学号 专业 姓名

理论课任务3

【任务名称】 更完整的扫雷程序

【任务目的】

增加一个菜单项，可以设置行数、列数和雷数

当点击的方块为0时，能自动打开其周围所有方块

【源码】

clear; close all; clc

global MineRow; %为了方便在函数中更改行数，定义全局变量

global MineColumn; %为了方便在函数中更改列数，定义全局变量

global MineNum; %为了方便在函数中更改雷数，定义全局变量

global fig; %定义全局变量fig

MineRow = 0; %初始值为0

MineColumn = 0;

MineNum = 0;

fig = uifigure('Name','扫雷','Position',[0 0 300 300]);

menu1 = uimenu(fig,'Enabl','on','Text','设置'); %设置菜单栏

menu1.MenuSelectedFcn = @(~,~)Fun2(fig); %菜单的回调函数Fun2

movegui(fig,'center'); %将fig移到屏幕中间

菜单的回调函数Fun2

function Fun2(fig) %定义菜单的回调函数fig

fig2 = uifigure("Position",[0 0 300 300]);

movegui(fig2,"center");

g = uigridlayout(fig2,'ColumnWidth',{'1x','1x'},'RowHeight',{'1x','1x','1x','1x'});

a = uilabel(g,'Text','行数');

a.Layout.Column = 1;a.Layout.Row = 1;

b = uilabel(g,'Text','列数');

b.Layout.Column = 1;b.Layout.Row = 2;

c = uilabel(g,'Text','雷数');

c.Layout.Column = 1;c.Layout.Row = 3;

d = uieditfield(g,'numeric','Limits',[1 inf],'RoundFractionalValues','on');

d.Layout.Column = 2;d.Layout.Row = 1;

e = uieditfield(g,'numeric','Limits',[1 inf],'RoundFractionalValues','on');

e.Layout.Column = 2;e.Layout.Row = 2;

f = uieditfield(g,'numeric','Limits',[1 inf],'RoundFractionalValues','on');

f.Layout.Column = 2;f.Layout.Row = 3;

h = uibutton(g,'Text','确定','ButtonPushedFcn',@Fun3);

h.Layout.Column = 2;h.Layout.Row = 4;

end

菜单中确认按钮的回调函数Fun3

function Fun3(app,event)

global MineRow;

global MineColumn;

global MineNum;

global fig;

MineRow = app.Parent.Children(4).Value;

MineColumn = app.Parent.Children(5).Value;

MineNum = app.Parent.Children(6).Value;

app.Parent.Parent.Visible = 'off';

grid = uigridlayout(fig,'ColumnWidth',repmat({'1x'},1,MineColumn),...

'RowHeight',repmat({'1x'},1,MineRow),...

'ColumnSpacing',0,'RowSpacing',0, ...

'Padding',[0 0 0 0]);

boom = rand(MineRow,MineColumn)<MineNum/(MineRow\*MineColumn);

for x = 1:MineRow

for y = 1:MineColumn

loc = [x-1,y-1;x-1,y;x-1,y+1;x,y-1;x,y+1;x+1,y-1;x+1,y;x+1,y+1];

loc(any(loc<1|loc>MineColumn|loc>MineRow,2),:) = [];

if boom(x,y)

data = 'X';

else

data = num2str(sum(boom((loc(:,2)-1)\*MineRow+loc(:,1))));

end

button(x,y) = uibutton(grid,'Text','','UserData',data);

end

end

% for x = 1:MineRow

% for y = 1:MineColumn

% loc = [];

% x1 = [-1 -1 0 1 1 1 0 -1];

% y1 = [0 -1 -1 -1 0 1 1 1];

% for k = 1:8

% if x+x1(k)<1 ||y+y1(k)<1||x+x1(k)>MineRow||y+y1(k)>MineColumn

% continue;

% end

% loc = [loc;x+x1(k) y+y1(k)];

% end

% if boom(x,y)

% data = 'X';

% else

% data = num2str(sum(boom(loc,'all')));

%

% end

% button(x,y) = uibutton(grid,'Text','','UserData',data);

% end

% end

for x = 1:MineRow

for y = 1:MineColumn

button(x,y).ButtonPushedFcn = @(~,~)Fun(button,x,y);

end

end

end

每个按钮的回调函数Fun

function Fun(button,x,y)

global MineRow;

global MineColumn;

button(x,y).Enable = 'off';

button(x,y).Text = button(x,y).UserData;

if button(x,y).Text = = '0'

x1 = [-1 -1 0 1 1 1 0 -1];

y1 = [0 -1 -1 -1 0 1 1 1];

for k = 1:8

if x+x1(k)<1 ||y+y1(k)<1||x+x1(k)>MineRow||y+y1(k)>MineColumn

continue;

end

button(x+x1(k),y+y1(k)).Enable = 'off';

button(x+x1(k),y+y1(k)).Text = button(x+x1(k),y+y1(k)).UserData;

end

end

if strcmp(button(x,y).Text,'X')

for i = 1:numel(button)

button(i).Enable = 'off';

if strcmp(button(i).UserData,'X')

button(i).Text = 'X';

end

end

end

end

【运行结果截图】

|  |  |
| --- | --- |
|  |  |



【任务总结】

经过多次调试，本程序的bug在于，当输入的行列不等时，如果用PPT的方法，UserData的计算会发生错误，如果用下面这种方法算UserData的话，逻辑上是没有问题的，但是当发现这个bug的时候已经晚了，由于很多函数封装的不够好，提前没有规划好变量类型，在写的时候又临时定义很多全局变量，很多地方定义的变量牵一发而动全身，所以再想改的话就会出现很多问题，

for x=1:MineRow

for y=1:MineColumn

loc=[];

x1=[-1 -1 0 1 1 1 0 -1];

y1=[0 -1 -1 -1 0 1 1 1];

for k=1:8

if x+x1(k)<1 ||y+y1(k)<1||x+x1(k)>MineRow||y+y1(k)>MineColumn

continue;

end

loc=[loc;x+x1(k) y+y1(k)];

end

if boom(x,y)

data='X';

else

data=num2str(sum(boom(loc,'all')));

这里计算UserData的方法用了C语言中处理二维矩阵的常用方法：构造位移方向向量，二维的矩阵共有8个方向，在if函数中判断是否超过范围，超过范围就换方向，虽然和matlab矩阵编程的思想不符合，运算速度也慢一些，但是至少可以保证逻辑正确。

之前不理解为什么PPT上面的代码要先把figure的Position放到左下角，然后再移到中间来，后来写matlab拼图的时候发现，Position的表示的是位置，如果想控制figure的长和宽的话需对Position的四个参数进行计算，但是如果把Position前两个参数设置为0，这时第3，4个参数就分别代表figure的长和宽。

对函数的封装一定要到位，高内聚低耦合，否则时候再想改就很麻烦。

在设置雷数的对话框的时候，其实可以直接用matlab自带的对话框inputdlg

，这样比自己创建figure简单很多，这也是写拼图的时候发现的，温故而知新。

由于我对回调函数对于形参实参的那些引用概念不清楚，所以本程序定义了很多全局变量，虽然会影响速度，但是大大方便了函数里面对变量的直接修改，无需担心输入参数不够之类的报错。

由于matlab2021b已经支持把函数之间定义到主函数的结尾不用新建M文件，此代码涉及到的3个函数都是回调函数，所以直接定义到了主函数的结尾以方便编写与理解。