

附件 1:

学 号	
成 绩	

移动终端开发技术 课程设计报告

题 目	欢乐写数字
班 级	软件技术二班
学 号	201825040225
姓 名	乔朝霞
小组成员	乔朝霞、张淇淘、郭海静、 李佳琪、李聪
指导教师	陈媛媛

2019 年 7 月 4 日

目录

第一章 需求分析.....	2
1.1 项目背景.....	2
第二章 相关技术.....	2
2.1 技术简介.....	2
2.2 移动平台开发.....	2
2.3 java 简介.....	2
2.5 开发工具简介.....	4
第三章 总体设计.....	4
3.1 系统功能模块设计.....	4
业务流程图.....	5
第四章 详细设计与实现.....	5
4.1 运行环境.....	5
4.2 开发工具及技术简介.....	5
4.3 系统首页设计.....	6
4.4 系统基本功能设计与实现.....	7
第五章 项目总结.....	11
5.1 项目特点.....	11
5.2 项目不足.....	11
5.3 项目体会.....	11

第一章 需求分析

1.1 项目背景

现在的小朋友认真学数字时的表情非常的痛苦，既然如此，我们做程序员的可以开发一个既可以玩又可以学习的 APP 软件，让小朋友们在娱乐的同时学习，大大增加了他们的学习乐趣与兴趣。

第二章 相关技术

2.1 技术简介

开发平台：采用 Android 平台并结合 java 语言进行软件的开发。

软件后台：采用 java 作为软件的后台语言。

2.2 移动平台开发

移动终端目前市场占有率最大的两大平台是 Google 的 Android 系统和苹果的 ios 系统，随着智能终端越来越便宜，而且 Android 平台有着应用的广泛性和开放性，占据了越来越多的终端产品，所以我们选择使用 Android 平台进行软件的开发和调试工作。

Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android 操作系统最初由 Andy Rubin 开发，主要支持手机。2005 年 8 月由 Google 收购注资。2007 年 11 月，Google 与 84 家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良 Android 系统。随后 Google 以 Apache 开源许可证的授权方式，发布了 Android 的源代码。第一部 Android 智能手机发布于 2008 年 10 月。Android 逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机等。2011 年第一季度，Android 在全球的市场份额首次超过塞班系统，跃居全球第一。2013 年的第四季度，Android 平台手机的全球市场份额已经达到 78.1%。

2.3 java 简介

Java 是一门面向对象编程语言，不仅吸收了 C++ 语言的各种优点，还摒弃了 C++ 里难以理解的多继承、指针等概念，因此 Java 语言具有功能强大和简单易用两个特征。Java

语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。

Java 具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点。Java 可以编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等。

Java 的优势：

(1) **简单性**：是 c++语法的纯净版。没有头文件，指针运算，结构，联合，操作符重载，虚基类等等。语法基于 c，因此学习起来完全不费力。

(2) **面向对象**：面向对象是一种程序设计技术，他将重点放在数据（即对象）和对象之间的接口上。

(3) **可移植性(跨平台性)**：这是 JAVA 的一个重要的优势。JAVA 代码或者说字节码、二进制码可以跨平台的移植，而不用管具体的操作系统和硬件环境。JAVA 本身就是面向网络的，只有在网络环境中才能显示出他的优势，比如：现在我有一个网络环境，要让我的笔记本和手机交互，笔记本环境是 windows 发出一个让手机定时录音的 JAVA 代码，手机只有简单 Linux 内核操作系统，照样可以完美的执行这个代码。“一次编写，随处运行”：“write once, run anywhere”

(4) **多线程**：多线程的使用可以带来更好的交互响应和实时行为。多线程的简单性是 Java 成为主流服务器端开发语言的主要原因之一。

(5) **安全性**：Java 适合于网络/分布式环境，为了达到这个目标，在安全性方面投入了很大的精力，使 Java 可以构建防病毒，防篡改的系统。

(6) **健壮性**：Java 是一种健壮的语言，吸收了 C/C++ 语言的优点，但去掉了其影响程序健壮性的部分（如：指针、内存的申请与释放等）。Java 程序不可能造成计算机崩溃。Java 系统仔细检测对内存的每次访问，确认它是合法的，而且不致引起任何问题。不过，即使 Java 程序也可能有错误。如果出现某种出乎意料之事，程序也不会崩溃，而是将该例外抛弃。再通过异常处理机制，程序就会发现这类例外，并加以处理。

