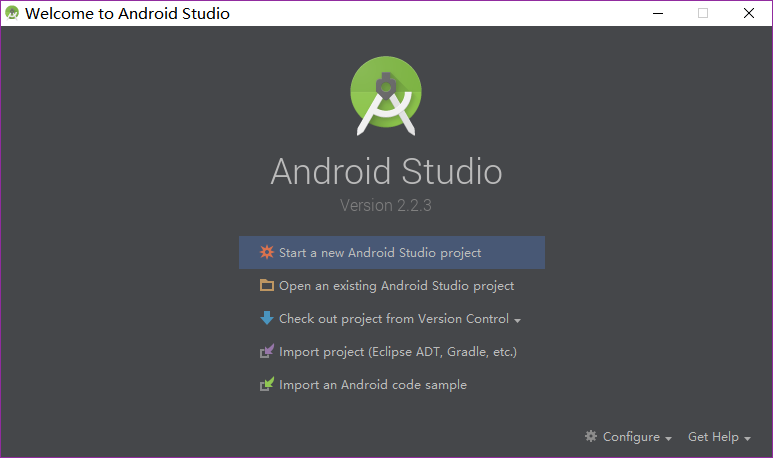
# Android课程设计——计算器

## 项目简介

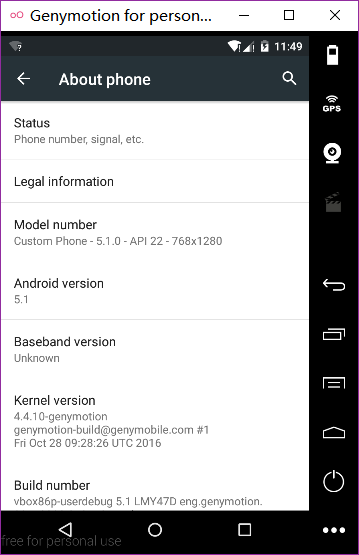
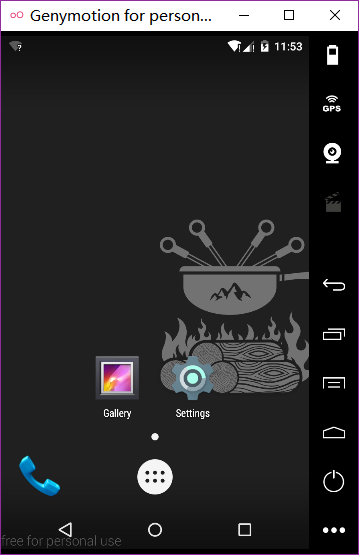
### 编写环境

Android Studio 2.2.3



### 运行环境

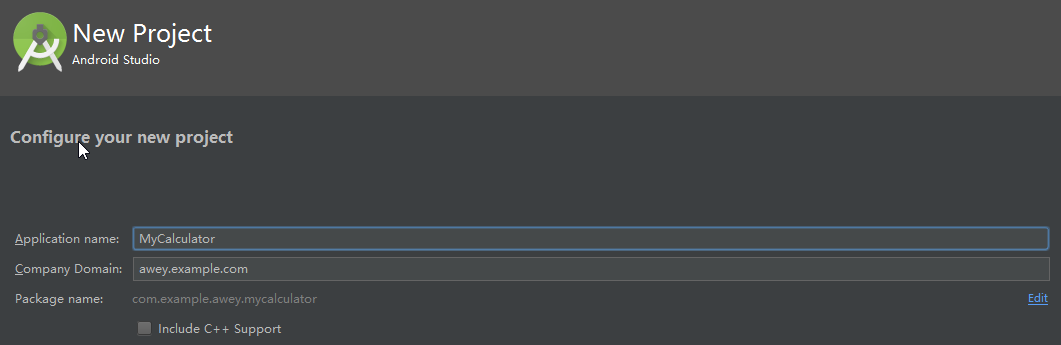
Genymotion模拟器 Custom Phone - 5.1.0 - API 22 768 × 1280



