实 验 报 告

课程名称 移动应用软件综合开发

开课学期 2020 春季

指导教师 林奕森

实 验 室 巡天楼 308

班　　级 17 软件工程 4 班

学　　号 2017070030429

姓　　名 钟祯

成绩：_____（五级）

实验课程 评分表标准

	全勤、学习态度端正、实验认真、积极回答问题、操作过程正确，结果准确，实验报告内容规范	偶有缺勤、实验认真、回答问题较积极、操作过程正确，结果准确，实验报告内容规范	旷课2次以内、偶有迟到、实验认真、回答问题较好、操作过程基本正确，结果基本准确，实验报告内容较规范	旷课2次以上、学习态度一般、基本能回答出问题、操作过程较正确，结果基本准确，实验报告内容基本规范	经常旷课，实验过程不认真、问题回答不积极、实验报告不符合要求或未交
	优秀（90-100）	良好（80-89）	中（70-79）	及格（60-69）	不及格(<59)
实验一					
实验二					
实验三					
实验四					
实验五					
实验成绩总评（五级制）					

说明：1．每次实验结束，学生完成一份实验报告，课程结束后汇总,加封面装订成册存档；2．各任课教师可在以上五项栏目的基础上，可根据实验课程和实验项目的具体需要，调整项目内容，但封面格式须统一；3．可根据实验数量自行添加行数。打印到封面背面

桂林航天工业学院学生实验报告

课程名称	移动应用软件综合开发	实验项目名称	实验一 菜单与对话框实例实验
开课教学单位及实验室	计算机科学与工程学院 巡天楼 308	实验日期	2020 年 03 月 06 日
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	林奕森	实验成绩	

一、实验目的

1、实现手势交互功能案例实验使用功能。

二、实验原理

1、使用 Android Studio 创建工程及类，编写程序。

2、调试程序，并在 Genymotion 中运行程序。

三、实验操作方法和步骤

1、完成实例 006 明日学院消息通知

2、完成实例 009 彩虹式菜单

四、实验结果与分析

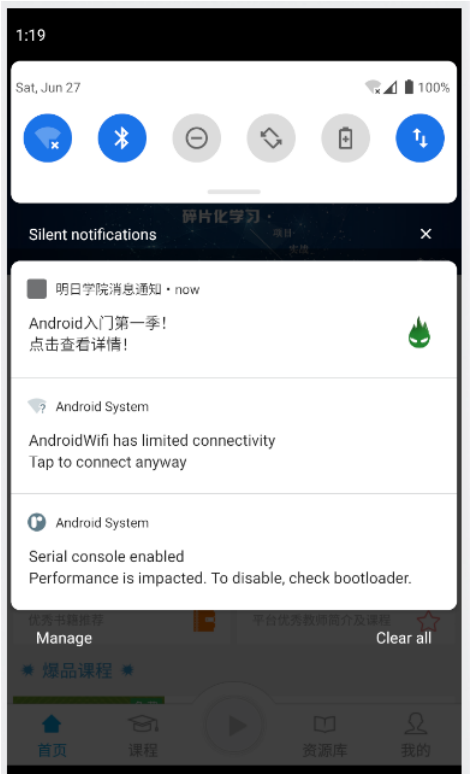
实例 006 部分关键源码：

```
public class MainActivity extends AppCompatActivity {

    final int NOTIFYID = 0x123;



    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        NotificationManager notificationManager = (NotificationManager)

getSystemService(NOTIFICATION_SERVICE);
```

```
        NotificationCompat.Builder notification = new
NotificationCompat.Builder(MainActivity.this);
        notification.setAutoCancel(true);
        notification.setTicker("安卓课程第一季上线啦！");
        notification.setSmallIcon(R.mipmap.ic_launcher);
        notification.setLargeIcon(BitmapFactory.decodeResource(getResources(),
R.mipmap.ic_launcher));
        notification.setContentTitle("Android 入门第一季！");
        notification.setContentText("点击查看详情！");
        notification.setWhen(System.currentTimeMillis());
        Intent intent = new Intent(MainActivity.this, MessageActivity.class);
        PendingIntent pi = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
        notification.setContentIntent(pi);
        assert notificationManager != null;
        notificationManager.notify(NOTIFYID, notification.build());
    }
}
```

实例 006 实验结果：


1:18
明日学院
体系课程
实战课程
伴你成长
碎片化学习
今日限免
每日一款趣味的练习先到先得！
成就排行
看前辈学分排行，继续努力学习吧！
工程宝典
优秀书籍推荐
明星讲师
平台优秀教师简介及课程
爆品课程
首页
课程
资源库
我的


1:19
Sat, Jun 27
100%
Silent notifications
明日学院消息通知 · now
Android入门第一季！
点击查看详情！
Android System
AndroidWifi has limited connectivity
Tap to connect anyway
Android System
Serial console enabled
Performance is impacted. To disable, check bootloader.
Manage
Clear all
爆品课程
首页
课程
资源库
我的


体系课程
明日学院
Android入门第一季
浏览完整体系
课程时长：28小时35分17秒
学习人数 252 人
收藏课程
课程简介
课程目录
第一章 搭建Android开发环境
1.1Android开发环境概述
免费
1.2JDK下载与安装
免费
1.3获取Android SDK
开始学习
1.4 下载并安装Eclipse
免费
1.5安装ADT插件
免费
1.6常见问题
免费
第二章 第一个Android程序
2.1一分钟学一章
免费
2.2创建Android应用程序
免费
2.3Android项目结构
免费

实例 009 部分关键源码：

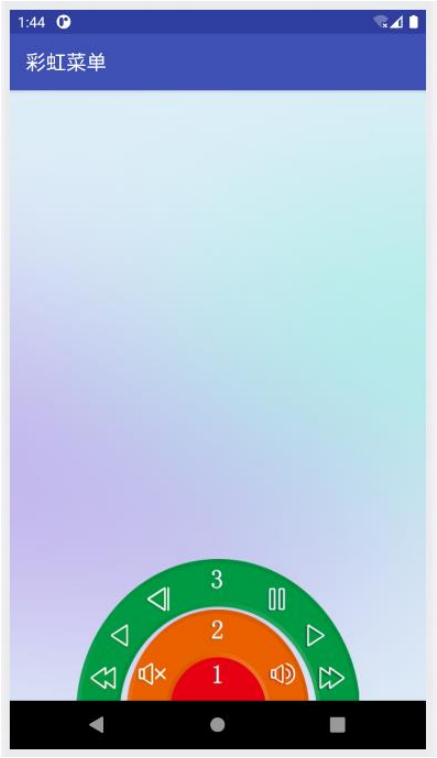
```
public class MainActivity extends AppCompatActivity {

    private RelativeLayout l2, l3;



    private boolean isl2Show = true;

    private boolean isl3Show = true;



    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);



        ImageButton imageButton_a1 = (ImageButton) findViewById(R.id.a_1);

        ImageButton imageButton_b2 = (ImageButton) findViewById(R.id.b_2);



        l2 = (RelativeLayout) findViewById(R.id.level_2);

        l3 = (RelativeLayout) findViewById(R.id.level_3);



        imageButton_b2.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                if (isl3Show) {
```

```
                MyAnimation.animationOUT(l3, 500, 0);
            } else {
                MyAnimation.animationIN(l3, 500);
            }

            isl3Show = !isl3Show;
        }
    });

    imageButton_a1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (!isl2Show) {
                MyAnimation.animationIN(l2, 500);
            } else {
                if (isl3Show) {
                    MyAnimation.animationOUT(l3, 500, 0);
                    MyAnimation.animationOUT(l2, 500, 500);
                    isl3Show = !isl3Show;
                } else {
                    MyAnimation.animationOUT(l2, 500, 0);
```

```
                    }

                }

                isl2Show = !isl2Show;

            }

        });

    }

}


public class MyAnimation {
    public static void animationIN(ViewGroup viewGroup, int duration) {
        for (int i = 0; i < viewGroup.getChildCount(); i++) {
            viewGroup.getChildAt(i).setVisibility(View.VISIBLE);
            viewGroup.getChildAt(i).setFocusable(true);
            viewGroup.getChildAt(i).setClickable(true);
        }

        Animation animation;
        animation = new RotateAnimation(
                -180, 0,
                Animation.RELATIVE_TO_SELF, 0.5f,
                Animation.RELATIVE_TO_SELF, 1.0f);
        animation.setFillAfter(true);
        animation.setDuration(duration);
        viewGroup.startAnimation(animation);
    }

    public static void animationOUT(final ViewGroup viewGroup, int duration, int
startOffSet) {
```

```
        Animation animation;
        animation = new RotateAnimation(
                0, -180,
                Animation.RELATIVE_TO_SELF, 0.5f,
                Animation.RELATIVE_TO_SELF, 1.0f);
        animation.setFillAfter(true);
        animation.setDuration(duration);
        animation.setStartOffset(startOffSet);
        animation.setAnimationListener(new Animation.AnimationListener() {
            @Override
            public void onAnimationStart(Animation animation) {
                // TODO Auto-generated method stub
            }

            @Override
            public void onAnimationRepeat(Animation animation) {
                // TODO Auto-generated method stub
            }

            @Override
            public void onAnimationEnd(Animation animation) {
                for (int i = 0; i < viewGroup.getChildCount(); i++) {
                    viewGroup.getChildAt(i).setVisibility(View.GONE);
                    viewGroup.getChildAt(i).setFocusable(false);
                    viewGroup.getChildAt(i).setClickable(false);
                }
            }
        });
        viewGroup.startAnimation(animation);
    }
}
```

