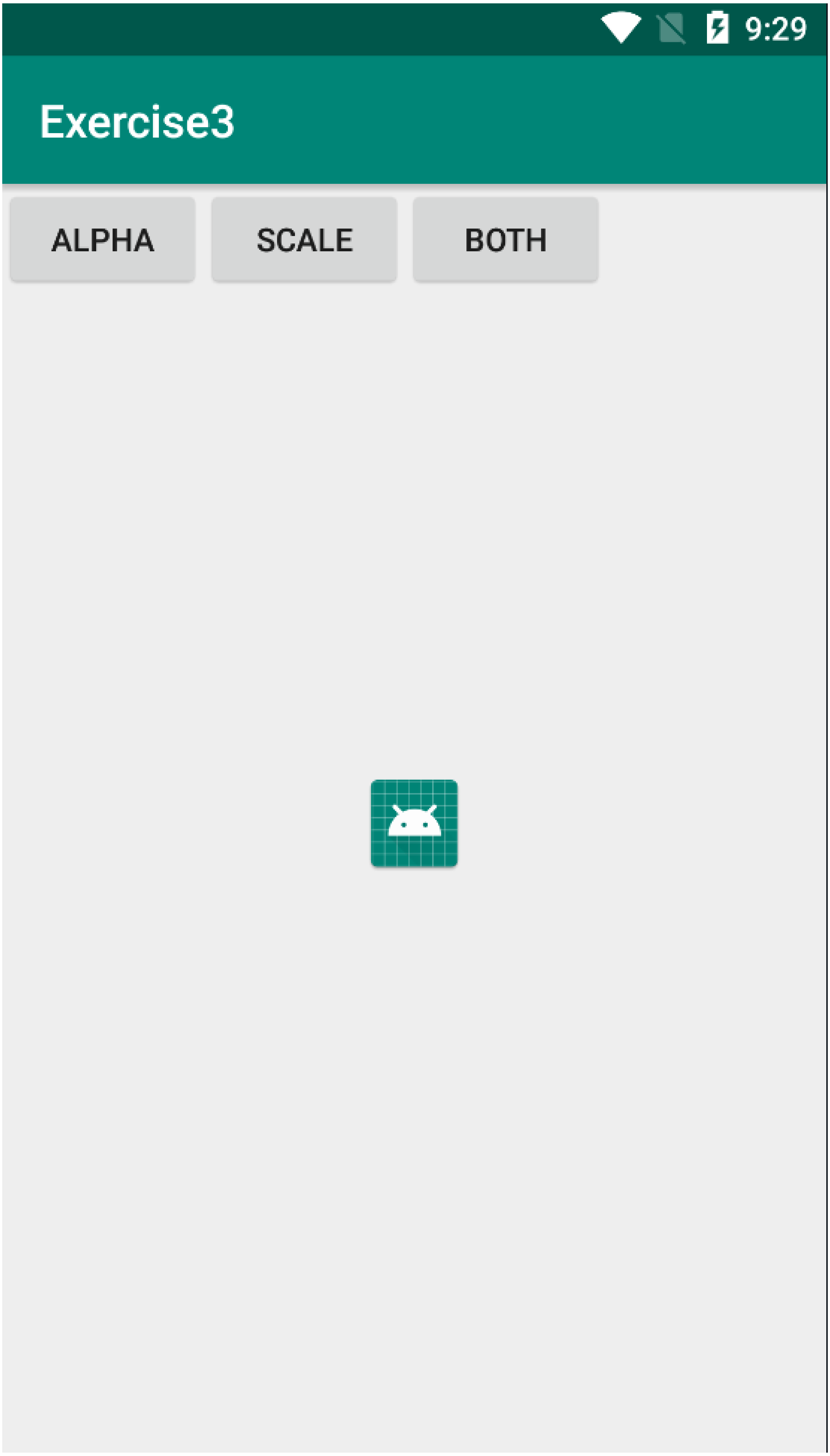
Exercise3

View Animation

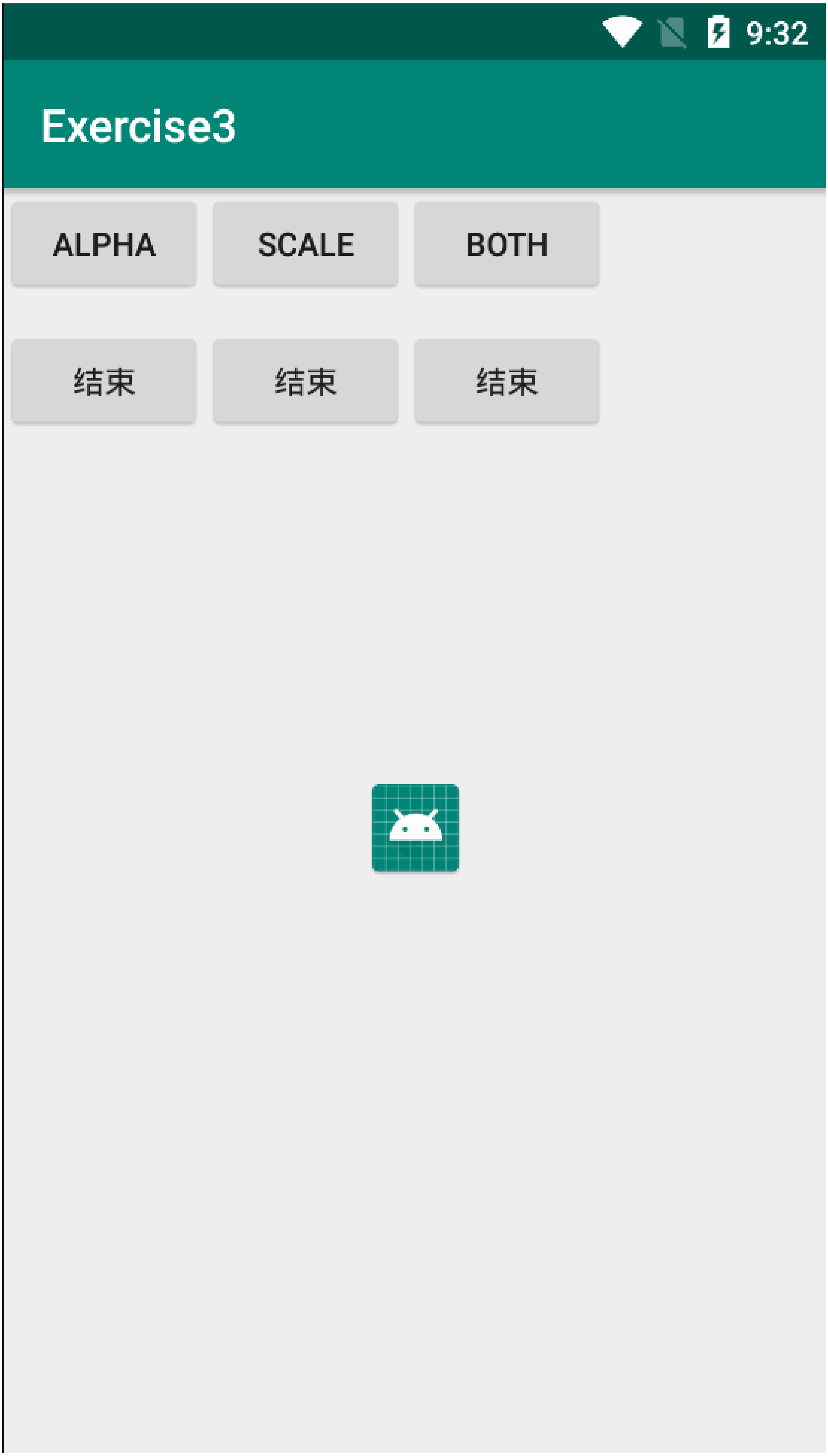

9:29
Exercise3
ALPHA
SCALE
BOTH

点击上方三个按钮，可播放对应动画

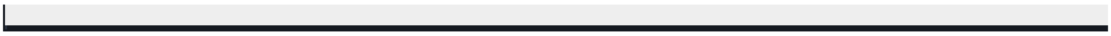
部分代码：（这里使用了AnimationSet实现两个动画的同时播放）

```
showBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AlphaAnimation alphaAnimation = new AlphaAnimation(0,1);
        ScaleAnimation scaleAnimation = new
ScaleAnimation(0f,1f,0f,1f,Animation.RELATIVE_TO_SELF,0.5f,Animation.RELATIVE_TO_SELF,0.5f)
;
        AnimationSet animationSet = new AnimationSet(true);
        animationSet.addAnimation(alphaAnimation);
        animationSet.addAnimation(scaleAnimation);
        animationSet.setDuration(1000);
