院 系 数据科学与计算机学院 学 号 　16337327　 姓 名 郑映雪

【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：RelativeLayout可以加上白云背景。

参考资料：动画.pdf

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：安卓程序设计（二）p26 或p41（TouchDown可以用于Activity） p14

参考源码：TweenAnimCases

3、利用ImageView.setRotateY()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

参考源码：Menu

参考资料：对话框和菜单.pdf

**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。

参考资料：动画.pdf

参考源码：FrameAnim

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，帧布局(p43)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。  
编程说明：（1）开始重放时要清除Path对象。  
 （2）最好把前面的飞行动画用*类方法*独立出来。

（3）动画通过onAnimationEnd()事件接续。

参考资料：安卓程序设计（二） p24

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：安卓程序设计（二） p56 p29 p62 p104

参考源码：CanvasDrawTest PathView.java

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。

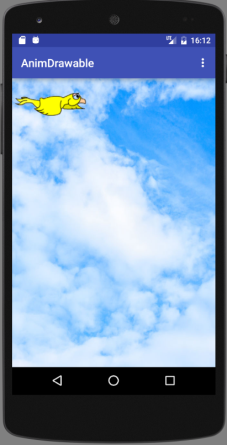
**其他控制功能**

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：



*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:

1. **package** com.example.a64635.birdfly;
3. **import** android.animation.Animator;
4. **import** android.animation.ObjectAnimator;
5. **import** android.graphics.drawable.AnimationDrawable;
6. **import** android.os.Bundle;
7. **import** android.support.design.widget.FloatingActionButton;
8. **import** android.support.design.widget.Snackbar;
9. **import** android.support.v7.app.AppCompatActivity;
10. **import** android.support.v7.widget.Toolbar;
11. **import** android.util.Log;
12. **import** android.view.MotionEvent;
13. **import** android.view.View;
14. **import** android.view.Menu;
15. **import** android.view.MenuItem;
16. **import** android.view.animation.Animation;
17. **import** android.view.animation.TranslateAnimation;
18. **import** android.widget.ImageView;
19. **import** android.widget.LinearLayout;
20. **import** android.widget.RelativeLayout;
22. **import** java.util.ArrayList;
23. **import** java.util.List;
24. **import** java.util.Timer;
25. **import** java.util.TimerTask;
27. **import** **static** android.R.attr.delay;
28. **import** **static** android.R.attr.fingerprintAuthDrawable;
29. **import** **static** android.R.attr.revisionCode;
30. **import** **static** java.lang.StrictMath.abs;
32. **public** **class** MainActivity **extends** AppCompatActivity {
33. **private** **boolean** isright = **true**, record = **false**, play = **false**, respond = **true**;
34. **private** **int** imagex,imagey;
35. **private** **float** nowx,nowy;
36. **private** ImageView imageView,imageView2;
37. **private** AnimationDrawable recording, birdfly;
38. **private** PathView pathView;
39. **private** ArrayList<Pos> list = **new** ArrayList<>();
40. **private** Pos last\_position;
41. RelativeLayout relativeLayout;
43. @Override
44. **protected** **void** onCreate(Bundle savedInstanceState) {
45. **super**.onCreate(savedInstanceState);
46. setContentView(R.layout.activity\_main);
47. Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
48. setSupportActionBar(toolbar);
49. imageView = (ImageView) findViewById(R.id.birdimg);
50. imageView2=(ImageView) findViewById(R.id.recordimg);