实例 009 实验结果：



结果分析：

实例 006 中，使用 NotificationManager 来管理 Notification.Builder 发送通知。

实例 009 中，使用 onClick 事件处理，还使用了动画方法。

桂林航天工业学院学生实验报告

课程名称	移动应用软件综合开发	实验项目名称	实验二 Android 常用控件应用
开课教学单位及实验室	计算机科学与工程学院 巡天楼 308	实验日期	2020 年 03 月 27 日
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	林奕森	实验成绩	

一、实验目的

1、实现手势交互功能案例实验使用功能。

二、实验原理

1、使用 Android Studio 创建工程及类，编写程序。

2、调试程序，并在 Genymotion 中运行程序。

三、实验操作方法和步骤

1、完成实例 019 弹出智能提示信息的搜索框

2、完成实例 020 页面垂直滚动公告条

四、实验结果与分析

实例 019 部分关键源码：

```
public class Book {

    private String bookname;



    public String getBookname() {

        return bookname;

    }



    public void setBookname(String bookname) {

        this.bookname = bookname;
```

```
    }

    @Override
    public String toString() {
        return bookname;
    }
}

public class MainActivity extends AppCompatActivity {
    private List<Book> booknames;

    private String[] bookList = {"书籍：Android 移动开发", "书籍：Java 从入门到精通", "书籍：Java Web 程序设计",
            "书籍:JSP 程序设计","书籍:PHP 程序设计","书籍:C# 程序设计","书籍:SQL Server 数据库管理与开发",
            "书籍：ASP.NET 开发实例大全","书籍：Visual Basic 开发实例大全","书籍：Visual C++ 开发实例大全",
            "书籍：Oracle 数据库管理与开发","书籍：Java 开发实例大全","书籍：Java Web 开发实例大全",
            "书籍：C# 开发实例大全", "书籍：Android 程序开发范例宝典", "书籍：HTML5 程序开发范例宝典"
```

```
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initData();
        initView();
    }

    private void initView() {
        AutoCompleteTextView autoCompleteTextView = (AutoCompleteTextView)
findViewById(R.id.autocomplete_country);
        MyAdapter bookAdapter = new MyAdapter(booknames, this);
        autoCompleteTextView.setAdapter(bookAdapter);
    }

    private void initData() {
        booknames = new ArrayList<>();
        for (String s : bookList) {
            Book book = new Book();
```

```
            book.setBookname(s);
            booknames.add(book);
        }
    }
}


public class MyAdapter extends BaseAdapter implements Filterable {
    private Context context;
    private List<Book> books;
    private ArrayFilter mArrayFilter;
    private ArrayList<Book> mFilterBooks;

    public MyAdapter(List<Book> mList, Context context) {
        this.books = mList;
        this.context = context;
    }

    @Override
    public int getCount() {
        return books == null ? 0 : books.size();
    }
```

```
@Override
public Object getItem(int position) {
    return books.get(position);
}

@Override
public long getItemId(int position) {
    return position;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View view;
    ViewHolder holder;
    if (convertView == null) {
        view = View.inflate(context, R.layout.list_item, null);
        holder = new ViewHolder();
        holder.tv_name = (TextView) view.findViewById(R.id.bookname);
        view.setTag(holder);
    } else {
```

```
            view = convertView;
            holder = (ViewHolder) view.getTag();
        }
        Book bookname = books.get(position);
        holder.tv_name.setText(bookname.getBookname());
        return view;
    }

    static class ViewHolder {
        public TextView tv_name;
    }

    @Override
    public Filter getFilter() {
        if (mArrayFilter == null) mArrayFilter = new ArrayFilter();
        return mArrayFilter;
    }

    private class ArrayFilter extends Filter {
        @Override
        protected FilterResults performFiltering(CharSequence constraint) {
```

```
        FilterResults results = new FilterResults();
        if (mFilterBooks == null) mFilterBooks = new ArrayList<>(books);
        if (constraint == null || constraint.length() == 0) {
            results.values = mFilterBooks;
            results.count = mFilterBooks.size();
        } else {
            List<Book> retList = new ArrayList<>();
            String str = constraint.toString().toLowerCase();
            for (Book book : mFilterBooks) {
                if (book.getBookname().contains(str)) retList.add(book);
            }
            results.values = retList;
            results.count = retList.size();
        }
        return results;
    }

    @Override
    @SuppressWarnings("unchecked")
    protected void publishResults(CharSequence constraint, FilterResults results) {
        books = (List<Book>) results.values;
```

```
            if (results.count > 0) {
                notifyDataSetChanged();
            } else {
                notifyDataSetInvalidated();
            }
        }
    }
}
```

