

A

猜猜鸡蛋放在哪只鞋子里

A.1 概述

猜猜鸡蛋放在哪只鞋子里小游戏就是在窗体上放置 3 只鞋子，单击其中的任意一只鞋子，将打开鞋子显示里面是否有鸡蛋，并且将没有被单击的鞋子设置为半透明显示，被单击的正常显示，同时根据单击的鞋子里面是否有鸡蛋显示对应的结果。例如，单击中间的那只鞋子，如果鸡蛋在这只鞋子里，将显示“恭喜您，猜对了，祝你幸福!”的提示文字，否则，将显示“很抱歉，猜错了，要不要再试一次?”的提示文字。

A.1.1 功能描述

猜猜鸡蛋放在哪只鞋子里是一个愉悦身心的小游戏，它的功能结构如图 A-1 所示。

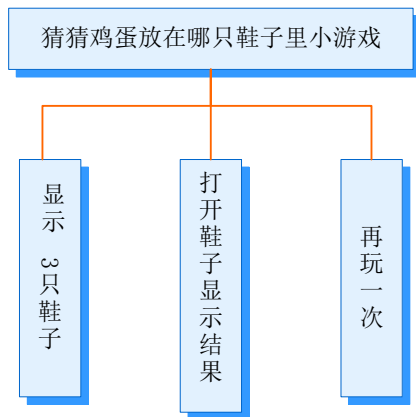


图 A-1 猜猜鸡蛋放在哪只鞋子里小游戏的结构图

A.1.2 系统流程

当玩家开始游戏时，屏幕上将显示 3 只鞋子，单击其中的任意一只鞋子，程序判断该鞋子中是否有鸡蛋，并且打开鞋子显示结果，此时可以通过单击“再玩一次”按钮重新开始游戏。具体的系统流程如图 A-2 所示。

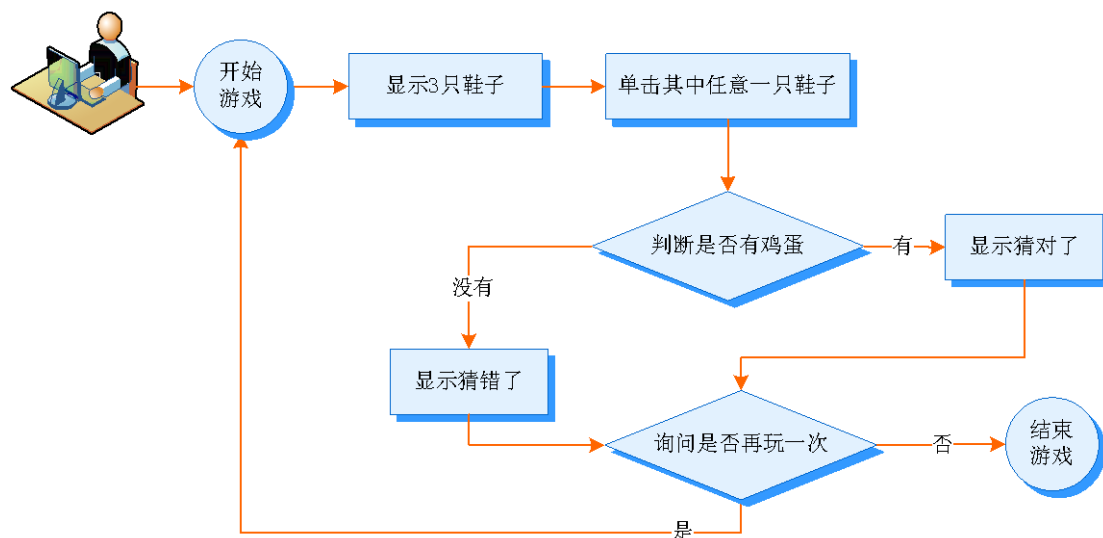


图 A-2 猜猜鸡蛋放在哪只鞋子里小游戏的系统流程图

A.1.3 主界面预览

为了使读者对本模块有一个基本的了解，下面给出“猜猜鸡蛋放在哪只鞋子里”游戏的主界面的预览效果，如图 A-3 所示。



图 A-3 游戏主界面

A.2 关键技术

在实现本实例时，最关键的技术就是如何随机的让 3 只鞋子中的一只里带有鸡蛋。这里我们通过一个 for 循环和 Math 类的 random()方法来实现，具体的代码如下：

```

for (int i = 0; i < 3; i++) {
    int temp = imageIds[i];           //将数组元素 i 保存到临时变量中
}
  