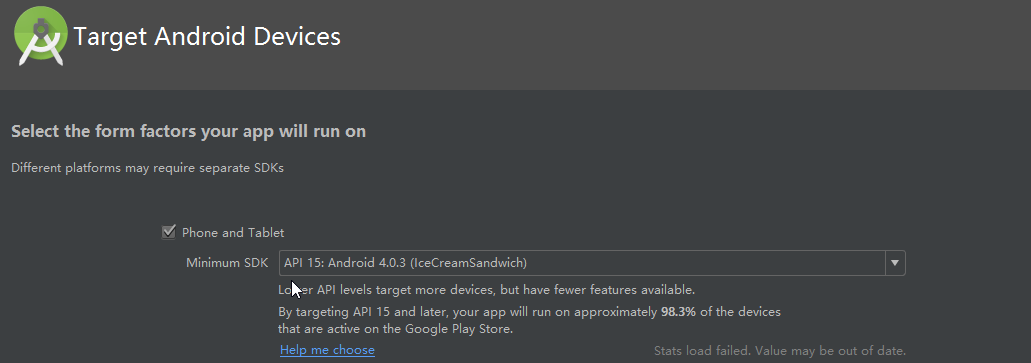
## 项目设计与实现

### 2.1创建项目

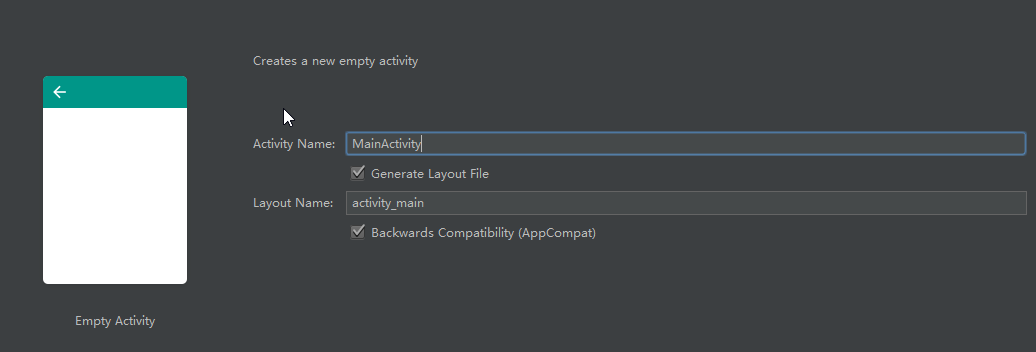
#### 新建工程MyCalculator



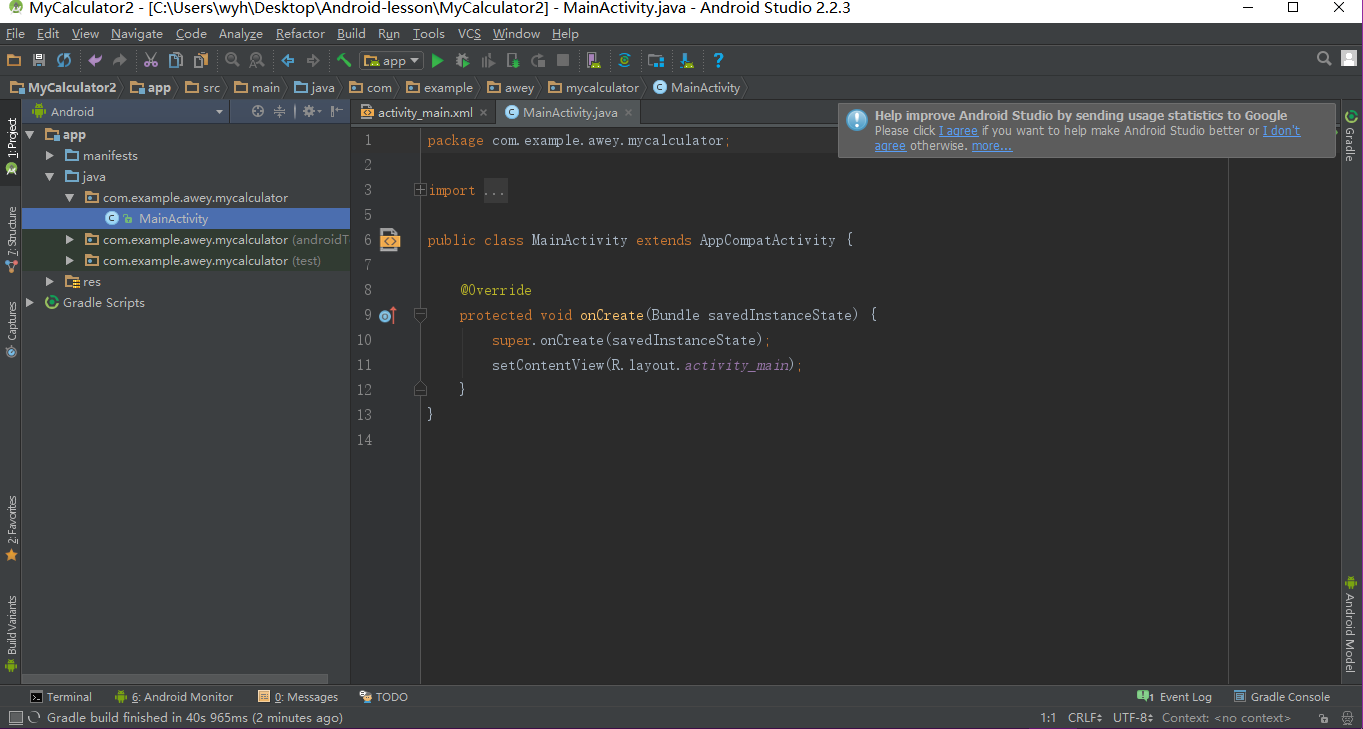
#### 设备选择



#### 模版选择



#### 完成项目创建

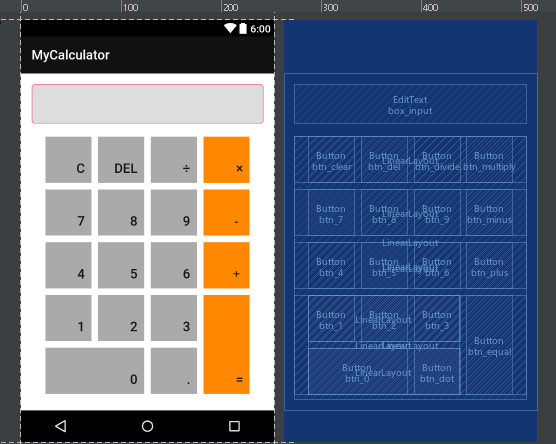


### 2.2页面布局

采用LinearLayout布局，上面为EditText框 ，下面是button键盘区域，

最终效果如图2-2

图 2-2 布局最终效果图



#### 1.设置文本框样式

在drawable文件夹下新建文件box.xml,修改样式，使其拥有圆角，背景填充色为白色，边框厚度为1dp、颜色为粉色

<!--圆角-->  
<corners android:radius="5dp" />  
<!--填充色-->  
<solid  
 android:color="#ddd"  
 />  
<!--边框-->  
<stroke  
 android:width="1dp"  
 android:color="@color/colorAccent"  
 />

在Activity\_main.xm引用box.xml

android:background="@drawable/box"

设置其editable属性为false，使其不能由键盘直接输入

android:editable="false"

其获取到按钮输入后应该从右下角开始呈现输入

android:gravity="bottom|right"

#### 键盘布局

整个键盘布局均为线性布局LinearLayout，与上面的EditText框有一定的外边距

1、2、3行中的四个按钮之间大小、左右边距相等，5、6行包含在一个LinearLayout中，其中Button\_equal（等号）左边包含一个LinearLayout，其有上下两层，如图2-2.2

