



华南理工大学
South China University of Technology

课程设计报告书

我家猫咪大养成

学 院：计算机科学与工程学院
专 业：计算机科学与技术专业
组长姓名学号：叶劲亨 201830582180

成员姓名学号：

陈澍豪 201830580070

李林文 201830580476

林滨 201830581107

冯肇鹏 201830580179

何麟 201830580247

林升升 201830581138

指导教师：张晶

起始日期：2019/12/12

我家猫咪大养成

一、选题背景

这是一款以猫咪养成+探险为主要玩法的 app，您可以与您的爱猫进行玩耍，同时也需要花时间去给 ta 喂食、治疗，并且还可以带 ta 出去探险哦！赶快开始您的猫咪养成之旅吧！

二、方案论证(设计理念)

我们大作业设计的是一款以小猫咪的养成和小猫咪的探险为主要玩法的游戏。我们实现的主要功能有：

（1）在主界面小猫咪会不停的讲话，还有不停摇尾巴的动画；给小猫咪投食猫粮和小鱼时小猫咪的饱食度上升，还会有进食的动画。随着时间的增加小猫咪的饱食度会降低，当小猫咪的饱食度降低到一定程度时，小猫咪的血量会降低；点击小猫咪时小猫咪的兴奋度上升随着时间的增加小猫咪的兴奋度也会降低；与小猫咪互动时小猫咪的亲密度会上升，长时间不与小猫咪互动则小猫咪的亲密度会下降；当点击医疗箱时，小猫咪的血量会上升；拖动小猫咪到屏幕顶端能让其做自由落体，还会有相应的动画。主页面还能为自己的小猫咪取好听的名字，点击问号还有关于如何玩这款 app 的简介。

（2）而对于探险功能只有当饱食度和血量高于一定程度时才能进入探险功能页面，当饱食度或者血量不足时将会提示饱食度或者健康值过低，无法进行探险。在探险页面，首先需要选择带什么物品去探险，选好后出发，在探险过程中小猫咪会遇到不同的随机事件，遇到不同事件会有不同的选择，对于不同的选择会有相应的奖励和惩罚。

三、过程论述

（一）美工设计

对于宠物养成游戏来说，一个吸引人的宠物形象尤为关键。我们综合考虑了猫，狗，等常规类型的宠物，以及例如鲲，史莱姆这样的猎奇宠物之后，选

择了近年来关注度和受欢迎度较高的猫咪作为宠物形象。

确定了选择猫作为形象之后，我们更需要确认宠物猫的画风。

我们采用了像素风作为我们宠物的绘画风格。一来，像素风的绘画风格比较符合我们的审美，再者，像素风的绘画比较适合我们这样绘画基础不高的程序员，其次，也致敬了 15 年前在中国流行的“数码宝贝”系列电子宠物。

在猫的形象方面，我们参考了许多与猫有关的动画，例如“罗小黑战记”等。采用 piskel 软件来进行猫咪的动画制作，每一笔每一画皆为手工画出，每个动作皆为绘制出的一系列图片合成而来。考虑到我们的猫咪形象要生动可爱，活灵活现，我们设计绘画了站立，被抓起，进食，掉落等猫咪动画。

