院 系 数据科学与计算机学院 学 号 16337233 姓 名 王凯祺

【实验题目】**Android动画实验**

【实验目的】学习Android的逐帧动画、补间动画、属性动画、Shape、菜单、触摸事件等 。

【实验准备】

直接从网上或从上传作业的网站上下载并安装Android Studio。

【实验说明】

先通过看视频birdFly.mp4了解本次实验要完成的任务。下面的实验内容把本次实验分成了18个功能来完成，可以只完成其中一部分功能，并在最后指明完成了哪些功能。

【实验内容】

**核心设计**

1、利用逐帧动画（FrameAnimation）实现小鸟动画（ImageView）。

编程说明：采用RelativeLayout，其background加上白云。

参考资料：动画.pdf（逐帧动画）

参考源码：FrameAnim

使用资源：sprite1.png~sprite8.png

2、利用TouchDown事件和TranslateAnimation实现在屏幕上引导小鸟飞行。

参考资料：动画.pdf(补间动画)

画布.pdf(画图的例子) -- Touch事件  
\* TouchDown可以用于Activity

参考源码：TweenAnimCases, DrawAllOnCanvas

3、利用ImageView的setRotationY()方法控制鸟飞行的方向。

4、控制鸟飞行的范围不要超出显示范围。

参考资料：动画.pdf(附录、获取屏幕尺寸)

\* 可以加一个宽高match\_parent的ImageView(后面13可以用到)，得到它的宽度和高度，或者直接使用屏幕尺寸（好像不准确）。

5、通过取得屏幕大小控制鸟飞行的速度。 如果整个屏幕对角飞行时间为5秒，按飞行距离设置每段飞行的时间，距离越短，时间越短。

6、采用属性动画（500ms）使得鸟飞行方向平滑变化（取代第3条）。注意：飞行动画的启动要延迟500ms以配合本动画。

参考资料：动画.pdf(属性动画)

参考源码：PropAnimCase

**菜单设计**

7、增加选项菜单：开始录像、结束录像、录像重放、结束回放

参考资料：对话框和菜单.pdf (菜单)

参考源码：Menu

**录像小圆点设计**

8、增加一个表示正在录制的小红点（shape）。设计一个0.5秒显示红点0.5秒显示灰色或透明点的逐帧动画，动画的图片直接使用drawable中的XML文件。

参考资料：动画.pdf（逐帧动画），可绘制类.pdf（ShapeDrawable）

参考源码：FrameAnim，ShapeDrawables

9、只有在录像时才显示小红点，结束录像时不显示小红点。

参考资料：列表与布局.pdf (p43,帧布局,setVisibility)

**录制设计**

10、用一个类Pos和一个ArrayList<Pos>记录所走路径的方向和停留地点。

11、每次开始录像时清除原来的记录。

**重播设计**

12、通过菜单“录像重放”重新显示飞行过程。  
编程参考：（1）开始重放时要清除Path对象。  
 （2）动画通过onAnimationEnd()事件接续。

参考资料：动画.pdf(属性动画)

