学号 WA2224013 专业 机器人工程 姓名 郭义月

理论课任务4

【任务名称】 用MATLAB实现拼图游戏

【任务目的】

1. 生成个按钮，点击任一个按钮可使之移动到空白区域（采用纯代码编写）
2. 点击“载入图片”选项，可打开图片并使之填充至按钮上
3. 点击“随机重置”选项，可随机打乱按钮的位置（保证打乱后的顺序是可还原的）
4. 点击“重设块数”选项，可重新设置按钮的数量
5. 点击“自动游戏”选项，可自动对当前乱序进行逐步还原（采用A星算法实现）

【源码】

主函数：

clc;clear;close all;

global fig;

global g;

global xv;

global yv;

global n;

n=4;

xv=n;

yv=n;

image=imread('E:\Matlab\111\ClassAssignment\JigsawPuzzle\Picture.png');

x=SplitArray(image,n);

for i=1:n

for j=1:n

imwrite(x{i,j},['E:\Matlab\111\ClassAssignment\JigsawPuzzle\Picture',num2str(n),num2str(i),num2str(j),'.png']);

end

end

fig=uifigure('Position',[0 0 400 400]);

g=uigridlayout(fig,'ColumnWidth',repmat({'1x'},1,n),...

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

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

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

movegui(fig,"center");

menu=uimenu(fig,'Text','功能');

for i=1:n

for j=1:n

if i==n&&j==n

continue;

end

data=[i j];

button(i,j)=uibutton(g,'Text','','UserData',data);

button(i,j).ButtonPushedFcn=@(~,~)ButtonMove(button,i,j);

end

end

% GenerateButton();

menuitem1=uimenu(menu,'Text','载入图片');

menuitem1.MenuSelectedFcn=@(~,~)LoadPicture(button);

menuitem2=uimenu(menu,'Text','随机重置');%可还原：存在的逆序为偶数

% 会留下9的那个位置

menuitem2.MenuSelectedFcn=@(~,~)RandomReset(button);

menuitem3=uimenu(menu,'Text','重设块数');

menuitem3.MenuSelectedFcn=@(~,~)ResetNumber();

menuitem4=uimenu(menu,'Text','自动游戏');

menuitem4.MenuSelectedFcn=@(~,~)AutoMove(button);

分割矩阵，把imread读到的二维矩阵分成n块

function [x]=SplitArray(Array,n)

x=cell(n,n);

row=size(Array,1);

column=size(Array,2);

for i=1:n

for j=1:n

x{i,j}=Array((i-1)\*fix(row/n)+1:fix(row/n)\*i,(j-1)\*fix(column/n)+1:j\*fix(column/n),:);

end

end

end

函数：按钮移动，点击任何一个按钮可以使之移动到空白区域

function ButtonMove(button,i,j)

global xv;

global yv;

temp=button(i,j).UserData;

button(i,j).UserData=[xv yv];

button(i,j).Layout.Row=xv;

button(i,j).Layout.Column=yv;

xv=temp(1);yv=temp(2);

end

菜单中，载入图片的回调函数

function LoadPicture(button)

global n;

for i=1:n

for j=1:n

if i==n&&j==n

continue;

end

button(i,j).Icon=['E:\Matlab\111\ClassAssignment\JigsawPuzzle\Picture',num2str(n),num2str(i),num2str(j),'.png'];

% button(i,j).Icon=['E:\Matlab\111\ClassAssignment\JigsawPuzzle\Picture411',num2str(n),num2str(i),num2str(j),'.png'];

end

end

end

菜单中，随机重置的回调函数

function RandomReset(button)

global n;

A1=kron((1:n)',ones(n,1));

A2=repmat((1:n)',n,1);

A=[A1,A2];

A(n\*n,:)=[];

C=ones(1,n\*n-1);

m=1;

global B;

while mod(m,2)~=0 %确保逆序为偶数，确保其可还原

rowrank = randperm(size(A, 1)); % 随机打乱的数字，从1~行数打乱

B = A(rowrank, :);%%按照rowrank打乱矩阵的行数

for i=1:size(B)

C(i)=(B(i,1)-1)\*n+B(i,2);

end

m=0;

for j=1:size(C,2)

for k=j:size(C,2)

if C(j)>C(k)

m=m+1;

end

end

end

end

for i=1:n

for j=1:n

if i==n&&j==n

continue;

end

button(i,j).UserData(1)=B((i-1)\*n+j,1);

button(i,j).UserData(2)=B((i-1)\*n+j,2);

button(i,j).Layout.Row=B((i-1)\*n+j,1);

button(i,j).Layout.Column=B((i-1)\*n+j,2);

end

end

end

菜单中，重设块数的回调函数

function ResetNumber()

global n;

global g;

answer=inputdlg('输入块数：');

n=sqrt(str2num(answer{1}));

image=imread('E:\Matlab\111\ClassAssignment\JigsawPuzzle\Picture.png');

x=SplitArray(image,n);

for i=1:n

for j=1:n

if i==n&&j==n

continue;

end

data=[i j];

button(i,j)=uibutton(g,'Text','','UserData',data);

button(i,j).ButtonPushedFcn=@(~,~)ButtonMove(button,i,j);

button(i,j).Layout.Row=i;

button(i,j).Layout.Column=j;

end

end

for i=1:n

for j=1:n

imwrite(x{i,j},['E:\Matlab\111\ClassAssignment\JigsawPuzzle\Picture',num2str(n),num2str(i),num2str(j),'.png']);

end

end

LoadPicture(button);

end

菜单中，自动游戏的回调函数

function AutoMove(button)

global B;

global xv;

global yv;

global n;

xv1=xv;

yv1=yv;

B1=B;

A1=kron((1:n)',ones(n,1));

A2=repmat((1:n)',n,1);