实例 019 实验结果：

实例 020 部分关键源码：

```
public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        VerTextView tv_ver = (VerTextView) findViewById(R.id.tv_ver);

        tv_ver.setListener(new VerTextView.onTouchListener() {

            @Override

            public void touchListener(String s) {

                Toast.makeText(MainActivity.this, s, Toast.LENGTH_LONG).show();

            }

        });

    }

}


public class VerTextView extends View {



    private Paint mPaint;

    private float x, startY, endY, firstY, nextStartY, secondY;

    private String[] text = {"Android 移动开发（慕课版）", "明日科技出新书了"};

    private float textWidth, textHeight;
```

```
private float speech = 0;

private static final int CHANGE_SPEECH = 0x01;

private boolean isScroll = false;


@SuppressLint("HandlerLeak")

private Handler mHandler = new Handler() {

    @Override

    public void handleMessage(Message msg) {

        super.handleMessage(msg);

        if (msg.what == CHANGE_SPEECH) speech = 1f;

    }

};


public VerTextView(Context context, AttributeSet attrs) {

    super(context, attrs);

    mPaint = new Paint();

    mPaint.setColor(Color.RED);

    mPaint.setTextSize(30f);

    Rect rect = new Rect();

    mPaint.getTextBounds(text[0], 0, text[0].length(), rect);

    textWidth = rect.width();
```

```
        textHeight = rect.height();
        x = getX() + getPaddingLeft();
        startY = getTop() + textHeight + getPaddingTop() - 5;
        endY = startY + textHeight + getPaddingBottom();
        nextStartY = getTop() - 5;
        firstY = startY;
        secondY = nextStartY;
    }

    @Override
    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
        super.onMeasure(widthMeasureSpec, heightMeasureSpec);
        int width = measureWidth(widthMeasureSpec);
        int height = measureHeight(heightMeasureSpec);
        setMeasuredDimension(width, height);
    }

    private int measureHeight(int heightMeasureSpec) {
        int result = 0;
        int size = MeasureSpec.getSize(heightMeasureSpec);
        int mode = MeasureSpec.getMode(heightMeasureSpec);
```

```
        if (mode == MeasureSpec.EXACTLY) {
            result = size;
        } else {
            result = (int) (getPaddingTop() + getPaddingBottom() + textHeight);
            if (mode == MeasureSpec.AT_MOST) {
                result = Math.min(result, size);
            }
        }
        return result;
    }

    private int measureWidth(int widthMeasureSpec) {
        int result = 0;
        int size = MeasureSpec.getSize(widthMeasureSpec);
        int mode = MeasureSpec.getMode(widthMeasureSpec);
        if (mode == MeasureSpec.EXACTLY) {
            result = size;
        } else {
            result = (int) (getPaddingLeft() + getPaddingRight() + textWidth);
            if (mode == MeasureSpec.AT_MOST) {
                result = Math.min(result, size);
```

```
        }
    }
    return result;
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    if (!isScroll) {
        mHandler.sendEmptyMessageDelayed(CHANGE_SPEECH, 2000);
        isScroll = true;
    }
    if (startY > endY || nextStartY > endY) {
        if (startY > endY) {
            startY = secondY;
            nextStartY = firstY;
        } else if (nextStartY > endY) {
            startY = firstY;
            nextStartY = secondY;
        }
        speech = 0;
```

```
            isScroll = false;
        } else {
            canvas.drawText(text[0], x, startY, mPaint);
            canvas.drawText(text[1], x, nextStartY, mPaint);
            startY += speech;
            nextStartY += speech;
        }
        invalidate();
    }

    public onTouchListener listener;

    public interface onTouchListener {
        void touchListener(String s);
    }

    public void setListener(onTouchListener listener) {
        this.listener = listener;
    }

    @Override
```

```
    public boolean dispatchTouchEvent(MotionEvent event) {

        if (event.getAction() == MotionEvent.ACTION_DOWN) {

            if (listener != null) {

                if (startY == firstY) {

                    listener.touchListener(text[0]);

                } else {

                    listener.touchListener(text[1]);

                }

            }

        }

        return true;

    }

}
```

实例 020 实验结果：

结果分析：

实例 019 中，我使用了迭代的方式（在 initData 方法中）替代了传统的逐行添加，增加了可读性，可扩展性。

实例 020 中，使用了 VerTextView，主要使用 Paint 画笔绘制页面。

桂林航天工业学院学生实验报告

课程名称	移动应用软件综合开发	实验项目名称	实验三 手势交互功能案例实验
开课教学单位及实验室	计算机科学与工程学院 巡天楼 308	实验日期	2020 年 04 月 24 日
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	林奕森	实验成绩	

一、实验目的

1、实现手势交互功能案例实验使用功能。

二、实验原理

1、使用 Android Studio 创建工程及类，编写程序。

2、调试程序，并在 Genymotion 中运行程序。

三、实验操作方法和步骤

1、完成 实例 044 手势打电话

2、完成 实例 046 图片的放大

3、完成 实例 047 长按碎屏效果

四、实验结果与分析

实例 044 部分关键源码：

```
public class MPermissionsActivity extends AppCompatActivity {
    private int REQUEST_CODE_PERMISSION = 0x00099;

    public void requestPermission(String[] permissions, int requestCode) {
        this.REQUEST_CODE_PERMISSION = requestCode;
        if (checkPermissions(permissions)) {
            permissionSuccess(REQUEST_CODE_PERMISSION);
        } else {
            List<String> needPermissions = getDeniedPermissions(permissions);
            ActivityCompat.requestPermissions(this,          needPermissions.toArray(new
String[0]), REQUEST_CODE_PERMISSION);
        }
```

```
    }

    private boolean checkPermissions(String[] permissions) {
        if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) return true;
        for (String permission : permissions) {
            if (ContextCompat.checkSelfPermission(this, permission) !=
PackageManager.PERMISSION_GRANTED) {
                return false;
            }
        }
        return true;
    }

    private List<String> getDeniedPermissions(String[] permissions) {
        List<String> needRequestPermissionList = new ArrayList<>();
        for (String permission : permissions) {
            if (ContextCompat.checkSelfPermission(this, permission) !=
PackageManager.PERMISSION_GRANTED ||
ActivityCompat.shouldShowRequestPermissionRationale(this, permission)) {
                needRequestPermissionList.add(permission);
            }
        }
        return needRequestPermissionList;
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == REQUEST_CODE_PERMISSION) {
            if (verifyPermissions(grantResults)) {
```

```
                permissionSuccess(REQUEST_CODE_PERMISSION);
            } else {
                permissionFail(REQUEST_CODE_PERMISSION);
                showTipsDialog();
            }
        }
    }

    private boolean verifyPermissions(int[] grantResults) {
        for (int grantResult : grantResults) {
            if (grantResult != PackageManager.PERMISSION_GRANTED) return false;
        }
        return true;
    }

    private void showTipsDialog() {
        new AlertDialog.Builder(this)
                .setTitle("提示信息")
                .setMessage("当前应用缺少必要权限，该功能暂时无法使用。如若需要，请单击【确定】按钮前往设置中心进行权限授权。")
                .setNegativeButton("取消", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                    }
                })
                .setPositiveButton("确定", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        startAppSettings();
                    }
                }).show();
```

```
    }

    private void startAppSettings() {
        Intent intent = new Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
        intent.setData(Uri.parse("package:" + getPackageName()));
        startActivity(intent);
    }

    public void permissionSuccess(int requestCode) {
        Log.e("------", "获取权限成功=" + requestCode);
    }

    public void permissionFail(int requestCode) {
        String TAG = "MPermissions";
        Log.d(TAG, "获取权限失败=" + requestCode);
    }
}

public class CreateGestureActivity extends Activity {
    private static final float LENGTH_THRESHOLD = 120.0f;

    private Gesture mGesture;
    private View mDoneButton;

    @SuppressLint("ResourceAsColor")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.create_gesture);
        mDoneButton = findViewById(R.id.done);
        GestureOverlayView          overlay          =          (GestureOverlayView)
```

```
findViewById(R.id.gestures_overlay);
        overlay.setGestureStrokeType(GestureOverlayView.GESTURE_STROKE_TYPE_MULTIPLE);
        overlay.setFadeOffset(2000);
        overlay.setGestureColor(R.color.black);
        overlay.setGestureStrokeWidth(8);
        overlay.addOnGestureListener(new GesturesProcessor());
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        if (mGesture != null) outState.putParcelable("gesture", mGesture);
    }

    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
        super.onRestoreInstanceState(savedInstanceState);
        mGesture = savedInstanceState.getParcelable("gesture");
        if (mGesture != null) {
            final      GestureOverlayView      overlay      =      (GestureOverlayView)
findViewById(R.id.gestures_overlay);
            overlay.post(new Runnable() {
                public void run() {
                    overlay.setGesture(mGesture);
                }
            });
            mDoneButton.setEnabled(true);
        }
    }

    public void addGesture(View v) {
```

```
        if (mGesture != null) {
            final TextView input = (TextView) findViewById(R.id.gesture_name);
            final CharSequence name = input.getText();
            if (name.length() == 0) {
                input.setError(getString(R.string.error_missing_name));
                return;
            }
            final GestureLibrary store = GestureBuilderActivity.getStore();
            store.addGesture(name.toString(), mGesture);
            store.save();
            setResult(RESULT_OK);
            final String path = new File(Environment.getExternalStorageDirectory(),
"gestures").getAbsolutePath();
            Toast.makeText(this, getString(R.string.save_success, path),
Toast.LENGTH_LONG).show();
        } else {
            setResult(RESULT_CANCELED);
        }
        finish();
    }

    public void cancelGesture(View v) {
        setResult(RESULT_CANCELED);
        finish();
    }

    private class GesturesProcessor implements GestureOverlayView.OnGestureListener {
        public void onGestureStarted(GestureOverlayView overlay, MotionEvent event) {
            mDoneButton.setEnabled(false);
            mGesture = null;
        }
```

```
        public void onGesture(GestureOverlayView overlay, MotionEvent event) {
        }

        public void onGestureEnded(GestureOverlayView overlay, MotionEvent event) {
            mGesture = overlay.getGesture();
            if (mGesture.getLength() < LENGTH_THRESHOLD) overlay.clear(false);
            mDoneButton.setEnabled(true);
        }

        public void onGestureCancelled(GestureOverlayView overlay, MotionEvent event) {
        }
    }
}
```