参考源码：PropAnimCase

13、在每条线路飞行前显示飞行路线。

编程说明：（1）增加一个自定义View作为背景（可以用第4步中增加的ImageView），并在其中定义一个Path对象

（2）每次开始一段线路时增加Path对象的点(lineTo)，并调用其方法invalidate()，使其重画Path对象。

（3）可以采用setStrokeJoin和setStrokeCap令路线圆滑。

（4）可以采用BlurMaskFilter令路线圆滑。

参考资料：画布.pdf （绘制路径）

参考源码：CanvasDrawTest，其中的PathView.java（运行时下拉选path\_view看运行结果）

14、菜单“结束重放”可以中途停止重放，并清除路径的显示。

15、重放完毕后回到播放前的状态。

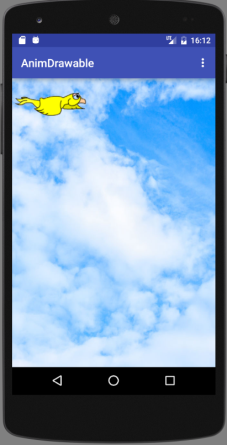
**其他控制功能**

16、播放录像时令触摸操作无效。

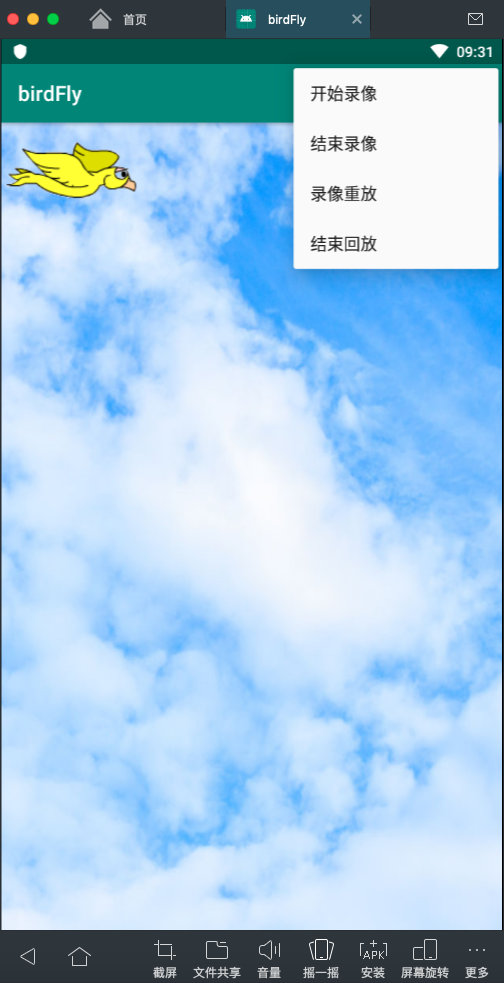
17、在飞行进行过程中令触摸操作无效。

18、录像时不能重放录像。

参考截屏：



我的截屏：



*完成后可以录制一个视频，直接用手机或者在教学网站的软件下载中下载Camtasia Studio。*

*按照完成的情况自己选择一些运行界面进行截屏：*

*完成后的主要源码(.java和.xml)：*

MainActivity.java

**package** com.tencent.birdfly;  
  
**import** android.animation.Animator;  
**import** android.animation.AnimatorSet;  
**import** android.animation.ObjectAnimator;  
**import** android.app.Activity;  
**import** android.graphics.Path;  
**import** android.graphics.drawable.AnimationDrawable;  
**import** android.graphics.drawable.Drawable;  
**import** android.os.Bundle;  
**import** android.support.design.widget.FloatingActionButton;  
**import** android.support.design.widget.Snackbar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.util.DisplayMetrics;  
**import** android.view.View;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.WindowManager;  
**import** android.view.animation.Animation;  
**import** android.view.animation.LinearInterpolator;  
**import** android.widget.ImageView;  
  
**import** java.util.ArrayList;  
**import** java.util.Collection;  
**import** java.util.Iterator;  
**import** java.util.List;  
**import** java.util.ListIterator;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** ArrayList<Line> **pos**;  
 **private boolean isRecording**;  
 **private** AnimationDrawable **recordingDrawable**;  
 ImageView **bird**, **bird2**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 Toolbar toolbar = findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
  
 **bird** = findViewById(R.id.***bird***);  
 AnimationDrawable birdDrawable = (AnimationDrawable)**bird**.getBackground();  
 birdDrawable.start();  
  
 **bird2** = findViewById(R.id.***bird2***);  
 AnimationDrawable bird2Drawable = (AnimationDrawable)**bird2**.getBackground();  
 bird2Drawable.start();  
  
 ImageView recording = findViewById(R.id.***recording***);  
 **recordingDrawable** = (AnimationDrawable)recording.getBackground();  
  