```

```
int index = (int) (Math.random() * 2); //生成一个随机数
imageIds[i] = imageIds[index]; //将随机数指定的数组元素的内容赋值给数组元素 i
imageIds[index] = temp; //将临时变量的值赋值给随机数组指定的那个数组元素
}
```

例如，在 for 循环中，第一次生成的随机数为 1，第二次生成的随机数为 0，第 3 次生成的随机数为 0，最后鸡蛋所在鞋子的变化过程如表 A-1 所示。其中，数组元素的值为 R.drawable.shoe_ok 时代表有鸡蛋。

表 A-1 随机指定鸡蛋所在鞋子的变化过程

i	临时变量 temp	index	imageIds[0]的值	imageIds[1]的值	imageIds[2]的值
0	temp=imageIds[0]	1	R.drawable.shoe_sorry	R.drawable.shoe_ok	R.drawable.shoe_sorry
1	temp=imageIds[1]	0	R.drawable.shoe_ok	R.drawable.shoe_sorry	R.drawable.shoe_sorry
2	temp=imageIds[2]	0	R.drawable.shoe_sorry	R.drawable.shoe_sorry	R.drawable.shoe_ok

A.3 实现过程

在实现猜猜鸡蛋放在哪只鞋子里游戏时，大致需要分为搭建开发环境、准备资源、布局页面和实现游戏规则代码等 4 个部分，下面进行详细介绍。

A.3.1 搭建开发环境

在开发本实例时，至少需要下载 Android SDK 5.0、Eclipse 4.4.2+ADT 插件。另外，在创建模拟器时，最好按照图 A-4 所示的参数进行配置。

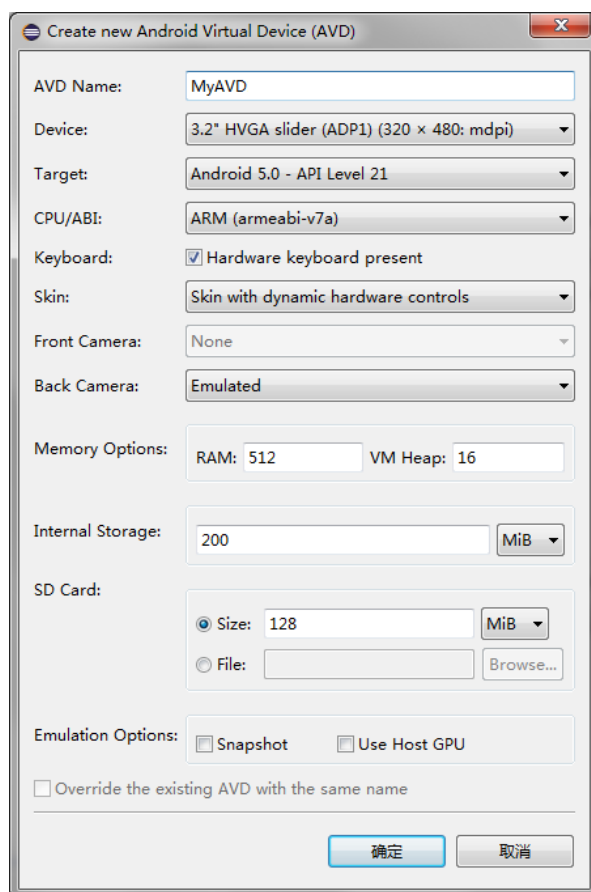


图 A-4 配置模拟器参数

A.3.2 准备资源

在实现本实例前，首先需要准备游戏中所需的图片资源，这里共包括游戏背景图片、图标、默认显示的鞋子、有鸡蛋的鞋子和没有鸡蛋的鞋子 5 张图片，如图 A-5 所示，并把它们放置在项目根目录下的 res/drawable-mdpi/文件夹中，放置后的效果如图 A-6 所示。



