**仿真建模与MATLAB**

**期末大作业**

1201003023 刘家彤

**目录**

[**一．** **整体框架** 3](#_Toc124268415)

[**二．** **主界面** 4](#_Toc124268416)

[**1)** **功能分析** 4](#_Toc124268417)

[**2)** **实现方法** 4](#_Toc124268418)

[**3)** **运行效果** 4](#_Toc124268419)

[**4)** **设计亮点** 5](#_Toc124268420)

[**三．** **“PhotoStore” 图像处理演示子系统** 6](#_Toc124268421)

[**1)** **功能分析** 6](#_Toc124268422)

[**2)** **图像读取与保存** 6](#_Toc124268423)

[**i.** **实现方法** 6](#_Toc124268424)

[**ii.** **运行效果** 6](#_Toc124268425)

[**3)** **功能1：灰度化** 9](#_Toc124268426)

[**i.** **原理** 9](#_Toc124268427)

[**ii.** **实现方法** 9](#_Toc124268428)

[**iii.** **运行效果** 9](#_Toc124268429)

[**4)** **功能2：RGB分量显示** 11](#_Toc124268430)

[**i.** **原理** 11](#_Toc124268431)

[**ii.** **实现方法** 11](#_Toc124268432)

[**iii.** **运行效果** 11](#_Toc124268433)

[**iv.** **设计亮点** 12](#_Toc124268434)

[**5)** **功能3：二值化** 12](#_Toc124268435)

[**i.** **原理** 12](#_Toc124268436)

[**ii.** **实现方法** 12](#_Toc124268437)

[**iii.** **运行效果** 13](#_Toc124268438)

[**iv.** **设计亮点** 13](#_Toc124268439)

[**6)** **功能4：反色** 13](#_Toc124268440)

[**i.** **原理** 13](#_Toc124268441)

[**ii.** **实现方法** 13](#_Toc124268442)

[**iii.** **运行效果** 14](#_Toc124268443)

[**7)** **功能5：镜像** 14](#_Toc124268444)

[**i.** **实现方法** 14](#_Toc124268445)

[**ii.** **运行效果** 14](#_Toc124268446)

[**8)** **功能6：倒影** 15](#_Toc124268447)

[**i.** **原理** 15](#_Toc124268448)

[**ii.** **实现方法** 15](#_Toc124268449)

[**iii.** **运行效果** 16](#_Toc124268450)

[**9)** **功能7：浮雕** 16](#_Toc124268451)

[**i.** **原理** 16](#_Toc124268452)

[**ii.** **实现方法** 16](#_Toc124268453)

[**iii.** **运行效果** 16](#_Toc124268454)

[**10)** **功能8：证件照** 17](#_Toc124268455)

[**i.** **原理** 17](#_Toc124268456)

[**ii.** **实现方法** 17](#_Toc124268457)

[**iii.** **运行效果** 17](#_Toc124268458)

[**11)** **退出** 18](#_Toc124268459)

[**i.** **实现方法** 18](#_Toc124268460)

[**ii.** **运行效果** 18](#_Toc124268461)

[**12)** **其他** 19](#_Toc124268462)

[**四．** **“蹦极模拟器” 蹦极演示子系统** 20](#_Toc124268463)

[**1)** **原理** 20](#_Toc124268464)

[**2)** **功能分析** 20](#_Toc124268465)

[**3)** **实现方法** 21](#_Toc124268466)

[**i.** **Simulink与App联动** 21](#_Toc124268467)

[**ii.** **Simulink设计** 21](#_Toc124268468)

[**iii.** **App设计** 21](#_Toc124268469)

[**4)** **运行效果** 23](#_Toc124268470)

[**5)** **设计亮点** 24](#_Toc124268471)

[**五．** **“扫雷”游戏子系统** 26](#_Toc124268472)

[**1)** **功能分析** 26](#_Toc124268473)

[**2)** **实现方法** 26](#_Toc124268474)

[**i.** **系统流程图** 27](#_Toc124268475)

[**ii.** **全局变量** 27](#_Toc124268476)

[**iii.** **bomb函数** 29](#_Toc124268477)

[**iv.** **gaming函数** 29](#_Toc124268478)

[**v.** **burybomb函数** 30](#_Toc124268479)

[**vi.** **CountBomb函数** 30](#_Toc124268480)

[**vii.** **leftclick函数** 30](#_Toc124268481)

[**viii.** **rightclick函数** 31](#_Toc124268482)

[**3)** **运行效果** 31](#_Toc124268483)

[**4)** **设计亮点** 33](#_Toc124268484)

[**六．** **总结与心得** 34](#_Toc124268485)

1. **整体框架**

本次大作业分为三部分组成，分别是：“PhotoStore” 图像处理演示子系统、“蹦极模拟器” 蹦极演示子系统、“扫雷”游戏子系统；三个子系统由主页面汇集在一起进行调用。

