学　号	201725040403
成　绩	

移动终端开发技术

课程设计报告

题　　目	闹钟
班　　级	软件技术四班
学　　号	**201725040403**
姓　　名	毕延珂
小组成员	毕延珂，宋睿智，盛兆玉，张世纪
指导教师	陈媛媛

2018 年 **7** 月 **5** 日

目 录

1 前言

1.1 研究背景

现在很多人由于需要处理各种事物，但是由于某些原因可能会忘记在某个时间段需要完成的工作和其他事情，因此需要一个闹钟来提醒自己，避免忘记这些应该做的事情。如果只有一个提醒事件，情况非常简单，但是对于多事件提醒的闹钟来说，如何解决提醒不冲突，如何使资源占据最小是一个迫切需要解决的问题。

1.2 研究意义

对于时间,闹钟是一个标度.时间不是一个在物质世界真实存在的东西,因而它需要某种实际物质作为指向,而闹钟就担当了这个角色。闹钟本身没有任何意义,是时间赋予了它意义,可以说闹钟存在的意义是依托时间而存在的。类似的关系还有尺规与空间等,它们本身没有意义,它们被它们所标度的物品赋予了意义。

闹钟的意义就是提醒人们注意一些容易忽略的时间

1.可以让人们养成良好的作息习惯，拒绝懒惰。

2.对一些时间掌握力不强的人，做到提醒通知，是家庭里面不错的小助手。

3.在照顾婴儿的时段,可以提醒妈妈们按时喂奶,以免错过时间。

1.3 主要内容

1.按照你设定的闹铃方式,准时响起;

2.可以根据用户的需求任意切换背景

3.响铃后，可以选择延时，可延迟闹钟响铃时间。

4.可设置重复响铃，不必用户重复设置。

1.4 软件特点

1. 简洁大方的控件界面。

2. 只要你选中了响铃,无论你手机音量设置怎样都会照常响铃。

3. 安装文件小,运行速度快。

4. 界面设计简单，易于新用户快速上手。

1.5 开发运行环境

Android Studio 是谷歌推出一个 Android 集成开发工具，基于 Inolil LIDEA 类以 Eclinse ADT， Android Studio 提供了集成的 Android 开发工具用于开发和调试，Android Studio 是谷歌推出了新的 Android 开发环境，开发者可以在编写程序的同时看到

自己的应用在不同尺寸屏罪中的样子。

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
- 提示工具以加获性能、可用性、版本兼容性等问题
- 支持 ProGuard 和应用签名
- 钻于模板的向导来生成常用的 Andrid 应用设计和组件
- 功能强大的布局编辑器，可以让你拖拉 ul 控件并进行效果预览

2 设计分析

2.1 题目重述

本次试验做的闹钟，主要有以下的几个要求：

1. 分析手机自带闹钟的功能实现，仿照并改良其功能结构。

2. 在实现其功能的同时添加了一些动画的背景，在使用的过程中，可以增加儿童使用的乐趣。

根据现代人的审美不断提高，所以考虑到闹钟的美观和易用性，我们精心挑选了一下动漫人物跟一些动漫的图标。

2.2 设计思路

设计思路流程：

1.进入的初始界面。

2.使用点击事件，选择设置闹钟响铃时间。

3.让闹钟能够读取系统时间，确保闹钟能准时响铃。

4.闹钟设置时间到时后，通过事件跳转，弹出响铃界面。

5.能够退出响铃界面。

6.优化页面布局，提升用户好感度。

2.3 待完善功能

1. 功能过于单一，只能实现闹钟功能、可以再进行设置一-下比较有新意的功能。

2. 不能实现多个闹钟的设置

3. 界面过于简陋，有很多地方比较差劲可以再进行改进一下。

4.由于个人或者团队的知识有一定缺陷，很多东西有些跟想象的不太一样，以后可以进行更多的知识积累，再进行这个小软件的改进。

3 程序说明

3.1 布局文件的代码及附图

android 中主要有六大布局，LinearLayout（线性布局），RelativeLayout（相对布局），AbsoluteLayout（绝对布局），GridLayout（网格布局），TableLayout（表格布局），FrameLayout（帧布局）。而为了更好的实现我们的界面我们使用了 LinearLavout（线性布局）。它包含的子控件将以横向或竖向的方式排列，按照相对位置来排列所有的 widgets 或者其他的 containers,超过边界时，某些控件将缺失或消失。因此一个垂直列表的每一行只会有一个 widget 或者是 container，而不管他们有多宽，而一个水平列表将会只有一个行高（高度为最高子控件的高度加上边框高度）。LinearLayout 保持其所包含的 widget 或者是 container 之间的间隔以及互相对齐（相对一个控件的右对齐、中间对齐或者左对齐）。