        mImage.startAnimation(animationSet);
    }
});
```

Property Animation



9:32
Exercise3
ALPHA
SCALE
BOTH
结束
结束
结束



点上方三个按钮可播放/暂停三种动画，下方按钮点击可结束播放动画

重要成员函数：

```
private void InitAnimator() {
    ObjectAnimator mScaleXanimator = ObjectAnimator.ofFloat(mImage,"scaleX",0f,1f);
    ObjectAnimator mScaleYanimator = ObjectAnimator.ofFloat(mImage,"scaleY",0f,1f);
    ObjectAnimator mAlphaAnimator = ObjectAnimator.ofFloat(mImage,"alpha",0,1);
    mScaleXanimator.setRepeatCount(ValueAnimator.INFINITE);
    mScaleYanimator.setRepeatCount(ValueAnimator.INFINITE);
    mScaleXanimator.setInterpolator(new LinearInterpolator());
    mScaleYanimator.setInterpolator(new LinearInterpolator());
    mScaleXanimator.setDuration(2000);
    mScaleYanimator.setDuration(2000);
    mScaleXanimator.setRepeatMode(ValueAnimator.REVERSE);
    mScaleYanimator.setRepeatMode(ValueAnimator.REVERSE);
    mAlphaAnimator.setRepeatCount(ValueAnimator.INFINITE);
    mAlphaAnimator.setInterpolator(new LinearInterpolator());
    mAlphaAnimator.setDuration(2000);
    mAlphaAnimator.setRepeatMode(ValueAnimator.REVERSE);
    animatorSet1 = new AnimatorSet();
    animatorSet2 = new AnimatorSet();
    animatorSet3 = new AnimatorSet();
    animatorSet1.play(mAlphaAnimator);
    animatorSet2.playTogether(mScaleXanimator,mScaleYanimator);
    animatorSet3.playTogether(mScaleXanimator,mScaleYanimator,mAlphaAnimator);
}
```