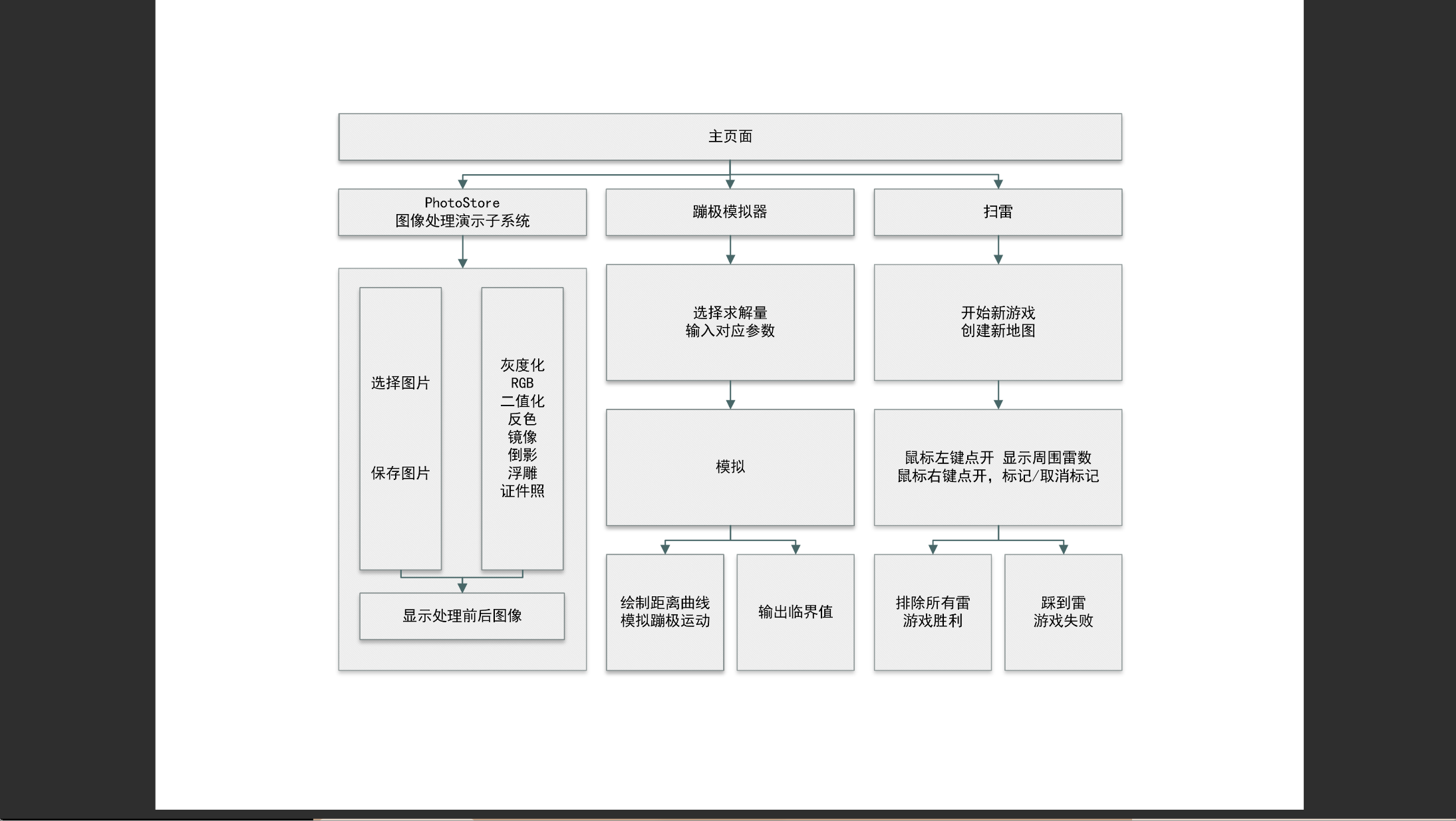


图 1 大作业系统框架图

1. **主界面**
   1. **功能分析**

主页面的功能需要实现：展示个人信息以及子系统的调用；同时作为呈现给用户的第一印象，需要用鲜明的设计吸引用户，图像化的标识指引用户。

* 1. **实现方法**

主页面由App Desinger设计，为了实现美观的需求，在UIFigure中插入提前设计好的照片作为背景，在此基础上插入三个按钮用于调用子系统，按钮采用图标的形式，直观的向用户展示对应的系统；按钮的回调函数由两条语句组成，**addpath**将对应的子系统所在文件夹路径添加进来，保证子系统的正常运行，**run**负责运行对应的子系统。

* 1. **运行效果**

****

图 2 主页面运行效果

进入系统，用户可以点击右下角三个图标之一进行子系统的选择调用

* 1. **设计亮点**
     1. 对于主界面进行了**图像化**的设计，提升交互性

1. **“PhotoStore” 图像处理演示子系统**
   1. **功能分析**

“PhotoStore” 图像处理演示子系统主要包含读取保存文件以及图像处理两部分的功能，进入系统提示用户进行待处理图像的上传，并选择对图像的处理方式和程度，完成图像的处理后可以选择保存当前处理后的图像到任意路径，并自主进行赋名。

* 1. **图像读取与保存**
     1. **实现方法**

图像的读取和保存由uigetfile、uiputfile和imread、imwrite函数实现；**uigetfile**和**uiputfile**函数用于记录用户选择的文件名称与路径，进行拼接后传入**imread、imwrite**，读入的原始图片记为**app.originalpicture**，后续进行处理。