图 A-5 准备的 5 张图片

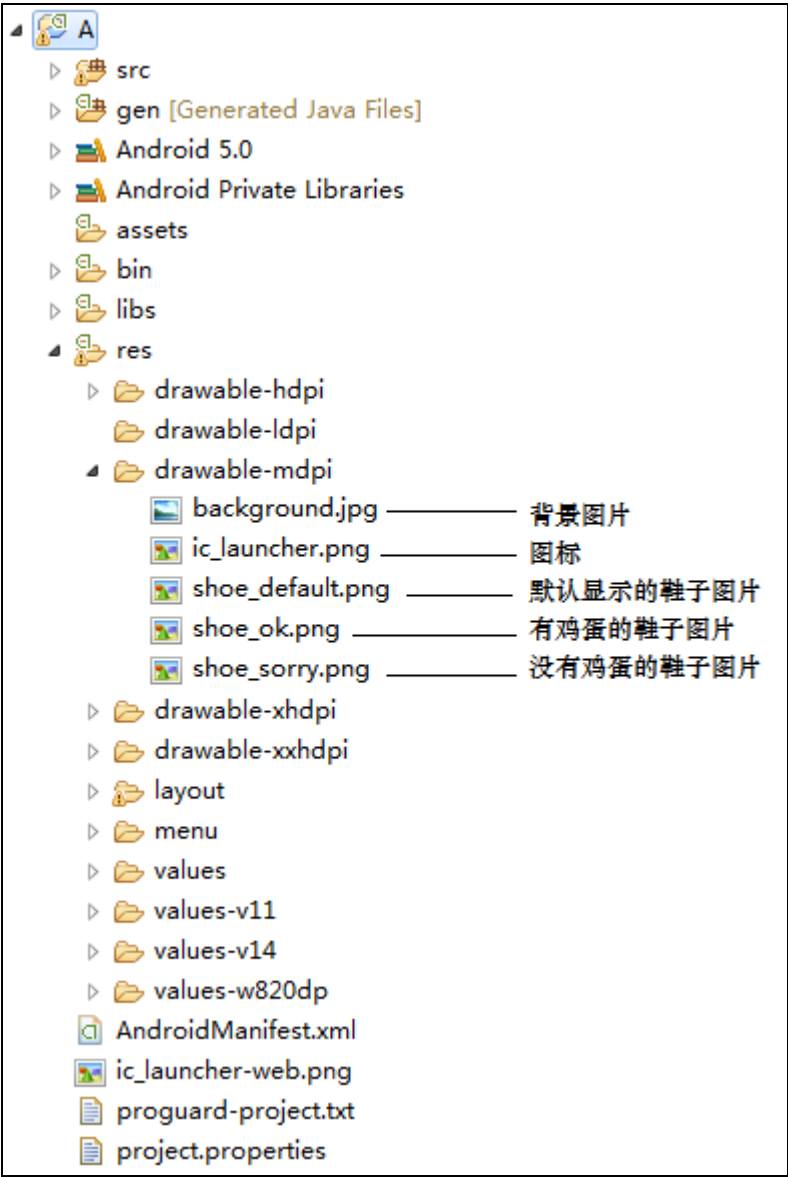


图 A-6 放置后的图片资源

将图片资源放置到 drawable-hdpi、drawable-ldpi 和 drawable-mdpi 文件夹后，系统将自动在 gen 目录下的 com.mingrisoft 包中的 R.java 文件中添加对应的图片 id。打开 R.java 文件，可以看到下面的图片 id：

```
public static final int background=0x7f020000;
public static final int ic_launcher=0x7f020001;
public static final int shoe_default=0x7f020002;
public static final int shoe_ok=0x7f020003;
public static final int shoe_sorry=0x7f020004;
```