暂停/结束实现逻辑：

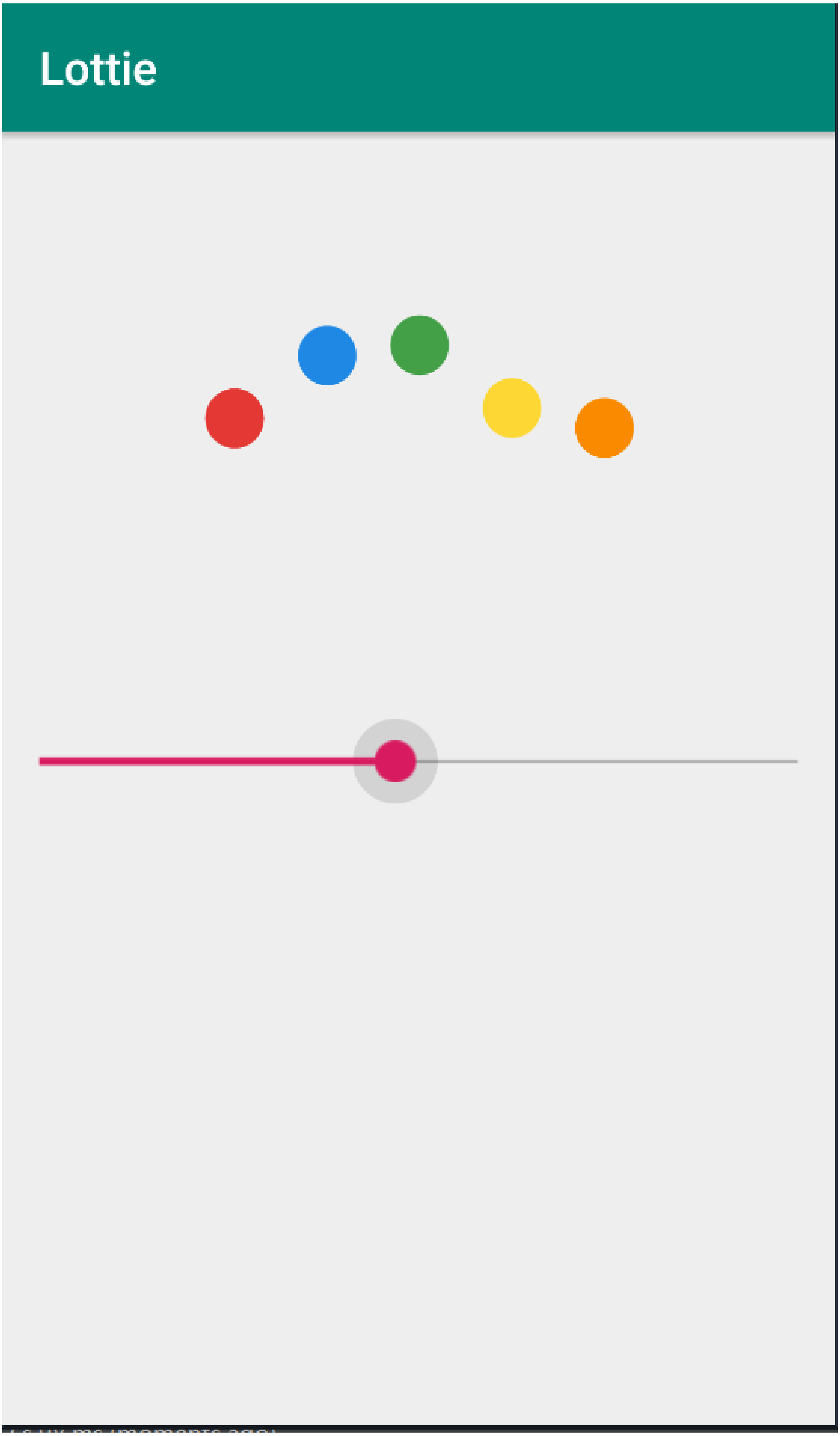
```
mShowScaleBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(!animatorSet2.isStarted()){
            animatorSet2.start();
        }
        else if(animatorSet2.isPaused()){
            animatorSet2.resume();
        }
        else{
            animatorSet2.pause();
        }
    }
});
```