在完成图像的读取后，使用**imread**函数，将原始图像展示在app.UIAxes。

图像的读取和保存由两种方式实现，第一种是页面上的‘’导入图片”与“保存照片”**按钮**，另一种是页面上方的**菜单栏**，点击文件选项进行功能的选择。

在上传和保存过程中，如果出现错误，系统进行**弹窗提示**。

* + 1. **运行效果**

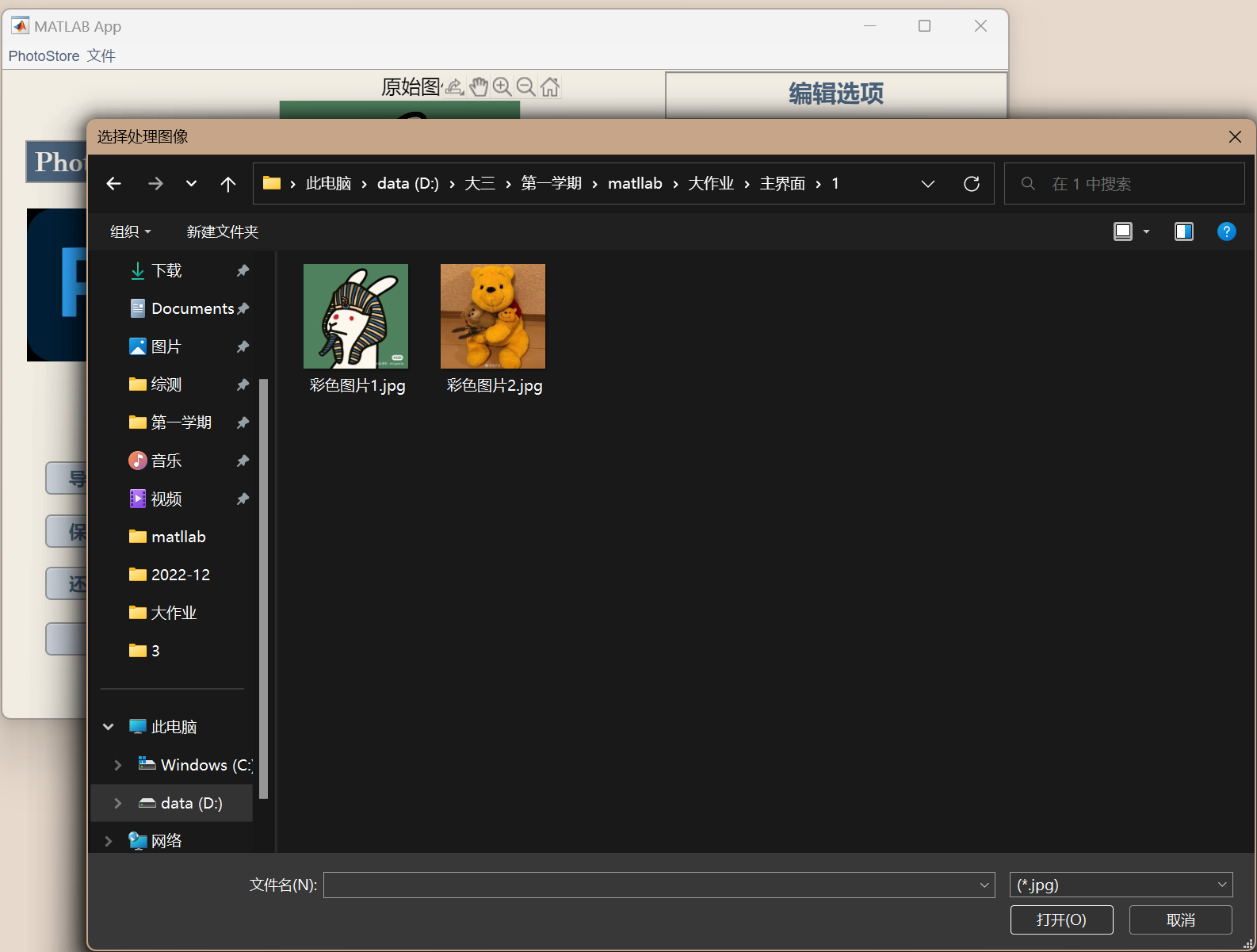


图 3 上传界面



图 4 上传失败提示

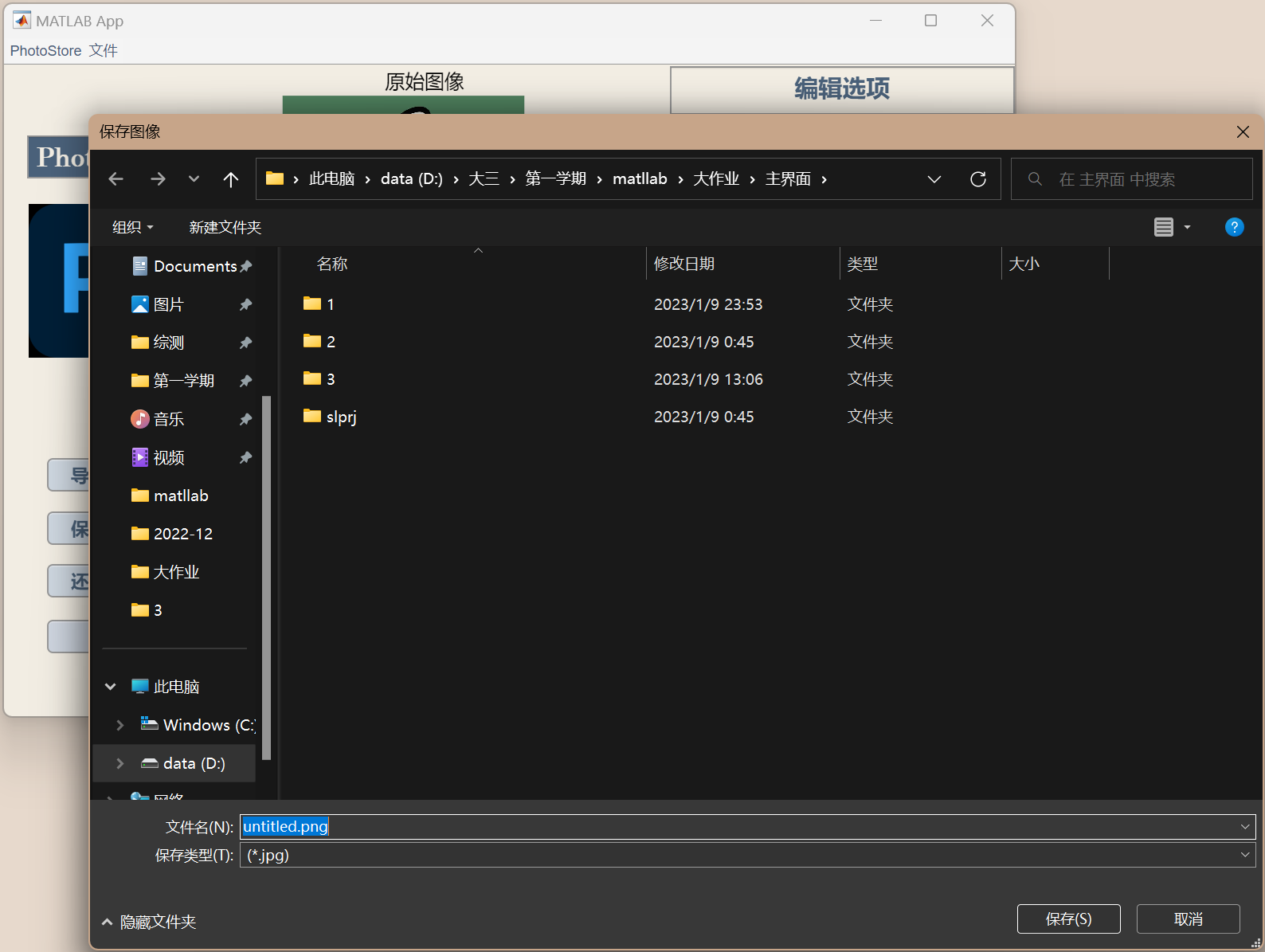


图 5 保存界面



图 6 保存失败界面

* 1. **功能1：灰度化**
     1. **原理**

灰度化处理就是将一幅色彩图像转化为灰度图像的过程。彩色图像分为R，G，B三个分量，分别显示出红绿蓝等各种颜色，灰度化就是使彩色的R，G，B分量相等。

* + 1. **实现方法**

在本系统中，用户可以选择三种灰度化的方式：

* **最大值法**：使R,G,B的值等于3值中最大的一个，**R=G=B=max（R,G,B）**，最大值法形成亮度很高的灰度图像。
* **平均值法**：是R,G,B的值求出平均值，**R=G=B=(R+G+B)/3**平均值法形成较柔和的灰度图像，本例中处理后人眼看不是很清晰。
* **加权平均值法**：根据重要性或其他指标给R,G,B赋予不同的权值，并使R,G,B的值加权平均，**R=G=B=WR+VG+UB**，实验和理论证明当W=0.30,V=0.59,U=0.11时，能得到最合理的灰度图像。

除了固定的最大值法与平均值法，用户可以自己输入加权平均值法的权重，系统默人权重为W=0.30,V=0.59,U=0.11。

灰度化功能由mygray函数实现：**y = mygaray(~,x,k,rp,gp,bp)**，用户完成选择和输入参数后调用函数，输出处理过的图像并绘制在app.UIAxes\_2.

进行灰度化对用户输入的原始图像要求为非灰度图，在“进行灰度”按钮出设置对于原始图像的判断，所原始图像为灰度图，系统将进行**弹窗提醒**，并**禁用**此功能。

* + 1. **运行效果**

****

图 7 用户选择彩色照片进行最大值灰度化

图 8 用户选择灰度照片进行最大值灰度化

* 1. **功能2：RGB分量显示**
     1. **原理**

RGB图像由格式为M×N×３的三维数组组成，其中的“3”可以理解为三幅M×N的二维图像（灰度值图像）。这三幅图像分别代表R、G、B分量，每个分量的像素点取值范围是[0,255]。单独输出对应的二维数组即可分离RGB分量

* + 1. **实现方法**

用户点击右侧选项中的“RGB”，系统进行**原始图片的判断**，如果用户上传的图片是灰度图，系统将弹窗提示，并锁定该页面。

若原始图片是彩色图，**绘制当前图像的RGB分量图**，并可以通过下方两种方式进行处理，用户勾选**“RGB分量显示”**多选组可以显示对应分量的图片，支持多分量重叠；用户还可以**滑动RGB滑动条**进行RGB分量的加减，调整范围为[-20,20]，在调整过程中，UIAxes\_2图窗实时显示调整后的图像，RGB分量图也实时变化。绘制RGB分量由**rgbshow**函数实现，函数由拆分三分量和绘制坐标图两个功能组成，在执行**slidervaluechanging**回调函数时进行调用。

* + 1. **运行效果**

****

图 10 用户屏蔽G分量并增加B分量效果

* + 1. **设计亮点**
       1. 添加**RGB分量分布图**让用户对于图片色彩分布有清晰的感受
       2. 设计**滑条**，用户可以体会不同色彩分量量值对图片的影响
       3. **实时显示**处理后的图片及其RGB分量分布图
  1. **功能3：二值化**
     1. **原理**

图像二值化（ Image Binarization）就是将图像上的像素点的灰度值设置为0或255，也就是将整个图像呈现出明显的黑白效果的过程。二值图像每个像素只有两种取值：要么纯黑，要么纯白。

* + 1. **实现方法**

Matlab提供了**imbinarize**函数，可以对输入的灰度图进行二值化处理，用户可以选择'global' 或 'adaptive'或自定义阈值的方式进行处理。

用户点击进入二值化模块后，系统**判断当前图像属性**，将彩色图利用**rgb2gray**函数转化为灰度图，获取用户选择的二值化方式与输入的阈值调用**imbinarize**进行二值化处理，将处理后的图像显示在UIAxes\_2图窗。

输入阈值采用滑动条的方式进行输入，调用**SliderValueChanged**函数将阈值实时显示在上方的文本框中。

* + 1. **运行效果**

****

图 11 用户选择输入阈值的方法对彩色图像进行二值化处理

* + 1. **设计亮点**
       1. 提供多种二值化的方法
  1. **功能4：反色**
     1. **原理**

反色，就是255-当前像素值。

* + 1. **实现方法**

反色由**y = Invert(~,x)**函数实现，函数先判断输入图像为彩色图或灰度图，再分别进行反色，传回在UIAxes\_2图窗显示处理后的图片。

* + 1. **运行效果**

****

图 12 用户选择彩色图进行反色

* 1. **功能5：镜像**
     1. **实现方法**

在Matlab中**flipud**和**fliplr**函数可以实现图像的镜像，用户点击“进行镜像”按钮，回调函数获取用户选择的镜像方向，进入对应的判断条件进行对应的镜像。

* + 1. **运行效果**

****

图 13 用户选择上下方向进行镜像

* 1. **功能6：倒影**
     1. **原理**

倒影的处理包括模糊和拼接，模糊常用的方法是在当前点的附近，如以当前点为左上顶点，距离为r的方矩形中任选一个点，用它的值替换当前点的值。完成模糊处理后和原图进行数组的合并。

* + 1. **实现方法**

倒影功能通过**y = reflection(x,r)**函数实现，传入用户输入的模糊距离和待模糊的照片，产生在模糊距离范围内的随机数，作为坐标的偏移，特别的，为了**避免距离过大产生明显的未处理的区域**，在完成横纵坐标的偏移后再进行一次横坐标的偏移。