2.5 开发工具简介

2.5.1 Android Studio

Android Studio 是谷歌推出一个 Android 集成开发工具，基于 IntelliJ IDEA. 类似 Eclipse ADT, Android Studio 提供了集成的 Android 开发工具用于开发和调试。

架构组成：在 IDEA 的基础上，Android Studio 提供基于 Gradle 的构建支持、Android 专属的重构和快速修复、提示工具以捕获性能、可用性、版本兼容性问题支持 ProGuard 和应用签名。

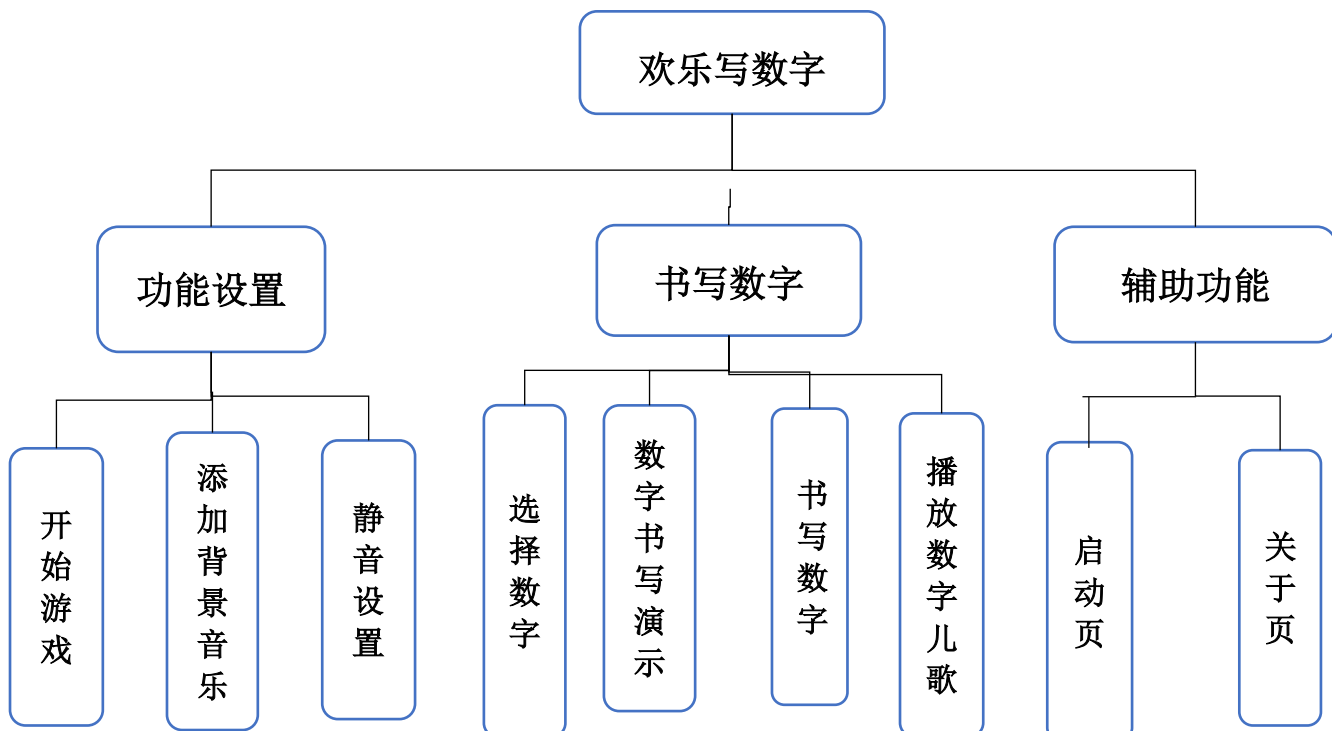
基于模板的向导来生成常用的 Android 应用设计和组件，功能强大的布局编辑器，可以让你拖拉 UI 控件并进行效果预览。