 **pos** = **new** ArrayList<>();  
 **isRecording** = **false**;  
  
 **bird**.setVisibility(View.***VISIBLE***);  
 **bird2**.setVisibility(View.***INVISIBLE***);  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***menu\_main***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.* **switch** (item.getItemId()) {  
 **case** R.id.***start\_video***:  
 **pos**.clear();  
 **isRecording** = **true**;  
 **recordingDrawable**.start();  
 **break**;  
 **case** R.id.***stop\_video***:  
 **isRecording** = **false**;  
 **recordingDrawable**.stop();  
 **break**;  
 **case** R.id.***start\_replay***:  
 **bird**.setVisibility(View.***INVISIBLE***);  
 **bird2**.setVisibility(View.***VISIBLE***);  
 **int** size = size\_pos();  
 AnimatorSet animatorSet = **new** AnimatorSet();  
 List<Animator> animatorList = **new** ArrayList<Animator>();  
 **bird2**.setX(get\_pos(0).startX());  
 **bird2**.setY(get\_pos(0).startY());  
 **for** (**int** i = 0; i < size; ++i) {  
 Path path = **new** Path();  
 path.moveTo(get\_pos(i).startX(), get\_pos(i).startY());  
 path.lineTo(get\_pos(i).endX(), get\_pos(i).endY());  
 **float** goX = get\_pos(i).endX() - get\_pos(i).startX();  
 **float** goY = get\_pos(i).endY() - get\_pos(i).startY();  
 Path path2 = **new** Path();  
 **if** (goX >= 0) {  
 path2.moveTo(1.0f, 1.0f);  
 } **else** {  
 path2.moveTo(-1.0f, 1.0f);  
 }  
 **double** dis = Math.*sqrt*(goX \* goX + goY \* goY);  
 WindowManager manager = getWindowManager();  
 DisplayMetrics outMetrics = **new** DisplayMetrics();  
 manager.getDefaultDisplay().getMetrics(outMetrics);  
 **int** width = outMetrics.**widthPixels**;  
 **int** height = outMetrics.**heightPixels**;  
 **double** disAll = Math.*sqrt*(1.0 \* width \* width + 1.0 \* height \* height);  
 **int** secs = (**int**)Math.*floor*(dis / disAll \* 5000);  
 ObjectAnimator dirAnimator = ObjectAnimator.*ofFloat*(**bird2**, **"scaleX"**, **"scaleY"**, path2);  
 ObjectAnimator objectAnimator = ObjectAnimator.*ofFloat*(**bird2**, **"translationX"**,**"translationY"**, path);  
 objectAnimator.setDuration(secs);  
 objectAnimator.setRepeatCount(0);  
 objectAnimator.setInterpolator(**new** LinearInterpolator());  
 animatorList.add(dirAnimator);  
 animatorList.add(objectAnimator);  
 }  
 animatorSet.playSequentially(animatorList);  
 animatorSet.start();  
  
 **break**;  
 **case** R.id.***stop\_replay***:  
 **bird**.setVisibility(View.***VISIBLE***);  
 **bird2**.setVisibility(View.***INVISIBLE***);  
 **break**;  
 }  
  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **public void** push\_pos(**float** startX, **float** startY, **float** endX, **float** endY) {  
 **pos**.add(**new** Line(startX, startY, endX, endY));  
 }  
  
 **public int** size\_pos() {  
 **return pos**.size();  
 }  
  
 **public** Line get\_pos(**int** index) {  
 **return pos**.get(index);  
 }  
}

ClickView.java

**package** com.tencent.birdfly;  
  
**import** android.app.Activity;  
**import** android.content.Context;  
**import** android.content.res.TypedArray;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.graphics.drawable.Drawable;  
**import** android.text.TextPaint;  
**import** android.util.AttributeSet;  
**import** android.util.DisplayMetrics;  
**import** android.util.Log;  
**import** android.view.MotionEvent;  
**import** android.view.View;  
**import** android.view.WindowManager;  
**import** android.view.animation.Animation;  
**import** android.view.animation.AnimationUtils;  
**import** android.view.animation.LinearInterpolator;  
**import** android.view.animation.TranslateAnimation;  
**import** android.widget.ImageView;  
**import** android.widget.Toast;  
  
**public class** ClickView **extends** View {  
  
 **public** ClickView(Context context) {  
 **super**(context);  
 **animationStarted** = **false**;  
 }  
  
 **public** ClickView(Context context, AttributeSet attrs) {  
 **super**(context, attrs);  
 **animationStarted** = **false**;  
 }  
  
 **public** ClickView(Context context, AttributeSet attrs, **int** defStyle) {  
 **super**(context, attrs, defStyle);  
 **animationStarted** = **false**;  
 }  
  