实例 044 实验结果：

实例 046 部分关键源码：

```
public class ZoomImageView extends View implements Observer {
    private final Paint mPaint = new Paint(Paint.FILTER_BITMAP_FLAG);
    private final Rect mRectSrc = new Rect();
    private final Rect mRectDst = new Rect();
    private final AspectQuotient mAspectQuotient = new AspectQuotient();
    private Bitmap mBitmap;
    private ZoomState mState;

    public ZoomImageView(Context context, AttributeSet attrs) {
        super(context, attrs);
        BasicZoomControl mZoomControl = new BasicZoomControl();
        BasicZoomListener mZoomListener = new BasicZoomListener();
        mZoomListener.setZoomControl(mZoomControl);
        setZoomState(mZoomControl.getZoomState());
        setOnTouchListener(mZoomListener);
        mZoomControl.setAspectQuotient(getAspectQuotient());
    }

    public void setImage(Bitmap bitmap) {
        mBitmap = bitmap;
        mAspectQuotient.updateAspectQuotient(getWidth(), getHeight(), mBitmap.getWidth(),
mBitmap.getHeight());
        mAspectQuotient.notifyObservers();
        invalidate();
    }

    private void setZoomState(ZoomState state) {
        if (mState != null) mState.deleteObserver(this);
        mState = state;
        mState.addObserver(this);
```

```
        invalidate();
    }

    private AspectQuotient getAspectQuotient() {
        return mAspectQuotient;
    }

    @Override
    protected void onDraw(Canvas canvas) {
        if (mBitmap != null && mState != null) {
            final float aspectQuotient = mAspectQuotient.get();
            final int viewWidth = getWidth();
            final int viewHeight = getHeight();
            final int bitmapWidth = mBitmap.getWidth();
            final int bitmapHeight = mBitmap.getHeight();
            final float panX = mState.getPanX();
            final float panY = mState.getPanY();
            final float zoomX = mState.getZoomX(aspectQuotient) * viewWidth / bitmapWidth;
            final float zoomY = mState.getZoomY(aspectQuotient) * viewHeight / bitmapHeight;
            mRectSrc.left = (int) (panX * bitmapWidth - viewWidth / (zoomX * 2));
            mRectSrc.top = (int) (panY * bitmapHeight - viewHeight / (zoomY * 2));
            mRectSrc.right = (int) (mRectSrc.left + viewWidth / zoomX);
            mRectSrc.bottom = (int) (mRectSrc.top + viewHeight / zoomY);
            mRectDst.left = 0;
            mRectDst.top = 0;
            mRectDst.right = getWidth();
            mRectDst.bottom = getHeight();
            if (mRectSrc.left < 0) {
                mRectDst.left += -mRectSrc.left * zoomX;
                mRectSrc.left = 0;
            }
```

```
            if (mRectSrc.right > bitmapWidth) {
                mRectDst.right -= (mRectSrc.right - bitmapWidth) * zoomX;
                mRectSrc.right = bitmapWidth;
            }
            if (mRectSrc.top < 0) {
                mRectDst.top += -mRectSrc.top * zoomY;
                mRectSrc.top = 0;
            }
            if (mRectSrc.bottom > bitmapHeight) {
                mRectDst.bottom -= (mRectSrc.bottom - bitmapHeight) * zoomY;
                mRectSrc.bottom = bitmapHeight;
            }
            mRectDst.left = 0;
            mRectDst.top = 0;
            mRectDst.right = viewWidth;
            mRectDst.bottom = viewHeight;
            canvas.drawBitmap(mBitmap, mRectSrc, mRectDst, mPaint);
        }
    }

    @Override
    protected void onLayout(boolean changed, int left, int top, int right, int bottom) {
        super.onLayout(changed, left, top, right, bottom);
        mAspectQuotient.updateAspectQuotient(right - left, bottom - top,
mBitmap.getWidth(), mBitmap.getHeight());
        mAspectQuotient.notifyObservers();
    }

    @Override
    public void update(Observable observable, Object data) {
        invalidate();
```

```
    }

    private static class BasicZoomListener implements OnTouchListener {
        private BasicZoomControl mZoomControl;
        private float mFirstX = -1;
        private float mFirstY = -1;
        private float mSecondX = -1;
        private float mSecondY = -1;
        private int mOldCounts = 0;

        public void setZoomControl(BasicZoomControl control) {
            mZoomControl = control;
        }

        @SuppressLint("ClickableViewAccessibility")
        public boolean onTouch(View v, MotionEvent event) {

            switch (event.getAction()) {
                case MotionEvent.ACTION_DOWN:
                    mOldCounts = 1;
                    mFirstX = event.getX();
                    mFirstY = event.getY();
                    break;
                case MotionEvent.ACTION_MOVE: {
                    float fFirstX = event.getX();
                    float fFirstY = event.getY();
                    int nCounts = event.getPointerCount();
                    if (1 == nCounts) {
                        mOldCounts = 1;
                        float dx = (fFirstX - mFirstX) / v.getWidth();
                        float dy = (fFirstY - mFirstY) / v.getHeight();
```

```
                    mZoomControl.pan(-dx, -dy);
                } else if (1 == mOldCounts) {
                    mSecondX = event.getX(event.getPointerId(nCounts - 1));
                    mSecondY = event.getY(event.getPointerId(nCounts - 1));
                    mOldCounts = nCounts;
                } else {
                    float fSecondX = event.getX(event.getPointerId(nCounts - 1));
                    float fSecondY = event.getY(event.getPointerId(nCounts - 1));
                    double nLengthOld = getLength(mFirstX, mFirstY, mSecondX,
mSecondY);
                    double nLengthNow = getLength(fFirstX, fFirstY, fSecondX,
fSecondY);
                    float d = (float) ((nLengthNow - nLengthOld) / v.getWidth());
                    mZoomControl.zoom((float) Math.pow(20, d), ((fFirstX + fSecondX) /
2 / v.getWidth()), ((fFirstY + fSecondY) / 2 / v.getHeight()));
                    mSecondX = fSecondX;
                    mSecondY = fSecondY;
                }
                mFirstX = fFirstX;
                mFirstY = fFirstY;
                break;
            }

        }

        return true;
    }

    private double getLength(float x1, float y1, float x2, float y2) {
        return Math.sqrt(Math.pow(x1 - x2, 2) + Math.pow(y1 - y2, 2));
    }
```

```
    }

    private static class BasicZoomControl implements Observer {
        private static final float MIN_ZOOM = 1;
        private static final float MAX_ZOOM = 16;
        private final ZoomState mState = new ZoomState();
        private AspectQuotient mAspectQuotient;

        public void setAspectQuotient(AspectQuotient aspectQuotient) {
            if (mAspectQuotient != null) mAspectQuotient.deleteObserver(this);
            mAspectQuotient = aspectQuotient;
            mAspectQuotient.addObserver(this);
        }

        public ZoomState getZoomState() {
            return mState;
        }

        public void zoom(float f, float x, float y) {
            final float aspectQuotient = mAspectQuotient.get();
            final float prevZoomX = mState.getZoomX(aspectQuotient);
            final float prevZoomY = mState.getZoomY(aspectQuotient);
            mState.setZoom(mState.getZoom() * f);
            limitZoom();
            final float newZoomX = mState.getZoomX(aspectQuotient);
            final float newZoomY = mState.getZoomY(aspectQuotient);
            mState.setPanX(mState.getPanX() + (x - .5f) * (1f / prevZoomX - 1f / newZoomX));
            mState.setPanY(mState.getPanY() + (y - .5f) * (1f / prevZoomY - 1f / newZoomY));
            limitPan();
            mState.notifyObservers();
        }
```

```
public void pan(float dx, float dy) {
    final float aspectQuotient = mAspectQuotient.get();
    mState.setPanX(mState.getPanX() + dx / mState.getZoomX(aspectQuotient));
    mState.setPanY(mState.getPanY() + dy / mState.getZoomY(aspectQuotient));
    limitPan();
    mState.notifyObservers();
}

private float getMaxPanDelta(float zoom) {
    return Math.max(0f, .5f * ((zoom - 1) / zoom));
}

private void limitZoom() {
    if (mState.getZoom() < MIN_ZOOM) {
        mState.setZoom(MIN_ZOOM);
    } else if (mState.getZoom() > MAX_ZOOM) {
        mState.setZoom(MAX_ZOOM);
    }
}

private void limitPan() {
    final float aspectQuotient = mAspectQuotient.get();
    final float zoomX = mState.getZoomX(aspectQuotient);
    final float zoomY = mState.getZoomY(aspectQuotient);
    final float panMinX = .5f - getMaxPanDelta(zoomX);
    final float panMaxX = .5f + getMaxPanDelta(zoomX);
    final float panMinY = .5f - getMaxPanDelta(zoomY);
    final float panMaxY = .5f + getMaxPanDelta(zoomY);
    if (mState.getPanX() < panMinX) mState.setPanX(panMinX);
    if (mState.getPanX() > panMaxX) mState.setPanX(panMaxX);
```

```
            if (mState.getPanY() < panMinY) mState.setPanY(panMinY);
            if (mState.getPanY() > panMaxY) mState.setPanY(panMaxY);
        }

        public void update(Observable observable, Object data) {
            limitZoom();
            limitPan();
        }
    }

    private static class AspectQuotient extends Observable {
        private float mAspectQuotient;

        public float get() {
            return mAspectQuotient;
        }

        public void updateAspectQuotient(float viewWidth, float viewHeight, float
contentWidth, float contentHeight) {
            final float aspectQuotient = (contentWidth / contentHeight) / (viewWidth /
viewHeight);
            if (aspectQuotient != mAspectQuotient) {
                mAspectQuotient = aspectQuotient;
                setChanged();
            }
        }
    }

    private static class ZoomState extends Observable {
        private float mZoom;
        private float mPanX;
```

```
    private float mPanY;

    public float getPanX() {
        return mPanX;
    }

    public float getPanY() {
        return mPanY;
    }

    public float getZoom() {
        return mZoom;
    }

    public float getZoomX(float aspectQuotient) {
        return Math.min(mZoom, mZoom * aspectQuotient);
    }

    public float getZoomY(float aspectQuotient) {
        return Math.min(mZoom, mZoom / aspectQuotient);
    }

    public void setPanX(float panX) {
        if (panX != mPanX) {
            mPanX = panX;
            setChanged();
        }
    }

    public void setPanY(float panY) {
        if (panY != mPanY) {
```

```
                mPanY = panY;
                setChanged();
            }
        }

        public void setZoom(float zoom) {
            if (zoom != mZoom) {
                mZoom = zoom;
                setChanged();
            }
        }
    }

}

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ZoomImageView zoomImg = (ZoomImageView) findViewById(R.id.image);
        Bitmap bitmap = BitmapFactory.decodeResource(this.getResources(), R.mipmap.c);
        zoomImg.setImage(bitmap);
    }
}
```