对于模糊距离，用户可以有三种方式进行输入：**编辑字段、微调器、滑块**；当使用某一种输入方式时，其他组件也显示当前输入的值。

* + 1. **运行效果**

****

图 15 用户对图像进行距离为4的倒影处理

* 1. **功能7：浮雕**
     1. **原理**

浮雕的算法是用当前点的像素值减去旁边像素点或斜上角（斜下角）对应点的像素值，以此生成浮雕图像。

* + 1. **实现方法**

浮雕效果由**y = relief(~,x,k,t)**函数实现，函数传入用户选择的浮雕方法和浮雕距离对图像进行浮雕处理，返回输出在**UIAxes\_2**图窗。

对于浮雕距离，用户可以有三种方式进行输入：**编辑字段、微调器、滑块**；当使用某一种输入方式时，其他组件也显示当前输入的值。

* + 1. **运行效果**

****

图 17 用户选择进行距离为2的彩色浮雕处理

* 1. **功能8：证件照**
     1. **原理**

Imread读入的图像是数组，将多个数组合并在一个数组中再通过imshow就可以实现证件照的排列效果。

* + 1. **实现方法**

按下“进行排列”按钮，系统获取用户输入的行列数，借助**cat**函数将数组循环拼接，最后输出在**UIAxes\_2**图窗。

* + 1. **运行效果**

****

图 18 用户选择进行2\*3的方式进行图像排列

* 1. **退出**
     1. **实现方法**

Matlab提供了**questdlg**函数，用于弹出提问的对话窗，获取用户的选择，当用户选择确定退出，调用**delete（app）**退出当前的app，否则return。

退出由两种方式实现，第一种是页面上的‘’退出””**按钮**，另一种是页面上方的**菜单栏**，点击PhotoStore选项进行功能的选择。

* + 1. **运行效果**



图 19 用户退出提示

* 1. **其他**

在页面上方的**菜单栏**，点击PhotoStore选项，可以出现**“帮助”、“作者”、“退出”**三个选项，点击“帮助”，可以弹出由**msgbox**函数生成的使用提示，点击“作者”，可以弹出出由msgbox函数生成的作者（笔者本人）信息。

1. **“蹦极模拟器” 蹦极演示子系统**
   1. **原理**

如图所示蹦极模型，蹦极者从起跳点自由下落，蹦极绳自然下垂长度为L，弹力系数为k，计基准面为0，基准面距离地面h米，蹦极者体重m千克，空气阻力a1=a2=1，重力加速度。

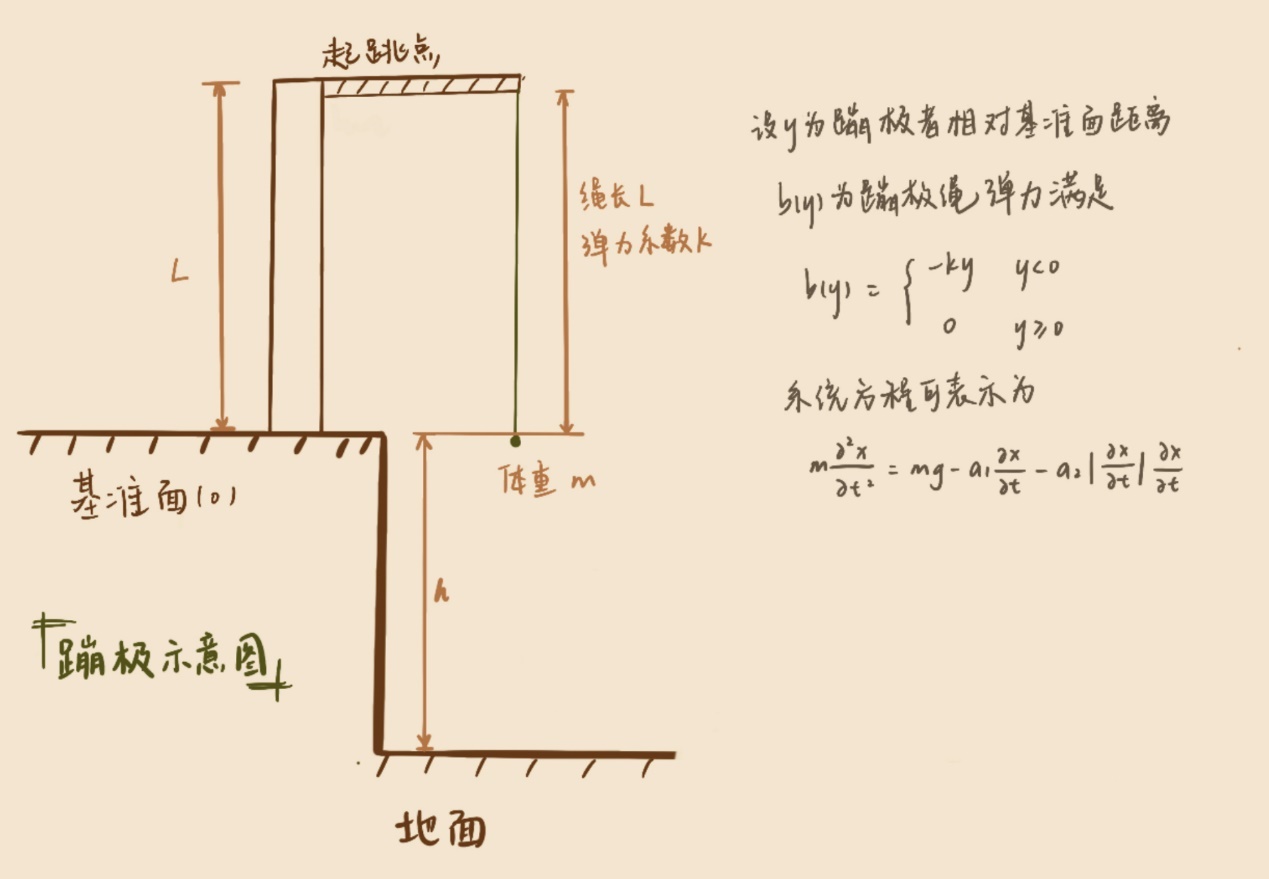


图 20 蹦极模拟器原理示意图

* 1. **功能分析**

“蹦极模拟器” 蹦极演示子系统基于simulink、 App Designer开发设计。在simulink的设计中，需要实现用户对于参数的输入，模型运算，结果的输出。在App Designer的设计中，需要实现整体界面的图形化，用户输入参数以及待求解变量的选择，模拟结果的输出。

* 1. **实现方法**
     1. **Simulink与App联动**

MATLAB提供了**assignin**函数：assignin(ws,var,val) 将值 val 赋给工作区 ws 中的变量 var。例如，assignin('base','x',42) 将值 42 赋予 MATLAB® 基础工作区中的变量 x。

在App的设计中，借助assignin函数，将用户输入的变量传递到工作区，接着利用sim函数调用编写好的simulink仿真，并将仿真结果传回工作区，App获取传递回来的仿真结果进行展示。

* + 1. **Simulink设计**

Simulink的设计如图所示，m、L、h、k变量从工作区进行获取，经过**积分模块、加法模块、选择模块、matlab fc模块**完成系统方程的计算，并借助**out模块**输出到工作区。