R.java 是系统自动派生的，最好不要进行修改。

A.3.3 布局页面

在实现猜猜鸡蛋放在哪只鞋子里小游戏时，只涉及到一个窗体布局页面，这里我们创建的是 main.xml 文件，在该文件中添加一个 3 行的表格，具体代码如下：

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/background"
    android:id="@+id/tableLayout1">
<!-- 第一行 -->
    <TableRow
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:gravity="center"
        android:layout_weight="2"
        android:id="@+id/tableRow1">
</TableRow>
<!-- 第二行 -->
    <TableRow
        android:id="@+id/tableRow2"
        android:layout_weight="1"
        android:gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
</TableRow>
<!-- 第三行 -->
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center_horizontal"
        >
        </LinearLayout>
</TableLayout>
```

在表格的第一行中，添加一个文本框组件，用于显示提示性文字，默认为字符串资源 title 指定的内容，具体代码如下：

```
<TextView
    android:text="@string/title"
    android:padding="10dp"
    android:gravity="center"
    android:textSize="20dp"
    android:textColor="#010D18"
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

在表格的第二行中，添加一个水平线性布局管理器，并在该水平线性布局管理器添加 3 个

ImageView 组件，用于显示代表 3 只鞋子的 3 张图片，具体代码如下：

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    >
    <ImageView android:id="@+id/imageView1"
        android:src="@drawable/shoe_default"
        android:paddingLeft="5dp"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"/>
    <ImageView
        android:id="@+id/imageView2"
        android:src="@drawable/shoe_default"
        android:paddingLeft="5dp"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"/>
    <ImageView
        android:id="@+id/imageView3"
        android:src="@drawable/shoe_default"
        android:paddingLeft="5dp"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"/>
</LinearLayout>
```

在表格的第三行中，添加一个用于实现再玩一次按钮的 Button 组件，并设置它的 android:id 属性值为"@+id/button1"，具体代码如下：

```
<Button
    android:text="再玩一次"
    android:textColor="#000"
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

A.3.4 实现游戏规则代码

实现游戏规则的代码全部编写在主活动 MainActivity 中，具体的实现步骤如下：

(1) 在主活动 MainActivity 中，定义一个保存全部图片 ID 的数组、3 个 ImageView 类型的对象和一个 TextView 类型的对象，具体代码如下：

```
int[] imageIds = new int[] { R.drawable.shoe_ok, R.drawable.shoe_sorry,
                             R.drawable.shoe_sorry }; // 定义一个保存全部图片 ID 的数组

private ImageView image1; //ImageView 组件 1
private ImageView image2; //ImageView 组件 2
private ImageView image3; //ImageView 组件 3
private TextView result; //显示结果
```

(2) 编写一个无返回值的方法 reset()，用于随机指定鸡蛋所在的鞋子，关键代码如下：

```
private void reset() {
    for (int i = 0; i < 3; i++) {
        int temp = imageIds[i]; //将数组元素 i 保存到临时变量中
        int index = (int) (Math.random() * 2); //生成 2 以内的一个随机整数
```

```

        imageIds[i] = imageIds[index]; //将随机数指定的数组元素的内容赋值给数组元素 i
        imageIds[index] = temp;        //将临时变量的值赋值给随机数组指定的那个数组元素
    }
}

```

(3) 由于 `ImageButton` 组件设置背景透明后, 将不再显示鼠标单击效果, 所以我们需要通过 `Drawable` 资源来设置图片的 `android:src` 属性。首先编写一个 `Drawable` 资源对应的 XML 文件 `button_state.xml`, 用于设置当鼠标按下时显示的图片, 以及鼠标没有按下时显示的图片, 具体代码如下:

```

image1 = (ImageView) findViewById(R.id.imageView1); //获取 ImageView1 组件
image2 = (ImageView) findViewById(R.id.imageView2); //获取 ImageView2 组件
image3 = (ImageView) findViewById(R.id.imageView3); //获取 ImageView3 组件
result = (TextView) findViewById(R.id.textView1);    //获取 TextView 组件
reset();                                           //将鞋子的顺序打乱