实例 046 实验结果：

实例 047 部分关键源码：

```
public class MainActivity extends AppCompatActivity {

    private BrokenView mBrokenView;
    private RelativeLayout parentLayout;
    private ImageView imageView;
    private boolean hasAlpha;
    private int[] resids;
    private BrokenTouchListener colorfulListener, whiteListener;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initView();
        mBrokenView = BrokenView.add2Window(this);
```

```
        Paint whitePaint = new Paint();
        whitePaint.setColor(0xffffffff);
        colorfulListener = new BrokenTouchListener.Builder(mBrokenView).build();
        whiteListener = new BrokenTouchListener.Builder(mBrokenView).setPaint(whitePaint).build();
        setOnTouchListener();
    }

    private void initView() {
        parentLayout = (RelativeLayout) findViewById(R.id.demo_parent);
        imageView = (ImageView) findViewById(R.id.demo_image);
        ImageButton imageButton = (ImageButton) findViewById(R.id.action_refresh);
        imageButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (v.getId() == R.id.action_refresh) {
                    mBrokenView.reset();
                    refreshDate();
                    setOnTouchListener();
                    setViewVisible();
                }
            }
        });
        @SuppressLint("Recycle") TypedArray ar = getResources().obtainTypedArray(R.array.imgArray);
        int len = ar.length();
        resids = new int[len];
        for (int i = 0; i < len; i++) resids[i] = ar.getResourceId(i, 0);
        refreshDate();
    }
```

```
    private void setViewVisible() {
        parentLayout.setVisibility(View.VISIBLE);
        imageView.setVisibility(View.VISIBLE);
    }

    public void refreshDate() {
        Random rand = new Random();
        int pos = rand.nextInt(resids.length);
        imageView.setImageResource(resids[pos]);
        hasAlpha = pos == 0 || pos == 1 || pos == 2;
    }

    @SuppressLint("ClickableViewAccessibility")
    public void setOnTouchListener() {
        parentLayout.setOnTouchListener(colorfulListener);
        if (hasAlpha) {
            imageView.setOnTouchListener(whiteListener);
        } else {
            imageView.setOnTouchListener(colorfulListener);
        }
    }
}

public class BrokenTouchListener implements View.OnTouchListener {
    private BrokenView brokenView;
    private BrokenConfig config;

    private BrokenTouchListener(Builder builder) {
        brokenView = builder.brokenView;
        config = builder.config;
    }
```

```
public static class Builder {
    private BrokenConfig config;
    private BrokenView brokenView;

    public Builder(BrokenView brokenView) {
        this.brokenView = brokenView;
        config = new BrokenConfig();
    }

    public Builder setPaint(Paint paint) {
        config.paint = paint;
        return this;
    }

    public BrokenTouchListener build() {
        return new BrokenTouchListener(this);
    }
}

@SuppressLint("ClickableViewAccessibility")
@Override
public boolean onTouch(View v, MotionEvent event) {
    if (brokenView.isEnable()) {
        BrokenAnimator brokenAnim;
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                if (config.childView != null) {
                    config.region = new Region(config.childView.getLeft(),
                            config.childView.getTop(),
                            config.childView.getRight(),
```

```
                                config.childView.getBottom());
                    }
                    if (config.region == null || config.region.contains((int) event.getX(),
(int) event.getY())) {
                        Point point = new Point((int) event.getRawX(), (int)
event.getRawY());
                        brokenAnim = brokenView.getAnimator(v);
                        if (brokenAnim == null)
                            brokenAnim = brokenView.createAnimator(v, point, config);
                        if (brokenAnim == null) return true;
                        if (!brokenAnim.isStarted()) {
                            brokenAnim.start();
                            brokenView.onBrokenStart(v);
                        } else if (brokenAnim.doReverse()) {
                            brokenView.onBrokenRestart(v);
                        }
                        v.getParent().requestDisallowInterceptTouchEvent(true);
                    } else {
                        return false;
                    }
                    break;
                case MotionEvent.ACTION_UP:
                    brokenAnim = brokenView.getAnimator(v);
                    if (brokenAnim != null && brokenAnim.isStarted()) {
                        if (brokenAnim.doReverse()) brokenView.onBrokenCancel(v);
                    }
                    break;
            }
            return true;
        } else {
            return false;
```

```
        }
    }
}
```

实例 047 实验结果：

结果分析：

实例 044 中，使用了 Paint 画笔，结合手机权限，申请存储空间，存储手势在手机内存。但在执行手势的过程中遇到了权限申请错误，研究了一会但还是没有成功。

实例 046 中，使用了 Paint 和 Bitmap 工具，实现对图片的放大缩小功能。

实例 047 中，使用了 Paint 和 Bitmap 工具，还涉及到了长按事件处理。

桂林航天工业学院学生实验报告

课程名称	移动应用软件综合开发		实验项目名称	实验四 图片处理	
开课教学单位及实验室	计算机科学与工程学院 巡天 308			实验日期	2020 年 05 月 12 日
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	林奕森		实验成绩		