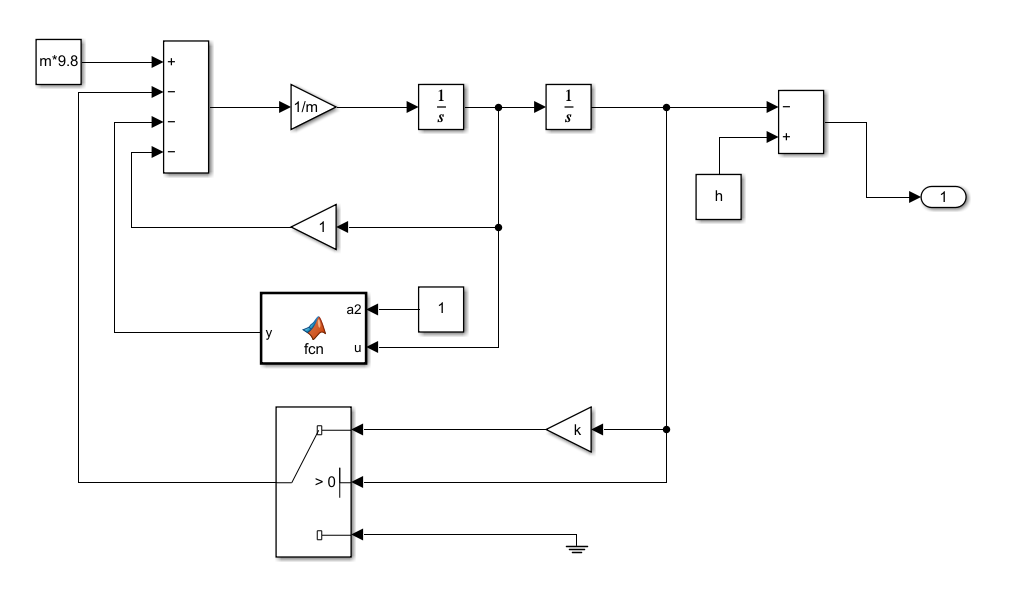
****

图 21 simulink设计图

* + 1. **App设计**

App的设计分为三部分，分别为输入参数模块，模拟结果模块，模拟控制模块。

* **输入参数模块**

本模块是由**单选按钮组**与**编辑字段**两部分结合形成，当用户选择使用**solve模式**时，通过单选按钮选择待求解的变量，选择后对应的编辑框设置**enable off**，其他变量可以使用系统默认参数，也可以用户在编辑框中自行输入；当用户选择使用**test模式**时，单选按钮**enable off**，编辑框**enable on**；锁定按钮由lockbutton(app)函数实现。

* **模拟结果模块**

模拟结果由两个坐标区域组成，左侧图窗用于显示当前模拟过程中离地距离与时间关系的曲线；右侧图窗用于模拟蹦极者运动状态，用圆圈表示蹦极者，当到达临界状态变为叉，运动状态用户可选择性输出，提高仿真速度。

全局变量**move**用于控制运动状态的选择性输出，运动状态的绘制运用到了**set**函数，读取仿真结果**yout**并赋给person；

* **模拟控制模块**

控制模块主要由四个按钮组成，分别用于开始**蹦极模拟，停止模拟，选择模拟模式，退出模拟系统**，除此之外还有一个**文字框**用于输出当前模拟参数，一个**指示灯**用于提示当前是否在进行模拟。