第三章 总体设计

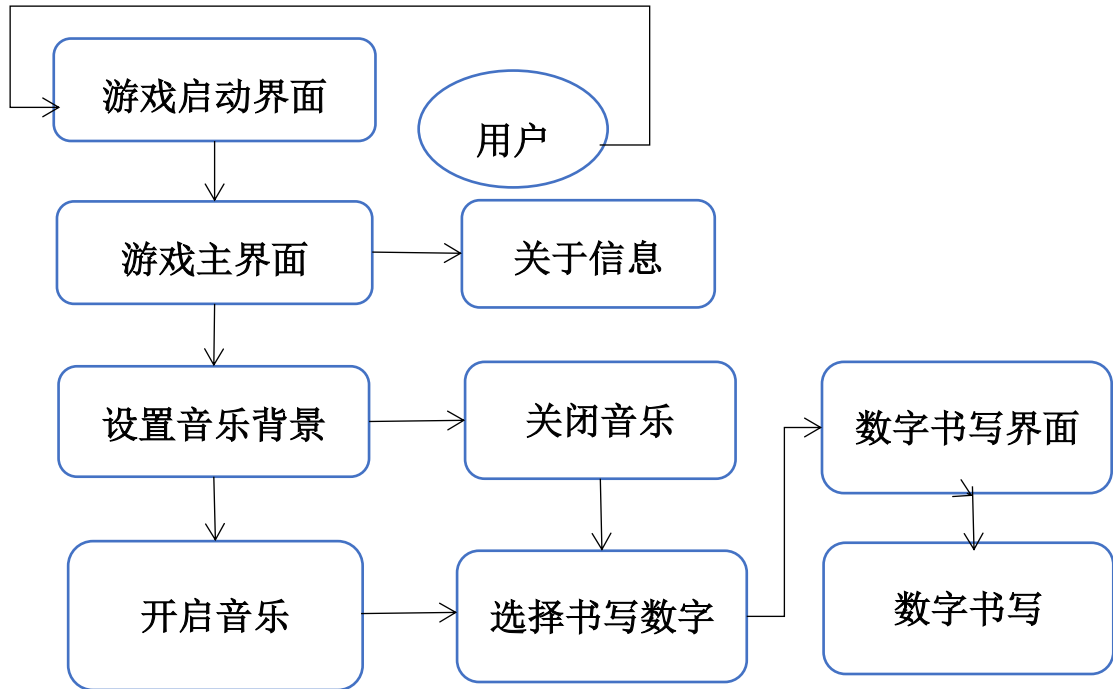
3.1 系统功能模块设计

系统功能模块功能设置模块、书写数字模块和辅助功能模块。

- (1) 功能设置模块：设置背景音乐的开关以及开始游戏。
- (2) 书写数字模块：选择书写的数字并书写，播放数字儿歌。
- (3) 辅助功能：启动页、关于页，用户的个人信息。



业务流程图



第四章 详细设计与实现

4.1 运行环境

1. 硬件环境

系统的硬件环境配置为处理器骁龙 800，运行内存 2G 以上。

2. 软件环境

系统的软件环境配置为 Android5.0 以上操作系统，Mysql 数据库，JDK1.5、MyEclipse5.1 开发工具包和 Tomcat6.0 服务器。

4.2 开发工具及技术简介

以下内容是对本系统所采用的开发工具和技术进行的简单介绍。

4.2.1 开发工具简介

Android Studio 是谷歌推出一个 Android 集成开发工具，基于 IntelliJ IDEA. 类似 Eclipse ADT, Android Studio 提供了集成的 Android 开发工具用于开发和调试。

架构组成：在 IDEA 的基础上，Android Studio 提供基于 Gradle 的构建支持、Android 专属的重构和快速修复、提示工具以捕获性能、可用性、版本兼容性问题支持 ProGuard 和应用签名。

基于模板的向导来生成常用的 Android 应用设计和组件，功能强大的布局编辑器，可以让你拖拉 UI 控件并进行效果预览。

4.2.2 技术简介

本系统采用 MVC 框架，MVC (Model-View-Controller) 是一种设计模式，它强制性地把应用程序的输入、处理和输出分开。MVC 把应用程序分成 3 个核心：模型层、视图层和控制层，它们分别担负不同的任务。其中视图层向用户显示相关的数据，并能接受用户的输入数据，但是它并不进行任何实际的业务处理。模型层表示业务数据和业务逻辑。控制层接收用户输入并调用模型和视图去完成用户的需求。