主界面：主界面我们使用了 LinearLayout（线性布局），通过代码跳转事件来时间对话框的弹出和跳转。

```
<LinearLayout  xmlns:android="
http://schemas.android.com/apk/res/android"
            android:id="@+id/layout"
        android:orientation="vertical"
```

```
  android:background="@drawable/background_a"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
  tools:context=".Start"  >
```

使用 android:layout_width 和 android:layout_height 来控制按钮的大小和对话框的大小。

```
<LinearLayout
          android:id="@+id/alarm_one"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_marginLeft="10dp"
          android:layout_marginTop="20dp">
        <TextView
                android:id="@+id/alarm"
          android:layout_width="wrap_content"
                android:layout_height="wrap_conten"
                android:layout_marginLeft="15dp"
                android:text="8:00"
                android:textSize="70px"/>
      </LinearLayout>
<LinearLayout
        android:layout_width="275dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="10dp"
        android:orientation="horizontal"  >
          <TextView
```

```
                android:layout_width="wrap_content"
               android:layout_height="wrap_content"
                     android:text="重复响铃"
          android:textSize="50px"/>
                <CheckBox
                       android:id="@+id/repeatingCheck"
                       android:layout_width="wrap_content"
                       android:layout_height="wrap_conten"
                       android:layout_marginLeft="190dp"
     </LinearLayout>
```

设置闹钟背景文本

```
    <TextView
           android:id="@+id/setBackground"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:layout_marginLeft="10dp"
      android:layout_marginTop="10dp"
           android:text="闹钟背景"
           android:textSize="50px"/>
```

使用 Button 按钮控件进行控制

```
    <Button
      android:id="@+id/cancel"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:layout_marginLeft="105dp"
      android:layout_marginTop="180dp"
           android:text="取消闹钟"/>
</LinearLayout>
```

利用布局文件来实现界面之间的跳转以及文本说明

```
<?xml  version="1.0"  encoding="utf-8"?>
<LinearLayout  xmlns:android="
http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"  >
        <ImageView
        android:id="@+id/header"
                android:layout_width="wrap_content"
                   android:layout_height="wrap_content"
                   android:paddingLeft="10dp"/>
```

使用 listview 来添加图片以及背景

3.2 主文件的代码及附图

主代码我们使用了点击事件和跳转来进行功能的实现

onclick 事件,即鼠标单击事件当鼠标点击事触发触发的内容根据我们的代码去执行。

所以我们广泛使用了 onclick 点击事件。

```
public class MainActivity extends Activity {
public static final String WALLPAPER_FILE="wallpaper_file";
```

设置访问权限：

```
private static int RESULT_LOAD_IMAGE = 1;
private LinearLayout layout;
private Button Cancel;
private TextView Alarm,SetBackground;
private LinearLayout AlarmOne;
private CheckBox RepeatingCheck;
boolean isClicked = false , isLate = false;
int count = 1,option = 0;;
private Calendar c = Calendar.getInstance();
AlertDialog builder = null;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
layout = (LinearLayout)findViewById(R.id.layout);
SetBackground = (TextView)findViewById(R.id.setBackground);
SetBackground.setOnClickListener(new TextView.OnClickListener(){
@Override
public void onClick(View arg0) {
```

```
// TODO Auto-generated method stub
final String[] PictureList={"更换背景","从相册选择照片"};
AlertDialog.Builder listDia=new AlertDialog.Builder(MainActivity.th
is);AlertDialog.Builder SystemDia=new AlertDialog.Builder(MainActiv
ity.this);
            SystemDia.setTitle("选择背景");
                                SystemDia.setItems(SystemPicture,
  new DialogInterface.OnClickListener() {
    @Override
   publicvoid onClick(DialogInterface dialog, int which) {
SharedPreferences.Editor editor = getSharedPreferences(WALLPAPER_FILE,
MODE_PRIVATE).edit();
```

