation

```
ImageView image = (ImageView) findViewById(R.id.image);
Animation hyperspaceJump = AnimationUtils.loadAnimation(this, R.anim.hyperspace_jump);
image.startAnimation(hyperspaceJump);
```