一、实验目的

1、实现图像处理使用功能。

二、实验原理

1、使用 Android Studio 创建工程及类，编写程序。

2、调试程序，并在 Genymotion 中运行程序。

三、实验操作方法和步骤

1、完成 实例 050 浏览朋友圈图片

四、实验结果与分析

实例 050 部分关键源码：

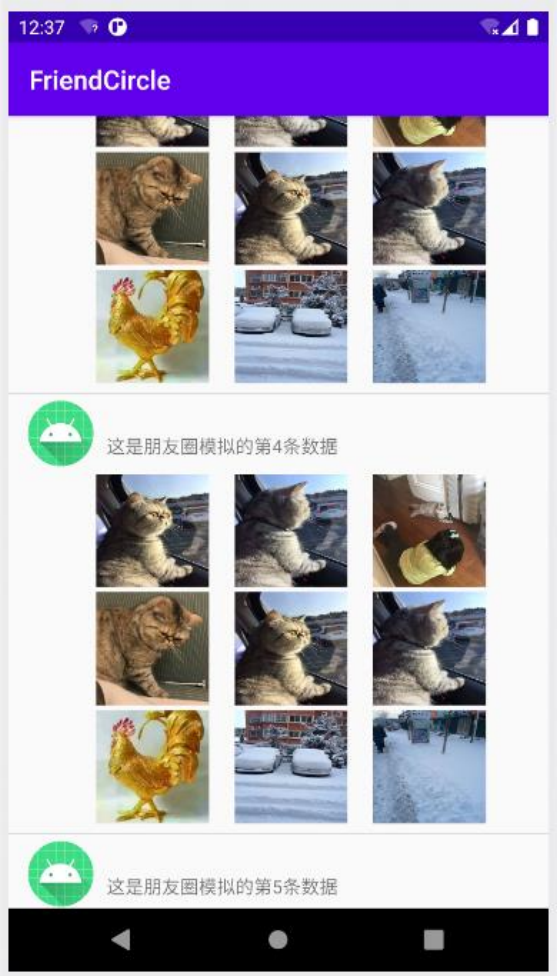
```
public class ShowImageActivity extends AppCompatActivity {
    private int index = 0;

    @RequiresApi(api = Build.VERSION_CODES.KITKAT)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.show_image_layout);
        init();
    }

    @RequiresApi(api = Build.VERSION_CODES.KITKAT)
```

```
    private void init() {
        final Intent intent = getIntent();
        ViewPager viewPager = (ViewPager) findViewById(R.id.show_view_pager);
        List<View> listViews = new ArrayList<>();
        for (int i = 0; i < Objects.requireNonNull(intent.getIntArrayExtra("image")).length; i++) {
            View view = LayoutInflater.from(getApplicationContext()).inflate(R.layout.view_pager_item, null);
            ImageView iv = (ImageView) view.findViewById(R.id.view_image);

            iv.setBackgroundResource(Objects.requireNonNull(intent.getIntArrayExtra("image"))[i]);
            listViews.add(view);
            iv.setOnLongClickListener(new View.OnLongClickListener() {
                @Override
                public boolean onLongClick(View v) {
                    Toast.makeText(ShowImageActivity.this, "这是第" + (index + 1) + "图片", Toast.LENGTH_SHORT).show();
                    return false;
                }
            });
        }
        MyPagerAdapter adapter = new MyPagerAdapter(listViews);
        viewPager.setAdapter(adapter);
        viewPager.setOnPageChangeListener(new PageChangeListener());
        viewPager.setCurrentItem(intent.getIntExtra("id", 0));
    }

    private class PageChangeListener implements ViewPager.OnPageChangeListener {

        @Override
        public void onPageScrollStateChanged(int arg0) {
```

```
        }

        @Override
        public void onPageScrolled(int arg0, float arg1, int arg2) {

        }

        @Override
        public void onPageSelected(int arg0) {
            index = arg0;
        }
    }
}

public class MainActivity extends AppCompatActivity {
    private ListView listView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findID();
        init();
    }

    private void init() {
        MyListViewAdapter listViewAdapter = new MyListViewAdapter(this);
        listView.setAdapter(listViewAdapter);
    }
```

```
    private void findID() {
        listView = (ListView) findViewById(R.id.list_view);
    }
}
```

实例 050 实验结果：



结果分析：

实例 050 中，重写了 adpter 的方法，实现朋友圈的布局。使用列表嵌套 fragment，实现图片单击事件，此外还使用了 ViewPage 实现列表滑动。

桂林航天工业学院学生实验报告

课程名称	移动应用软件综合开发	实验项目名称	实验五 摄像头案例实验
开课教学单位及实验室	计算机科学与工程学院 巡天 308	实验日期	2020 年 06 月 24 日
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	林奕森	实验成绩	

一、实验目的

1、实现图像处理使用功能。

二、实验原理

1、使用 Android Studio 创建工程及类，编写程序。

2、调试程序，并在 Genymotion 中运行程序。

三、实验操作方法和步骤

1、完成 实例 066 调用系统相机获取图像

2、完成 实例 070 扫描与生成二维码

四、实验结果与分析

实例 066 部分关键源码：

```
public class MainActivity extends AppCompatActivity {
    private TextView txt;
    private ImageView image;
    private File file;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txt = (TextView) findViewById(R.id.txt);
        image = (ImageView) findViewById(R.id.show_image);
        Button btn = (Button) findViewById(R.id.button);
```