除了猫咪的动画以外，我们还绘制了图标，并对背景进行了修改和优化，使得猫咪和背景看起来更加协调。

（二）动画实现

基于整个游戏对猫猫的互动需求以及减轻工作负担，我们决定选用一个开源的外部件 GifImageView 作为猫猫控件的基类。

采用这个控件的最大好处是：美工的资源可以直接做成 GIF 格式，而不是以逐帧播放 JPG 的动画形式实现猫猫的动作。

此外为了实现与用户的互动，我们使用了为猫猫添加了 onTouchListener 以获取用户对猫猫的点击和拖动等操作。

对应用户不同的动作我们设计了不同的动画，如：用户拖动猫猫时，猫猫会更换至挣扎的状态并跟随用户的手指移动，模拟出被抓住的感觉。

如果用户将猫猫抓取到了一定高度再释放，猫猫会以掉落状下落回到原位。

而为了能够令猫猫移动时依然能够响应用户的动作，要求猫猫在屏幕上发生真实的移动，而不是只有显示上的移动。

所以我们采用了 ObjectAnimator 实现猫猫在某方向的线性运动，同时通过 AnimatorSet 的封装能够实现猫猫在屏幕上进行曲线运动。

在这个基础上我们让猫猫下落归位时更自然和更有趣味性。

另外当用户进行喂食或治疗等操作时，猫猫也会表现出不同的状态。

（三）实体类设计

该 app 的主要实体类为 Kitten（猫咪）、Player（玩家）、Food（食物）、Medicine（医疗物品）、Toy（玩具）等。

（1）Kitten 类

```
public class Kitten extends LitePalSupport {  
    @Column(defaultValue = "lovely kitten")  
    private String name;  
  
    //饱食度  
    @Column(defaultValue = "100")  
    private double hungry;  
  
    //兴奋值  
    @Column(defaultValue = "100")  
    private double happy;  
  
    //亲密度  
    @Column(defaultValue = "0")  
    private double intimacy;  
  
    //健康值  
    @Column(defaultValue = "100")  
    private double health;  
  
    //出生日期  
    private Date birthDay;  
  
    //时间  
    private Date time;  
}
```

考虑到整个 app 内只可能存在一个 Kitten 对象，所以采用了工厂设计模式，

实现一个 DBUtil 工厂类来获取数据库中的唯一一个 Kitten 对象（如果没有，则创建一个 Kitten 对象后返回），下面的 Player 类同理。

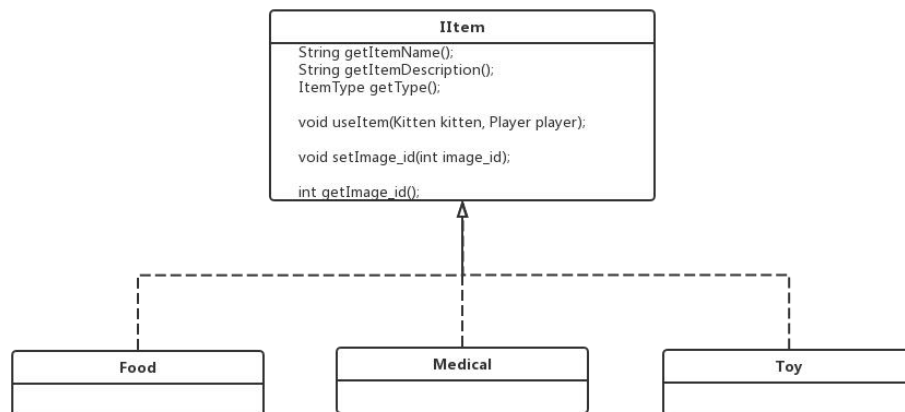
（2）Player 类

```
public class Player extends LitePalSupport {  
  
    //食物数量  
    @Column(defaultValue = "10")  
    private int food;  
  
    //爱心值  
    @Column(defaultValue = "80")  
    private double heart;  
  
    //金钱  
    @Column(defaultValue = "500")  
    private int money;  
  
    //物品栏  
    private List<Object> items;  
}
```

（3）IItem 接口

```
public interface IItem {  
    String getItemName();  
  
    String getItemDescription();  
  
    void useItem();  
  
    ItemType getType();  
  
    void setImage_id(int image_id);  
  
    int getImage_id();  
}
```

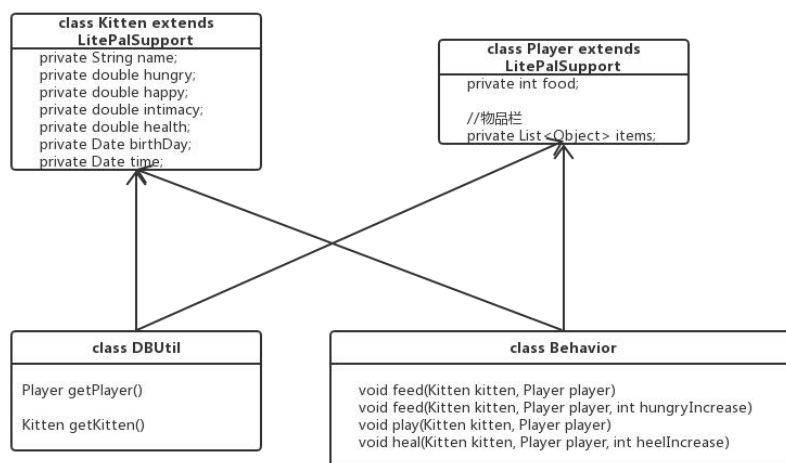
定义 IItem 接口相当于定义了 Food 类、Medicine 类、Toy 类的实现标准。其具体子类这里便不再赘述。