51. pathView = (PathView) findViewById(R.id.path\_view);
52. relativeLayout=(RelativeLayout) findViewById(R.id.content\_main);
53. birdfly = (AnimationDrawable) imageView.getBackground();
54. recording = (AnimationDrawable) imageView2.getBackground();
55. birdfly.start();
56. recording.start();
57. }
58. **public** **boolean** onTouchEvent(MotionEvent event) {
59. **final** **int** maxheight=relativeLayout.getHeight(),maxwidth=relativeLayout.getWidth();
60. imagex=imageView.getWidth();
61. imagey=imageView.getHeight();
63. // 在这里判断一下如果是按下操作就获取坐标然后执行方法
64. **if** (event.getAction() == MotionEvent.ACTION\_DOWN && !play && respond) {
65. nowx=imageView.getX();
66. nowy=imageView.getY();
67. respond = **false**;
68. **float** oldx,oldy;
69. **final** **float** X,Y,newx,newy;
70. **if** (event.getX() + imagex/2 > maxwidth) oldx=maxwidth - imagex/2;**else** oldx=event.getX();
71. **if** (event.getY() - 100 + imagey/2 > maxheight) oldy=maxheight - imagey/2;**else** oldy=event.getY() - 100;
72. **if** (oldx < imagex/2) X=imagex; **else** X=oldx;
73. **if** (oldy < imagex/2) Y=imagey; **else** Y=oldy;
74. newx = X - nowx - imagex/2;
75. newy = Y - nowy - imagey/2;
76. **if** (newx < 0 && isright) {
77. isright = **false**;
78. ObjectAnimator rotateY = ObjectAnimator.ofFloat(imageView, "rotationY", 0, 180);
79. rotateY.setDuration(500);
80. rotateY.start();
81. }
82. **else** **if** (newx > 0 && !isright) {
83. isright = **true**;
84. ObjectAnimator rotateY = ObjectAnimator.ofFloat(imageView, "rotationY", 180, 0);
85. rotateY.setDuration(500);
86. rotateY.start();
87. }
88. TranslateAnimation animation = **new** TranslateAnimation(0, newx, 0, newy);
89. animation.setStartOffset(500);
90. **double** maxLength = Math.sqrt(Math.pow(maxheight - imagey, 2) + Math.pow(maxwidth - imagex, 2));
91. animation.setDuration(Double.valueOf(5000 \* (Math.sqrt(Math.pow(newx, 2) + Math.pow(newy, 2)) / maxLength)).longValue());
92. imageView.startAnimation(animation);
93. animation.setAnimationListener(**new** Animation.AnimationListener() {
94. @Override
95. **public** **void** onAnimationStart(Animation animation) {
96. }
98. @Override
99. **public** **void** onAnimationEnd(Animation animation) {
100. imageView.clearAnimation();
101. imageView.setX(X - imagex/2);
102. imageView.setY(Y - imagey/2);
103. **if** (record) {
104. Pos pos = **new** Pos(nowx, nowy, isright);
105. list.add(pos);
106. }
107. respond = **true**;
108. }
110. @Override
111. **public** **void** onAnimationRepeat(Animation animation) {
112. }
113. });
114. }
115. **return** **super**.onTouchEvent(event);
116. }
118. **private** **void** play(Pos firstPos, **final** **int** inoldx) {
119. **final** **int** maxheight=relativeLayout.getHeight(),maxwidth=relativeLayout.getWidth();
120. **if** (inoldx >= list.size()) {
121. **return**;
122. }
123. **final** Pos nextPos = list.get(inoldx);
124. pathView.set(firstPos, nextPos, imagex, imagey);
125. **float** newx = nextPos.getx() - firstPos.getx();
126. **float** newy = nextPos.gety() - firstPos.gety();
127. **if** (newx < 0 && isright) {
128. isright = **false**;