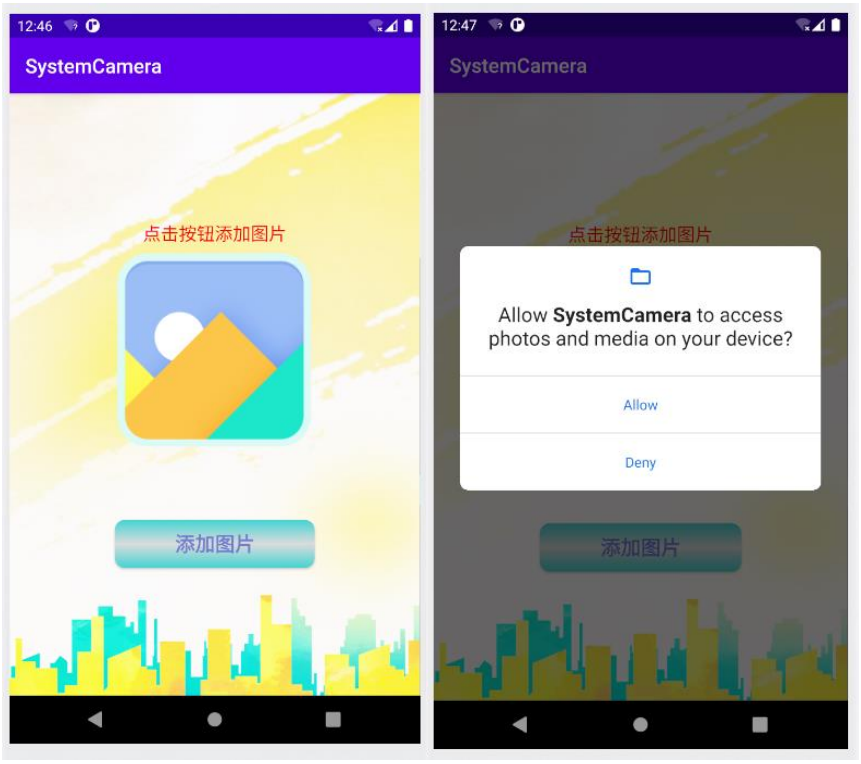
```
        btn.setOnClickListener(v -> {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                if (ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
                    if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
                        AlertDialog.Builder alert = new AlertDialog.Builder(this);
                        alert.setTitle("重要提示");
                        alert.setMessage("该权限用于SD卡的读写操作,如果取消将影响程序的
功能使用!");
                        alert.setPositiveButton("我知道了", (dialog, which) ->
applyForPermission());
                        alert.create();
                        alert.show();
                    } else {
                        applyForPermission();
                    }
                } else {
                    setDialog();
                }
            } else {
                setDialog();
            }
        });
    }

    private void applyForPermission() {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE}, 1000);
    }
```

```
    private void setDialog() {
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(this);
        alertDialog.setTitle("提示");
        alertDialog.setMessage("是否使用系统相机进行拍照？");
        alertDialog.setPositiveButton("确定",
                (dialog, which) -> {
                    Toast.makeText(this, "点击了确认", Toast.LENGTH_SHORT).show();
                    getPictureFromCamera();
                });
        alertDialog.setNeutralButton("取消", (dialog, which) -> Toast.makeText(this, "点击了取消", Toast.LENGTH_SHORT).show());
        alertDialog.create();
        alertDialog.show();
    }

    private void getPictureFromCamera() {
        if (!hasSdcard()) {
            Toast.makeText(this, "储存卡不可用", Toast.LENGTH_SHORT).show();
            return;
        }
        Intent intent = new Intent("android.media.action.IMAGE_CAPTURE");
        file = new File(Environment.getExternalStorageDirectory(), "my_image");
        intent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(file));
        startActivityForResult(intent, 1000);
    }

    private boolean hasSdcard() {
        return Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED);
    }
```

```
@SuppressLint("SetTextI18n")
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode != RESULT_OK) return;
    switch (requestCode) {
        case 1000:
            crop(Uri.fromFile(file));
            break;
        case 2000:
            image.setImageBitmap(BitmapFactory.decodeFile(file.getPath()));
            txt.setText("图片的保存路径【" + file.getAbsolutePath() + "】");
            break;
        default:
            break;
    }
}


private void crop(Uri uri) {
    Log.e("URI", Objects.requireNonNull(uri.getPath()));
    Log.e("URI", uri.toString());
    Intent intent = new Intent("com.android.camera.action.CROP");
    intent.setDataAndType(uri, "image/*");
    intent.putExtra("crop", "true");
    intent.putExtra("aspectX", 1);
    intent.putExtra("aspectY", 1);
    intent.putExtra("outputX", 150);
    intent.putExtra("outputY", 150);
    intent.putExtra("scale", true);
    intent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(file));
```

```
        intent.putExtra("outputFormat", Bitmap.CompressFormat.JPEG.toString());
        intent.putExtra("noFaceDetection", true);
        intent.putExtra("return-data", true);
        startActivityForResult(intent, 2000);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == 1000) {
            Log.i("申请权限个数", grantResults.length + "");
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                setDialog();
            } else {
                Toast.makeText(this, "未获得权限", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

实例 066 实验结果：


12:46
SystemCamera
点击按钮添加图片
添加图片
12:47
SystemCamera
点击按钮添加图片
Allow SystemCamera to access photos and media on your device?
Allow
Deny
添加图片


12:47
SystemCamera
点击按钮添加图片
提示
是否使用系统相机进行拍照?
取消
确定
添加图片

实例 070 部分关键源码：

```
public class MainActivity extends MPermissionsActivity {

    private static final int SCANCODE = 1;
    private TextView scanMessage;
    private ImageView enCodeImage1, enCodeImage2;
    private EditText editText;

    @SuppressLint("ClickableViewAccessibility")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        scanMessage = (TextView) findViewById(R.id.scan_txt);
        enCodeImage1 = (ImageView) findViewById(R.id.code_image1);
        enCodeImage2 = (ImageView) findViewById(R.id.code_image2);
        editText = (EditText) findViewById(R.id.input_txt);
        LinearLayout linearLayout = (LinearLayout) findViewById(R.id.ll);
        linearLayout.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                if (event.getAction() == MotionEvent.ACTION_DOWN) {
                    View view = getCurrentFocus();
                    if (view != null) {
                        ((InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE)).hideSoftInputFromWindow(view.getWindowTo
ken(), InputMethodManager.HIDE_NOT_ALWAYS);
                    }
                }
                return false;
            }
```

```
        });
    }

    public void scanCode(View view) {
        requestPermission(new String[]{Manifest.permission.CAMERA}, 1000);
    }

    @Override
    public void permissionSuccess(int requestCode) {
        super.permissionSuccess(requestCode);
        if (requestCode == 1000) {
            startActivityForResult(new Intent(this, CaptureActivity.class), SCANCODE);
        }
    }

    public void enCode1(View view) {
        if ("".equals(editText.getText().toString())) {
            Toast.makeText(this, "请在输入框中输入内容", Toast.LENGTH_SHORT).show();
            return;
        }
        Bitmap codeBitmap = EncodingUtils.createQRCode(editText.getText().toString(), 500,
500, null);
        enCodeImage1.setImageBitmap(codeBitmap);
    }

    public void enCode2(View view) {
        Bitmap logoBitmap = BitmapFactory.decodeResource(getResources(), R.mipmap.mrkj);
        Bitmap codeBitmap = EncodingUtils.createQRCode("http://www.mingrisoft.com/", 500,
500, logoBitmap);
        enCodeImage2.setImageBitmap(codeBitmap);
    }
```

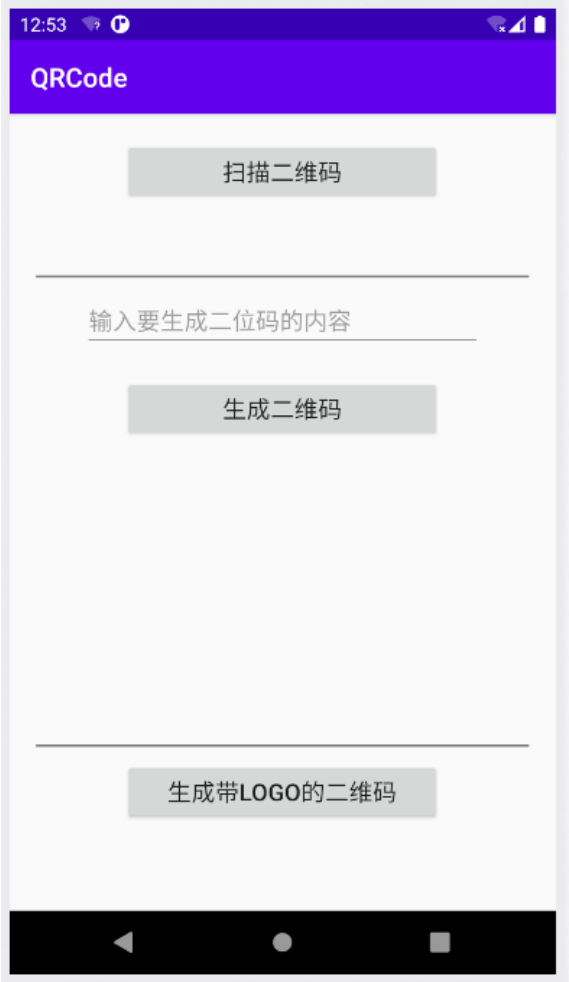
```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode != RESULT_OK) return;
        if (requestCode == SCANCODE) scanMessage.setText(data.getStringExtra("result"));
    }
}

public class MPermissionsActivity extends AppCompatActivity {
    private final String TAG = "MPermissions";
    private int REQUEST_CODE_PERMISSION = 0x00099;

    public void requestPermission(String[] permissions, int requestCode) {
        this.REQUEST_CODE_PERMISSION = requestCode;
        if (checkPermissions(permissions)) {
            permissionSuccess(REQUEST_CODE_PERMISSION);
        } else {
            List<String> needPermissions = getDeniedPermissions(permissions);
            ActivityCompat.requestPermissions(this,           needPermissions.toArray(new
String[0]), REQUEST_CODE_PERMISSION);
        }
    }