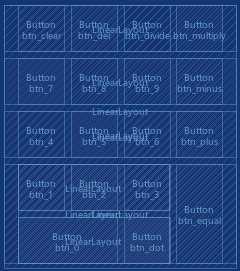


图 2-2.2 键盘布局

#### 3.修改按钮样式

按钮中的字符或数字均在按钮靠右下角

android:gravity="bottom|right"

按钮颜色“×”、“＋”、“－”和“=”为暗橙色，其余为暗灰色

android:background="@android:color/darker\_gray"

android:background="@android:color/holo\_orange\_dark"

### 2.3计算功能实现

#### 按钮和输入框

首先定义按钮和输入框

Button btn\_0; //数字0  
Button btn\_1; //数字1  
Button btn\_2; //数字2  
Button btn\_3; //数字3  
Button btn\_4; //数字4  
Button btn\_5; //数字5  
Button btn\_6; //数字6  
Button btn\_7; //数字7  
Button btn\_8; //数字8  
Button btn\_9; //数字9  
Button btn\_dot; //小数点  
  
Button btn\_plus; //加号  
Button btn\_minus; //减号  
Button btn\_multiply; //乘号  
Button btn\_divide; //除号  
Button btn\_equal; //等号  
  
Button btn\_clear; //清除  
Button btn\_del; //删除  
EditText box\_input; //显示框

然后实例化按钮和输入框，并给button加上（Button）进行强制转化

//实例化按钮

btn\_0 = (Button) findViewById(R.id.*btn\_0*);

btn\_1 = (Button) findViewById(R.id.*btn\_1*);

...