 @Override  
 **public boolean** onTouchEvent(MotionEvent event) {  
 **switch** (event.getAction()) {  
 **case** MotionEvent.***ACTION\_DOWN***:  
 **if** (**animationStarted**) **break**;  
 Log.*d*(**"A"**, String.*valueOf*(event.getX()) + **" "** + String.*valueOf*(event.getY()));  
 **final** ImageView bird = ((Activity)getContext()).findViewById(R.id.***bird***);  
 **final float** birdX = bird.getX() + ((**float**)bird.getWidth() / 2);  
 **final float** birdY = bird.getY() + ((**float**)bird.getHeight() / 2);  
 **final float** goX = event.getX() - birdX;  
 **final float** goY = event.getY() - birdY;  
 ((MainActivity)getContext()).push\_pos(bird.getX(), bird.getY(),  
 event.getX() - ((**float**)bird.getWidth() / 2), event.getY() - ((**float**)bird.getHeight() / 2));  
 **final double** dis = Math.*sqrt*(goX \* goX + goY \* goY);  
 WindowManager manager = ((Activity)getContext()).getWindowManager();  
 DisplayMetrics outMetrics = **new** DisplayMetrics();  
 manager.getDefaultDisplay().getMetrics(outMetrics);  
 **int** width = outMetrics.**widthPixels**;  
 **int** height = outMetrics.**heightPixels**;  
 **final double** disAll = Math.*sqrt*(1.0 \* width \* width + 1.0 \* height \* height);  
 **final int** secs = (**int**)Math.*floor*(dis / disAll \* 5000);  
 **if** (goX >= 0) {  
 bird.setScaleX(1.0f);  
 } **else** {  
 bird.setScaleX(-1.0f);  
 }  
 TranslateAnimation anim = **new** TranslateAnimation(Animation.***RELATIVE\_TO\_SELF***, 0.0f,  
 Animation.***ABSOLUTE***, goX,  
 Animation.***RELATIVE\_TO\_SELF***, 0.0f,  
 Animation.***ABSOLUTE***, goY);  
 anim.setFillBefore(**true**);  
 anim.setFillAfter(**true**);  
 anim.setStartOffset(0);  
 anim.setDuration(secs);  
 anim.setRepeatMode(Animation.***ABSOLUTE***);  
 anim.setRepeatCount(0);  
 anim.setInterpolator(**new** LinearInterpolator());  
 anim.setAnimationListener(**new** Animation.AnimationListener() {  
 @Override  
 **public void** onAnimationStart(Animation animation) {  
 **animationStarted** = **true**;  
 }  
  
 @Override  
 **public void** onAnimationEnd(Animation animation) {  
 bird.clearAnimation();  
 bird.setX(bird.getX() + goX);  
 bird.setY(bird.getY() + goY);  
 **animationStarted** = **false**;  
 }  
  
 @Override  
 **public void** onAnimationRepeat(Animation animation) {  
  
 }  
 });  
 bird.startAnimation(anim);  
 **break**;  
 }  
 **return true**;  
 }  
  
 @Override  
 **protected void** onDraw(Canvas canvas) {  
 **super**.onDraw(canvas);  
 }  
 **private boolean animationStarted**;  
}

Line.java

**package** com.tencent.birdfly;  
  
**public class** Line {  
 Line(**float** sX, **float** sY, **float** eX, **float** eY) {  
 **this**.**sX** = sX;  
 **this**.**sY** = sY;  
 **this**.**eX** = eX;  
 **this**.**eY** = eY;  
 }  
 **public float** startX() {  
 **return sX**;  
 }  
 **public float** startY() {  
 **return sY**;  
 }  
 **public float** endX() {  
 **return eX**;  
 }  
 **public float** endY() {  
 **return eY**;  
 }  
 **private float sX**, **sY**, **eX**, **eY**;  
}

【完成情况】

完成了哪些内容？(√完成 ×未做或未完成)

1 [√] 2 [√] 3[√] 4[ ] 5 [√] 6 [ ] 7[√] 8[√] 9[√] 10[√]

11 [√] 12 [√] 13[ ] 14[√] 15 [√] 16 [√] 17[ ] 18[ ]

【实验体会】

写出实验过程中遇到的问题，解决方法和自己的思考；并简述实验体会（如果有的话）。

学会了怎么做逐帧动画、补间动画，然后还会用AnimatorSet来将一段动画的集合串联起来，实现录像功能。

【交实验报告】

(a) 每位同学独立完成本实验内容并填写实验报告。

(b) 截止时间：2019年7月20日23:00

上交作业网站：<http://172.18.187.9/netdisk/default.aspx?vm=16and>

文件夹: /实验上交/03、动画实验

上传文件：学号\_姓名\_动画实验.doc (实验报告)   
学号\_姓名\_动画实验.rar (按项目压缩，注意删除build和app/build目录)

学号\_姓名\_动画实验.mp4 (可选)