    private boolean checkPermissions(String[] permissions) {
        if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) return true;
        for (String permission : permissions) {
            if       (ContextCompat.checkSelfPermission(this,       permission)       !=
PackageManager.PERMISSION_GRANTED) {
                return false;
            }
```

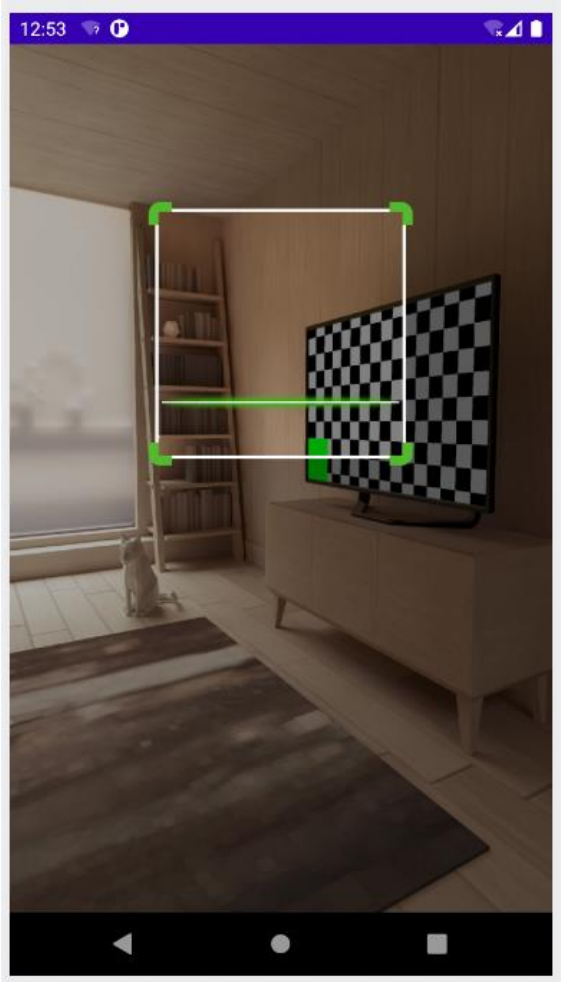
```
        }
        return true;
    }

    private List<String> getDeniedPermissions(String[] permissions) {
        List<String> needRequestPermissionList = new ArrayList<>();
        for (String permission : permissions) {
            if (ContextCompat.checkSelfPermission(this, permission) !=
                    PackageManager.PERMISSION_GRANTED ||
                    ActivityCompat.shouldShowRequestPermissionRationale(this,
permission)) {
                needRequestPermissionList.add(permission);
            }
        }
        return needRequestPermissionList;
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == REQUEST_CODE_PERMISSION) {
            if (verifyPermissions(grantResults)) {
                permissionSuccess(REQUEST_CODE_PERMISSION);
            } else {
                permissionFail(REQUEST_CODE_PERMISSION);
                showTipsDialog();
            }
        }
    }
```

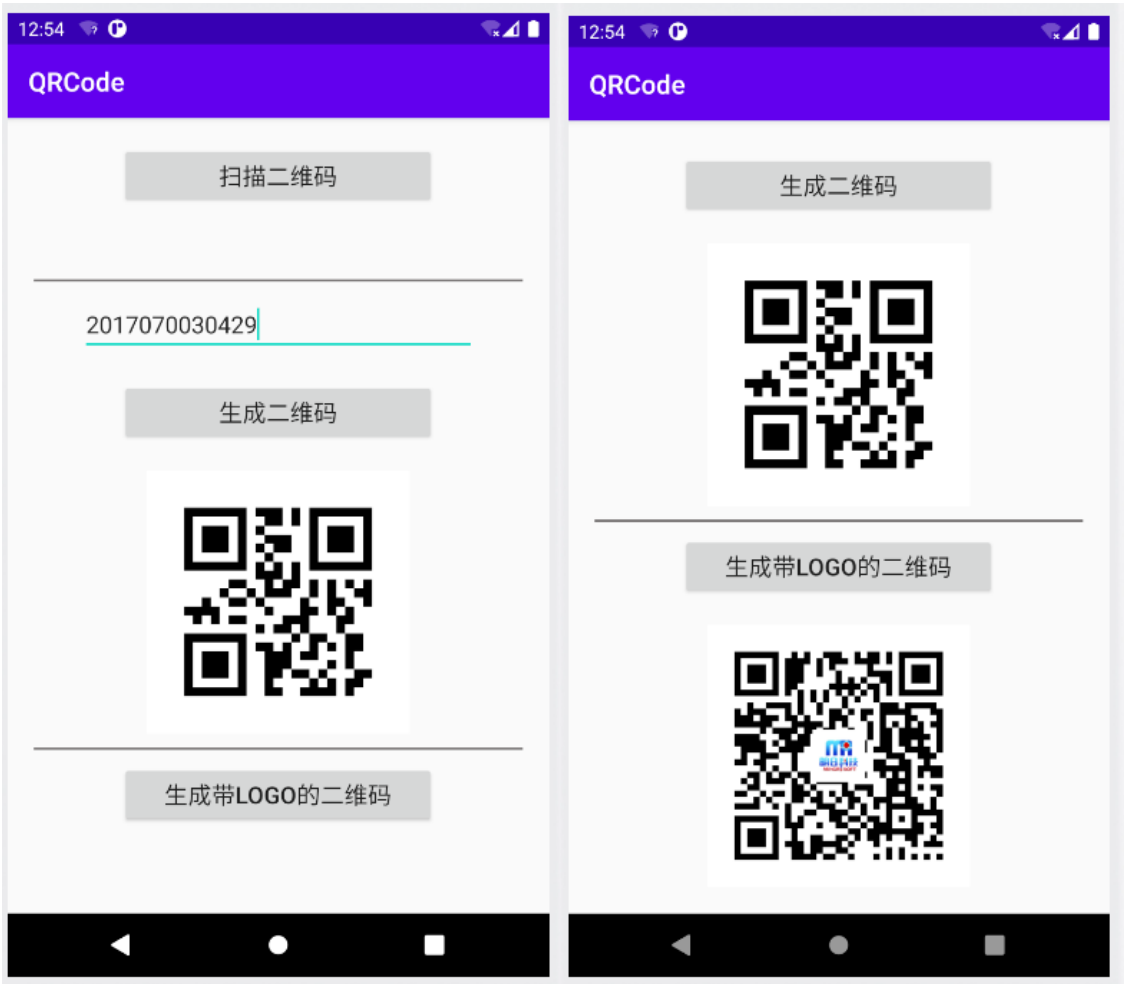
```
    private boolean verifyPermissions(int[] grantResults) {
        for (int grantResult : grantResults) {
            if (grantResult != PackageManager.PERMISSION_GRANTED) return false;
        }
        return true;
    }

    private void showTipsDialog() {
        new AlertDialog.Builder(this)
                .setTitle("提示信息")
                .setMessage("当前应用缺少必要权限，该功能暂时无法使用。如若需要，请单击【确
定】按钮前往设置中心进行权限授权。")
                .setNegativeButton("取消", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                    }
                })
                .setPositiveButton("确定", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        startAppSettings();
                    }
                }).show();
    }

    private void startAppSettings() {
        Intent intent = new Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
        intent.setData(Uri.parse("package:" + getPackageName()));
        startActivity(intent);
    }
```

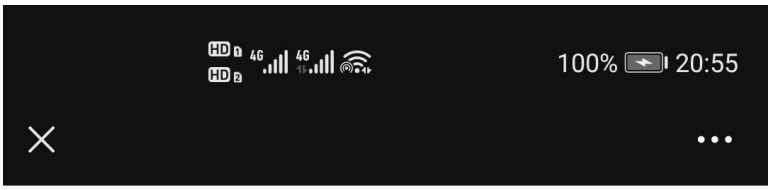
```
    public void permissionSuccess(int requestCode) {
        Log.d(TAG, "获取权限成功=" + requestCode);

    }

    public void permissionFail(int requestCode) {
        Log.d(TAG, "获取权限失败=" + requestCode);
    }
}
```

实例 070 实验结果：









2017070030429

结果分析：

实例 066 中，使用了权限请求功能，在用户允许软件的权限请求后，理论可以调用系统相机拍照。但在实际使用中，还是遇到了权限请求失败问题。

实例 070 中，使用了第三方库：zing 包，通过 zing 包来识别和生成二维码。