A=[A1,A2];

A(n\*n,:)=zeros(1,2);

B1=[B1;zeros(1,2)];

while(~isequal(A(1:n\*n-1,:),B1(1:n\*n-1,:)))

for i=1:n

for j=1:n

pause(3);

if i==n&&j==n

continue;

end

if button(i,j).UserData(1)==i&& button(i,j).UserData(2)==j

continue;

end

xx=button(i,j).UserData(1);

yy=button(i,j).UserData(2);

% xt=B1((i-1)\*n+j,1);

% yt=B1((i-1)\*n+j,2);

button(B1((i-1)\*n+j,1),B1((i-1)\*n+j,2)).UserData=[xv1 yv1];

% button(B1((i-1)\*n+j,1),B1((i-1)\*n+j,2)).UserData(2)=yv1;

button(B1((i-1)\*n+j,1),B1((i-1)\*n+j,2)).Layout.Row=xv1;

button(B1((i-1)\*n+j,1),B1((i-1)\*n+j,2)).Layout.Column=yv1;

button(i,j).Layout.Row=i;

button(i,j).Layout.Column=j;

button(i,j).UserData=[i j];

B1((xv1-1)\*n+yv1,:)=B1((i-1)\*n+j,:);

% Temp=B((i-1)\*n+j,:);

% B(Temp(1),Temp(2))=[xv1 yv1];

B1((i-1)\*n+j,:)=[i j];

xv1=xx;yv1=yy;

% B1((xv1-1)\*n+yv1,:)=[0 0];

end

end

end

% pause(1.5);

% goal=[1 1;1 2;1 3;2 1;2 2;2 3;3 1;3 2];

% D2=[1 1;1 2;1 3;2 1;3 3;2 3;3 1;3 2];

% for i=1:size(goal,1)

% button(goal(i,1),goal(i,2)).UserData(1)=button(D2(i),1);

% button(goal(i,1),goal(i,2)).UserData(2)=button(D2(i),2);

% button(goal(i,1),goal(i,2)).Layout.Row=D2(i,1);

% button(goal(i,1),goal(i,2)).Layout.Column=D2(i,2);

% end

end

【运行结果截图】

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

【任务总结】

本程序虽能实现基本功能，但存在多处bug，当重设块数之后，随机重置时，就会显示'matlab.graphics.Graphics'的属性 'UserData' 无法识别。但是UserData再每次重置块数时都是重新定义的，所以不知道为什么会无法识别。但是本程序已经把块数n设置为全局变量，直接在程序中更改n的值即可。

在随机重置中，如何保证每次的重置都能还原？查阅资料可得，当随机序列的逆序数为偶数时，拼图游戏可以还原，当为奇数时，游戏不能还原。即每次随机数都有1/2的概率能够还原，只要在while循环里判断逆序数时候为偶数就可以保证随机重置的拼图能复原。

while mod(m,2)~=0 %确保逆序为偶数，确保其可还原

rowrank = randperm(size(A, 1)); % 随机打乱的数字，从1~行数打乱

B = A(rowrank, :);%%按照rowrank打乱矩阵的行数

for i=1:size(B)

C(i)=(B(i,1)-1)\*n+B(i,2);

end

m=0;

for j=1:size(C,2)

for k=j:size(C,2)

if C(j)>C(k)

m=m+1;

end

end

end

end

对于拼图这类简单路径规划问题，不用A\*算法，直接做也可以实现自动游戏。

用B1来记录每个时刻各个按钮所在的位置，依次循环行与列，对于不在自己位置上的按钮，将霸占他位置的按钮移到空白位置，再将该按钮移到对应位置，这样该按钮之前的位置就是需要更新的空位的位置。虽然思路简单，但是劣势在于，只循环一次大概率是得不到完成的拼图的，这个时候可以在while循环里面判断B1矩阵与目标矩阵是否相等，但是如果加入while循环，matlab就非常容易死机，该方法存在很多弊端。如果不加while循环，可以一直点自动游戏，最后总会完成拼图。（A\*算法真的写不出来，自己想了一个暴力简单的思路，在优化中遇到了问题）

在循环中，如果已经循环到最后或者按钮已经在原位置，结束本次循环

if i==n&&j==n

continue;

end

if button(i,j).UserData(1)==i&& button(i,j).UserData(2)==j

continue;

end

xx ，yy表示按钮I，j目前所在位置

xx=button(i,j).UserData(1);

yy=button(i,j).UserData(2);

% xt=B1((i-1)\*n+j,1);

% yt=B1((i-1)\*n+j,2);

把霸占当前按钮的按钮移到空位

button(B1((i-1)\*n+j,1),B1((i-1)\*n+j,2)).UserData=[xv1 yv1];

% button(B1((i-1)\*n+j,1),B1((i-1)\*n+j,2)).UserData(2)=yv1;

button(B1((i-1)\*n+j,1),B1((i-1)\*n+j,2)).Layout.Row=xv1;

button(B1((i-1)\*n+j,1),B1((i-1)\*n+j,2)).Layout.Column=yv1;

button(i,j).Layout.Row=i;

button(i,j).Layout.Column=j;

button(i,j).UserData=[i j];

更新按钮的状态B1

B1((xv1-1)\*n+yv1,:)=B1((i-1)\*n+j,:);

% Temp=B((i-1)\*n+j,:);

% B(Temp(1),Temp(2))=[xv1 yv1];

B1((i-1)\*n+j,:)=[i j];

更新空位的坐标 xv1，yv1

xv1=xx;yv1=yy;

% B1((xv1-1)\*n+yv1,:)=[0 0];

写了好久，之前写的过几天不看就又忘了，本程序依然使用了很多全局变量，导致后来很多变量都在函数中被改了但是我不知道，简直就是一坨。