129. ObjectAnimator rotateY = ObjectAnimator.ofFloat(imageView, "rotationY", 0, 180);
130. rotateY.setDuration(500);
131. rotateY.start();
132. }
133. **else** **if** (newx > 0 && !isright) {
134. isright = **true**;
135. ObjectAnimator rotateY = ObjectAnimator.ofFloat(imageView, "rotationY", 180, 0);
136. rotateY.setDuration(500);
137. rotateY.start();
138. }
139. TranslateAnimation animation = **new** TranslateAnimation(0, newx, 0, newy);
140. animation.setStartOffset(500);
141. **double** maxLength = Math.sqrt(Math.pow(maxheight - imagey, 2) + Math.pow(maxwidth - imagex, 2));
142. **double** speed = Math.sqrt(Math.pow(newx, 2) + Math.pow(newy, 2)) / maxLength;
143. animation.setDuration(Double.valueOf(5000 \* speed).longValue());
144. imageView.startAnimation(animation);
145. animation.setAnimationListener(**new** Animation.AnimationListener() {
146. @Override
147. **public** **void** onAnimationStart(Animation animation) {
148. }
150. @Override
151. **public** **void** onAnimationEnd(Animation animation) {
152. imageView.clearAnimation();
153. imageView.setX(nextPos.getx());
154. imageView.setY(nextPos.gety());
155. play(nextPos, inoldx + 1);
156. }
158. @Override
159. **public** **void** onAnimationRepeat(Animation animation) {
160. }
161. });
162. }
163. @Override
164. **public** **boolean** onCreateOptionsMenu(Menu menu) {
165. // Inflate the menu; this adds items to the action bar if it is present.
166. getMenuInflater().inflate(R.menu.menu\_main, menu);
167. **return** **true**;
168. }
170. @Override
171. **public** **boolean** onOptionsItemSelected(MenuItem item) {
172. // Handle action bar item clicks here. The action bar will
173. // automatically handle clicks on the Home/Up button, so long
174. // as you specify a parent activity in AndroidManifest.xml.
175. **int** id = item.getItemId();
177. //noinspection SimplifiableIfStatement
178. **if** (id == R.id.start) {
179. imageView2.setVisibility(View.VISIBLE);
180. record = **true**;
181. list.clear();
182. list.add(**new** Pos(nowx, nowy, isright));
183. }
184. **if** (id == R.id.end) {
185. imageView2.setVisibility(View.GONE);
186. record = **false**;
187. }
188. **if** (id == R.id.replay && !record) {
189. pathView.clear();
190. play = **true**;
191. last\_position = **new** Pos(nowx, nowy, isright);
192. **if** (list.size() != 0) {
193. Pos firstPos = list.get(0);
194. imageView.setX(firstPos.getx());
195. imageView.setY(firstPos.gety());
196. **if** (isright != firstPos.isisRight())
197. **if** (firstPos.isisRight()) imageView.setRotationY(0);**else** imageView.setRotationY(180);
198. isright = firstPos.isisRight();
199. play(firstPos, 1);
200. }
201. }
202. **if** (id == R.id.end\_replay) {
203. pathView.clear();
204. imageView.clearAnimation();
205. imageView.setX(last\_position.getx());
206. imageView.setY(last\_position.gety());
207. **if** (isright != last\_position.isisRight())
208. **if** (last\_position.isisRight())
209. imageView.setRotationY(0);
210. **else**
211. imageView.setRotationY(180);
212. isright = last\_position.isisRight();
213. play = **false**;
214. }
215. **return** **super**.onOptionsItemSelected(item);
216. }