IItem 接口和其具体实现子类

(4) Behavior 类

Behavior 类是一个用来描述 Kitten 和 Player 对象行为的类，通过调用该类的 feed、play、heal 等方法可以修改 Kitten 和 Player 对象的状态。Behavior 类和其他类的关系如下 uml 图所示。



Behavior 类和其他类的关系图

(四) 逻辑实现

(1) 主页面简单按钮监听

以点击右上角的饱食度按钮为例：

点击按钮---》弹出对话框---》再次点击屏幕取消对话框。

(2) Kitten 对象改名 (图片展示在“结果分析”处, 下同)

通过弹出一个用于修改猫咪名字 Dialog 来实现该功能。

设计: 点击修改名字按钮弹出对话框, 提示输入猫咪新名字, 点击确定后判断输入字符是否合法, 合法则修改。

实现: 生成一个对话框, 插入可编辑文本组件, 设置文本的提示, 设置对话框的确定和取消按钮, 点击确定后响应, 利用 `trim()` 方法对输入的文本进行去除无效字符处理, 然后判断字符串长度是否为 0, 为 0 则无效输入, 不修改猫咪, 否则输入有效, 修改猫咪名字, 保存到数据库。

用户交互逻辑:

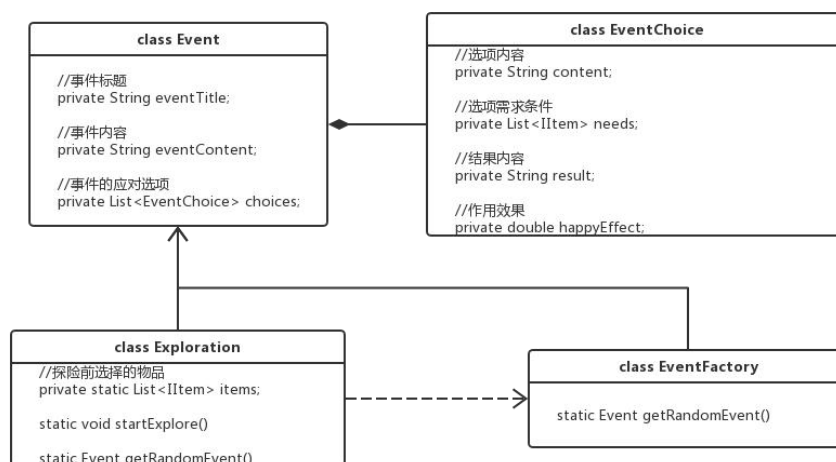
点击改名按钮---》调用 `MyDialog.showSettingDialog()`, 弹出对话框---》输入新名字---》调用 `DBUtil.getKitten()` 得到 Kitten 对象---》调用 Kitten 的 `setName()` 和 `save()` 方法。

(3) 探险模块

该 app 具有一个带猫咪外出探险的功能, 本质是一个文字游戏, 玩家可以在探险之前携带所需的物品进行一场文字游戏大冒险: 探险过程会发生随机事件, 例如猫咪受到袭击或者猫咪得到意外奖励, 玩家可以通过点击不同的选项来触发不同的奖励或惩罚, 当然不同的选项对携带的物品也有不同的要求, 有些选项需要携带特定的物品才能触发。

探险界面由宠物图像、探险随机事件标题、探险随机事件内容和用户选择组成。用户每次进入探险, 将随机进入一个探险情景, 根据用户携带的物品, 用户可以点击选项做出不同选择, 不同选择将会有不同的结果。

为了实现上述功能, 该 app 主要定义了四个类, 如下 uml 图所示。



Event 类：定义随机事件内容，包含事件的（四个）应对选项。

EventChoice 类：定义事件的应对选项。

EventFactory 类：随机事件产生工厂。

Exploration 类：整个探险过程类，包含探险携带的物品等。

总的探险功能可以简要概括为：

在主页面中点击探险按钮---》如果猫咪处于健康状态---》进入探险前的选择物品页面---》选择物品后点击出发探险按钮---》遭遇随机事件---》点击选项，如果不满足当前选项条件，则重新选择---》如果满足当前选项条件，则弹出对话框显示选择结果并更新 Kitten 对象的状态（掉血或者增加开心值等）---》点击关闭或者确定按钮回到主页面。