```

(4) 为 3 个显示鞋子的 `ImageView` 组件添加单击事件监听器, 用于将鞋子打开, 并显示猜猜看的结果, 关键代码如下:

```

// 为第一只鞋子添加单击事件监听
image1.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        isRight(v, 0);        // 判断结果
    }
});
// 为第二只鞋子添加单击事件监听
image2.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        isRight(v, 1);        // 判断结果
    }
});
// 为第三只鞋子添加单击事件监听
image3.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        isRight(v, 2);        // 判断结果
    }
});

```

(5) 编写 `isRight()` 方法, 用于显示打开的鞋子, 并显示判断结果, 具体代码如下:

```

/**
 * 判断猜出的结果
 *
 * @param v
 * @param index
 */
private void isRight(View v, int index) {
    // 使用随机数组中图片资源 ID 设置每个 ImageView
    image1.setImageDrawable(getResources().getDrawable(imageIds[0]));
    image2.setImageDrawable(getResources().getDrawable(imageIds[1]));
    image3.setImageDrawable(getResources().getDrawable(imageIds[2]));
}

```



```

// 为每个 ImageView 设置半透明效果
image1.setAlpha(100);
image2.setAlpha(100);
image3.setAlpha(100);
ImageView v1 = (ImageView) v; // 获取被单击的图像视图
v1.setAlpha(255); // 设置图像视图的透明度
if (imageIds[index] == R.drawable.shoe_ok) { // 判断是否猜对
    result.setText("恭喜您, 猜对了, 祝你幸福!");
} else {
    result.setText("很抱歉, 猜错了, 要不要再试一次? ");
}
}

```

(6) 获取“再玩一次”按钮, 并为该按钮添加单击事件监听器, 在其单击事件中, 首先将标题恢复为默认值、然后设置 3 个 `ImageView` 的透明度为完全不透明, 最后再设置这 3 个 `ImageView` 的图像内容为默认显示图片, 具体代码如下:

```

Button button = (Button) findViewById(R.id.button1); // 获取“再玩一次”按钮
// 为“再玩一次”按钮添加事件监听器
button.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        reset();
        result.setText(R.string.title); // 将标题恢复为默认值
        image1.setAlpha(255);
        image2.setAlpha(255);
        image3.setAlpha(255);
        image1.setImageDrawable(getResources().getDrawable(
R.drawable.shoe_default));

        image2.setImageDrawable(getResources().getDrawable(R.drawable.shoe_default));

        image3.setImageDrawable(getResources().getDrawable(R.drawable.shoe_default));
    }
});

```

A.4 运行项目

项目开发完成后, 就可以在模拟器中运行该项目了。此时, 如果您没有创建模拟器, 那么需要先创建并启动模拟器, 然后再按照以下步骤运行项目。

(1) 在“项目资源管理器”中选择项目名称节点, 并在该节点上单击鼠标右键, 在弹出的快捷菜单中选择“运行方式”/“Android Application”菜单项, 即可在创建的 AVD 模拟器中运行 Android 程序。

(2) 程序成功在模拟器中运行后, 将显示如图 A-7 所示的游戏主界面。单击其中的任意一只鞋子, 将打开鞋子显示里面是否有鸡蛋, 并且将没有被单击的鞋子设置为半透明显示, 被单击的正常显示, 同时根据单击的鞋子里面是否有鸡蛋显示对应的结果。例如, 单击中间的那只鞋子, 如果鸡蛋在这只鞋子里, 将显示如图 A-8 所示的运行结果, 否则, 将显示如图 A-9 所示的效果。单击“再玩一次”按钮, 重新开始游戏。



图 A-7 游戏主界面



图 A-8 猜对了时的效果



图 A-9 猜错了时的效果

A.5 小结

本章通过一个猜猜鸡蛋放在哪只鞋子里小游戏，向读者介绍了 Android 开发小游戏的基本流程，以及页面布局 and Android 基本组件 Button 和 ImageView 的具体应用。通过本章的学习，读者应该掌握 Android 页面布局以及基本组件 Button 和 ImageView 的具体应用，以及实现随机指定鸡蛋所在鞋子的方法。