4.3 系统首页设计

本系统网站的主界面设计结构采用了上中下的结构，上、中部分为视图页面，下部分为游戏开始键、静音设置键以及查看用户信息按钮，界面设计如图所示。



4.4 系统基本功能设计与实现

本节对系统主要功能设计进行详细的说明。

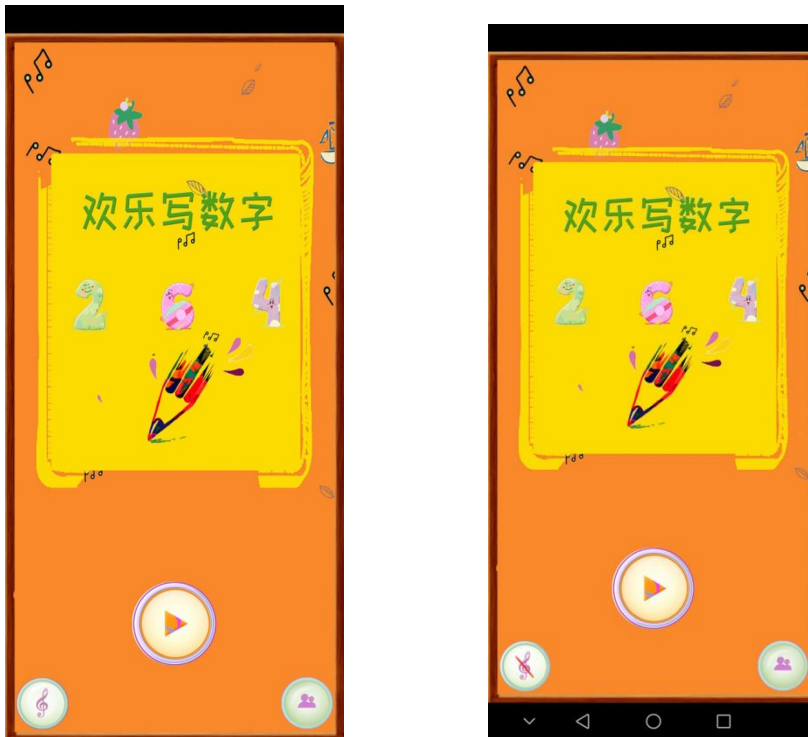
4.4.1 功能设置模块设计与实现

包括设置背景音乐的开关以及开始游戏的设计。

1. 背景音乐

(1) 实现目标

点击背景音乐图标后静音，再次点击播放。



(2) 实现过程

```
private void PlayMusic() {  
    mediaPlayer = MediaPlayer.create(this, R.raw.number_music);  
    mediaPlayer.setLooping(true);  
    mediaPlayer.start();  
}  
  
public void OnPlay(View v) {  
    startActivity(new Intent(MainActivity.this, SelectActivity.class));  
}
```



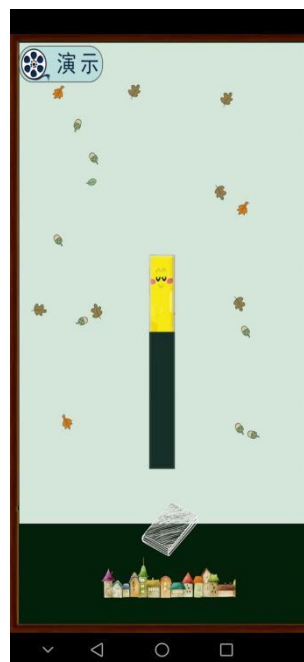
```
public void OnMusic(View v) {
    if (isPlay == true) {
        if (mediaPlayer != null) {
            mediaPlayer.stop();

            music_btn.setBackgroundResource(R.drawable.btn_music2);
            isPlay = false;
        }
    } else {
        PlayMusic();

        music_btn.setBackgroundResource(R.drawable.btn_music1);
        isPlay = true;
    }
}
```

2. 开始游戏

(1) 实现目标：点击按钮，开始游戏。



(2) 实现过程

```
public class StartActivity extends Activity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_start);

        Timer timer=new Timer();

        TimerTask timerTask=new TimerTask() {

            @Override

            public void run() {

                startActivity(new

Intent(StartActivity.this,MainActivity.class));

                finish();

            }

        };timer.schedule(timerTask,2000);

    }

}
```