定义 layout.setBackgroundResource 设置背景图片。使用 case 和

break 控件实现改变背景图片。

```
switch(which){case 0:
layout.setBackgroundResource(R.drawable.xiaohuangren);
editor.putInt("wellpaper", R.drawable.xiaohuangren);
editor.commit();
break;
case 1
layout.setBackgroundResource(R.drawable.background_a);
break;
case 2:
layout.setBackgroundResource(R.drawable.xiaowanzi);
```

```
                                    break;
                                    case 3:
layout.setBackgroundResource(R.drawable.duola);
                                        break;
                                        default:
                                        break;
                                          }
                                      }
                                   });
                        SystemDia.create().show();
                                 break;
                               case 1:
                                 Intent i = new Intent(
                                  Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                          startActivityForResult(i,RESULT_LOAD_IMAGE);
                               }}});
                       listDia.create().show();}
```

```
});
```

用 AlarmManager 控件监听系统时间，先获取 AlarmManager am，然后 am.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), pendingIntent);pendingIntent 启动的是一个广播意图，里面的 intent 将会定时发送到相应的 receiver，然后做出相应操作。

```
        RepeatingCheck = (CheckBox)findViewById(R.id.repeatingCheck);
        RepeatingCheck.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener(){
            @Override
```

```
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                isClicked = true;
            }
        });
Cancel = (Button)findViewById(R.id.cancel);
        Cancel.setOnClickListener(new Button.OnClickListener(){
            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                Intent intent = new Intent(MainActivity.this,
RingAlarm.class);
                PendingIntentpi=
PendingIntent.getBroadcast(MainActivity.this, 0,intent, 0);
                AlarmManageram=
(AlarmManager)getSystemService(ALARM_SERVICE);
                am.cancel(pi);
                Alarm.setText("");
    stopService(newIntent("com.example.alarm1.MUSIC"));
            }
        });
        Alarm = (TextView)findViewById(R.id.alarm);
        AlarmOne =
(LinearLayout)findViewById(R.id.alarm_one);AlarmOne.setOnClickListene
r(newLinearLayout.OnClickListene(){
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                c.setTimeInMillis(System.currentTimeMillis());
                 int mHour_0=c.get(Calendar.HOUR_OF_DAY);
```