JSON animation + SeekBar

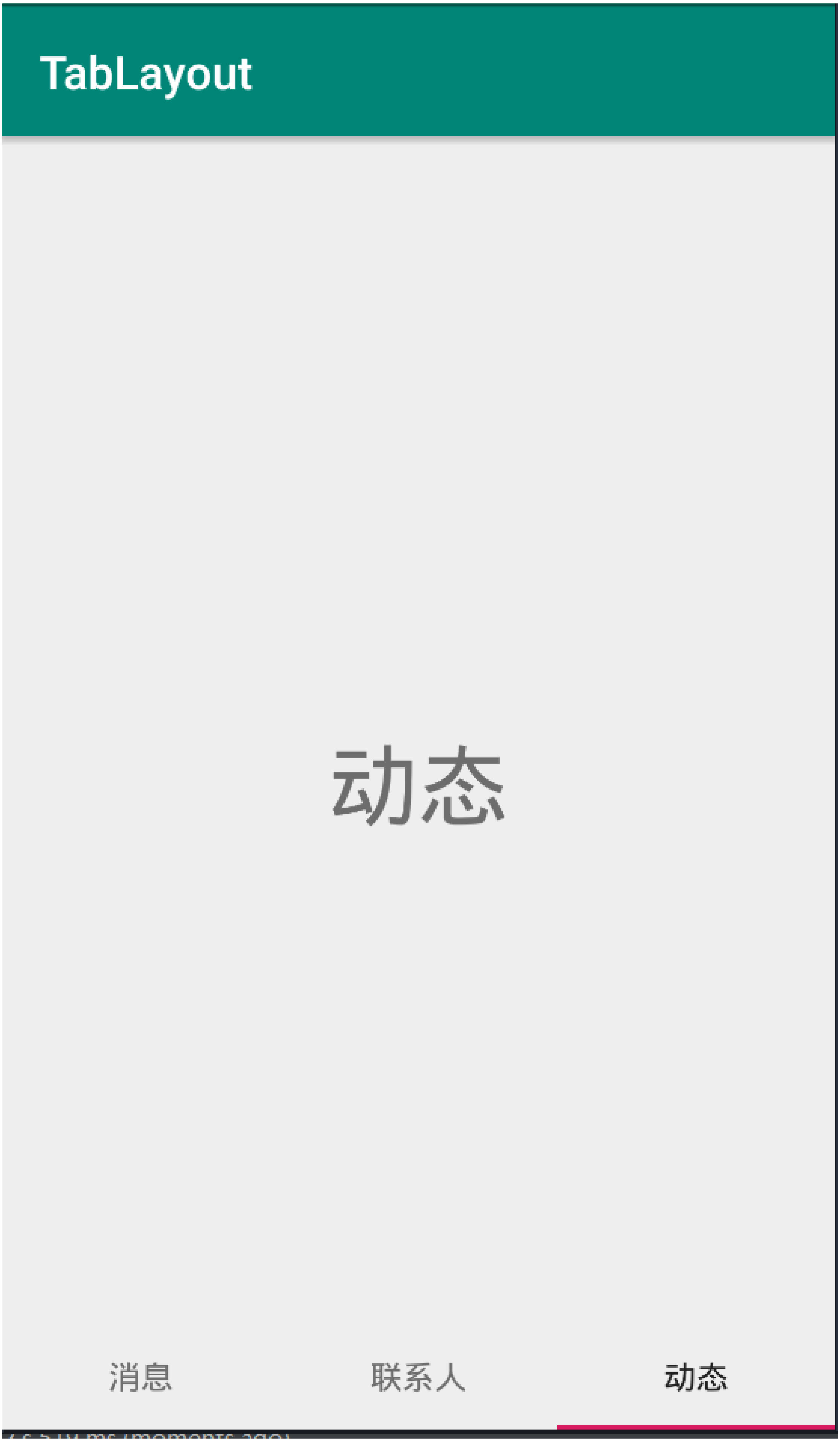
拖动进度条可将动画停到对应位置

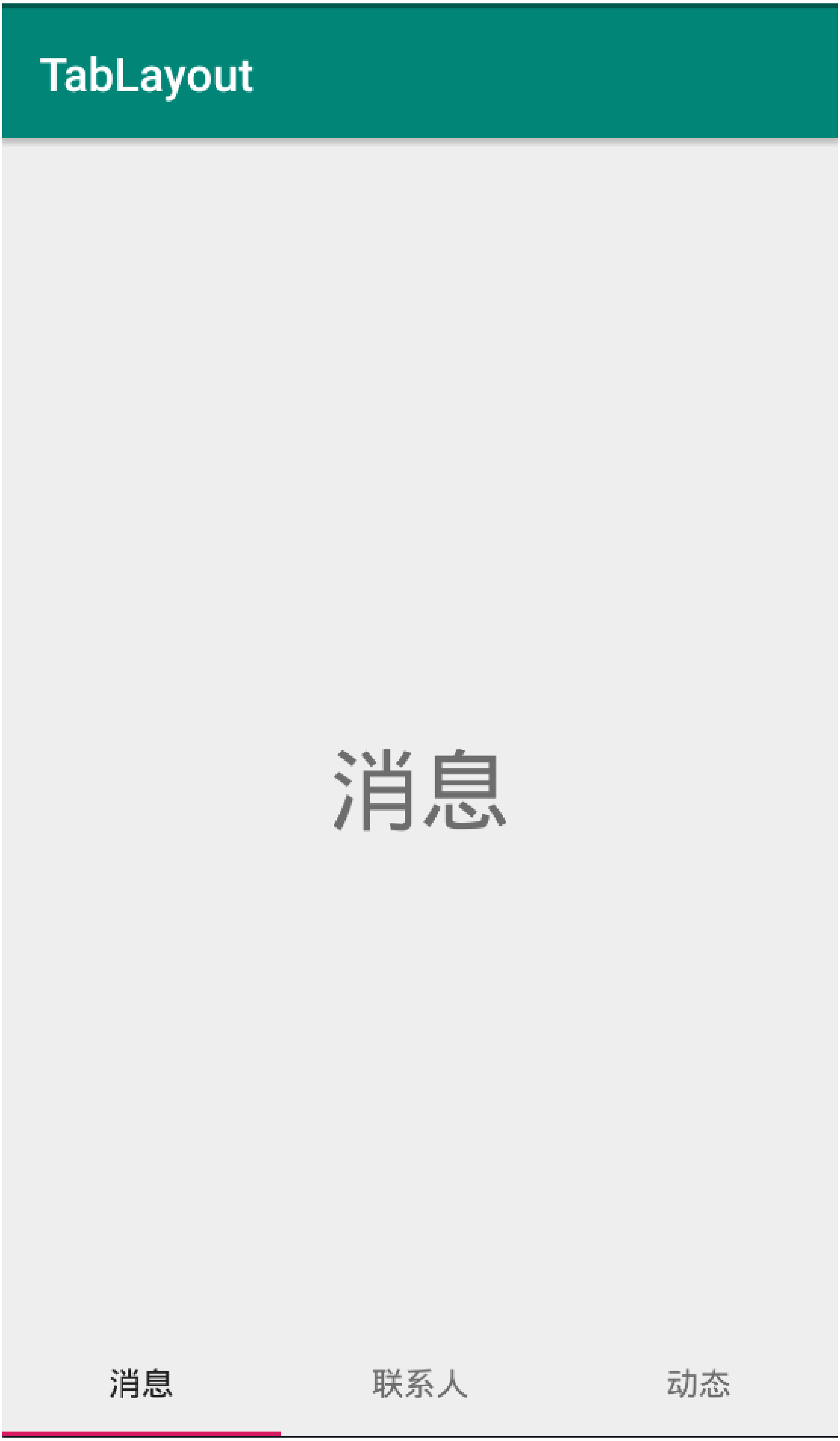
SeekBar.OnSeekBarChangeListener重要成员函数：

```
public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
    mlv.pauseAnimation();
    mlv.setProgress(progress/100f);
}
```


Lottie

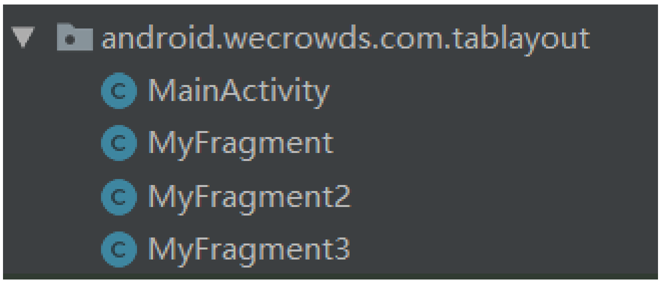
TabLayout + ViewPager


TabLayout
动态
消息
联系人
动态



TabLayout
消息
消息
联系人
动态

可实现tab切换，显示不同的fragment



android.wecrowds.com.tablayout
MainActivity
MyFragment
MyFragment2
MyFragment3

共建立了3个Fragment类，分别对应三个tab的视图

重要代码：

```
mTitle = new ArrayList<>();
mTitle.add("消息");
mTitle.add("联系人");
mTitle.add("动态");

mFragment = new ArrayList<>();
mFragment.add(new MyFragment());
mFragment.add(new MyFragment2());
mFragment.add(new MyFragment3());

mViewPager.setAdapter(new FragmentPagerAdapter(getSupportFragmentManager()) {
    @Override
    public Fragment getItem(int position) {
        return mFragment.get(position);
    }

    @Override
    public int getCount() {
        return mFragment.size();
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return mTitle.get(position);
    }
});

mytab.setupWithViewPager(mViewPager);
```