...  
btn\_equal = (Button) findViewById(R.id.*btn\_equal*);// 等于  
  
//实例化输入框  
box\_input = (EditText) findViewById(R.id.*box\_input*);

##### (1)给按钮设置点击事件

因为当前Activity已经实现了OnClickListener，所以在setOnClickListener传入this即可

//给按钮设置点击事件  
btn\_0.setOnClickListener(this);

btn\_1.setOnClickListener(this);  
...

...  
btn\_equal.setOnClickListener(this);

##### (2)实现按钮的点击

用swich判断在onClick方法里的当前点击的按钮，并把id对应的按钮显示到输入框中

数字0~9以及小数点按钮：

box\_input.setText("");

四则运算：

box\_input.setText(str + " " + ((Button) v).getText() + " ");

清除：直接把输入框内容设为空

box\_input.setText("");

删除：如果输入框不为空，则长度-1

if (str!=null&& !str.equals("")) {  
 box\_input.setText(str.substring(0,str.length()-1));  
 }

等号：用getResult方法计算记过

getResult();

#### 计算方法

private void getResult() { }

先取输入框的内容

String exp = box\_input.getText().toString();

然后根据内容对各种情况进行判断

如果当前内容为空或没有东西就直接返回

if (exp == null || exp.equals("")) {  
 return;  
}

因为每条运算式都包含有空格，所以以此来判断

##### 没有空格

比如用户点了数字直接点等号，因为没有运算符，直接return，不进行运算

if (!exp.contains(" ")) {//如果不包含空格（运算符前面有空格），直接返回（比如点了数字，没有运算符）  
 return;  
}

##### 有空格

首先对第一个数字、运算符和第二个数字进行定义

String s1 = exp.substring(0, exp.indexOf(" "));//运算符前面的字符串  
String op = exp.substring(exp.indexOf(" ") + 1, exp.indexOf(" ") + 2);//运算符  
String s2 = exp.substring(exp.indexOf(" ") + 3);//运算符后面的字符串

接下来s1和s2的情况进行分析

###### ①s1、s2非空

if (!s1.equals("") && !s2.equals("")) {  
 double d1 = Double.*parseDouble*(s1);  
 double d2 = Double.*parseDouble*(s2);  
 if (op.equals("＋")) {  
 result = d1 + d2;  
 } else if (op.equals("－")) {  
 result = d1 - d2;  
 } else if (op.equals("×")) {  
 result = d1 \* d2;  
 } else if (op.equals("÷")) {  
 if (d2 == 0) {  
 Toast.*makeText*(MainActivity.this, "除数不能为0！！！", Toast.*LENGTH\_LONG*).show();  
 result = 0;  
 } else {  
 result = d1 / d2;  
 }  
 }

｝

###### ②s1空，s2非空

else if(s1.equals("") && !s2.equals("")){  
 double d2 = Double.*parseDouble*(s2);  
 if (op.equals("＋")) {  
 result = 0 + d2;  
 } else if (op.equals("－")) {  
 result = 0 - d2;  
 } else if (op.equals("×")) {  
 result = 0 ;  
 } else if (op.equals("÷")) {  
 result = 0;  
 }  
}

###### ③s1、s2均为空

直接将内容清空

box\_input.setText("");

##### (3)清空

进行了一次运算之后，需要将上一次运算遗留下来的结果清空，否则将影响到下一次的运算

解决：在进行下一轮计算前将其str=””

至此，简单计算器功能已实现

### 2.4 夜间模式

## 项目心得

。。。。。

## 参考资料

1. 慕课网课程Android攻城狮的第一门课（入门篇） www.xxx.com
2. W3C SCHOOL: <http://www.w3school.com.cn/json/json_intro.asp>
3. 百度文档 Json结构、实例：

<http://wenku.baidu.com/view/b4c6a00e581b6bd97f19ea31.html>