219. }

Pos.java:

1. **package** com.example.a64635.birdfly;
3. /\*\*
4. \* Created by 64635 on 2019/7/13.
5. \*/
7. **public** **class** Pos {
8. **private** **float** x, y;
9. **private** **boolean** isright;
10. **public** **boolean** isisRight() {
11. **return** isright;
12. }
13. **public** **float** getx() {
14. **return** x;
15. }
16. **public** **float** gety() {
17. **return** y;
18. }
19. **public** Pos(**float** x, **float** y, **boolean** isright) {
20. **this**.x = x;
21. **this**.y = y;
22. **this**.isright = isright;
23. }
25. }

Activity\_main.xml:

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<android.support.design.widget.CoordinatorLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:app="http://schemas.android.com/apk/res-auto"
4. xmlns:tools="http://schemas.android.com/tools"
5. android:layout\_width="match\_parent"
6. android:layout\_height="match\_parent"
7. android:fitsSystemWindows="true"
8. tools:context="com.example.a64635.birdfly.MainActivity"**>**
10. **<android.support.design.widget.AppBarLayout**
11. android:layout\_width="match\_parent"
12. android:layout\_height="wrap\_content"
13. android:theme="@style/AppTheme.AppBarOverlay"**>**
15. **<android.support.v7.widget.Toolbar**
16. android:id="@+id/toolbar"
17. android:layout\_width="match\_parent"
18. android:layout\_height="?attr/actionBarSize"
19. android:background="?attr/colorPrimary"
20. app:popupTheme="@style/AppTheme.PopupOverlay" **/>**
22. **</android.support.design.widget.AppBarLayout>**
24. **<include** layout="@layout/content\_main" **/>**
26. **</android.support.design.widget.CoordinatorLayout>**

Content\_main.xml:

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:app="http://schemas.android.com/apk/res-auto"
4. xmlns:tools="http://schemas.android.com/tools"
5. android:id="@+id/content\_main"
6. android:layout\_width="match\_parent"
7. android:layout\_height="match\_parent"
8. android:background="@drawable/cloud"
9. app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"
10. tools:context="com.example.a64635.birdfly.MainActivity"
11. tools:showIn="@layout/activity\_main"**>**
13. **<ImageView**
14. android:id="@+id/birdimg"
15. android:layout\_width="wrap\_content"
16. android:layout\_height="wrap\_content"
17. android:background="@drawable/bird"
18. **/>**
20. **<ImageView**
21. android:visibility="gone"
22. android:id="@+id/recordimg"
23. android:layout\_width="wrap\_content"
24. android:layout\_height="wrap\_content"
25. android:background="@drawable/record"
26. android:layout\_alignParentTop="true"
27. android:layout\_alignParentEnd="true" **/>**

30. **<com.example.a64635.birdfly.PathView**
31. android:layout\_width="match\_parent"
32. android:layout\_height="match\_parent"
33. android:id="@+id/path\_view"**/>**
34. **</RelativeLayout>**

Menu\_main.xml:

1. **<menu** xmlns:android="http://schemas.android.com/apk/res/android"
2. xmlns:app="http://schemas.android.com/apk/res-auto"
3. xmlns:tools="http://schemas.android.com/tools"
4. tools:context="com.example.a64635.birdfly.MainActivity"**>**
5. **<item**
6. android:id="@+id/start"
7. android:orderInCategory="100"
8. android:title="开始录像"
9. app:showAsAction="never" **/>**
10. **<item**
11. android:id="@+id/end"
12. android:orderInCategory="100"
13. android:title="结束录像"
14. app:showAsAction="never" **/>**
15. **<item**
16. android:id="@+id/replay"
17. android:orderInCategory="100"
18. android:title="录像回放"
19. app:showAsAction="never" **/>**
20. **<item**
21. android:id="@+id/end\_replay"
22. android:orderInCategory="100"
23. android:title="结束回放"
24. app:showAsAction="never" **/>**
25. **</menu>**

【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】

1、动画中在移动的时候有闪动的情况，这时候需要添加imageView.clearAnimation();语句。

2、在动画结束之后保存新的起点时有错乱的现象。是因为onTouchEvent的响应是有按下和抬起两步的，单独判断是否为down事件再进行动画即可。另外，animation的四个参数默认是相对于自身而说的。在调节模式时吃过坐标类型的亏（差不多是调节了一整天后来发现直接在默认值赋值即可）。同时，imageview进行动画与本身的坐标是没有关系的，它只是进行了一个动画，但是元件本身还是在原坐标，所以需要使用setX和setY修改一下元件的位置，不然下一次还需要重新调节。

3、本项目中红点部分有bug，动画不能闪动，至今没有解决，深表遗憾。后来发现在别人的电脑上可以显示（难道是版本问题？）

4、本次实验里的“录制视频”和“视频回放”其实是一种“虚假”的录制，关键在于使用一个数组存储每次点击的坐标，然后在点击回放的时候，依次从数组中读出数据，然后“视频回放”功能则根据读出的数据一步一步进行动画即可。