按下**开始蹦极按钮**（蹦极图标）后，系统读取用户选择的模拟模式、模拟参数等数据，调用**gobungee**函数。

按下**停止模拟按钮**，全局变量**ka**赋为1，对正在进行的模拟进行**中断**。

**选择模拟模式**是一个开关，用户可以选择模拟模式：**“solve”**或**“test”**；全局变量**swt**用于获取用户选择的仿真模式，执行对应输入参数的**按钮锁定**。

**退出模拟系统**按钮用于退出“蹦极模拟器”，在用户按下按钮后系统会弹出**questdlg提示框**，确认用户是否确认退出。

**文字框用**于显示在运行“solve”模式中，显示当前模拟测试参数的具体数值；**指示灯**用于提示当前是否在进行模拟。

* 1. **运行效果**

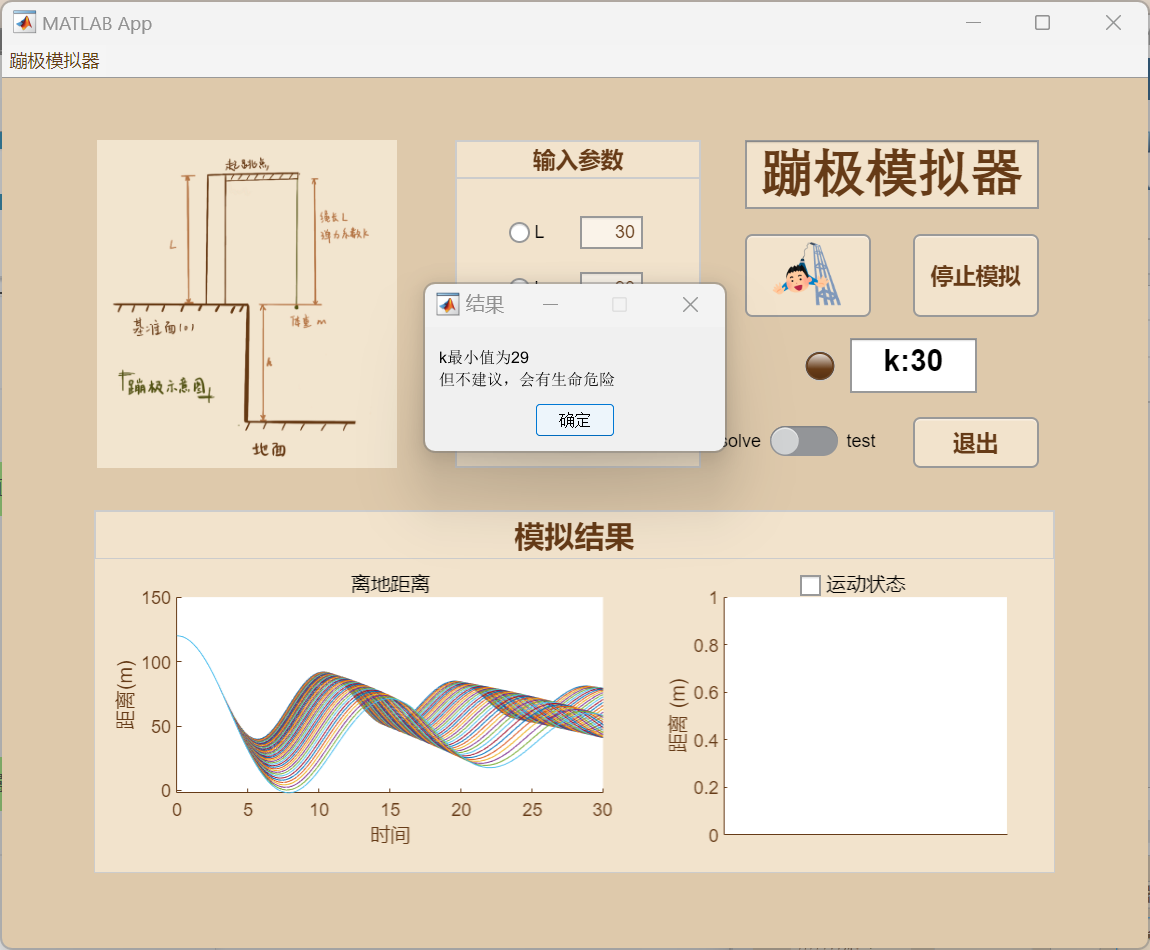
****

图 23 用户选择“solve ”模式对k进行求解

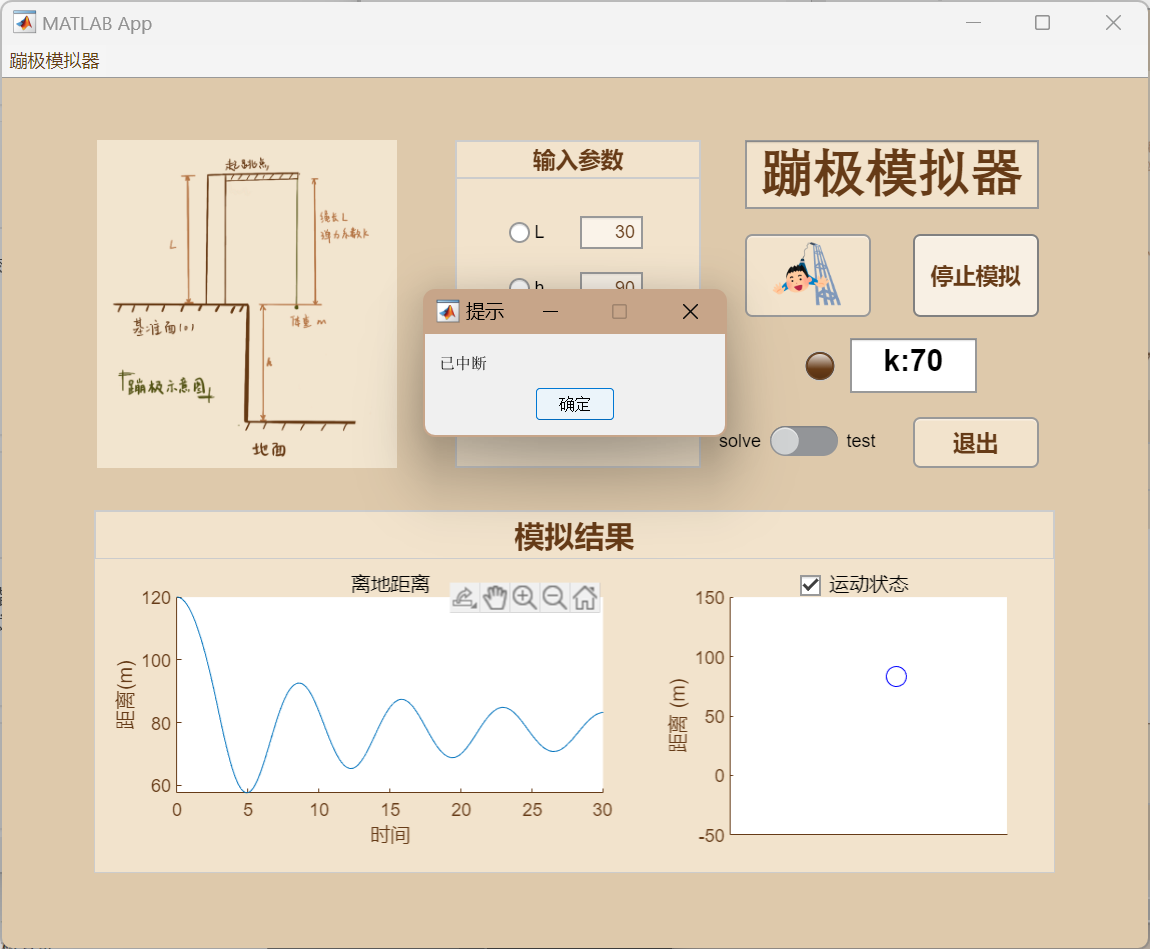
****

图 24 用户对模拟进行中断

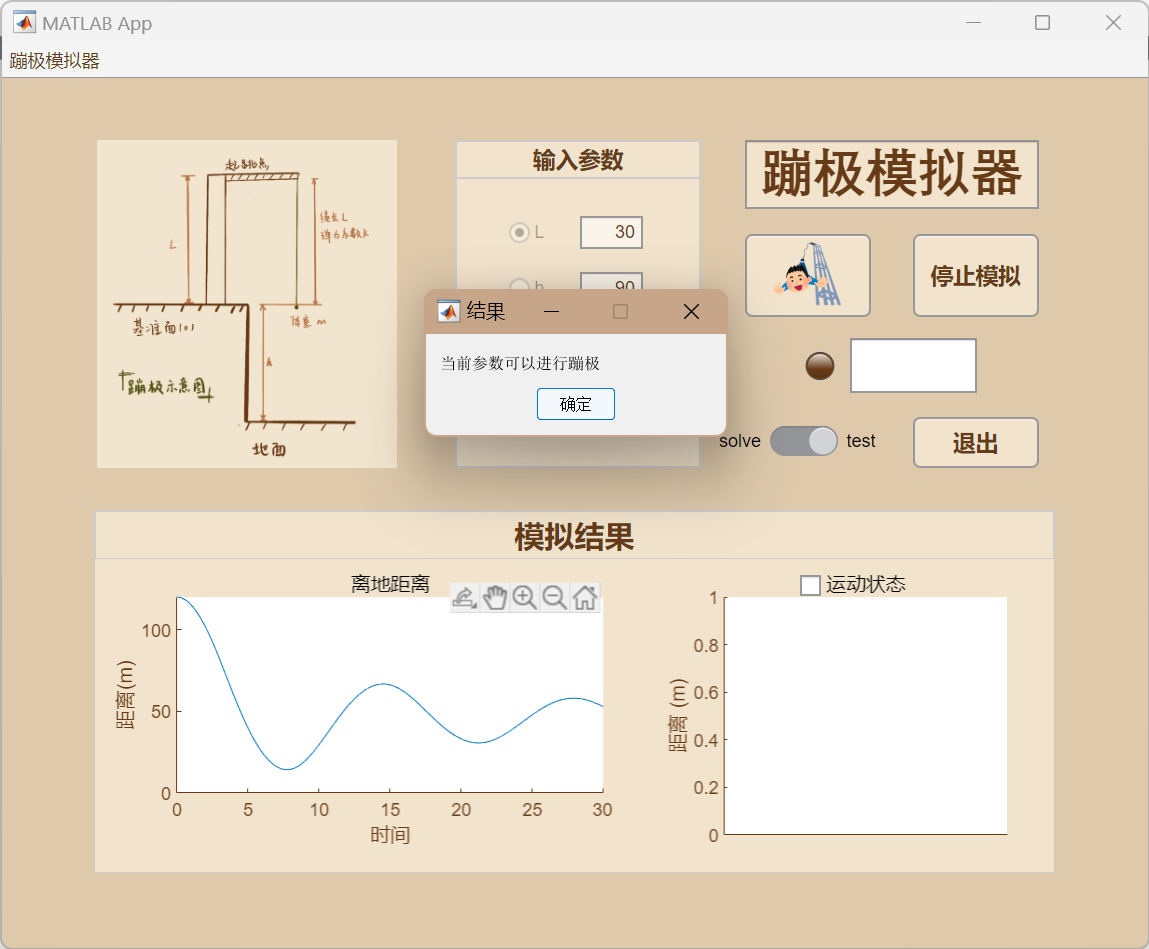
****

图 25 用户使用“test”模式进行参数测试

* 1. **设计亮点**
     1. 针对不同的模拟模式对输入参数进行限制，避免模拟产生输入性错误
     2. 进行多种人性化设计：利用文字框和指示灯显示当前模拟进程和模拟状况，运动状态的选择性输出可以帮助提高模拟速度。

1. **“扫雷”游戏子系统**
   1. **功能分析**

扫雷是一款电脑游戏。游戏目标是找出所有没有地雷的方格，完成游戏；要是按了有地雷的方格，游戏失败。

游戏开始时，玩家可看到空白方块排成的矩形，如果是第一次点开方块则不会踩到地雷。如果玩家点开没有地雷的方块，会有一个数字显现其上，这个数字代表著邻近方块有多少颗地雷（数字至多为8），玩家须运用逻辑来推断哪些方块含或不含地雷。

玩家可在推测有地雷的方块上点鼠右键，以放置标记来标明地雷的位置；若再次点击右键，成为空白的方块。若在已标明标记的方块点击左键，方块不会有任何的变动。

因此在“扫雷”的设计中，分为以下几步：创建gameboard，随机生成n个雷，根据玩家点击方式和点击位置对当前方块进行变形，判断玩家点击的位置是否含雷，判断当前玩家游戏状态，对游戏结果进行输出，重新开始游戏或退出游戏。

* 1. **实现方法**

在matlab中可以通过多种方法来构建 MATLAB® App：

* 使用 **App 设计工具**以交互方式构建
* 使用 **MATLAB 函数**以编程方式构建

在前两个系统的设计中，均采用了第一种方式：使用 App 设计工具以交互方式构建，而对于“扫雷”系统，需要更加开放的方式进行设计，因此**本系统使用 MATLAB 函数以编程方式构建**：通过figure 函数创建一个图窗以用作 UI 的容器。然后，以编程方式向其中添加组件。