（4）前台服务

服务几乎都是在后台运行的，一直以来它都是默默地做着辛苦的工作。但是服务的系统优先级还是比较低的，当系统出现内存不足的情况时，就有可能回收掉正在后台运行的服务。如果希望服务可以一直保持运行状态，而不会由于系统内存不足的原因导致被回收，就可以考虑使用前台服务，而我们项目小猫咪的养成如果在后台运行时被回收了的话小猫咪的饱食度、兴奋度和亲密度就不会随时间降低了。所以我们选择使用了前台服务来完成这个目的。前台服务和普通服务最大的区别就在于，它会一直有一个正在运行的图标在系统的

状态栏显示，下拉状态栏后可以看到更加详细的信息，非常类似于通知的效果。为了实现前台服务，我们使用了 Notification 对象来创建服务，然后调用 startForeground() 方法将前台服务在系统状态栏显示出来。

(5) 计时器实现数值降低

使用 Timer() 方法实现计时功能以及 Handler 来改变饱食度、兴奋度和亲密度数值 UI，从而实现饱食度兴奋度等随时间降低。

(6) 玩家与猫咪的互动

1、“纽扣”背包的实现

如何实现背包界面的美化是困扰我们的一个问题，经过深思熟虑我们决定使用开源库。我们引用开源库‘com.nightonke:boommenu:2.1.1’的布局，设计了酷炫的背包界面及背包打开动画。在主页面的左下角，有一个“纽扣”按钮，用户点击该按钮后，将会出现爆炸效果的动画弹出效果，并且背包物品也进行了相关美化。

2、猫咪进食

设计：饱食度较低时直接回复到 40，否则每次增加 60%，上限 100，这样最多只需 3 次即可回复到 100。

实现：判断当前饱食度，然后根据设定更新饱食度，保存到数据库。

```
public void feed(Kitten kitten, Player player) {
    if (player.getFood() != 0 && kitten.getHungry() != 100) {
        player.setFood(player.getFood() - 1);
        if (kitten.getHungry() < 40) {
            kitten.setHungry(max(40.0, 1.6 * kitten.getHungry()));
        } else {
            kitten.setHungry(min(100.0, 1.6 * kitten.getHungry()));
        }
        kitten.save();
        player.save();
    }
}
```

3、猫咪玩耍：

设计：提升兴奋值和亲密度和健康值，兴奋值上限为 100，每次增加 10 点，亲密度每次增加 2 点，健康值每次增加 1 点。

实现：更新兴奋值、亲密度、健康值，保存到数据库。

```
public void play(Kitten kitten, Player player) {
    kitten.setHappy(min(100, kitten.getHappy() +
        mContext.getResources().getInteger(R.integer.play_happy_increase)));
    kitten.setIntimacy(kitten.getIntimacy() +
        mContext.getResources().getInteger(R.integer.play_intimacy_increase));
    player.setHeart(player.getHeart() + 1);
    kitten.save();
    player.save();
}
```

4. 猫咪治疗：

设计：因为事件和饱食度过低会降低健康值，所以可以使用背包里的医疗箱来回复健康值。

实现：更新健康值，每次增加 40，上限 100，保存到数据库。

```
public void heal(Kitten kitten, Player player, int healIncrease) {
    kitten.setHealth(min(100, kitten.getHealth() + healIncrease));
    kitten.save();
    player.save();
}
```

（7）对话框

用户点击饱食度、兴奋度时会弹出对话框提示对应的详细介绍，我们使用了 Android 原生组件 AlertDialog，但是由于这个组件过于简陋不美观，我们修改了 Style 中的 DialogTheme，以改变 AlertDialog 的样式，达到美观的效果。

（8）猫咪讲话气泡：

设计：猫咪会不定时的讲话，我们需要一个组件来显示文字内容，微信的

聊天气泡就很简单美观，所以我们也采用聊天气泡来显示文字。

实现：因为气泡是会拉伸的，自己制作图片的话很可能效果不好，所以去网上查找了实现方法，正好在 github 上找到了一个不错的实现 (<https://github.com/wangyiwy/BubbleLayout>)，这是一个自定义 Layout，背景是气泡框，内部可以放 TextView 等组件，测试发现随文字长度变化背景拉伸正常，可以使用。使用计时器和修改组件的 visibility 属性实现气泡弹出一会儿后消失，文本修改直接改 TextView 的 text 属性即可。