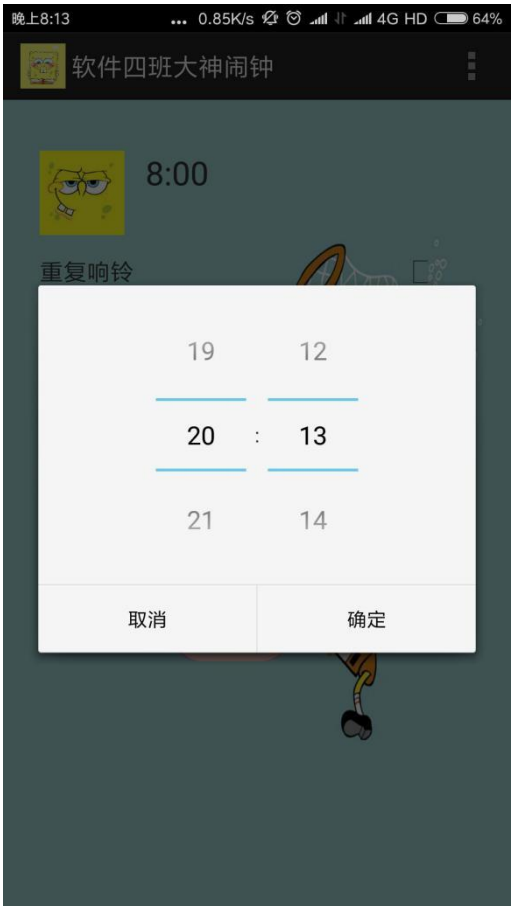
```
                int mMinute_0=c.get(Calendar.MINUTE);
                   new TimePickerDialog(MainActivity.this,
                     new TimePickerDialog.OnTimeSetListener()
                     {
public void onTimeSet(TimePicker view,int hourOfDay,  int minute)  {
c.setTimeInMillis(System.currentTimeMillis());
c.set(Calendar.HOUR_OF_DAY,hourOfDay);
c.set(Calendar.MINUTE,minute);
c.set(Calendar.SECOND,0);
c.set(Calendar.MILLISECOND,0);
long date = c.getTimeInMillis();
if(c.getTimeInMillis() < System.currentTimeMillis()){
c.set(Calendar.DAY_OF_YEAR, c.get(Calendar.DAY_OF_YEAR) + 1); }
Intentintent=newIntent(MainActivity.this,RingAlarm.class);
PendingIntentsender=PendingIntent.getBroadcast(
MainActivity.this,0, intent, 0);
AlarmManager am;
am=(AlarmManager)getSystemService(ALARM_SERVICE);
am.set(AlarmManager.RTC_WAKEUP, c.getTimeInMillis(),  sender  );
Stringstr=format(hourOfDay)+":"+format(minute);
                 Alarm.setText(str);
                SharedPreferences textChange = getPreferences(0);
                SharedPreferences.Editor editor_1 = textChange.edit();
                              editor_1.putString("TIME1", str);
                              editor_1.commit();
SharedPreferences timeShare = getPreferences(0);
              SharedPreferences.Editoreditor= timeShare.edit();
                              editor.putString("TIME1", str);
                              editor.commit();
```

```
Toast.makeText(MainActivity.this,"设置成功"+str,
                                    Toast.LENGTH_SHORT)
                                    .show();
                            if(!isClicked){
am.setRepeating(AlarmManager.RTC_WAKEUP, c.getTimeInMillis(),
          1 * 60 * 1000, sender);
                                isClicked = false;}}
                    },mHour_0,mMinute_0,true).show();
          }   });}
```

使用 onKeyUp 事件控制退出 APP

```
    private String format(int x)
    {
       String s=""+x;
      if(s.length()==1) s="0"+s;
      return s;  }
    @Override
    public boolean onKeyUp(int keyCode, KeyEvent event) {
              if(keyCode == KeyEvent.KEYCODE_BACK){
     builder = new AlertDialog.Builder(MainActivity.this)
       .setIcon(R.drawable.icon)
       .setTitle("c")
       .setMessage("确认退出")
       .setPositiveButton("退出",
         new DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog,
            int whichButton) {
             Intent intent = new Intent();
             intent.setAction(Intent.ACTION_MAIN);
             intent.addCategory(Intent.CATEGORY_HOME);
```

```
        startActivity(intent);}  })
    .setNegativeButton("返回",
      new DialogInterface.OnClickListener() {
       public void onClick(DialogInterface dialog,
         int whichButton) {
        builder.dismiss();
       }
      }).show();  }
           return true;
```



这是我们的主代码，主要是实现主界面的点击事件和事件闹钟的倒计时功能。

4 总结

记得在开学第一节课的时候老师让我们安装程序，我的没有装成功，是老师不断帮我才能让我在接下来的学习中有所进步，也让我感受到安卓的吸引力，从小喜欢玩游戏的我，对于安卓程序也是比较热爱，通过编程可以让人有一种自豪感，虽然一个学期的学习并不算长，但是我在一个学期中的收获不少，尤其是在安卓方面。学习其实是一种体会过程和分享成功的媒介，在学习中不断探索新的知识并且不断旧的知识，从而享受学习带给自己的快乐。良好的学习习惯是学习成败的关键。对于一个进入大学的我来说，学习习惯的好坏决定自己是否能取得满意的成绩。每一天坚持去自习便是一个十分好的习惯，选取自习的场所也因人而异，能够选取去图书馆或者公教区。每一天都要有一种信念：“我要坚持自习”。一旦构成了这种信念，你便会继续下去，无论是在大学还是以后工作了它都会伴随你。我就是构成这种自习的习惯，最终带给我许多的惊喜，我还要在接下来的生活中继续学习，继续努力，继续进步，最后我要感谢一下老师这一个学期对我们谆谆教诲，下学期的我将会更加努力！