* + 1. **系统流程图**

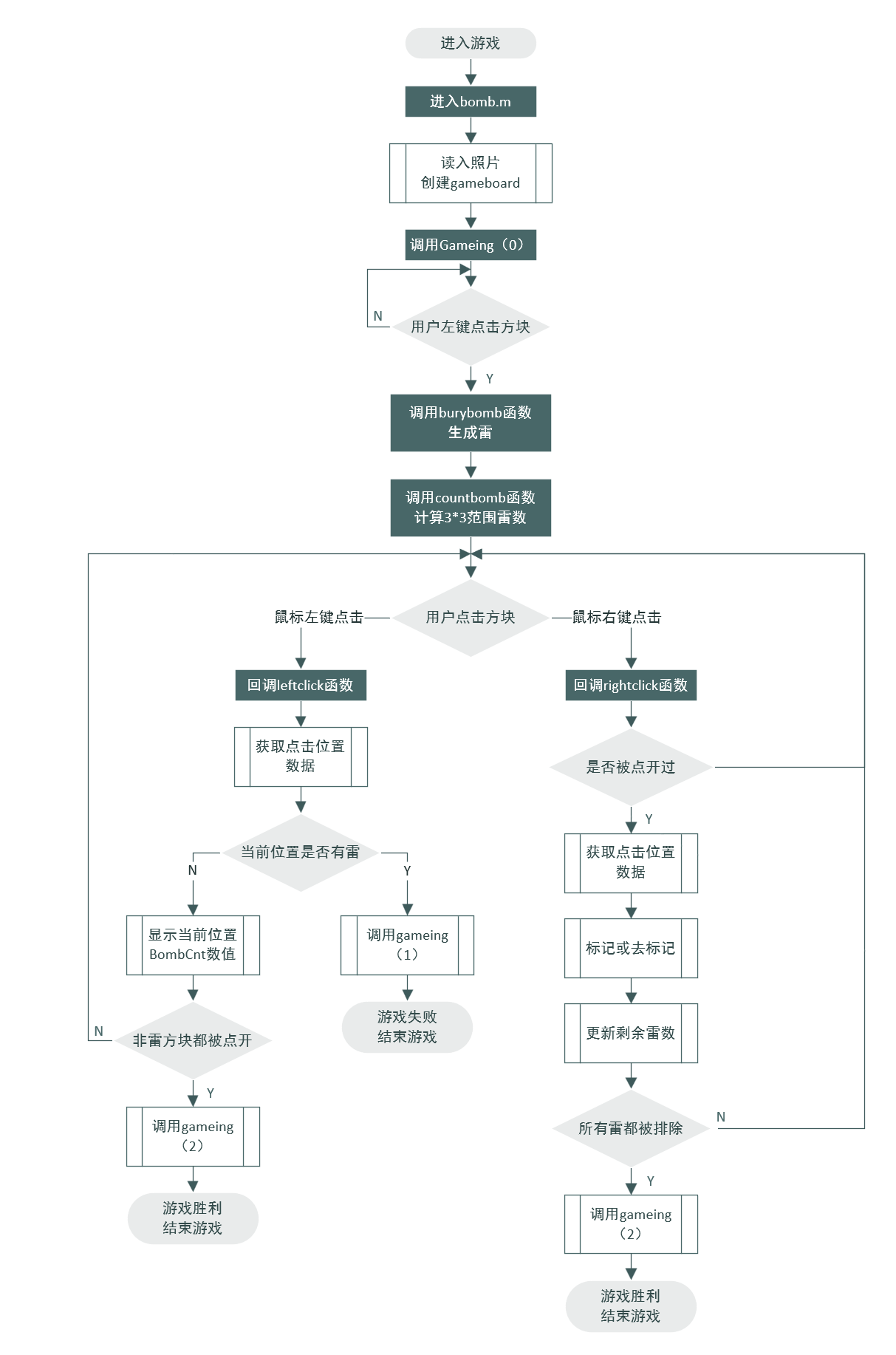
****

图 26 扫雷流程图

* + 1. **全局变量**
* **GameSize：**地图大小，设置为7
* **BombNum：**雷的个数，设置为15
* **Buttongroup：**gameboard按钮矩阵
* **UIFigure：**游戏图窗
* **Startbutton：**开始游戏按钮
* **Leavebutton：**离开游戏按钮
* **BombSet：**bomb分布矩阵 0：没有雷 1：有雷
* **BombCnt：** bomb接临矩阵 数字代表周围九宫格含雷数量
* **Chack：**左键点击矩阵 0：没有被玩家左键点开 1：已经被玩家左键点开
* **Mark：**右键标记矩阵 0：没有被玩家右键标记 1：已经被玩家右键标记
* **Init：**首次点击标志
* **StartPoint：**初次点击位置
* **Nowbomb：**当前剩余雷数（显示文本）
* **ResBomb：**当前剩余类数
* **Bombimg：**炸弹图标
* **dieimg ：**游戏结束图标
* **liveimg ：**游戏正常图标
* **markimg ：**右键标记图标
* **successimg ：**游戏成功图标
* **cyrimg：**游戏失败图标
  + 1. **bomb函数**

bomb函数是“扫雷”游戏的主函数，在本函数中，执行了以下功能：读取照片素材，初始化gameboard参数，创建游戏图窗，生成开始游戏与离开游戏按钮，调用gameing函数。

其中，**创建游戏图窗**使用了**figure**函数，设置了'Name'、'position' 、'Color'三个属性；**生成开始游戏与离开游戏按钮**使用了**uicontrol**函数，设置了'Parent'， 'Style'， 'Position'，'CallBack'， 'FontSize'，'FontWeight'， 'BackgroundColor'， 'CData'属性。

特别的，UIControl 对象包含**'CallBack'**和**'CData'**属性，前者可以设置左键按下按钮时的主回调函数，后者可以将按钮的图标设置为彩色图片。

* + 1. **gaming函数**

gaming函数是一个switch case otherwise结构，输入参数为0、1、2。

* **Case 0**：新游戏设置

开始按钮图标重置：使用**set**函数设置startbutton的**‘Cdata’**属性；

矩阵置零：将记录数据的矩阵利用zeros函数置零；

剩余雷数恢复：使用**uicontrol**函数，设置'Parent'，'BackgroundColor'，'Style'，'String'，'Position'， 'FontWeight'，'FontSize'， 'ForegroundColor'属性；

生成按钮矩阵：循环使用**uicontrol**函数，特别的，**'ButtondownFcn',**

**'CallBack'**，'两个属性分别为右键按下时的回调函数和左键按下时的回调函数，相比单一回调函数的App设计工具，编程方法有更多样的选择，可以更好的区分玩家鼠标的左、右键，除此之外，UIControl还有**‘UserData’**属性，可以指定为任何数组，在生成按钮的过程中，将‘UserData‘设置为**当前循环次数**，并做一定的处理用于记录当前生成按钮的位置，**这些特殊的属性是本系统选择用 MATLAB 函数以编程方式构建的主要原因。**

* Case 1：玩家踩到雷，游戏失败

开始按钮图标变换：开始按钮的图标由笑脸编程骷髅，表示本场游戏失败。

标记所有雷：循环整个gameboard，将是雷的方块利用**set**函数将**‘Cdata’**属性设置为雷图标

弹出游戏失败弹窗：利用**msgbox**提示玩家游戏失败。

* Case 2：所有雷都被标记/所有没有雷的方块都被点开，游戏胜利

弹出游戏胜利弹窗：利用**msgbox**提示玩家游戏胜利。

* + 1. **burybomb函数**

burybomb顾名思义，用来“埋雷”，在玩家进行第一次点击后调用次函数，函数实现**随机生成BombNum个雷记录到BombSet数组中**，为了实现玩家第一次点击避免雷，“埋雷”函数还实现了**移走用户第一次点击位置及其周围九宫格的雷**的功能，实现如下：

* + 1. **CountBomb函数**

本函数在完成“埋雷”后调用，通过双层循环计算每个方块3\*3范围雷的数目，记录在BombCnt数组中，计算方法使用sum函数：**sum(sum(BS(i-1:i+1,j-1:j+1)))**。