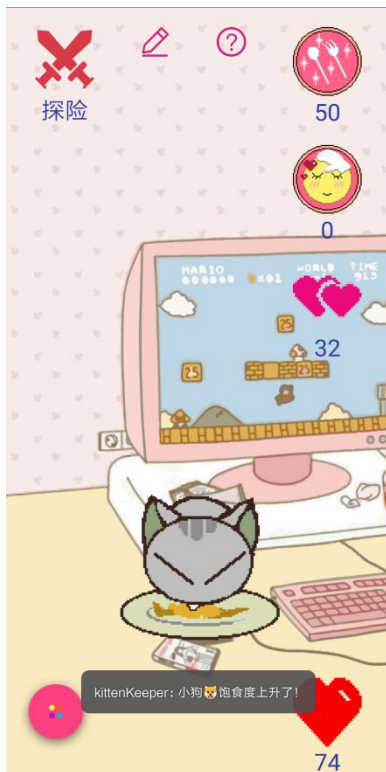
四、结果分析



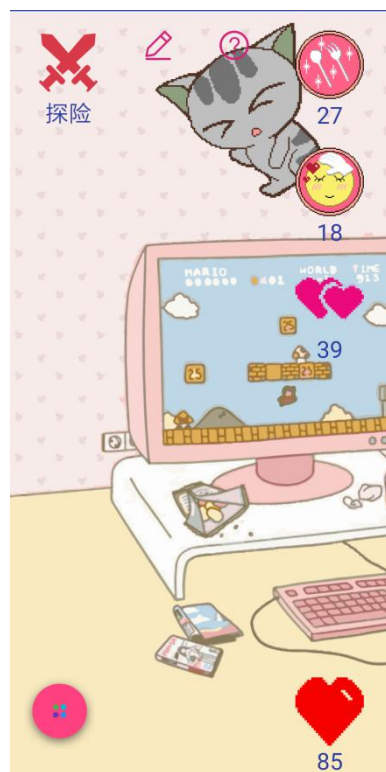
进入主页面



点击上方的提示按钮弹出游戏提示



喂食动画



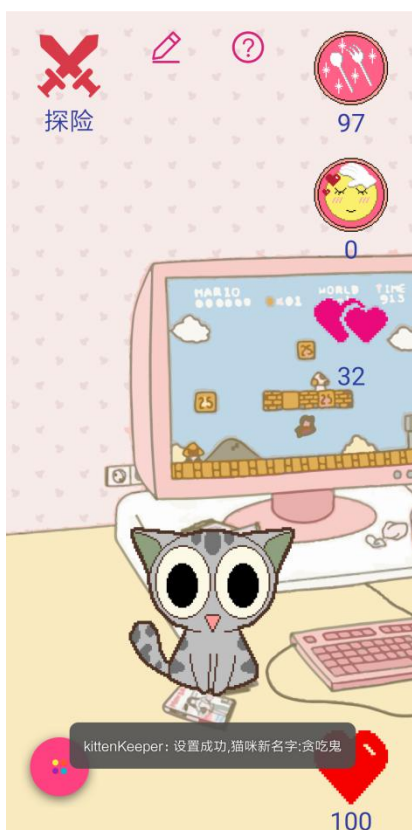
点击并拖动猫咪播放拉动动画



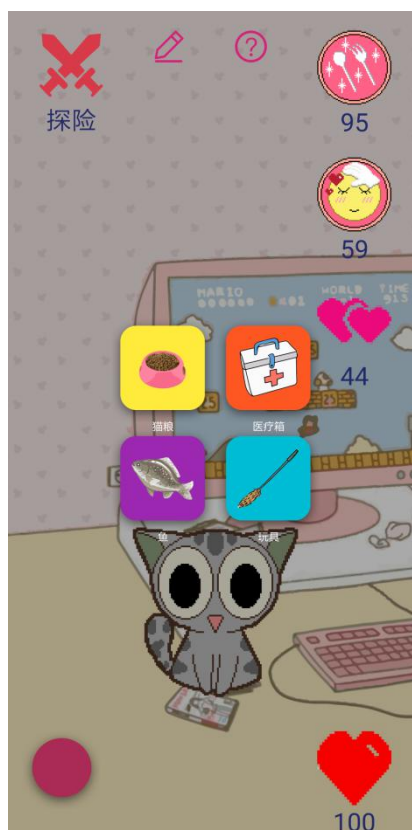
松开手指播放掉落动画



点击屏幕上方的改名按钮



输入新名字改名成功



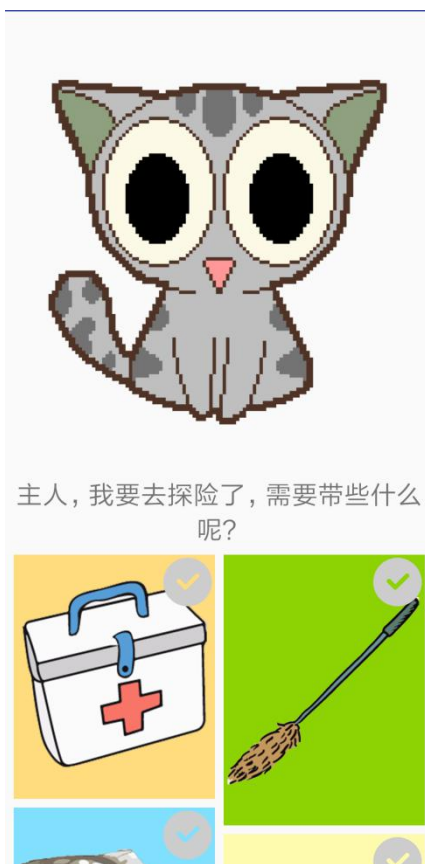
点击右下方的纽扣按钮弹出背包界面



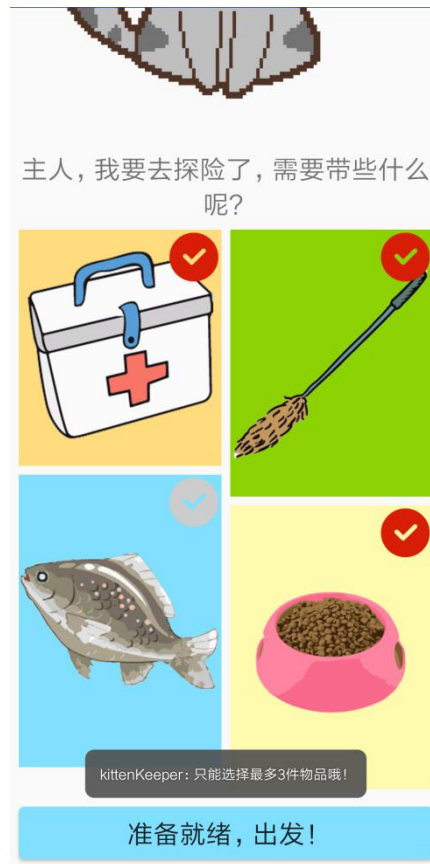
点击医疗包播放治疗动画



说话气泡展示



探险前选择物品



点击右上角的按钮选择物品



点击物品进入介绍页面



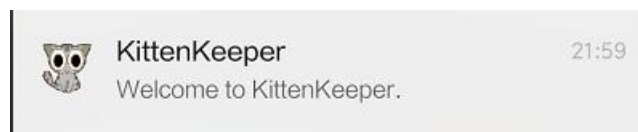
点击出发按钮正式探险



特定选项有特定的物品要求



点击可选的按钮弹出结果对话框



前台服务, 点击后进入主页面

五、项目成员工作内容分配

叶劲亨: 负责主要的逻辑实现, 包括探险功能、数据库调用、主要业务逻辑、修复 bug、报告汇总等。

陈澍豪: 负责美工设计, 定制背景图和动画等。

林滨: 负责美工设计, 定制背景图和动画等。

冯肇鹏: 负责动画实现。

何麟: 负责逻辑实现和页面布局, 包含说话气泡、改名 dialog 逻辑等。

李林文: 负责逻辑实现, 包含前台服务和定时器实现等。

林升升: 负责逻辑实现和页面布局, 包含界面设计、纽扣按钮、对话框显示等。