4.4.2 书写数字的设计与实现

1. 实现目标



2. 实现过程

```
public class OneActivity extends Activity{
```

```
    MediaPlayer mediaPlayer;
```

```
    private ImageView iv_frame;
```

```
    int i = 1;
```

```
    float x1;
```

```
    float y1;
```

```
    float x2;
```

```
    float y2;
```

```
    float x3;
```

```
    float y3;
```

```
    int igvx;
```

```
    int igvy;
```

```
    int type = 0;
```

```

int widthPixels;
int heightPixels;
float scaleWidth;
float scaleHeight;
Timer touchTimer = null;
Bitmap arrdown;
boolean typedialog = true;
private LinearLayout linearLayout = null;

@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_number);
    if (MainActivity.isPlaying==true)
        PlayMusic();
    initView();
}

private void PlayMusic() {
    mediaPlayer = MediaPlayer.create(this, R.raw.music1);
    mediaPlayer.setLooping(true);
    mediaPlayer.start();
}

@Override
protected void onStop() {
    super.onStop();
    if (mediaPlayer != null) {
        mediaPlayer.stop();
    }
}

@Override

```

```

protected void onDestroy() {
    super.onDestroy();
    if (mediaPlayer != null) {
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
    }
}

private void initView() {
    iv_frame = (ImageView) findViewById(R.id.iv_frame);

    linearLayout = (LinearLayout) findViewById(R.id.LinearLayout1);

    LinearLayout write_layout= (LinearLayout)
findViewById(R.id.LinearLayout_number);

    write_layout.setBackgroundResource(R.drawable.bg1);

    widthPixels = this.getResources().getDisplayMetrics().widthPixels;

    heightPixels = this.getResources().getDisplayMetrics().heightPixels;

    scaleWidth = ((float) widthPixels / 720);
    scaleHeight = ((float) heightPixels / 1280);
    try {

        InputStream is = getResources().getAssets().open("onl_1.png");

        arrdown = BitmapFactory.decodeStream(is);
    } catch (IOException e) {
        e.printStackTrace();
    }

    LinearLayout.LayoutParams layoutParams = (LinearLayout.LayoutParams)

```

```

iv_frame.getLayoutParams();

layoutParams.width = (int) (arrdown.getWidth() * scaleHeight);

layoutParams.height = (int) (arrdown.getHeight() * scaleHeight);

iv_frame.setLayoutParams(layoutParams);
lodimagep(1);
linearLayout.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:

                x1 = event.getX();
                y1 = event.getY();
                igvx = iv_frame.getLeft();
                igvy = iv_frame.getTop();

                if (x1 >= igvx && x1 <= igvx + (int) (arrdown.getWidth()
* scaleWidth)
                    && y1 >= igvy & y1 <= igvy + (int)
(arrdown.getWidth() * scaleWidth)
                ) {
                    type = 1;
                } else {
                    type = 0;
                }
                break;
            case MotionEvent.ACTION_MOVE:
                igvx = iv_frame.getLeft();
                igvy = iv_frame.getTop();
                x2 = event.getX();
                y2 = event.getY();

```

4.4.3 关于页的设计与实现

1. 实现目标



2. 实现过程

```
public class AboutActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_about);  
    }  
  
    public void OnBack(View view) {  
        AboutActivity.this.finish();  
    }  
}
```

第五章 项目总结

5.1 项目特点

欢乐写数字软件包括功能设置模块、数字书写模块以及辅助功能模块。

可以播放音乐，随着鼠标书写，点击静音按钮对音乐进行静音，使用到了轮播和数据库的知识，大量运用了条件语句，使用到大量图片，使用到了 mediaPlayer 播放器。

5.2 项目不足

只能对数字“1”进行书写，“演示按钮”没有添加方法，点击没有反应。我们中也存在许多不足，很多代码不能理解，导致很多功能无法实现（其中包括后台）。

5.3 项目体会

在 Android 的学习过程中，可能会遇到形形色色的问题不容易解决，应多去专业论坛了解相关的知识，书本上的知识有限。要学会从网上搜索有用的信息加以整理，促进学习的深入和知识水平的提高。在学习的过程中，最好不是仅仅停留在 Java 表层，不是抄书上的例子运行出结果就可以。要注意，即便对一个简单的例子也要有耐心去琢磨、调试、改动。