* + 1. **leftclick函数**

本函数为按钮的**左键回调函数**，当用户用鼠标左键点击方块时调用。

当用户第一次点击方块时，全局变量init为0，使用**get**函数获取用户点击句柄的**‘UserData’：get(gco,'UserData')**，记为全局变量StartPoint，随后调用burybomb函数和CountBomb函数，并将init赋为1，标志结束第一次点击。

当用户继续用鼠标左键点击，依旧调用leftclick函数，为pos赋值为**get(gco,'UserData')**，记录点击位置；根据点击位置进行判断，如果所在方块不是雷，通过set函数设置当前句柄的‘string’和‘enable’属性**set(gco,'string',BombCnt(pos(1),pos(2)))，set(gco,'Enable','off')，**并将此位置对应的chack数组被标记为1；如果所在方块是雷，通过set函数设置当前句柄的‘**Cdata’为雷图标**，调用**gameing（1）**结束本次游戏；判断本次点击后**是否将所有非雷的方块全部点开**，如果符合，调用**gameing（2）**结束本次游戏。

* + 1. **rightclick函数**

当用户用鼠标右键点击方块，调用此函数。进入函数，首先判断当前按钮是否被点开，如果没有被点开，通过**get**函数获取该方块的**‘string’**和**‘UserData’**赋值给str和pos，判断str决定对该方块标记或去标记。完成标记后更新剩余雷的个数，判断当前**mark矩阵和Bombset矩阵**是否相同，如果相同调用**gameing（2）**结束本次游戏。

* 1. **运行效果**

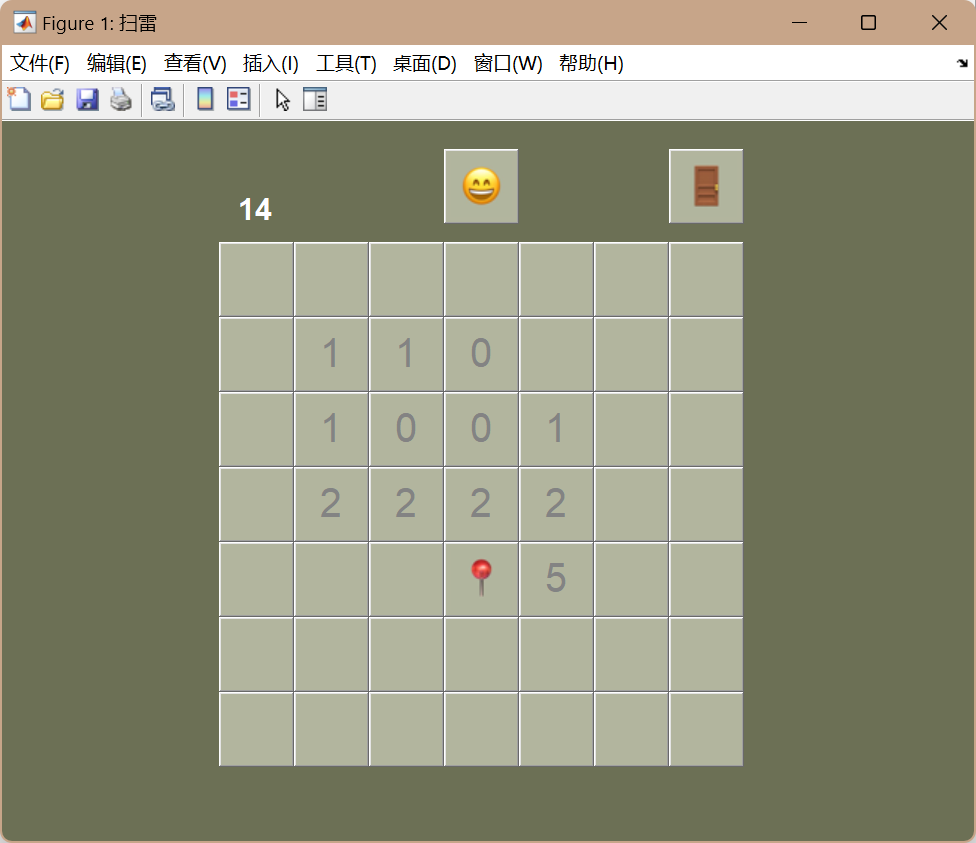
****

图 31 扫雷游戏界面

****

图 32 玩家踩到雷 游戏失败

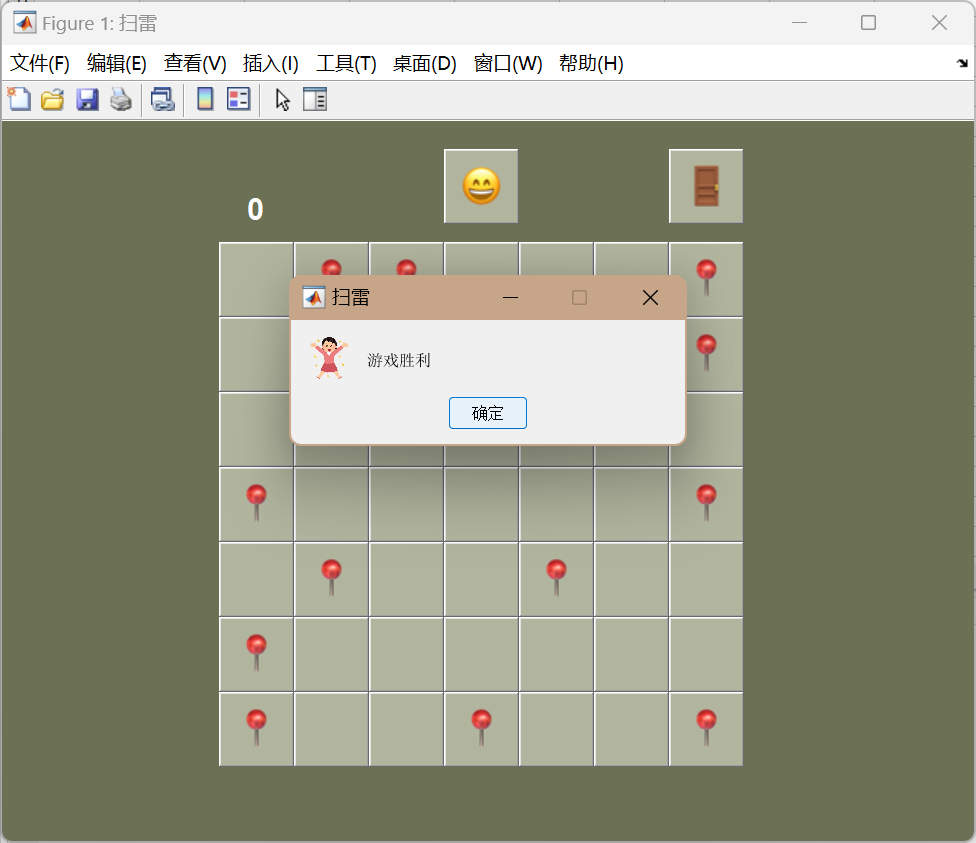
****

图 33 玩家排除所有雷 游戏胜利

* 1. **设计亮点**
     1. 还原了经典游戏“扫雷”的基本游戏设计
     2. 有较好的交互性，图形化

1. **总结与心得**

本次大作业完成了图像处理系统、蹦极模拟系统以及扫雷游戏系统，通过大作业的撰写，有效的串联起本门课一学期的学习内容，将他们综合运用：通过设计系统，掌握App designer的使用；通过编写图像处理功能，掌握本学期学过的图像操作；通过蹦极模拟系统，掌握simulink的使用；通过扫雷游戏的设计，锻炼解决问题的能力。通过这次大作业系统的设计与开发，让我对MATLAB有了更深入的了解，对于MATLAB的使用有了更进一步的练习；特别是，当我想实现某一项功能时，借助MATLAB提供的帮助，一步步解决问题，实现效果。感谢这门课让我了解了这样一个极为强大且易于掌握的工具。